CURE: An Effective COVID-19 Remedies based on Machine Learning Prediction Models

Poonam Phogat[a], Rajat Chaudhary[b]

[a]Computer Science & Engineering, SGT University, Gurugram, Haryana (India)
[b]Computer Science & Engineering, Bharat Institute of Engineering & Technology, Hyderabad, Telangana (India)

Abstract

Coronavirus disease (COVID-19) is a severe pandemic infectious virus that enters into healthy cells of a living body. COVID-19 virus makes copies in the organs of the host body by multiplying itself which ultimately leads to the death of some healthy cells and therefore weakens the immune system. In a mild stage, it mainly affects the respiratory tract and leads to pneumonia, organ failure, and death reaching the last stage. This paper focused on the early detection of the COVID-19 patient based on the positive symptoms of the disease. In this paper, the COVID-19 Remedies (CURE) scheme is proposed based on machine learning prediction models for the treatment of COVID patients. For experimental results, the performance analysis of the CURE scheme is evaluated on the Python platform which is tested using the Kaggle dataset from Johns Hopkins University.

Keywords

COVID-19, Machine Learning, Prediction Model

1. Introduction

The virus that induces COVID-19 is a severe acute respiratory syndrome coronavirus-2 (COVID-2) that was first diagnosed in late December 2019 during an investigation into an outbreak in Wuhan, China. As the cases were increasing rapidly throughout the world, the WHO declared the disease pandemic on March 11, 2020. Currently, the transmission of COVID-19 becomes uncontrollable because the number of cases has reached the threshold limit [1]. The virus enters into healthy cells of a living body and makes copies in the organs of the host body by multiplying itself which ultimately leads to the death of some healthy cells and therefore weakens the immune system. In a mild stage, it mainly affects the respiratory tract and leads to pneumonia, organ failure, and death reaching the last stage [2]. The disease is prominent in old age people with a weak immune system and already having other primitive diseases like diabetes, high blood pressure, cardiovascular and respiratory diseases [3]. Figure 1 shows the global statistics till July 30, 2020, on the total confirmed cases, active cases, total deaths, and total cured cases on the COVID-19 virus. Figure 1(a) presents the total number of coronavirus cases across different countries which shows that the virus is spreading rapidly with the highest cases in the USA followed by India. The total confirmed positive cases across the world are 2,18,69,976 out of which 26,47,663 cases are of India. Figure 1(b) shows the statistics of the active cases, where there are 65,04,303 active cases occurred globally.

Figure 1(c) presents the total death cases, and finally, Figure 1(d) shows the total cured cases [4]. This is a communication spreading virus that spreads through respiratory droplets present in the air. These aerosols come to an open environment when an infected person sneezes and coughs and enter in other persons through the mouth and nostrils and reach to lungs. There is no precise treatment to cure COVID-19. Some steps are being taken to eliminate the virus using different medicines like Hydroxychloroquine which is an antimalarial antibiotic. Currently, it is used to treat coronavirus patients, it helps in inhibition of infection by increasing the endosomal pH which provides enough strength to the immune system to fight against the viral disease [5].

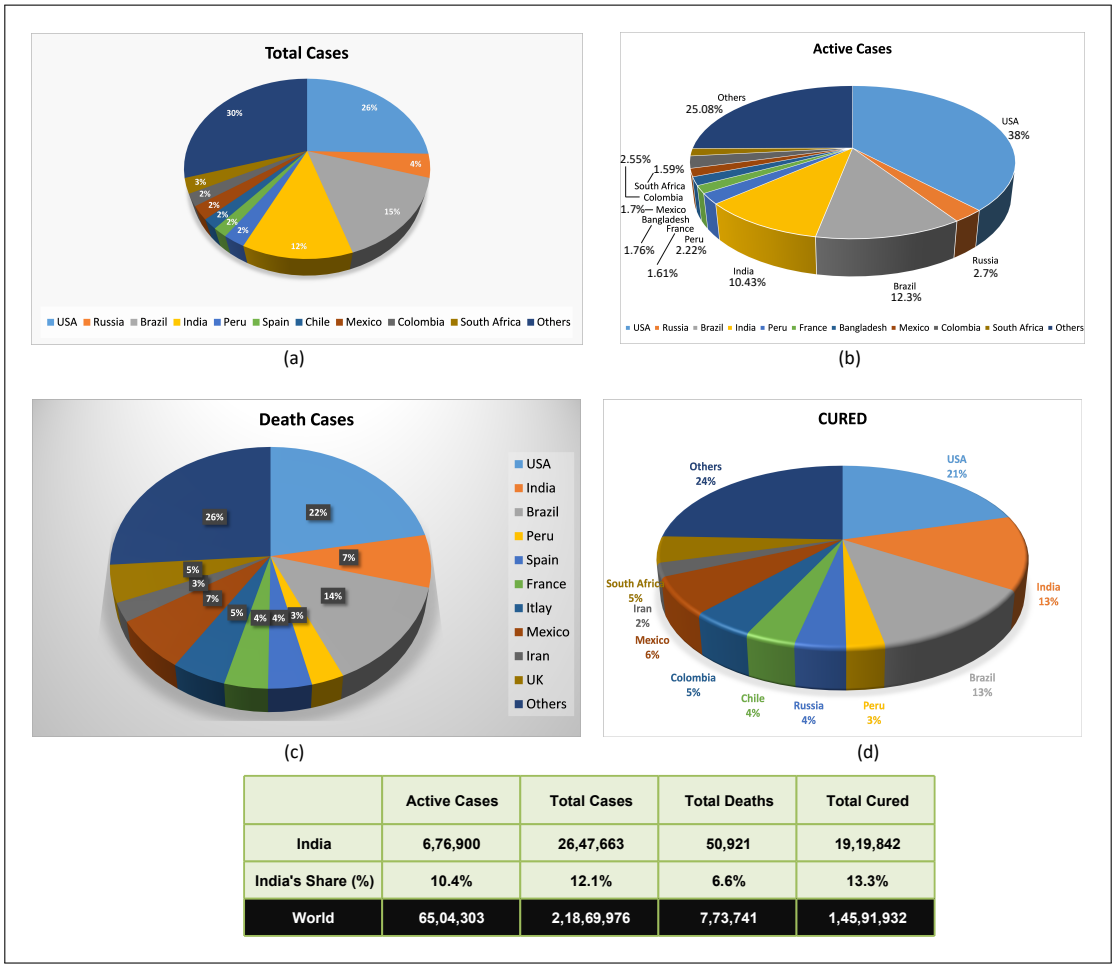
Some preventions are necessary for the treatment of this pandemic. From the very beginning of COVID-19, the government of almost all the countries has taken strict actions such as complete lockdown, social distancing, use of sanitizer, and masks to reduce all the causing elements [6]. By exploring various studies, Machine Learning seems to be the best prediction model for forecasting the increasing COVID-19 infected cases. Regression and classification approach of ML work according to the availability of data to diagnose this problem.

1.1. Contributions

The contributions of the paper are summarized below.

- Diagnose the symptoms of COVID-19 patients based on the classification of the diseases.
- To recover the COVID-19 patients, CURE scheme

ISIC'21: International Semantic Intelligence Conference, February 25-27, 2021, Delhi, India
poonamphogat07@gmail.com (P. Phogat); rajat@biet.ac.in (R. Chaudhary)
0000-0002-6554-918X (R. Chaudhary)



	Active Cases	Total Cases	Total Deaths	Total Cured
India	6,76,900	26,47,663	50,921	19,19,842
India's Share (%)	10.4%	12.1%	6.6%	13.3%
World	65,04,303	2,18,69,976	7,73,741	1,45,91,932

Figure 1: Data statistics of total, active, death, and cured cases on COVID-19.

is proposed scheme based on machine learning prediction model to forecast the best suitable treatment for COVID disease.

- For simulation, the proposed scheme is tested using the Kaggle dataset.
- Finally, the performance evaluation is compared with the five classifiers and predicts the most efficient outcome using the Python platform.

1.2. Paper Organization

The rest of the paper is structured as follows: Section II discusses the literature review of the existing schemes. Section III presents the system model followed by the proposed CURE scheme in Section IV. Section V comprises of the prediction performance evaluation, and finally, Section VI concludes the paper.

2. Literature Review

The researchers introduce some methods of Machine Learning for classification. The easiest classification is the Linear Regression method which is used to reduce the sum of squared differences between real and predicted data. The drawbacks of this model are its non-effectiveness with non-alignment data and sensitiveness to deviation [7]. Through the Logistic Regression Model, it is shown that the contingency of conclusion is Logistic function-based. The positiveness of this model is that it is free of complications. But it fails to assume linearity. By Naive Bayes Model, it is proposed that it confined

training data to calculate inevitable parameters and efficaciously deals with real-world data. One another model K-Nearest Neighbour shows that it works efficiently with modest data and relevant with multi-class problems [8], [9].

Pinter *et al.* [10] proposed Machine Learning approaches multi-layered perceptron-imperialist competitive algorithm (MLP-ICA) and adaptive network-based fuzzy interference system (ANFIS) for prediction of the COVID-19 confirmed positive and death cases. This model is used to maintain accuracy for the next 9 days which gives the reassuring results [11]. The government and the public have to appreciate the researchers and help in lowering the data by maintaining social distancing and following other precautions [12]. Hamzeh *et al.* [13] works on Susceptible-Exposed-Infectious-Recovered (SEIR) model which predicts that it performs well on moderate data. The outbreak of this infectious disease may cause variations in the data prediction.

Jia *et al.* [14] defines four stages for COVID-19 cases. In the first stage, there comes travel history of a person having COVID-19 symptoms which leads to lockdown. When the infected person comes in contact with other persons, the virus reached in the second stage. To prevent the increasing data social distancing is applied. Next, the third stage in which there is neither travel history nor contact with an infected person. So the chances of viral spreading through the respiratory droplets become high. Hence, the use of masks and sanitizers is necessary. The next and last stage is an uncontrollable stage where the cases reached the threshold limit. Tuli *et al.* [15] improved COVID-19 prediction by using a model of Machine Learning. In this model data-driven approach is used to help the government and the public. After covering data with ML and AI, researchers can forecast the time scale and regions where the possibility of spreading of this disease is maximum. This is predicted that using different models of ML, COVID-19 cases can be controlled or eliminated from all the countries of the world which are facing this critical situation.

3. System Model

Figure 2 presents the workflow of the proposed CURE scheme for the treatment of COVID patient. Initially, the input is the dataset that is taken from Johns Hopkins University dataset. Then the symptoms of positive cases are analyzed which are categorized into 3 sub-parts: Severe, Moderate, and Mild symptoms. A patient having severe symptoms which includes throttling must face a harsh period. Moderate symptoms include shortness of breath, fever, cough. Mild symptoms include fever, cough, and headache. The proposed scheme for COVID-19 outbreak analysis is trained and tested on real-time data using the

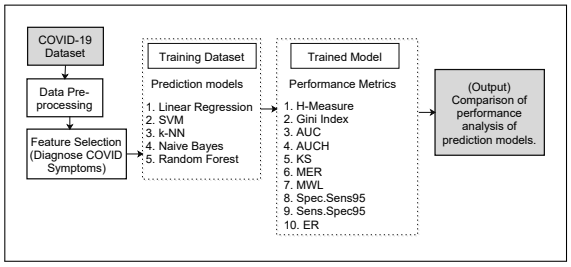


Figure 2: Workflow of the proposed CURE Scheme for the treatment of COVID patient.

symptoms of COVID-19 patients.
Problem is heightened with the unbalancing of data. In medical data the class imbalance problem is frequent which occurs with the dominancy of more cases of some classes over others. To handle the imbalanced dataset, several elucidations are appropriate at both algorithmic and data level. In this paper, the performances of 5 classifiers and regressions are compared on imbalanced dataset which is obtained while studying on the prediction of COVID-19. On the bases of attainment of these regression and classifiers, impact of SMOTE (Synthetic Minority Oversampling Technique) - an approach which deals with imbalanced dataset, is thoroughly evaluated.

With the comfort of the algorithms used in this method, k samples are finding out which are in proximity to the minority samples in minority classes and standard Euclidean distance method is used to attain this distance. With the number of cases in minority and majority classes, imbalanced dataset is taken. Based on the independent variable, the original dataset is partitioned into two sets – training set (80%) and test sets (20%) using stratified random sampling. By applying SMOTE technique, training set is over samples to find out the distribution of class suited best to the dataset and 8 training sets obtain among which 1 is original set other than 7 over sampled set having different rates.

4. Proposed CURE Scheme

The proposed CURE scheme uses wide range of methods and tools are used for prediction. With the combination of different models- SVM (Support Vector Machine), LR (Linear Regression), k-NN (k- Nearest Neighbors), Classification Naïve Bayes and R tool, a machine learning model is proposed for forecasting of COVID-19 infection rate. Collected Dataset is cleaned before further processing and is considered as first step in knowledge discovery in databases. For written characters classification problems this data cleansing process is applied using Machine Learning techniques. The process that implements methods to detect missing and incorrect data, error correction

and explore data bases is called data cleaning in which reassembling and disintegrating of data is involved. Data cleansing is practiced on numerous merged data bases in which appearance of duplicate records takes place. Four dimensional qualities are proposed which includes certainty, correctness, integrity and consistency.

Primary symptoms of this disease include loss of taste and smell, headache, fever, dizziness, tiredness and shortness of breath. Since seriousness, symptoms are classified into three categories i.e. mild, moderate, and severe. Mild symptoms possess fever, cough, headache. The frequency of seriousness is low at this stage. Then comes the moderate stage in which shortness of breath is the main symptom along with high fever and cough. In severe stage, the patient reach into critical situation and becomes profoundly serious. Respiratory problem is the main problem the patients must face. The virus mainly affects the lungs which damages alveoli responsible for supply of oxygen to all parts of body through blood vessels and RBCs, respectively. The virus damages the alveolus wall and results into its thickening due to which transfer of oxygen to RBCs lowers down which ultimately leads to hypoxia. Due to insufficient intake of oxygen, chances of organ failure remain high. Collected data is first trained and then tested using different models - SVM (Support Vector Machine), LR (Linear Regression), k-NN (k- Nearest Neighbors), Classification and Naïve Bayes. The explanation of these prediction methods are listed below.

4.1. Linear Regression (LR)

LR is the most usable statistical technique for predictive analysis in Machine Learning. Based on supervised learning, Linear regression is a Machine Learning algorithm which performs a regression task. LR prediction model use the given data points to obtain the optimal fit line to train the dataset. A simple equation of a line is $y = mx + c$, where y is a dependent variable, x is independent variable, and m, c are constant whose values are computed by using the calculus theories. Figure 3(a) shows an example of LR prediction model that consider the features as input and predict a continuous output as a result by obtaining a linear curve for a given problem. The output of LR model is computed by using the equation.

$$y = \mu_0 + \mu_1 x_1 + \epsilon, \tag{1}$$

where μ_0 represents y intercept, μ_1 represents slope, x_1 is the input value, ϵ represents error term, and y is the output value of the model. Initially at the start of the training, β is initialized randomly but we correct μ during the training specified to each feature such that the loss (deviation between the desired and predicted output) is minimized. The metric of loss is calculated by using mean squared error (MSE). The pros of using LR are easy, simple implementation, fast training, regularized to avoid over fitting, easily updated with new data using gradient descent. The disadvantages of LR model is that it performs poorly for non-linear relationships, not flexible to capture complex patterns, polynomials can be time consuming. However to generate a discrete output i.e., 0 or 1, the logistic regression (binary classification) model is used. Figure 3(b) shows an example of Logistic regression which calculates the aggregate sum of the input variables similar to LR model but it runs the output through non-linear sigmoidal function to generate the output.

$$y = \frac{1}{1 + e^{-x}}, \tag{2}$$

where x is the input value, y is the output value of the model, and e is exponential. LR prediction model can be implemented on Python.

4.2. Support Vector Machine method (SVM)

SVM is a supervised ML algorithm used for both classification and regression. An example of SVM classifier is shown in Figure 3(c) which is a representation of different classes in a decision plane or hyperplane in n-dimensional space. In this figure, support vector are the datapoints that are nearest to the hyperplane. These data points are divided into classes by using separating line (H_1, H_2, H_3). Here, a margin is defined as the gap or perpendicular distance from the line to the support vectors. The objective of SVM is to separate the datasets into classes to calculate maximum marginal hyperplane. Initially, SVM find hyperplanes iteratively that isolate the classes based on that SVM select the hyperplane that divides the classes in best way. SVM can perform efficiently on non-linear classification while performing linear classification. With dimensional spaces and the cases having number of dimensions greater than number of samples, it is extremely effective. SVM tranform the input vector to n-dimensional space known as a feature space (f) by using non-linear function then a linear function of linear regression is performed to space. It is implemented in Python by using SVM kernels. The types of SVM kernels are linear kernel, polynomial kernel, and raial bias function (RBF) kernel.

Linear Kernel: It is the dot product between two observations and the linear kernel function is defined by using the equation.

$$f(v, v_i) = sum(v * v_i), \tag{3}$$

where v, v_i are two vectors.

Polynomial Kernel: It discriminate curved or non-linear input space which is defined by using the equation.

$$f(v, v_i) = 1 + sum(v * v_i)^d, \tag{4}$$

where d is the degree of polynomial which is manually set in the learning algorithm.

Radial Bias Function (RBF) Kernel: It transform input space into multi-dimensional space which is defined by using the equation.

$$f(v, v_i) = exp(-\gamma * sum(v * v_i)^2), \tag{5}$$

where γ lies between 0 and 1 which is set manually and its default value is 0.1.

The steps to be followed in implementing SVM classifier for text classification are as follows: (i) import *svm* packages. (ii) load the input dataset. (iii) select features from the dataset. (iv) plot SVM boundaries with original data. (v) generate the values of regularization parameter. (vi) SVM classifier object are created by using kernel (linear, polynomial, RBF). (vii) text final output is the text classification. The advantage of using SVM classifiers are high accuracy with multi-dimensional space, stores very less memory and use a subset of training points. The disadvantage of SVM classifiers is that the performance of SVM does not scale for larger datasets due to high training time, and does not perform good with overlapping classes. Thus, decision tree are usually preferred over SVM for large datasets.

4.3. k-NN (k-Nearest Neighbors)

k-nearest neighbors (k-NN) algorithm is supervised ML technique which is generally used for classification problems. It can be used for both classification as well as regression. k-NN method classifies documents based on resemblance measurements which estimating the factors such as distance and proximity, the similarity between two data points is quantified and classified based on nearest neighbors of each data point. Figure 3(d) shows an example of k-NN model which assumes the closeness of two data points (similar data points). k-NN works on the principle of feature similarity in order to predict the values of new datapoints. Thus, the new data point allocates a value based on the proximity as it matches the data points in the training set. The steps involved in k-NN algorithm are as follows: (i) Load the training and testing dataset. (ii) Select the value of k (integer) i.e. the closest data points. (iii) For each point in the test data, compute the distance between test data and each row of training data with the help of Euclidean or Hamming distance and sort the distance values in ascending order. (iv) Select the top k rows from the sorted array. Next, allocate a class to the test point based on most frequent class of these rows. (v) final output.

k-NN algorithm can be implemented in Python by using the following approach: (i) importing necessary python packages, (ii) download the Kaggle COVID-19 dataset, (iii) assign column names to the dataset, (iv) read dataset to pandas dataframe, (v) perform data preprocessing, (vi) split the data into train and test dataset (60% training data and 40% of testing data), (vii) perform data scaling, (viii) train the model using K-nearest neighbors classifier class of sklearn, (ix) obtain prediction, (x) output results- confusion matrix, classification report, and accuracy. The benefits of k-NN algorithms are simple, useful for nonlinear data, high accuracy. The limitations of k-NN algorithm is that it is costly algorithm as it stores all the training data. In addition, it requires more memory storage, and prediction is slow in case of large dataset.

4.4. Naïve Bayes

Naïve Bayes is a classification method based on bayes theorem which works on the principle of strong assumptions of conditional independence that the existence of a feature in a class is independent to the existence of any other feature in the same class. Let us consider an example of smart 4K TV, a smart TV is considered into the category of smart if covers the features such as Internet connection, high definition, bluetooth, USB ports, HDMI connectivity, support multiple applications. However, these are dependent on each other but individual feature contribute independently to the probability of the smart 4K TV is a smart TV. Naïve Bayes is a highly scalable algorithm that can be certainly train on small dataset. Figure 3(e) shows an example of Naïve Bayes model that classify the data points based on posterior probability of class into three different classes i.e., classifier 1 (red data points), classifier 2 (orange data points), and classifier 3 (blue data points). The expression of Naïve Bayes algorithm based on bayes theorem is defined as follows.

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)}, \tag{6}$$

where $P(A|B)$ indicates the posterior probability of class, $P(B|A)$ indicates likelihood probability of predictor given class, while P(A) refers to prior probability of class, and P(B) refers to marginal probability or prior probability of predictor. For building the prediction model using Naïve Bayes classifier, the model is categorized into three types: (i) Gaussian Naïve Bayes (GNB), (ii) Bernoulli Naïve Bayes (BNB), and (iii) Multinomial Naïve Bayes (MNB). Python library, Scikit learn is the most useful library that helps us to build a Naïve Bayes model in Python. We have the following three types of Naïve Bayes model under Scikit learn Python library.

GNB Classifier: It is based on the consideration that the data from each label is drawn from a simple Gaussian distribution. *MNB Classifier*: Here, the features are considered to be drawn from a simple Multinomial distribution which is most suitable for the features that represents discrete counts. *BNB classifier*: BNB consider

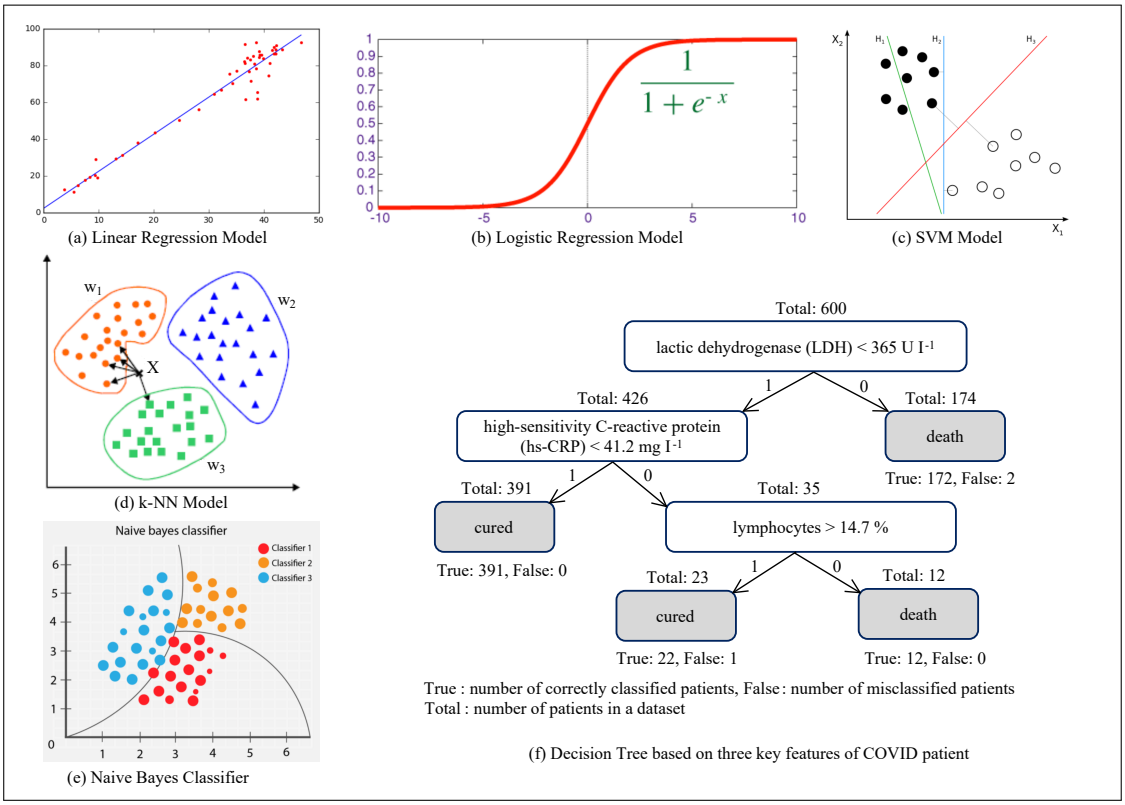


Figure 3: Prediction models: (a) Linear Regression model, (b) Logistic Regression, (c) SVM model, (d) k-NN classifier, (e) Naive Bayer Classifier, and (f) Decision Tree Induction model.

the features to be binary (0s and 1s). For example, text classification with 'bag of words' model.

The steps involved in implementing the GNB classifier in Python are as follows: (i) import the GNB packages under Scikit learn Python library. (ii) obtain blobs of points by using *make_blobs*() function of Scikit with Gaussian distribution. (iii) for GNB model, we need to import GaussianNB and make its object. (iv) perform prediction after obtaining some new data. (v) plot new data to find its boundaries. (vi) using line of codes compute posterior probabilities of labels. (vii) output array. The benefits of using Naïve Bayes classifier are fast and easy implementation, less training data, converge faster than discriminative models like logistic regression, and suitable for both continuous as well as discrete data. The limitations of Naïve Bayes classifier are zero frequency in case a variable is assigned with a category but not being observed in training data set, then Naïve Bayes classifier set a zero probability and does not give a prediction, feature independence as in real life application it is difficult to have a set of features which are completely independent of each other. The applications of Naïve Bayes classifier are real-time prediction, multi-class prediction, text classification.

4.5. Decision Tree Induction Classifier

is a simple, easy understandable non parametric classifier which is based on flexible decision tree algorithm. It can perform both classification and regression with the help of algorithms used to formulate this model from the original dataset, unpremeditated selection of training data is accomplished. The steps to be involved in the working of decision tree algorithm are as follows. (i) selection of random samples from a given dataset. (ii) construct a decision tree for every sample and compute the prediction result from every decision tree. (iii) voting is done for every predicted result. (iv) choose the most voted prediction result as the output of the prediction algorithm.

The decision tree is implemented in Python by using the following approaches. (i) importing necessary Python packages, (ii) download the Kaggle dataset, (iii) assign column names to the dataset, (iv) read dataset to

pandas dataframe, (v) perform data pre-processing by using script lines, (vi) divide the data into train and test split (suppose, split the dataset into 70% training data and 30% of testing data), (vii) train the decision tree model with the help of RandomForest Classifier class of sklearn, (viii) generate prediction by using script, and (ix) final output is the confusion matrix and classification Report. Figure 3(f) shows an example of the rule based on three key features disease of COVID-19 patient dataset i.e., lactic dehydrogenase (LDH), high-sensitivity C-reactive protein (hs-CRP), and lymphocytes. The decision tree was obtained by a random split of total 600 patients at the root of the forest which is the number of patients to training and validation datasets, whereas the leaf node returns the outcome as the number of cured and death patients.

The key benefits of using decision tree model are it is suitable for large range of datasets, overcomes the problem of overfitting by merging the results of different decision trees, flexible and possess very high accuracy, scaling of data is not required. The limitations of decision tree algorithm are high complexity, harder and time-consuming in comparison to other prediction models, and requires more computational resources.

5. Prediction Models Performance Evaluation

The performance of prediction models can be assessed using a variety of metrics listed as follows:
(1) H-measure, (2) Gini-Index, (3) Area Under Curve (AUC), (4) Area Under the convex Hull of the ROC Curve (AUCH), (5) Kolmogorov-Smirnoff statistic (KS), (6) Minimum Error Rate (MER), (7) Minimum Cost Weighted Error Rate (MWL), (8) Specificity when Sensitivity is held fixed at 95% (Spec.Sens95), (9) Sensitivity when Specificity is held fixed at 95% (Sens.Spec95), and (10) Error Rate (ER).

H-measure: H-measure is an important measure of classification performance that measures the accuracy of the model. The primary statistics of interest are the so-called mis-classification counts, i.e., the number of False Negatives (FN) and False Positives (FP). There are four scenarios in prediction modeling. (i) *True positives (TP)*: In case of true positives (TP), actuals are positives and are predicted as positives. (ii) *False positives (FP)*: In case of false positives (FP), actuals are negatives and are predicted as positives. (iii) *False negatives (FN)*: In case of false negatives (FN), actuals are positives and are predicted as negatives. (iv) *True negatives (TN)*: In case of true negatives, actuals are negatives and are predicted as positives. An example of false positive is occurrences where a disease is mistakenly diagnosed, and an example of false negatives is occurrences where the presence of a disease is missed.

Accuracy (A_C): The accuracy in a given datasets with data points (TP + TN) is the ratio of total correct predictions by the classifier to the total data points. The value of A_C lies between 0 and 1.

$$A_c = \frac{(TP + TN)}{(TP + TN + FP + FN)} * 100. \tag{7}$$

Area Under Curve (AUC): AUC measures the quality of models used for classification problems. It is a metric for binary calculation which calculates the area under the curve of a given performance measure whose value lies between 0.5 and 1.

Gini-Index (GI): GI is used for comparison of models which is the difference of a distribution is calculated by using Gini-coefficient and its values lies between 0 and 1.

$$GI = (2 * AUC - 1). \tag{8}$$

KS: KS chart measures performance of classification models. More accurately, K-S is a measure of the degree of separation between the positive and negative distributions.

$$KS = |cumulative\% + ve - cumulative\% - ve| \tag{9}$$

Error Rate (ER): ER is defined as the ratio of the total mis-classification count (FP + FN) divided by the number of samples.

$$ER = \frac{FP + FN}{n} = \frac{FP + FN}{FN + FP + TN + TP}. \tag{10}$$

MER: It represents the Minimum Error Rate. Here threshold value act as a free parameter.

MWL: It is related to the KS statistics. Here, cost guides the threshold value in this measure.

Specificity and Sensitivity: True Positive Rate (TPR) or Sensitivity (Sens), and True Negative Rate (TNR), or called Specificity (Spec.)

$$Sens = \frac{TP}{TP + FN}, \quad Spec. = \frac{TN}{TN + FP}. \tag{11}$$

Figure 7 computes the H measure by using five classifiers. The normalised cost is computed on X-axis. Let us assume that $c \in [0, 1]$ denote the cost of misclassifying a class 0 object as class 1 (FP), and $1 - c$ represensts the cost of misclassifying a class 1 object as class 0 (FN). This asymmetry can be seen to underlie the KS statistic, which is a simple linear transformation of the MWL when $c = \pi_1, 1 - c = \pi_0$. The severity ratio (SR) is defined as the ratio between the two costs, where SR = 1 that represents the symmetric costs.

$$SR = \frac{c}{1 - c}, \quad Normalised\ Cost = \frac{SR}{1 + SR}. \tag{12}$$

where, the Y-axis represents the weighted cost. The H-measure is computed for all the five classifiers and finally, the mean value of Severity Ratio (SR) is 1.12. We pre-process the data to make the experimental data more efficient and remove redundancy.

5.1. Dataset

To validate the performance of the proposed CURE scheme, the dataset is being collected from the Kaggle COVID-19 patient pre-condition dataset [16]. The Kaggle dataset is provided by the Johns Hopkins University through Github repository which contains the real-time updated record of the total active cases, death cases, recovered cases of the COVID-19 pandemic. In the modern time of advancement in technology and all rounded progress, to make human beings as well as the medical science more mentally and physically prepared and attentive, such type of health issues or threatening disease will prove very helpful and challenging. As per the reports disclosed by World Health Organization (WHO), the health curve (infectious cases and cured cases) remains changing abruptly every day, it becomes burdensome for the medical and other departments engaged in this kind act to serve the world medical facilities and other necessary things to make an estimate of total requirements of the health related equipment's and resources. It becomes very helpful for the entire medical department and other concerned authorities if the corona patients be accommodated all the resources which will prove a blessing for them to fight the lethal disease. In this context, the data collected contains 23 features of 5,66,603 patients.

5.2. Results and Discussion

The implementation of the experimental results are performed in Python. The results are computed based on finding the missing values, heatmap function, feature selection, and comparison of the machine learning models. The discussion related to the results are summarized below.

5.2.1. Missing Values

The initial step is to find the missing values in the Kaggle dataset [16] and plot these missing values. Figure 4 visualized the histogram of the missing values in COVID dataset. As a substitute to these, we computed the mean and replaced the missing value with its mean. The default input is a numeric array with levels 0 and 1, where the minimum value is 0 and the maximum value is 1.

5.2.2. Heatmap Representation

As the Kaggle COVID-19 dataset, we collected does not contain any missing or redundant value, so we represented the complete dataset in Figure 5. It is drawn using the heatmap function of python and capable to presenting the diagrammatically view of the dataset. The parameters of the COVID patients are considered on the X and Y axis.

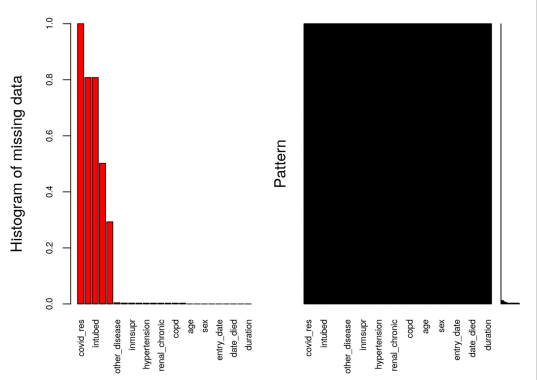


Figure 4: Histogram of missing values.

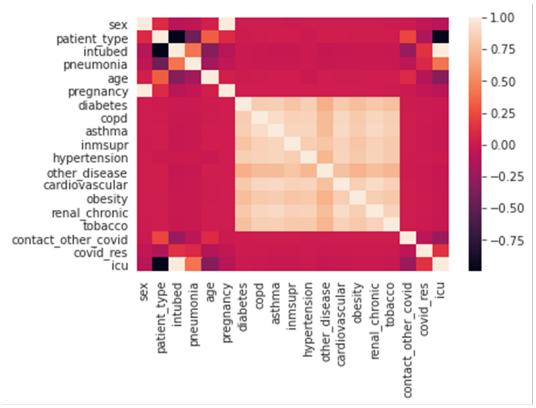


Figure 5: Heatmap of all the features of COVID-19 dataset.

5.2.3. Feature selection

As shown in Figure 6, We have selected 10 features among 23 features from the COVID patient dataset. This selection is being made by analyzing the features after computing the feature importance score in the form of Gini-index through the implementation of decision tree method.

5.2.4. Machine Learning Model

As discussed in the CURE scheme, the machine models are being used on the pre-processed data. However,

Table 1
Comparison of the performance analysis of various ML prediction models.

Models	H	Gini Index	AUC	AUCH	KS	MER	MWL	Spec.Sens95	Sens.Spec95	ER
SVM	0.687	0.802	0.901	0.901	0.802	0.099	0.098	0.443	0.447	0.46
LR	0.672	0.791	0.896	0.896	0.791	0.104	0.104	0.421	0.506	0.482
k-NN	0.655	0.781	0.891	0.891	0.781	0.109	0.109	0.478	0.49	0.469
Naïve Bayes	0.632	0.765	0.882	0.882	0.765	0.117	0.117	0.494	0.52	0.47
Random Forest	0.675	0.794	0.897	0.897	0.794	0.103	0.103	0.448	0.475	0.476

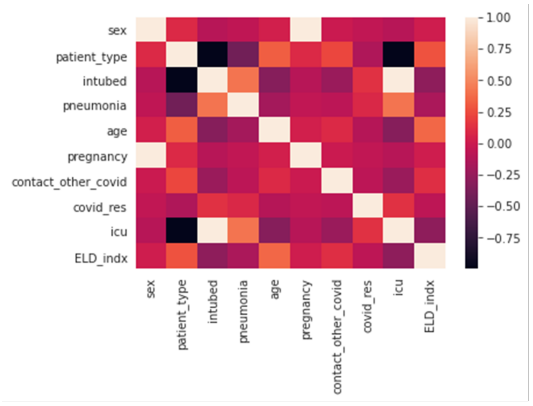


Figure 6: Representation of selected values of dataset.

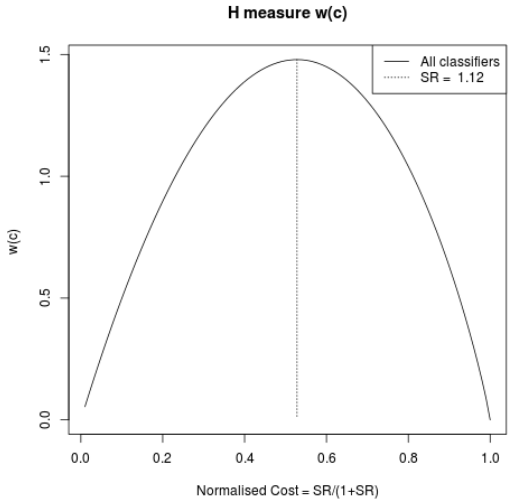


Figure 7: H-measure of ensembled model.

there are different methods to enhance the performance of the prediction models which dependent on the technique involved. One such technique is to construct the ensemble models in order to obtain a score for a particular outcome, we can start integrating them to produce ensemble scores. Figure 7 computes H-measure of ensembled model which can be used to improve the area under the curve for these models even further. Let us assume, a decision tree classifier and a logistic regression model, both predicting standard risks. A new score can be calculated as the average of these two classifiers and then assess it as a further model. Usually the area under the curve improves for these ensemble models.

After experimentation, the results are computed in Table 1.

6. Conclusion

In this paper, a CURE scheme is proposed based on machine learning prediction models for the treatment of the COVID patients through remote e-heathcare. The performance analysis of the proposed scheme is evaluated on Python platform which is tested using Kaggle dataset from Johns Hopkins University on COVID-19 patient pre-condition. Then, the features are extracted from the datasets of the COVID patient for diagnosing the symptoms of the coronavirus. Next, the collected data is first trained and then tested using different machine learning prediction models (such as SVM, LR, k-NN, , and Naive Bayes) that classify the features of the COVID patient for forecasting of infection rate. Finally, the performance of the prediction models are assessed using a variety of metrics listed as follows: (1) H-measure, (2) Gini Index, (3) Area Under Curve (AUC), AUCH, KS, Minimum Error Rate (MER), Minimum Cost Weighted Error Rate (MWL), Spec.Sens95, Sens.Spec95, Error Rate (ER). The performance evaluation shows that the CURE scheme outperforms the existing approach which deals with imbalanced dataset.

In future, we will ensure the secrecy of the corona virus data as the patients sensitive credentials can be leaked during data transmission through wireless channels (Internet).

References

[1] Punn, Narinder Singh, Sanjay Kumar Sonbhadra, and Sonali Agarwal. "COVID-19 Epidemic Analysis using Machine Learning and Deep

Learning Algorithms" medRxiv (2020), doi: https://doi.org/10.1101/2020.04.08.20057679.

[2] Jamshidi, M., Lalbakhsh, A., Talla, J., Peroutka, Z., Hadjilooei, F., Lalbakhsh, P., Jamshidi, M., La Spada, L., Mirmozafari, M., Dehghani, M. and Sabet, A. "Artificial Intelligence and COVID-19: Deep Learning Approaches for Diagnosis and Treatment" IEEE Access, vol. 8, pp.109581-109595, Jun. 2020.

[3] Yan, Li, Hai-Tao Zhang, Yang Xiao, Maolin Wang, Chuan Sun, Jing Liang, Shusheng Li et al. "Prediction of survival for severe Covid-19 patients with three clinical features: development of a machine learning-based prognostic model with clinical data in Wuhan" medRxiv (2020).

[4] "COVID-19 Worldwide Dashboard - WHO Live World Statistics" Online available: https://covid19.who.int/, accessed on 31 July, 2020.

[5] Rehman, Suriya, Tariq Majeed, Mohammad Azam Ansari, Uzma Ali, Hussein Sabit, and Ebtesam A. Al-Suhaimi. "Current scenario of COVID-19 in pediatric age group and physiology of immune and thymus response." Saudi Journal of Biological Sciences (2020).

[6] Nguyen, Thanh Thi. "Artificial intelligence in the battle against coronavirus (COVID-19): a survey and future research directions." Preprint, DOI 10 (2020).

[7] Zhang, Jian, and Yiming Yang. "Robustness of regularized linear classification methods in text categorization." In Proceedings of the 26th annual international ACM SIGIR conference on Research and development in informaion retrieval, pp. 190-197. 2003.

[8] Tan, Yuxuan. "An improved KNN text classification algorithm based on K-medoids and rough set." In 2018 10th International Conference on Intelligent Human-Machine Systems and Cybernetics (IHMSC), vol. 1, pp. 109-113. IEEE, 2018.

[9] Samuel, Jim, G. G. Ali, Md Rahman, Ek Esawi, and Yana Samuel. "Covid-19 public sentiment insights and machine learning for tweets classification." Information, vol. 11, no. 6 Jun. (2020).

[10] Pinter, Gergo, Imre Felde, Amir Mosavi, Pedram Ghamisi, and Richard Gloaguen. "COVID-19 Pandemic Prediction for Hungary; a Hybrid Machine Learning Approach." Mathematics, vol. 8, no. 6 (2020):890.

[11] Yan, Li, Hai-Tao Zhang, Yang Xiao, Maolin Wang, Chuan Sun, Jing Liang, Shusheng Li et al. "Prediction of criticality in patients with severe Covid-19 infection using three clinical features: a machine learning-based prognostic model with clinical data in Wuhan." MedRxiv (2020).

[12] Lin, Leesa, Rachel F. McCloud, Cabral A. Bigman, and Kasisomayajula Viswanath. "Tuning in and catching on? Examining the relationship between pandemic communication and awareness and knowledge of MERS in the USA." Journal of Public Health 39, no. 2 (2017): 282-289.

[13] Hamzah, FA Binti, C. Lau, H. Nazri, D. V. Ligot, G. Lee, and C. L. Tan. "CoronaTracker: worldwide COVID-19 outbreak data analysis and prediction." Bull World Health Organ 1 (2020): 32.

[14] Jia, Lin, Kewen Li, Yu Jiang, and Xin Guo. "Prediction and analysis of Coronavirus Disease 2019." arXiv preprint arXiv:2003.05447 (2020).

[15] Tuli, Shreshth, Shikhar Tuli, Rakesh Tuli, and Sukhpal Singh Gill. "Predicting the Growth and Trend of COVID-19 Pandemic using Machine Learning and Cloud Computing." Internet of Things (2020): 100222.

[16] "COVID-19 patient pre-condition dataset", 2020. Online Available: https://www.kaggle.com/tanmoyx/covid19-patient-precondition-dataset/notebooks