# Predicting Covid Cases through Machine Learning Algorithms

*By: Loh, Nathan; Schultze, Nikolaus*

## Introduction and Problem Description

Hospital staffing is very minimal, and doctors are very busy. With the addition of Covid, this already limited hospital staffing is spread very thin, and they have to work extensive hours, causing them to get tired and worn out. The staff is already responsible for the care of their patients, but with the addition of Covid, there are more patients than ever, resulting in the staff caring for more patients and forcing them to spend less time per patient. As a result of the influx of Covid patients, the hospital staff needs more time to visit every patient and determine whether or not the person has Covid. Not only is this influx of Covid patients requiring these new Covid patients to be attended by the hospital staff, but it also requires extra resources. This influx also takes the hospital staff away from its other patients and uses resources that could be designated for the other patients. While there are other alternatives for testing for Covid, these alternatives, such as Rapid Testing, have flaws. While these rapid Covid tests give the Covid test results back quickly, they have an issue with inaccurate and false-positive results. There are more accurate Covid tests than rapid tests, but they take over 24 hours for the patient to get their result back. These long wait times result in patients having to quarantine while they await the results; however, only some patients will quarantine. The long wait time can result in a person with Covid continuing to adventure out into public and spread Covid to others. For these lab tests, if there is a higher number of patients sending in Covid tests, then the wait time for the results can be increased. The algorithms we used include Tree Regression (DT), Logistic Regression (LR), Regular Regression (RR), K-Nearest Neighbors (KNN), and two methods of a combination of KNN, LR, and DT. We found that the first method of a combination of KNN, LR, and DT had the best accuracy with an accuracy of 97.19%, Regular Regression had the best sensitivity with a sensitivity of 99.01%, and overall the second method of a combination of KNN, LR, and DT worked the best.

## Description of the Data

The data comes from Kaggle called “Symptoms and COVID Presence (May 2020 data)” by Hemanth Harikrishnan. The data contains a list of symptoms, including Breathing Problems, Fever, Dry Cough, Sore Throat, Running Nose, Asthma, Chronic Lung Disease, Headache, Heart Disease, Diabetes, Hypertension, Fatigue, and Gastrointestinal. It also includes possible events in which a person could have received Covid, including Abroad travel, Contact with COVID Patients, Attended Large Gathering, Visited Public Exposed Places, and Family working in Public Exposed Places. The data also includes a COVID-19 column, indicating whether the patient has Covid. Each column in the original dataset represents “Yes” and “No,” which we converted into binary values, with Yes being converted to 1 and No being converted to 0. For the symptoms column, 1 represents the person having the symptom, and 0 represents the person not

having the symptom. For the events columns, 1 represents the person attending an event of that type, and 0 representing not attending the event. For the Covid-19 column, 1 represents the person having Covid, and 0 represents the person not having Covid. There is a total of 5,434 data points.

Not all symptoms may be impactful in determining whether a patient has Covid. As such, we perform feature selection on the data to remove any unnecessary predictors. Upon first analyzing the data description, it is noticed that the predictors, Wearing Masks and Sanitization, had a minimum and maximum value of 0, indicating that there is only a single unique input. Thus, these two predictors will have no impact when training the dataset, so their columns can be removed. Afterward, we examine the correlation between the predictors and response variables. This is shown in the following image.

```
correlation_matrix = df2.corr()
correlation_matrix
correlation_matrix.style.background_gradient(cmap='Blues', axis=None)
```

Out[5]:

| | Breathing Problem | Fever | Dry Cough | Sore throat | Running Nose | Asthma | Chronic Lung Disease | Headache | Heart Disease | Diabetes | Hyper Tension | Fatigue | Gastrointestinal | Abroad travel | Contact with COVID Patient | Attended Large Gathering | Visited Public Exposed Places | Family working in Public Exposed Places | COVID-19 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Breathing Problem | 1.000000 | 0.089903 | 0.159562 | 0.303768 | 0.055190 | 0.075318 | -0.098291 | -0.062172 | -0.073366 | 0.055427 | 0.045256 | 0.000561 | -0.075390 | 0.117795 | 0.214634 | 0.200304 | 0.066688 | 0.018295 | 0.443764 |
| Fever | 0.089903 | 1.000000 | 0.127580 | 0.322235 | 0.081758 | 0.073953 | -0.025160 | -0.035416 | -0.031462 | 0.050286 | 0.079001 | -0.060458 | -0.008067 | 0.128726 | 0.164704 | 0.070490 | 0.002252 | 0.012102 | 0.352891 |
| Dry Cough | 0.159562 | 0.127580 | 1.000000 | 0.213907 | -0.030763 | 0.086843 | -0.043664 | -0.035912 | 0.047566 | -0.006593 | 0.081989 | -0.039909 | 0.008251 | 0.331418 | 0.128330 | 0.117963 | 0.086176 | 0.163102 | 0.464292 |
| Sore throat | 0.303768 | 0.322235 | 0.213907 | 1.000000 | 0.039450 | 0.081377 | -0.050440 | -0.015971 | 0.002177 | 0.001938 | 0.042811 | -0.023290 | 0.025886 | 0.205986 | 0.189251 | 0.216438 | 0.079055 | 0.104378 | 0.502848 |
| Running Nose | 0.055190 | 0.081758 | -0.030763 | 0.039450 | 1.000000 | -0.022763 | -0.014376 | 0.068479 | -0.056750 | 0.042961 | -0.020445 | 0.007026 | -0.014673 | 0.034526 | 0.003776 | 0.061099 | 0.032568 | -0.061323 | -0.005657 |
| Asthma | 0.075318 | 0.073953 | 0.086843 | 0.081377 | -0.022763 | 1.000000 | -0.033771 | 0.037064 | 0.076783 | -0.012060 | 0.017707 | 0.006564 | 0.101909 | 0.068286 | 0.005046 | -0.044592 | 0.020941 | -0.115679 | 0.089930 |
| Chronic Lung Disease | -0.098291 | -0.025160 | -0.043664 | -0.050440 | -0.014376 | -0.033771 | 1.000000 | -0.050480 | -0.039860 | 0.046789 | -0.010331 | -0.047655 | -0.050333 | -0.088854 | -0.062482 | -0.020548 | -0.093049 | 0.038343 | -0.056837 |
| Headache | -0.062172 | -0.035416 | -0.035912 | -0.015971 | 0.068479 | 0.037064 | -0.050480 | 1.000000 | 0.048471 | 0.032390 | -0.207489 | 0.052035 | 0.097778 | 0.043589 | -0.082101 | -0.162992 | -0.005790 | -0.012625 | -0.027793 |
| Heart Disease | -0.073366 | -0.031462 | 0.047566 | 0.002177 | -0.056750 | 0.076783 | -0.039860 | 0.048471 | 1.000000 | -0.032956 | 0.049139 | -0.058925 | 0.004121 | -0.020761 | -0.025593 | -0.045437 | 0.086169 | 0.035000 | 0.027072 |
| Diabetes | 0.055427 | 0.050286 | -0.006593 | 0.001938 | 0.042961 | -0.012060 | 0.046789 | 0.032390 | -0.032956 | 1.000000 | 0.042543 | -0.043903 | 0.040651 | 0.039013 | -0.085696 | -0.061650 | -0.078212 | 0.097696 | 0.040627 |
| Hyper Tension | 0.045256 | 0.079001 | 0.081989 | 0.042811 | -0.020445 | 0.017707 | -0.010331 | -0.207489 | 0.049139 | 0.042543 | 1.000000 | -0.027605 | -0.067972 | -0.016382 | 0.027307 | 0.002911 | 0.019174 | 0.048152 | 0.102575 |
| Fatigue | 0.000561 | -0.060458 | -0.039909 | -0.023290 | 0.007026 | 0.006564 | -0.047655 | 0.052035 | -0.058925 | -0.043903 | -0.027605 | 1.000000 | 0.009356 | -0.068401 | -0.027383 | -0.031058 | -0.009562 | -0.025623 | -0.044188 |
| Gastrointestinal | -0.075390 | -0.008067 | 0.008251 | 0.025886 | -0.014673 | 0.101909 | -0.050333 | 0.097778 | 0.004121 | 0.040651 | -0.067972 | 0.009356 | 1.000000 | 0.099577 | 0.025277 | -0.017251 | -0.061885 | -0.027603 | -0.003367 |
| Abroad travel | 0.117795 | 0.128726 | 0.331418 | 0.205986 | 0.034526 | 0.068286 | -0.088854 | 0.043589 | -0.020761 | 0.039013 | -0.016382 | -0.068401 | 0.099577 | 1.000000 | 0.080210 | 0.113399 | 0.069609 | 0.143094 | 0.443875 |
| Contact with COVID Patient | 0.214634 | 0.164704 | 0.128330 | 0.189251 | 0.003776 | 0.005046 | -0.062482 | -0.082101 | -0.025593 | -0.085696 | 0.027307 | -0.027383 | 0.025277 | 0.080210 | 1.000000 | 0.234649 | 0.079800 | 0.006909 | 0.357122 |
| Attended Large Gathering | 0.200304 | 0.070490 | 0.117963 | 0.216438 | 0.061099 | -0.044592 | -0.020548 | -0.162992 | -0.045437 | -0.061650 | 0.002911 | -0.031058 | -0.017251 | 0.113399 | 0.234649 | 1.000000 | 0.083795 | 0.063776 | 0.390145 |
| Visited Public Exposed Places | 0.066688 | 0.002252 | 0.086176 | 0.079055 | 0.032568 | 0.020941 | -0.093049 | -0.005790 | 0.086169 | -0.078212 | 0.019174 | -0.009562 | -0.061885 | 0.069609 | 0.079800 | 0.083795 | 1.000000 | 0.028486 | 0.119755 |
| Family working in Public Exposed Places | 0.018295 | 0.012102 | 0.163102 | 0.104378 | -0.061323 | -0.115679 | 0.038343 | -0.012625 | 0.035000 | 0.097696 | 0.048152 | -0.025623 | -0.027603 | 0.143094 | 0.006909 | 0.063776 | 0.028486 | 1.000000 | 0.160208 |
| COVID-19 | 0.443764 | 0.352891 | 0.464292 | 0.502848 | -0.005657 | 0.089930 | -0.056837 | -0.027793 | 0.027072 | 0.040627 | 0.102575 | -0.044188 | -0.003367 | 0.443875 | 0.357122 | 0.390145 | 0.119755 | 0.160208 | 1.000000 |

*Figure 1: Correlation matrix of the dataset.*

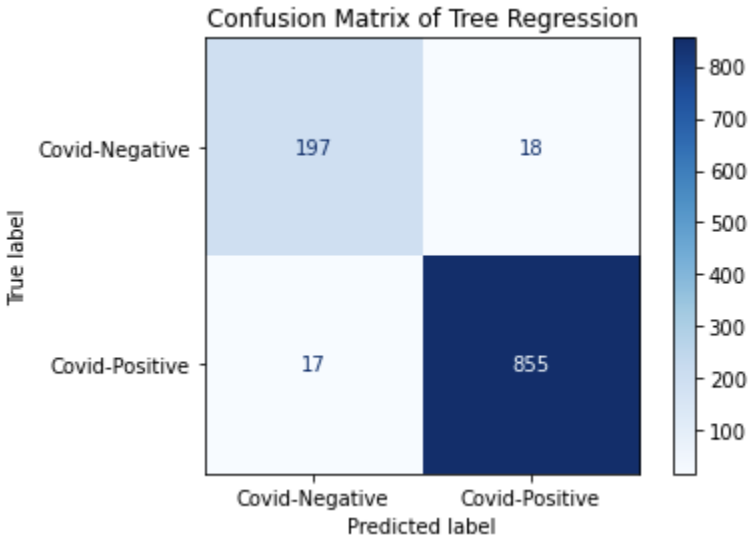
As seen in Figure 1, each predictor produced a correlation coefficient between different variables. In the last row and column, we get the correlation coefficient between the predictors and the response variable. Values that are closer to positive ones indicate that there is a positive correlation between the two variables. In contrast, values closer to negative ones indicate a negative correlation between the two. Coefficient values close to zero indicate little to no correlation between the two variables. The correlation matrix shows that Runny Nose, Asthma, Chronic Lung Disease, Headache, Heart Disease, Diabetes, Hyper Tension, Fatigue, and Gastrointestinal predictors had correlation coefficient values close to 0. As such, we removed these from our dataset.

## Algorithm Description and Observations

In our project, we utilize four algorithms and a combination of them. The algorithms are Tree Regression (Decision Tree or DT), Logistic Regression (LR), Regular Regression (Linear Regression), and K-Nearest Neighbors (KNN). Our last methods involve a combination of KNN, LR, and DT.

Tree Regression Algorithm

Tree Regression, specifically Decision Trees, is a classification algorithm presenting a "decision" at each node. After the machine trains the dataset, it creates a tree. Each non-leaf node provides multiple subtrees that are traversed based on the decision of certain predictors. For example, let the prediction start at the root node with the patient having a positive symptom of fever. The prediction traverses the left subtree if the patient does not have a fever and goes right if they do have a fever. Thus, according to the patient's symptoms, the machine would traverse down the right subtree.

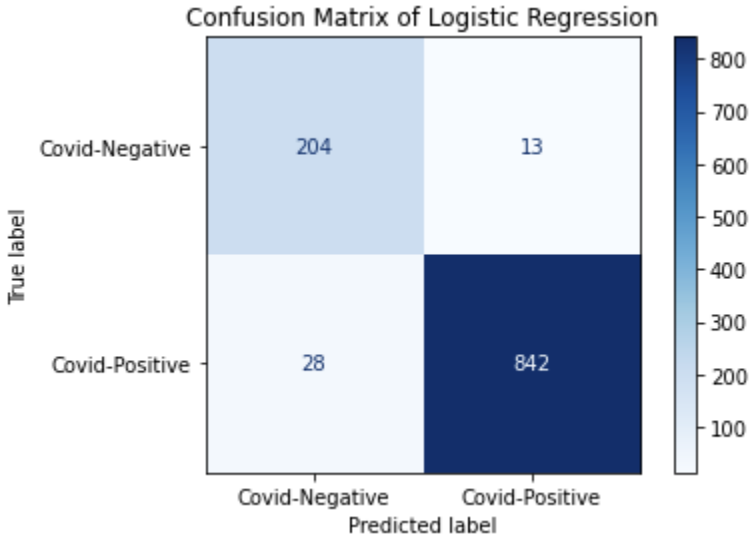



| Test Number | Accuracy | Precision | Sensitivity | Specificity |
|---|---|---|---|---|
| Test 1 | 96.78% | 97.94% | 98.05% | 91.63% |
| Test 2 | 96.87% | 98.52% | 97.63% | 93.50% |
| Test 3 | 97.24% | 98.26% | 98.26% | 93.30% |
| Test 4 | 96.50% | 97.59% | 98.04% | 90.50% |
| Test 5 | 97.06% | 98.87% | 97.55% | 94.71% |
| Average | 96.89% | 98.24% | 97.91% | 92.73% |

*Table 1: Results of the Tree Regression after 5 runs.*

Logistic Regression Algorithm

Logistic Regression is a type of classification algorithm. It utilizes the sigmoid activation function to predict the classes or labels of the data points. The output of each prediction is a value between 0 and 1. As such, any value estimated to have a value above 0.5 is predicted to be of class 1, and values at 0.5 and below are set to class 0.

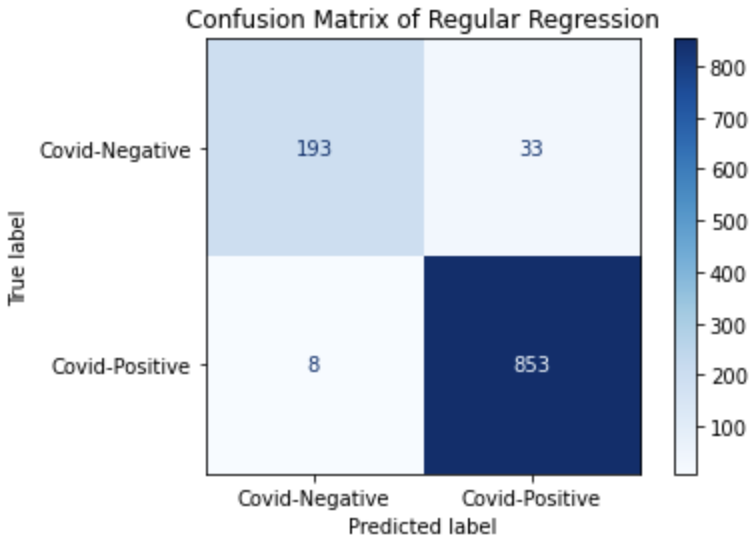



| Test Number | Accuracy | Precision | Sensitivity | Specificity |
|---|---|---|---|---|
| Test 1 | 96.78% | 98.63% | 97.41% | 93.94% |
| Test 2 | 96.50% | 98.14% | 97.47% | 92.69% |
| Test 3 | 96.41% | 98.40% | 97.18% | 92.96% |
| Test 4 | 96.23% | 98.15% | 97.14% | 92.49% |
| Test 5 | 96.14% | 97.71% | 97.49% | 90.52% |
| Average | 96.41% | 98.21% | 97.34% | 92.52% |

*Table 2: Results of the Logistic Regression after 5 runs.*

Regular Regression Algorithm

Regular Regression, also known as Linear Regression, is an algorithm used to predict continuous values. It utilizes a straight line (y=mx+b) to fit the data points. This project is a classification problem, meaning that the Linear Regression algorithm needs to be slightly altered. Since every feature and output is a binary value, the predicted value will be close to 0 and 1. Thus, we set a threshold value where any output greater than it would be considered class 1, or it would be considered class 0. The threshold we chose was 0.5.

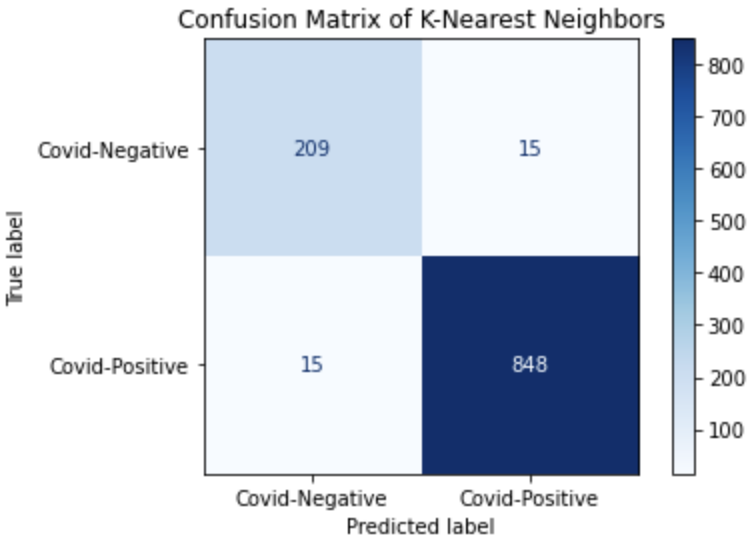



| Test Number | Accuracy | Precision | Sensitivity | Specificity |
|---|---|---|---|---|
| Test 1 | 96.23% | 96.28% | 99.07% | 85.40% |
| Test 2 | 96.23% | 96.29% | 99.08% | 85.14% |
| Test 3 | 95.22% | 95.52% | 98.76% | 79.60% |
| Test 4 | 95.95% | 96.24% | 99.01% | 80.66% |
| Test 5 | 96.69% | 96.95% | 99.11% | 85.26% |
| Average | 96.06% | 96.26% | 99.01% | 83.21% |

*Table 3: Results of the Linear Regression after 5 runs.*

K-Nearest Neighbors Algorithm

K-Nearest Neighbors is a supervised algorithm that utilizes distances to classify data points. It takes “k” closest data points, where “k” is set to be five as default. In our algorithm, we set “k” to be 15, an odd number to avoid dealing with tiebreakers. As such, our algorithm takes an unlabeled data point and finds the 15 closest labeled data points. Out of the 15 data points, it will count which class occurs the most and set that value as the final prediction.

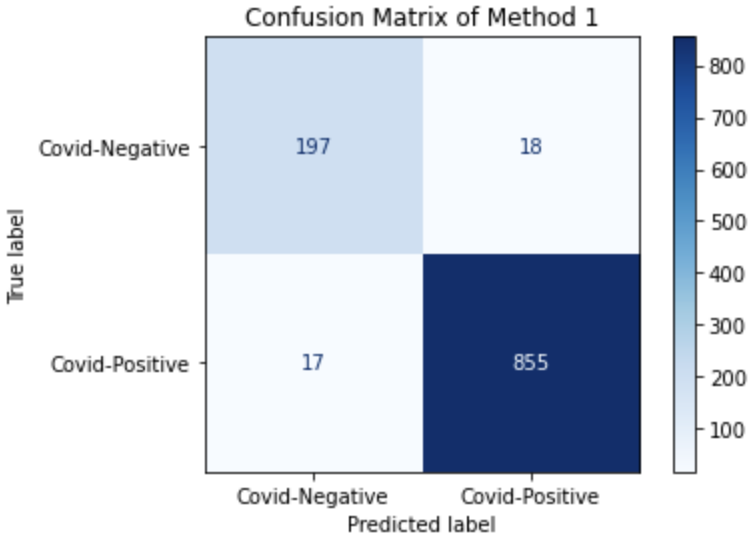



| Test Number | Accuracy | Precision | Sensitivity | Specificity |
|---|---|---|---|---|
| Test 1 | 97.24% | 98.26% | 98.26% | 93.30% |
| Test 2 | 96.14% | 95.91% | 99.29% | 84.81% |
| Test 3 | 96.04% | 97.84% | 97.29% | 90.55% |
| Test 4 | 97.24% | 97.39% | 99.19% | 89.64% |
| Test 5 | 97.42% | 97.78% | 99.10% | 89.85% |
| Average | 96.82% | 97.44% | 98.63% | 89.63% |

*Table 4: Results of the K-Nearest Neighbors after 5 runs.*

KNN+LR+DT Algorithms - Method 1

We combined a few of these to improve on the individual algorithms of Linear Regression, Logistic Regression, Decision Trees, and K-Nearest Neighbors. Specifically, we used a combination of K-Nearest Neighbors, Logistic Regression, and Decision Trees. We first trained each of the three algorithms on the same training set before creating separate prediction lists. We then take each of the predictions and compare them with each other, setting the final prediction to be the majority.

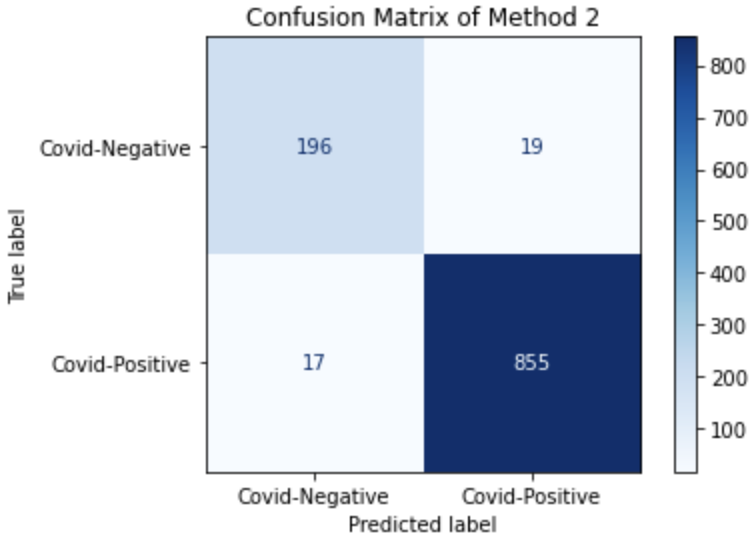



| Test Number | Accuracy | Precision | Sensitivity | Specificity |
|---|---|---|---|---|
| Test 1 | 96.78% | 97.94% | 98.05% | 91.63% |
| Test 2 | 97.24% | 98.19% | 98.41% | 92.16% |
| Test 3 | 96.60% | 97.83% | 97.94% | 91.12% |
| Test 4 | 97.15% | 98.43% | 98.10% | 92.63% |
| Test 5 | 98.16% | 98.86% | 98.86% | 95.31% |
| Average | 97.19% | 98.25% | 98.27% | 92.57% |

*Table 5: Results of Method 1 after 5 runs.*

KNN+LR+DT Algorithms - Method 2

Although the accuracy had increased in Method 1, the sensitivity had slightly decreased. As such, we decided to find another method that may increase the sensitivity. Our second method utilizes the same algorithms as the first and follows the same setup. The main difference with Method 2 is that it is based on the KNN algorithm. We took the KNN predictions and iterated through them. If any output equaled 0, we would then compare the predictions of the Decision Tree and Logistic Regression at that index. If either the Decision Tree or Logistic Regression predicted a positive Covid result, it sets the final prediction as 1.




| Test Number | Accuracy | Precision | Sensitivity | Specificity |
|---|---|---|---|---|
| Test 1 | 96.69% | 97.83% | 98.05% | 91.16% |
| Test 2 | 97.15% | 97.33% | 99.21% | 88.24% |
| Test 3 | 96.78% | 96.45% | 99.66% | 85.05% |
| Test 4 | 97.15% | 97.79% | 98.77% | 89.47% |
| Test 5 | 97.70% | 98.18% | 98.97% | 92.49% |
| Average | 97.09% | 97.52% | 98.93% | 89.28% |

*Table 6: Results of Method 2 after 5 runs.*

## Final Comparisons

| Algorithm | Tree | Logistic | Regular | KNN | KNN+LR+DT Method 1 | KNN+LR+DT Method 2 |
|---|---|---|---|---|---|---|
| Accuracy | 96.89% | 96.01% | 96.06% | 96.82% | 97.19% | 97.09% |
| Precision | 98.24% | 98.14% | 96.26% | 97.44% | 98.25% | 97.52% |
| Sensitivity | 97.91% | 98.14% | 99.01% | 98.63% | 98.27% | 98.93% |
| Specificity | 92.73% | 92.11% | 83.21% | 89.63% | 92.57% | 89.28% |

*Table 7: Average outcomes of each algorithm.*

In the final comparison for each algorithm, we will mainly examine the accuracy and the sensitivity because it deals with false negative reports. False negative reports are important as we want to avoid predicting a person to be Covid-negative when they are Covid-positive. On the other hand, it is safer to predict if a person has Covid when they do not. In short, we want to deal with sensitivity because it calculates the percent that the algorithm correctly guesses actual Covid-positive symptoms. From our results, linear regression has the lowest accuracy; however, it is still high for a regression algorithm that predicts continuous values. It has the highest sensitivity at the cost of having the lowest specificity. We then compare the two combination methods with KNN, which had the highest average accuracy compared to the other individual algorithms. The two combination methods have higher accuracy than the KNN algorithm, meaning that our methods showed slight improvements. The first method had a lower sensitivity, while the second one had a higher one. If we had to choose between the two methods, the second method would be better, as it sacrifices a small amount of accuracy for higher sensitivity. In other words, the second method had fewer false negative predictions. In general, KNN Method 1 has the best accuracy, KNN Method 1 has the best precision, Regular Regression has the best sensitivity, and the Decision Tree has the best specificity.

## Main Contributions

Nathan's main contributions to this project include completing the Logistic Algorithm, the K-Nearest Neighbors Algorithm, and the algorithm constituting K-Nearest Neighbors, Logistic Regression, and Decision Tree. In addition, Nathan completed the description and comparison of the algorithms for the report and presentation. Nikolaus's main contributions to this project include completing the Tree Algorithm and the Regular Algorithms. In addition, Nikolaus completed the algorithms' testing, collected the tests' results for each algorithm, and determined potential future uses. Both Nathan and Nikolaus contributed to completing the problem description and data description.

## Lessons Learned

Some lessons that we learned from this project include time management, work distribution, communication, and project management. For this project, we had various deadlines already set, such as the completion of the proposal, the intermediate project report, the presentation, and the final. We also set deadlines for ourselves, including completing the different algorithms. We had to balance working on this project with classes, other classwork, and other work for this class and ensure that we met each of these deadlines. We also had to decide the best way to split the work for this project. Originally we only had three algorithms: tree, logistic, and regular. However, we decided to add a K-Nearest Neighbors Algorithm and an algorithm combining K-Nearest Neighbors, Logistic Regression, and Decision Tree. In addition, we also had to work on the proposal, the intermediate project report, the presentation, and the final. For this project, we divided the workload evenly: one group member (Nathan) did more work on the algorithms, and the other group member (Nikolaus) did more work on the presentation and reports. We also

had to manage the project both long-term and day-to-day. To do this, we had regular communication to see the other group member's progress on their work for the project. We communicated through Discord, checking in several times a week, asking questions, and determining the split of future work. This allowed us to manage the project very well day-to-day to complete the smaller deadlines such as the proposal, intermediate project report, and the algorithms, as well as long-term in completing the presentation, the final project report, and the overall project itself.

## Future Work

Some potential future work includes adapting the algorithms to fit different datasets. The algorithms we created were designed for a specific dataset. This is because the data includes several columns containing various symptoms and whether the patient had that symptom. To adapt the algorithms, we would have to import another dataset and then go through the dataset, observing what the columns do. We will have to see if the columns contain the symptoms like the dataset we are currently using. If the column format is similar, we could remove any symptoms we deem irrelevant after looking at the correlation matrix. Similar to what we did with our current dataset for Running Nose, Asthma, Chronic Lung Disease, Headache, Heart Disease, Diabetes, Hypertension, Fatigue, and Gastrointestinal. If the database does not have a column format similar to our current dataset, then we would have to remove the dataset and not use it or adapt our algorithm further so it works for the dataset.

Another potential future work would include implementing a website or an app for our project. We would upload our most accurate algorithm to a website or mobile app to do this. This would allow people to stay at home and take a test that predicts if they might have Covid or not. The app could work by having various checkable boxes for each symptom and fever, and the user would check any boxes for the symptoms and features they have. By allowing users to stay home and get an accurate prediction, ideally, fewer people go out into public with Covid, which would help prevent the spreading of Covid. In addition to users taking a predicted Covid test, the website or mobile app can show users weekly updates and current trends in Covid-19 for their area. This will depend on if we can secure more data for other regions besides the current data, which was focused on India.

The final piece of future work would be to work with the CDC, WHO, or another health organization to use their collected data on our algorithms to generate reports. These health organizations contain datasets from several countries rather than just one country. This would allow us to complete the reports for several countries while simultaneously allowing us to do a more significant number of countries and help keep a larger part of the general population informed. Rather than create a website or mobile app, we could put our reports on the websites and mobile apps of the CDC, WHO, and any other health organization. This would allow us to reach our target audience better, as many people looking into Covid-19 already use these websites and apps.

## Github

https://github.com/nschultze/cs584project