PERCENTAGE-PROBABILITY-EVALUATION-FOR-COVID-19-USING-MACHINE-LEARNING

A PREPRINT

Aman Kumar
Department of Mathematics
IISER Bhopal
amank17@iiserb.ac.in

Pratik Ingle
Department of EECS
IISER Bhopal
pratik17@iiserb.ac.in

May 3, 2021

ABSTRACT

Coronavirus disease (COVID-19) is a highly contagious disease and has proved to be a disastrous threat to humanity. It is caused by infection from coronavirus. It started on Nov. 17, 2019 in Wuhan, China and now is transformed into a pandemic. First person to have contracted COVID-19 is a 55-year-old individual from Hubei province in China on Nov. 17, 2019. Researchers are consistently working on the vaccine for this disease and as for now proper testing and isolation is the only key against this disease. As of this time, globally there are 130 million confirmed cases of COVID-19. Due to the massive number of possible patients, it is difficult to test each and every person and even a single positive case can lead to the spreading of the virus at a very huge scale. Previously the approach for allotment of the test was based on the travel history of the person or if the person is suffering from COVID-19 symptoms or if he/she was in contact with an infected person but this is not applicable in the current scenario. Also, the limitations in the number of test kits are debarring us from quantitative testing and hence we need to improve qualitative aspects while choosing a person for testing. Thus we need some more strong decisive aspects for possible corona testing of a person. In this study, we have predicted the percent probability of a person for being infected with coronavirus with the help of some basic clinical features of their body by applying various machine learning algorithms. This prediction can help us to prioritize our testing to the most prone cases of the corona. We can further test less prone cases by dividing them into small groups and mixing their blood samples together. Thus if a sample tests negative then the whole group of people tests negative and we need not perform any further testing. Thus our model for probability prediction is highly helpful in targeting a better audience for testing as well as reducing the number of possible tests.

Keywords Machine learning · Clinical features · COVID-19 · Blood test.

1 Introduction

Early identification of COVID-19 could be a milestone to fight against this highly contagious disease. It can be spread easily by being in contact with the infected person. And the same is the reason for the recent outbreak of the disease all over the world. It started in China and now has turned into a pandemic killing more than 2.84M people globally. Testing each and every person is not something that can be done easily. Therefore it is logical to find another way to predict whether a person is infected or not. These predictions can also be done for COVID-19. Before the prediction, it is important to pick out the right clinical features to work with or distinguish COVID-19 from other diseases diagnosed

from that feature. There has been a study distinguishing COVID-19 and viral pneumonia from chest.These predictions are also being taken a step further and used for prediction of survival of a patient who is being found COVID-19 positive or trying to implement a strategy for test kit allocation. Scholars have also tried to find which chemical or medicine is useful for lowering the risk such as study about the effectiveness of hydroxychloroquine and azithromycin in recovery from COVID -19. Our model works on the prediction of percent probability of a person being infected by COVID-19 and an innovative way of performing the test in an efficient way such that not only all the people are tested but also the use of the test kits are minimized.

2 Dataset

This dataset contains anonymized data from "patients seen at the Hospital Israelita Albert Einstein in Brazil".The samples were collected to carry out the "SARS-CoV-2 RT-PCR" and additional laboratory tests in the hospital.

These data were anonymized in accordance with proper international enactment and guidance. The clinical data were normalized such that their mean is zero and has a unit standard deviation. We downloaded the original dataset from "Kaggle" which is an online community for data science practitioners. The original dataset consisted of 5644 rows of patients and 111 columns which consisted of clinical features (eg.Hematocrit, Hemoglobin, Platelets etc)including patient ID and test results. Before applying our model to the dataset we cleaned the dataset by removing rows and columns with very less or no information. Finally, we applied our model with 602 rows of patients and 17 columns which included patient ID and test results.

2.1 Cleaning of Data

Firstly, we deal with missing values by marking them as 'NaN' and these values thus get debarred from any operation and instead mean of that column's data gets filled in their place. Now after dividing the dataset into training and test set, we did feature scaling so as to standardize the independent features in a definite range to handle the large variation in data.

3 Applying Machine learning models

3.1 Naive Bayes Analysis

Here we represent each person to be tested (t) as a set of clinical features associated with them. Let x and y be the set of positive and negative cases respectively. Let t be a person to be tested then $t = f_1 + f_2 + ... + f_n$ where $f_1, f_2, ..., f_n$, etc are different clinical features of a person to be tested. Now calculation of probability for t to be tested positive

$$p\left(\frac{x}{t}\right) = p\left(\frac{t}{x}\right) * p(t)/p(x)$$

Here 'Naive' condition is assumed to be true i.e.each clinical feature of t is independent of other clinical features in it. Thus,

$$p(t) = p(f_1) * p(f_2) * ..... * p(f_n)$$

Now implementing Bayes Theorem,

$$p\left(\frac{t}{x}\right) = p\left(\frac{f_1}{x}\right) * p\left(\frac{f_2}{x}\right) * ..... * \left(\frac{f_n}{x}\right)$$

Hence our model is ready for prediction.

3.2 Linear Regression Analysis

Multiple Linear Regression is one of the important regression algorithms which models the linear relationship between a single dependent continuous variable and more than one independent variable.

In Multiple Linear Regression, the target variable(t) is a linear combination of multiple predictor variables $f_1, f_2, f_3, ..., f_n$. Since it is an enhancement of Simple Linear Regression, so the same is applied for the multiple linear regression equation, the equation becomes:

$$t = b_0 + b_1 * f_1 + b_2 * f_2 + ... + b_n * f_n$$

Where,

t = Output of test result

$b_0, b_1, b_2, b3, ..., b_n$ = Coefficients of the model.

$f_1, f_2, f_3, ..., f_n$ = clinical feature

Assumptions for Multiple Linear Regression:

- A linear relationship should exist between the Target and predictor variables.
- The regression residuals must be normally distributed.
- MLR assumes little or no multicollinearity (correlation between the independent variable) in data

3.3 Logistic Regression Analysis

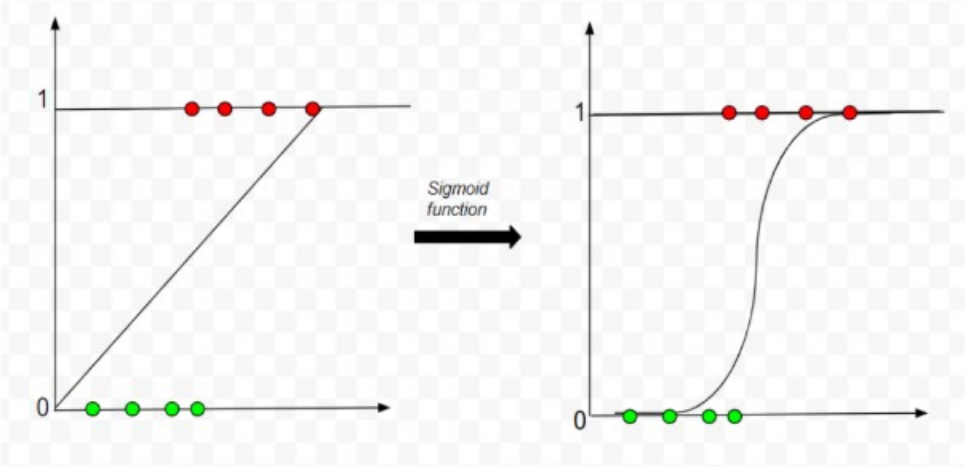
In normal regression, we would have used a function to predict the probability of a per son to be tested positively. But, in logistic regression, the idea is the same but with that function, we combine another function named sigmoid or logistic function. All the clinical features act as the independent variables and are used to predict the dependent variable which in our case is whether a person is infected with COVID-19 or not.

$$y = a_0 + a_1 x$$

$$Sigmoid function \Rightarrow \quad p = \frac{1}{1 + e^y}$$

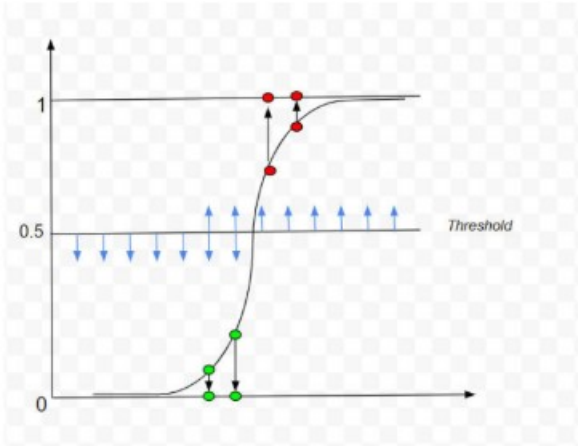
$$\ln\left(\frac{p}{1-p}\right) = a_0 + a_1 x$$

Figure 1: Change by Sigmoid Function



As we can see figure 1 shows the change that occurs after applying the logistic function on the data set and figure 2 shows how the probability threshold works in predicting the percent probability.

Figure 2: Logistic regression with threshold



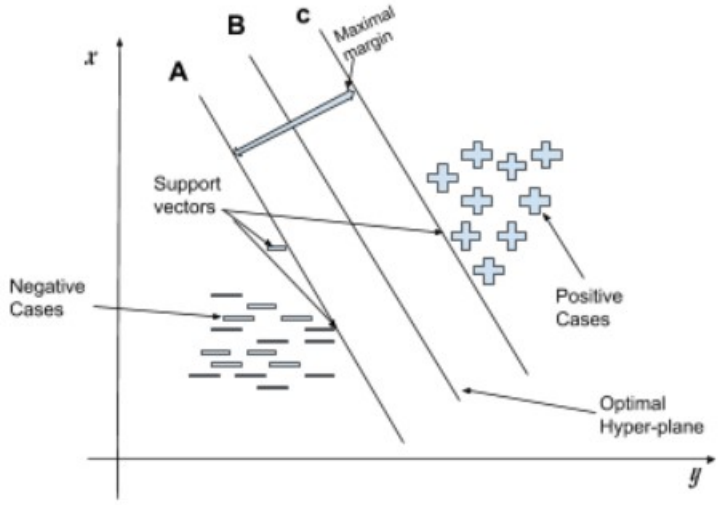
3.4 Support Vector Machine(SVM) Analysis

For the prediction of probability through SVM we need to plot the data of each patient in a m dimensional space where m is the number of clinical features used for prediction. This classifier works by searching the optimal hyper-plane so as to perfectly differentiate the plotted classes in m dimensional space. Each coordinate is represented as a vector. Out of many hyper-planes that one is selected which precisely differentiates between positive and negative cases in accordance with a given probability threshold. The equation for separating hyper-plane,

$$w^T \cdot x^{(i)} + b = 0$$

Where $w =$ Normal to the hyper-plane, $x^{(i)} = i^t$ patient in the training set on which model is to be trained. Here we choose the hyperplane with a maximum margin so as to minimize the possibility of wrong test detection.

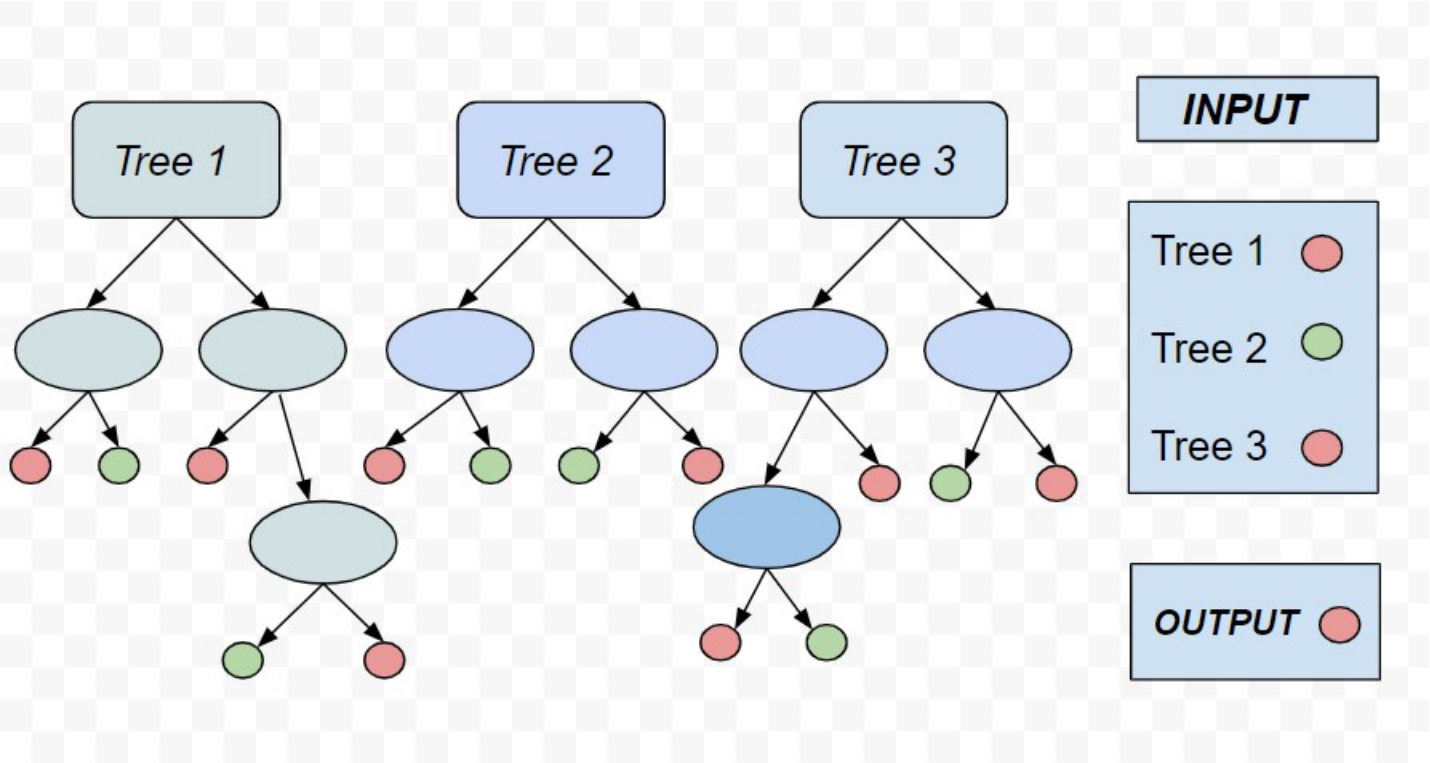
Figure 3: Support Vector Machine Classification



3.5 Random Forest Analysis

Random Forest, as the name suggests, consists of many decision trees. It builds a number of decision trees on the randomly selected data sample. The number of decision trees can be set in accordance with the dataset we are using for our model. Then it gets predictions from each tree and by means of majority voting, it selects the decision which gets the majority vote.

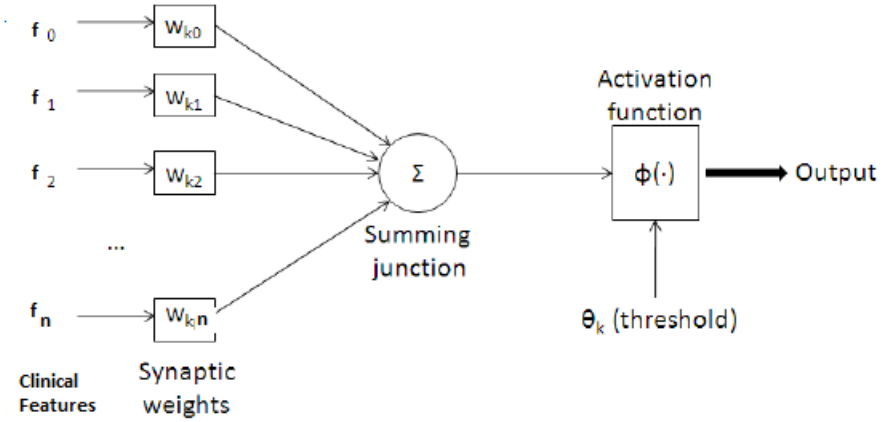
Figure 4: Random Forest Classification



3.6 Neural Network Analysis

An ANN has hundreds or thousands of artificial neurons called processing units, which are interconnected to each other in terms of layers. each processing unit takes input data and depending upon requirement it gives output mimicking function for given data. The input receive various forms and structures of information based on an internal weighting system, and the neural network attempts to learn about the information presented to produce one output. Artificial Neural Network primarily consists of three layers: Input layer, Hidden layer and Output layer. Input layer take input data(same format or several different formats). The hidden layer presents in-between input and output layers. It performs all the calculations to find hidden features and patterns. finally the input goes through a series of transformations using the hidden layer, which finally results in output. The ANN takes input and computes the weighted sum of the inputs and includes a bias. This computation is represented in the form of a transfer function.

Figure 5: Mathematical-model-of-an-artificial-neural-network-ANN



$$t = \sum_{i=1}^{n} W_i f_i + b$$

t = Output of test result

$b_0, b_1, b_2, b3, ..., b_n$ = Coefficients of the model.

$f_1, f_2, f_3, ..., f_n$ = clinical feature

3.7 Unsupervised Learning Analysis

Unsupervised Learning is a machine learning technique in which the users do not need to supervise the model. Instead, it allows the model to work on its own to discover patterns and information that was previously undetected. It mainly deals with the unlabelled data.Unsupervised Learning Algorithms allow users to perform more complex processing tasks compared to supervised learning. Although, unsupervised learning can be more unpredictable compared with other natural learning methods. Unsupervised learning algorithms include clustering, anomaly detection, neural networks, etc.

Advantages of Unsupervised Learning

- Unsupervised learning is used for more complex tasks as compared to supervised learning because, in unsupervised learning, we don't have labeled input data.
- Unsupervised learning is preferable as it is easy to get unlabeled data in comparison to labeled data.

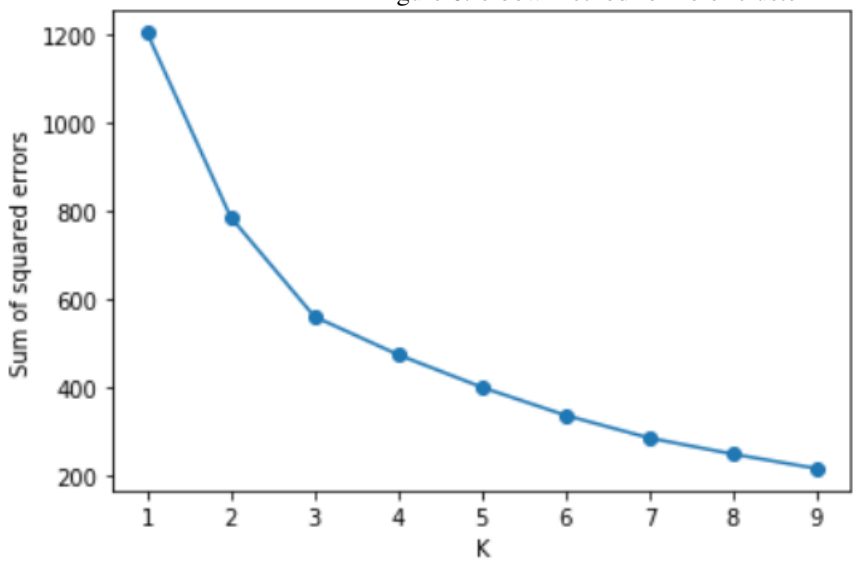
Disadvantages of Unsupervised Learning

- Unsupervised learning is intrinsically more difficult than supervised learning as it does not have corresponding output.
- The result of the unsupervised learning algorithm might be less accurate as input data is not labeled, and algorithms do not know the exact output in advance.

K-means: First, "K" refers to the number of clusters you want. That is, K = n means n number of clusters to be identified. Then there's something called "centroid", which is an imaginary/artificial data point (an average of data points) around which each cluster of data is partitioned. So K = 2 means that the algorithm will partition the observations (data) into 2 clusters such that the distances between the centroids and observations are minimized.

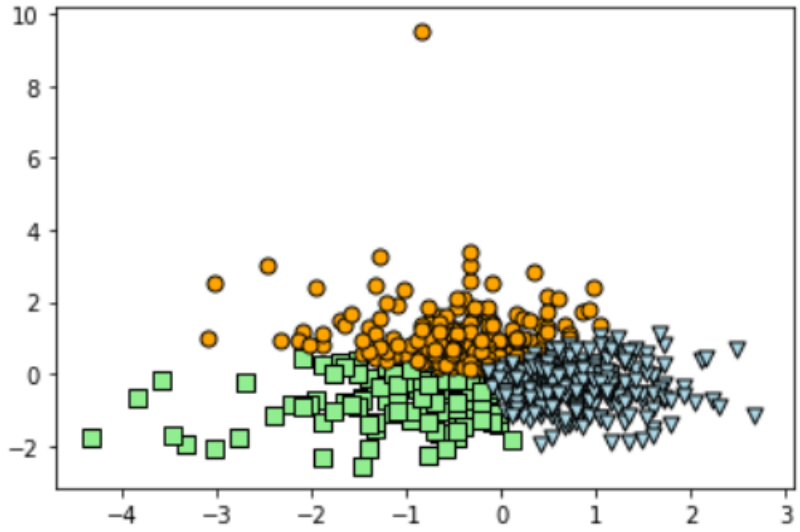
Determine number of clusters In K-means algorithm you need to define the number of clusters you want. The so-called "elbow method" can help determine that by minimizing the sum of squared errors.

Figure 6: elbow method for no of cluster



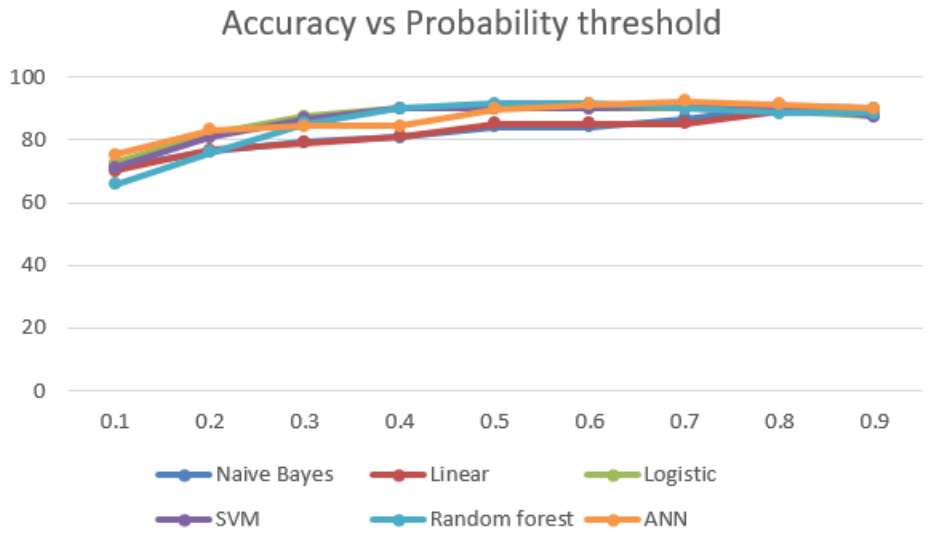
Model implementation Once you have made a determination on the only required parameter in the previous step, you are good to fit the model, visualize the number of clusters in a two-dimensional plot and do further analysis to answer the research question you are looking for.

Figure 7: Clusters



3.8 Comparative Analysis of the various methods used

Our model predicts the percent probability of whether a patient is suffering from COVID-19 or not. This percent probability p can further help us to reduce the number of tests that are to be performed. The purpose of using percent probability rather than a binary output is that if binary output gives even a single False Negative can lead to damage on a very large scale as it all started with one patient and now has turned into a pandemic. On the other hand with the help of percent probability, we never rule out the possibility of False Negative so there is a surety of no negative impact on practical use.



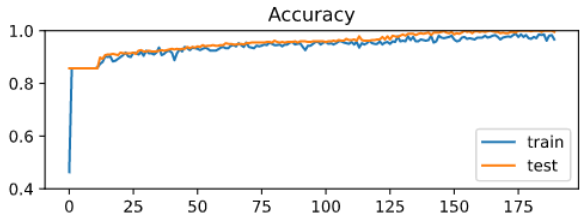


Figure 8: ANN Accuracy

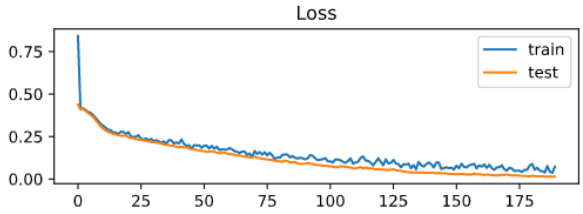


Figure 9: ANN Loss

The above graph shows the change of accuracy with respect to different thresholds of probability for different machine learning classifier algorithms. As it is evident from the above graph that with the increase in the probability threshold, the accuracy increases and reaches a maximum value and then starts decreasing. This pattern occurs because when the threshold is very low the value of False Positives is very high in the prediction which leads to low accuracy whereas when the threshold increases the value of False Positive decreases and accuracy increases. But as we go on increasing the probability threshold, the value of False Negative starts increasing, near the maxima there is a balance between the two values and the accuracy is highest but after that False Negative increases rapidly thus decreasing the accuracy at that point.

Figure 10: Accuracy Table(NB,SVM,RF,LR,linear,ann)

Theshold	Naive Bayes	Linear	Logistic	SVM	Random forest	ANN
0.1	71.07	70.11	72.72	71.07	65.8	75.2
0.2	76.3	76.9	81.81	80.99	76.03	83.3
0.3	79.33	79.08	87.6	86.77	85.12	84.54
0.4	80.99	81.12	90.08	90.08	90.08	84.54
0.5	84.29	85.2	90.08	90.08	91.73	89.7
0.6	84.29	85.2	90.08	90.08	91.73	91.3
0.7	86.77	85.2	90.08	90.9	90.08	92.3
0.8	90.9	89.2	89.25	90.9	88.43	91.3
0.9	90.08	89.2	87.6	87.6	88.43	90

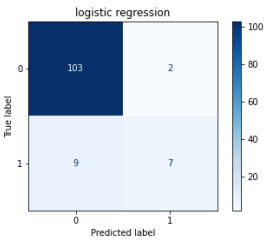


Figure 11: Logistic Regression

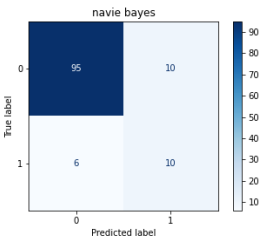


Figure 12: Naive Bayes

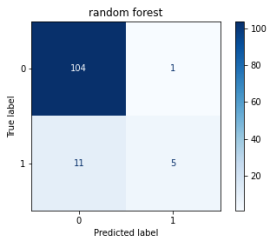


Figure 13: Random Forest

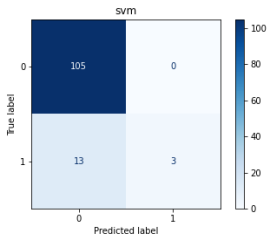


Figure 14: SVM

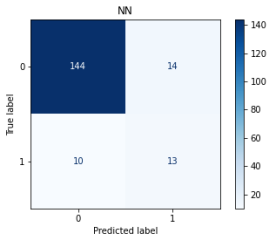


Figure 15: Neural Network

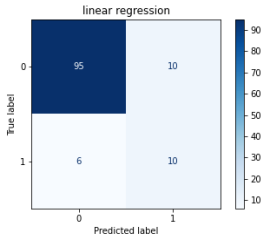


Figure 16: Linear Regression

4 Discussion

A machine learning algorithm in general works on a default value of probability threshold for classification. Thus it can only classify test data but cannot predict the percent probability. We have modified the probability threshold such that our model can not only classify the data but it can also predict the percent probability.

For data to be tested, we first run it on our model such that probability threshold > 0.9 and those who result positive after this are removed from the test data. Thus we have removed those people who are having the probability of more than 90 percentage of being infected with Coronavirus. Now we are left with data of people whose percent probability of being infected is less than or equal to 90 percentage in test data. Now we will again run the test data on our model by keeping the probability threshold > 0.8 and so on till the probability threshold > 0.1,so that we could segregate the cases on the basis of different probability thresholds.

Hence we got the cases which are most and least critical. Now we can perform the testing for most critical cases first and for the less critical cases we could divide them into small groups and perform testing by mixing the blood sample of all the people in that group and if that sample tests negative then the whole group tests negative and we need not perform any further testing. If this sample tests positive then we divide this into two subgroups and then again perform the testing for these two subgroups in a similar manner and we keep on doing this till we find the positive case. It is like performing Binary Search for finding the positive case from that small group. With the proper application of machine learning algorithms like Naive Bayes, SVM, linear regression , Logistic Regression ,Random Forest, ann and unsupervised learning we have proved our model a successful one.

Our study is not only limited to the present study there can be further studies on our model so that we could increase our accuracy, sensitivity and specificity for wider acceptance of our model. There could be furthermore algorithms of machine learning that could be applied to our model so that we could evaluate our model more efficiently and that might give us better or surprising results. We still haven't tested our dataset on any neural network or any other deep learning model so further work could be done in this field. We have set the probability threshold randomly, there could be some algorithm made to identify the optimal selection of probability threshold for a particular dataset. This model can be trained better when a larger dataset is fed into it. Also, we have used a limited number of clinical features for training our model so we can increase the number of clinical features and be more selective in considering a clinical feature for the dataset for better results.

5 Conclusion

It is so evident from our study that this model could be highly beneficial for large scale testing during this pandemic as testing is the prime key against this disease. There has been a very limited study to predict the percent probability for a person to be infected with Coronavirus. Thus our study could be a milestone in this field and could be highly helpful in properly targeting our potential COVID-19 patients and testing them with higher priority. Also, we have described a method to test persons with lower risks after their identification by mixing their blood samples and testing them collectively such that a minimum number of tests are done. Different classification models gave different accuracy for the same data-set. We achieved an accuracy of 90.9 from Naive Bayes and SVM classifiers while 90.8 and 91.73 from Logistic Regression and Random Forest classifiers respectively also 85.3, 92.8 ,55.3 from linear regression , ann and unsupervised learning respectively which can further be improvised when trained on a bigger dataset and with a greater number of clinical features in the dataset.