# Covid-19 Predictions

*A COVID-19 self diagnosing program based on Machine Learning Algorithms


1st Muhammad Shahzeb Kayani
*Dept. of computer engineering*
*Air University*
Islamabad,Pakistan
171209@students.au.edu.pk

2nd Usman Aziz
*Dept. of computer engineering*
*Air University*
Islamabad,Pakistan
171206@students.au.edu.pk

3rd Usama Ahmad
*Dept. of computer engineering*
*Air University*
Islamabad,Pakistan
171479@students.au.edu.pk



***Abstract*—COVID has proved to be very disastrous threat to all mankind. It all started about a year ago in Wuhan,China. It then transformed itself into a global pandemic. People all around the world are working on its cure. There are a few vaccines which are distributed among the countries as of this moment but it'll take a lot of time to manufacture and distribute those vaccines among everyone. So, proper testing and isolation is the only solution we have for now.**

**As of this time, 49 million people have died due this disease and there 38 million active confirmed cases of COVID all around the globe. Due to the massive number of patients, it is difficult to test everyone and even one unidentified positive person who has COVID can infect a lot of people.**



***Index Terms*—Covid, blood test, symptoms, artificial inteligence, clinical features**


## I. Introduction

As testing everyone for COVID is not feasible so it is logical to think of any other way to diagnose a person. In this project, we used clinical text features of patients who were tested for COVID to train a model and then predict an unseen patient once his data is presented.

Previously many scholars have done the same for different other diseases such as using machine learning classifiers for detection of influenza with the help of data about symptoms faced by the person, detection of Exocrine pancreatic insufficiency (EPI), a condition in which the body does not have the right amount of pancreatic enzymes, with the help of machine learning algorithms.

Academics have also tried to diagnose dengue with the help of artificial neural networks while some has used Simple Classification and Regression Tree for ordering the patients in dengue hazards. The models were also built for prediction of Sepsis, a life-threatening disease, with the help of machine learning model built with restricted clinical features. [1]

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. Many intellectuals have worked on these predictions and used different clinical features and classification techniques such as using the X-ray images, radiographs and applying deep learning algorithms for classification or using chest CT and deep neural network for predicting the probability of being infected or simply using X-ray images for detection of COVID19. In some cases, blood samples are used for the prediction whereas some scholars analysed cough samples using AI-based engine for prediction.

Researchers had also tried to stratify the patients into low and high-risk groups. 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.

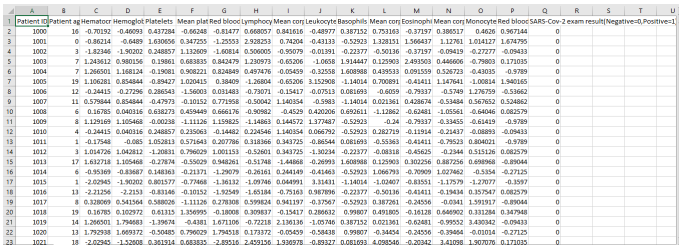
Scholars have also tried to find which chemical or medicine is useful for lowering the risk. We have implemented various supervised machine learning algorithms and found Naive Bayes, random forest and logistic regression algorithm to be the most useful.The performance of the models was evaluated on their accuracy of predicting correctly the type of sentiment on unseen data.

## II. Dataset

The dataset we used for this project was taken from "Kaggle" [7] which is an online data community. 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.

The original dataset consisted of 5644 rows of patients and 111 columns which consisted of clinical features i.e. 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.

Firstly, we deal with missing values by marking them



| Patient ID | Patient age | Hematocrit | Hemoglobin | Platelets | Mean platelet | Red blood | Lymphocyte | Mean corpuscular | Leukocyte | Basophils | Mean corpuscular | Eosinophil | Mean corpuscular | Monocyte | Red blood | SARS-Cov-2 exam result | [Negative=0,Positive=1] |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1000 | 16 | -0.70192 | -0.46093 | 0.437284 | -0.66248 | -0.81477 | 0.668057 | 0.841616 | -0.48977 | 0.387152 | 0.753163 | -0.37197 | 0.386517 | 0.4626 | 0.967144 | 0 | |
| 1001 | 0 | -0.86214 | -0.6489 | 1.630656 | 0.347255 | -1.25553 | 2.928253 | 0.74204 | -0.43133 | -0.52923 | 1.328151 | 1.566437 | 1.12761 | 1.014127 | 1.674795 | 0 | |
| 1002 | 3 | -1.82346 | -1.90202 | 0.248857 | 1.132609 | -1.60814 | 0.506005 | -0.95079 | -0.01191 | -0.22377 | -0.56136 | -0.37197 | -0.09419 | -0.27277 | -0.09433 | 0 | |
| 1003 | 7 | 1.243612 | 0.980156 | 0.19861 | 0.683835 | 0.842479 | 1.230973 | -0.65206 | -1.0658 | 1.914447 | 0.125903 | 2.493503 | 0.446606 | -0.79803 | 0.171035 | 0 | |
| 1004 | 7 | 1.266501 | 1.168124 | -0.19081 | 0.908221 | 0.824849 | 0.497476 | -0.05459 | -0.32558 | 1.608988 | 0.439533 | 0.091559 | 0.526723 | -0.43035 | -0.9789 | 0 | |
| 1005 | 19 | 1.106281 | 0.858844 | -0.89427 | 1.020415 | 0.38409 | -1.26804 | -0.65206 | 3.152908 | -1.14014 | 0.700891 | -0.41411 | 1.147641 | -1.00814 | 1.940165 | 0 | |
| 1006 | 12 | -0.24415 | -0.27296 | 0.286543 | -1.56003 | 0.031483 | -0.73071 | -0.15417 | -0.07513 | 0.081693 | -0.6059 | -0.79337 | -0.5749 | 1.276759 | -0.53662 | 0 | |
| 1007 | 11 | 0.579844 | 0.858844 | -0.47973 | -0.10152 | 0.771958 | -0.50042 | 1.140354 | -0.5983 | -1.14014 | 0.021361 | 0.428674 | -0.53484 | 0.567652 | 0.524862 | 0 | |
| 1008 | 6 | 0.16785 | 0.040316 | 0.638273 | 0.459449 | 0.666176 | -0.90982 | -0.4529 | 0.420206 | 0.692611 | -1.12862 | -0.62481 | -1.05561 | -0.64046 | 0.082579 | 0 | |
| 1009 | 8 | 1.129169 | 1.105468 | -0.00238 | -1.11126 | 1.159825 | -1.14863 | 0.144572 | 1.177467 | -0.52923 | -0.24 | -0.79337 | -0.33455 | -0.61419 | -0.9789 | 0 | |
| 1010 | 4 | -0.24415 | 0.040316 | 0.248857 | 0.235063 | -0.14482 | 0.224546 | 1.140354 | 0.066792 | -0.52923 | 0.282719 | -0.11914 | -0.21437 | -0.08893 | -0.09433 | 0 | |
| 1011 | 1 | -0.17548 | -0.085 | 1.052813 | 0.571643 | 0.207786 | 0.318366 | 0.343725 | -0.86544 | 0.081693 | -0.55363 | -0.41411 | -0.79523 | 0.804021 | -0.9789 | 0 | |
| 1012 | 3 | 1.014726 | 1.042812 | -1.20831 | 0.796029 | 1.001153 | -0.52601 | 0.343725 | -1.30234 | -0.22377 | -0.08318 | -0.45625 | -0.2344 | 0.515126 | 0.082579 | 0 | |
| 1013 | 17 | 1.632718 | 1.105468 | -0.27874 | -0.55029 | 0.948261 | -0.51748 | -1.44868 | -0.26993 | 1.608988 | 0.125903 | 0.302256 | 0.887256 | 0.698968 | -0.89044 | 0 | |
| 1014 | 6 | -0.95369 | -0.83687 | 0.148363 | -0.21371 | -1.29079 | -0.26161 | 0.244149 | -0.41463 | -0.52923 | 1.066793 | -0.70909 | 1.027462 | -0.5354 | -0.27125 | 0 | |
| 1015 | 1 | -2.02945 | -1.90202 | 0.801577 | -0.77468 | -1.36132 | -1.09746 | 0.044991 | 3.31431 | -1.14014 | -1.02407 | -0.81551 | -1.17579 | -1.27077 | -0.3597 | 0 | |
| 1016 | 13 | -2.21256 | -2.2153 | -0.83146 | -0.10152 | -1.92549 | -1.65184 | -0.75163 | 0.987896 | -0.22377 | -0.50136 | -0.41411 | -0.19434 | 0.357547 | 0.082579 | 0 | |
| 1017 | 8 | 0.328069 | 0.541564 | 0.588026 | -1.11126 | 0.278308 | 0.599824 | 0.941197 | -0.37567 | -0.52923 | 0.387281 | -0.24556 | -0.0341 | 1.591917 | -0.89044 | 0 | |
| 1018 | 19 | 0.16785 | 0.102972 | 0.61315 | 1.356995 | -0.18008 | 0.309837 | -0.15417 | 0.286632 | 0.99807 | 0.491805 | -0.16128 | 0.646902 | 0.331284 | 0.347948 | 0 | |
| 1019 | 14 | 1.266501 | 1.794681 | -1.39674 | -0.4381 | 1.671106 | -0.72218 | 2.136136 | -1.05746 | 0.387152 | 0.021361 | -0.62481 | -0.99552 | 3.430342 | -0.09433 | 0 | |
| 1020 | 13 | 1.792938 | 1.669372 | -0.50485 | 0.796029 | 1.794518 | 0.173372 | -0.05459 | -0.59418 | 0.99807 | -0.34454 | -0.24556 | -0.19464 | -0.01014 | -0.27125 | 0 | |
| 1021 | 18 | -2.02945 | -1.52608 | 0.361914 | 0.683835 | -2.89516 | 2.459156 | 1.936978 | -0.89327 | 0.081693 | 4.098546 | -0.20342 | 3.41098 | 1.907076 | 0.171035 | 0 | |

Fig. 1. Dataset Sample

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.

## III. Models

### A. Naive Bayes Algorithm

Naive Bayes, a supervised learning algorithm is based on Bayes theorem and is used to solve classifications problems. It works on probabilities i.e. it predicts on the basis of probability of an object. It is called "Naive" because it assumes that the occurrence of certain features is independent of other features. The only issue with Naive Bayes algorithm is that it can not learn the relationship between different features.

Here we represent each person to be tested(t) as a set of clinical features associated with them. Let x an 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 f1,f2,fn etc are different clinical features of a person to be tested. Now calculation of probability for t to be tested positive:

$$p(x/t) = p(t/x) * p(t)/p(x)$$

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 if we implement Naive theorem on the above equation then we get:

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

This is the final model we used in our code.

### B. Logistic Regression Algorithm

In statistics, the logistic model is used to model the probability of a certain class or event existing such as pass/fail, win/lose, alive/dead or healthy/sick. This can be extended to model several classes of events such as determining whether an image contains a cat, dog, lion, etc.

In normal regression, we would have used a function to predict the probability of a person 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.

$$y = a_0 + a1$$

$$p = \frac{1}{1 + e^y}$$

Which is the signmoid function.

$$ln(\frac{p}{a - p}) = a_0 + a1$$

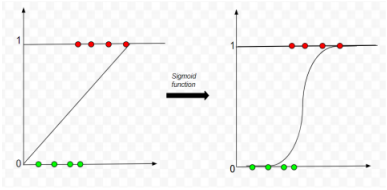



Fig. 2. Change due to Sigmoid function

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.

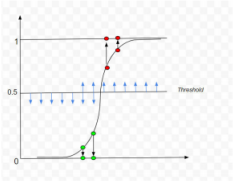



Fig. 3. Logistic Regression with threshold

### C. Random Forest Algorithm

Random forest is a flexible, easy to use machine learning algorithm that produces, even without hyper-parameter tuning, a great result most of the time. It is also one of the most used algorithms, because of its simplicity and diversity (it can be used for both classification and regression tasks).

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.

Random forest adds additional randomness to the model, while growing the trees. Instead of searching for the most important feature while splitting a node, it searches for the best feature among a random subset of features. This results in a wide diversity that generally results in a better model.

Therefore, in random forest, only a random subset of

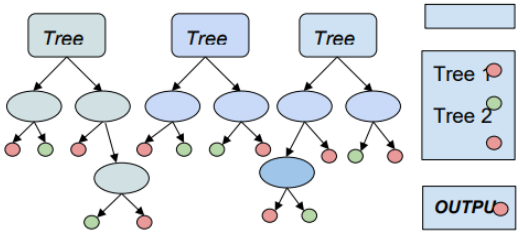



Fig. 4. Random Forest Classification

the features is taken into consideration by the algorithm for splitting a node. You can even make trees more random by additionally using random thresholds for each feature rather than searching for the best possible thresholds (like a normal decision tree does).

## IV. Tests and Results

This project was coded on Spyder platform of Anaconda Software. The model is tested with random data after training and testing. Individual results of the algorithms are given below:

### A. Naive Bayes Algorithm

We implemented Naive Bayes algorithm on our dataset and it achieved accuracy of 90.90 percent with 80 percent of the data as training data and 20 percent as testing data. It identified 89 false negatives and 13 true positives after performing 121 tests.

### B. Logistic Regression Algorithm

After testing Naive Bayes algorithm, we moved onto implementing and testing logistic regression and in the results it gained the accuracy of 90.08 percent which is almost similar to Naive Bayes algorithm. It identified 100 false negatives and 9 true positives successfully after performing 121 tests.

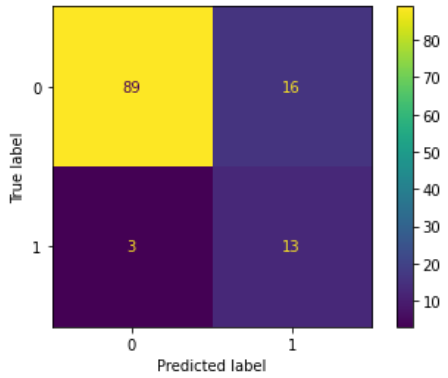



Fig. 5. Naive Bayes Confussion Matrix

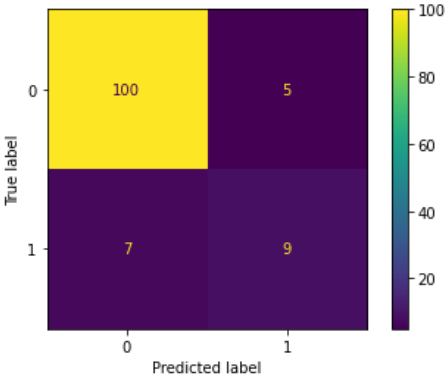



Fig. 6. Logistic Regressio Confussion Matrix

### C. Random Forest Algorithm

After Naive bayes and logistic regression, we implemented and tested Random forest algorithm and it got the accuracy of 88.42 percent which is the least of the 3 algorithms. It successfully identified 104 false negatives and 7 true positives after performing 121 tests.

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.

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

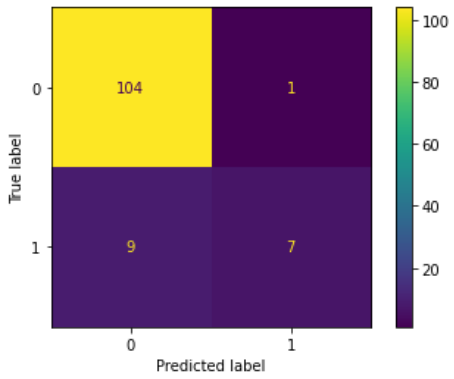



Fig. 7. Random Forest Confussion Matrix

## V. 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.

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, Logistic Regression and Random Forest we have proved our model a successful one.

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.

## VI. Software Engineering Strategy

The software engineering model we used while doing this project is "Iterative Model". With Iterative development software changes on each iteration, evolves and grows. As each iteration builds on the previous one, software design remains consistent.

As software is delivered in parts, there is no need for a full specification from the project's start and small changes to requirements are possible in the course of the development process. However, the requirements can't change radically – major ones must be defined in the beginning, especially those for system design in case of Incremental development as further integration of the delivered software parts can become an issue.

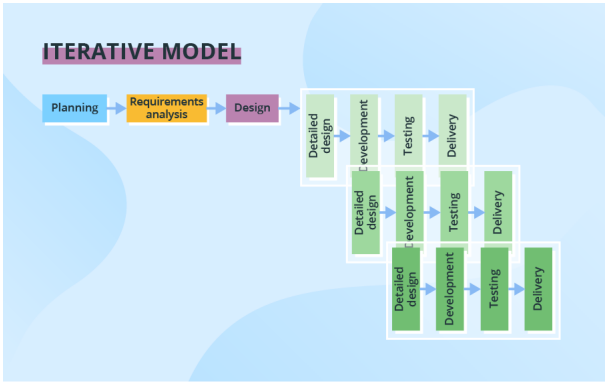



Fig. 8. Iterative Model

## VII. 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 dataset. We achieved an accuracy of 90.90 from Naive Bayes while 90.8 and 88.42 from Logistic Regression and Random Forest classifiers respectively which can further be improvised when trained on a bigger dataset and with a greater number of clinical features in the dataset.

## REFERENCES


[1] Yadav, Milind, Murukessan Perumal, and M. Srinivas. "Analysis on novel coronavirus (COVID-19) using machine learning methods." Chaos, Solitons and Fractals 139 (2020): 110050.

[2] Barstugan, M., Ozkaya, U. and Ozturk, S., 2020. Coronavirus (covid-19) classification using ct images by machine learning methods. arXiv preprint arXiv:2003.09424.

[3] Alimadadi, Ahmad, et al. "Artificial intelligence and machine learning to fight COVID-19." (2020): 200-202.

[4] Elaziz, Mohamed Abd, et al. "New machine learning method for image-based diagnosis of COVID-19." Plos one 15.6 (2020): e0235187.

[5] Assaf, Dan, et al. "Utilization of machine-learning models to accurately predict the risk for critical COVID-19." Internal and emergency medicine 15.8 (2020): 1435-1443.

[6] Kassani, S.H., Kassasni, P.H., Wesolowski, M.J., Schneider, K.A. and Deters, R., 2020. Automatic Detection of Coronavirus Disease (COVID-19) in X-ray and CT Images: A Machine Learning-Based Approach. arXiv preprint arXiv:2004.10641.

[7] www.kaggle.com/einsteindata4u/covid19 "Link to dataset"