Detection of COVID-19 on Chest CT Scans using Machine Learning Algorithms

By

Raj Asarpota

CS 4395 – Senior Project

Spring 2021

University of Houston – Downtown

Houston, TX

Faculty Advisor

Dr. Dvijesh Shastri

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Class Instructor

Dr. Kenneth Oberhoff

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Table of Contents**

1. Abstract…………………………………………………………………………………………3

2. Acknowledgement……………………………………………………………………………...4

3. Introduction……………………………………………………………………………………..5

4. Methodology

4.1 Tools/Dataset………………………………………………………………………….6

4.2 Logistic Regression…………………………………………………………………....7

4.3 K-Nearest Neighbor…………………………………………………………………...9

4.4 Decision Tree………………………………………………………………………...12

4.5 Random Forest……………………………………………………………………….14

5. Results

5.1 Logistic Regression Model Results………………………………………………….17

5.2 KNN Model Results………………………………………………………………….18

5.3 Decision Tree Model Results………………………………………………………...19

5.4 Random Forest Model Results……………………………………………………….20

5.5 Utilized Machine Learning Algorithms’ Results Table……………………………...21

6. Conclusion…………………………………………………………………………………….22

7. References……………………………………………………………………………………..23

**Abstract**

The new Coronavirus (COVID-19) is an acute deadly disease that originated from Wuhan province, China in December 2019 and spread globally. The biological structure of COVID-19 comprises of a positive-oriented single-stranded RNA-type, and it is difficult to treat the disease owing to its mutating feature. Medical professionals globally at the time were undergoing intensive research to develop an effective cure for the disease. Presently, COVID-19 is the primary cause of thousands of deaths globally, and major deaths are in the USA, Spain, Italy, China, the UK, Iran, etc. Many types of coronavirus exist, and these viruses are commonly seen in animals. COVID-19 has been discovered in human, bat, pig, cat, dog, rodent, and poultry. Symptoms of COVID-19 include sore throat, headache, fever, runny nose, and cough. The virus can provoke the death of people with weakened immune systems. People infected by COVID-19 may suffer from pneumonia because the virus spreads to the lungs. COVID-19 is transmitted from person to person mostly by physical contact. Generally, healthy people can be infected through breath contact, hand contact, or mucous contact with people carrying COVID-19. [7]

Recently, artificial intelligence (AI) has been widely used for the acceleration of biomedical research. Using deep learning approaches, AI has been used in many applications such as image detection, data classification, image segmentation. I believe there too is a role for artificial intelligence approaches for the detection and characterization of the virus on medical imaging chest CT scans.

In this project, I used both COVID-19 CT chest scans image dataset and normal CT chest scans image dataset provided by the Cancer Imaging Archive. Each image was preprocessed with resizing and plotting, before being appended to an array alongside another array consisting of labels for each training image. Due to scikit-learn’s specifications, the shape of the image array had to be reshaped from a 3-D array to a 2-D array; the only difference being the image dimensions were multiplied together. Then, I trained and tested the datasets using four classic Machine Learning algorithms: Logistical Regression, K-Nearest Neighbors, Decision Tree, and Random Forest. Each model provided varying but overall positive results, given that the image dataset provided was very concise and restrictive. Due to time/hardware constraints, only the Logistical Regression and K-Nearest Neighbors models have been optimized.

**Acknowledgements**

I would like to acknowledge the University of Houston-Downtown and everyone part of it: faculty, staff, mentors, advisors, and administrations that created an excellent educational environment that allowed me to successfully learn and complete my undergraduate courses.

A special acknowledgement to Dr. Dvijesh Shastri, for providing outstanding guidance with this project. He has been very encouraging and extremely supportive throughout the entire duration of working with me this semester. Dr. Shastri has been an excellent mentor to overview this project; he shared all his knowledge and experiences on the subject matter that benefited and developed my skills as a computer scientist. I am proud and honored having Dr. Shastri as my professor and faculty mentor.

In addition, I would like to thank both Dr. Kenneth Oberhoff for being my instructor for the senior project course and Dr. Pablo Guillen-Rondon for excellently covering the topic of Machine Learning in his course, which was heavily utilized in this project. Lastly, I would like to thank my friends and family for all of the unconditional love and support.

**Introduction**

At the end of the year in 2019, the World Health Organization announced a mysterious coronavirus-related pneumonia in Wuhan, China. The first case of the new virus in the United States, COVID-19, was confirmed by the Center of Disease Control at the end of January in 2020. With this new virus rapidly spreading worldwide, the way of life suddenly changed. This took a major toll on how all of society’s pillars functioned. In my case, this pandemic severely altered how I conducted my academics while studying at this University. For my senior project, I wanted to create a program that reflects on the challenges faced during this turbulent times.

The early detection and diagnosis of COVID-19 and the accurate separation of non-COVID-19 cases at the lowest cost and in the early stages of the disease are among the main challenges in the current COVID-19 pandemic. Concerning the novelty of the disease, diagnostic methods based on radiological images suffer from shortcomings despite their many applications in diagnostic centers. COVID-19 tests are currently hard to come by, there aren’t enough and they cannot be manufactured fast enough, which is causing panic. Given that there are limited COVID-19 testing kits, medical professionals need to rely on other diagnosis measures. [7] For the purposes of this project, I thought to explore CT scans images as doctors frequently use X-rays and CT scans to diagnose pneumonia, lung inflammation, abscesses, and/or enlarged lymph nodes. Since COVID-19 attacks the epithelial cells that line our respiratory tract, I can use CT scans to analyze the health of a patient’s lungs. And given that nearly all hospitals have CT scans imaging machines, it could be possible to use X-rays to test for COVID-19 without the dedicated test kits. A drawback is that X-ray analysis requires a radiology expert and takes significant time, which is precious when people are sick around the world. **Therefore developing an automated analysis system is required to save medical professionals valuable time.**

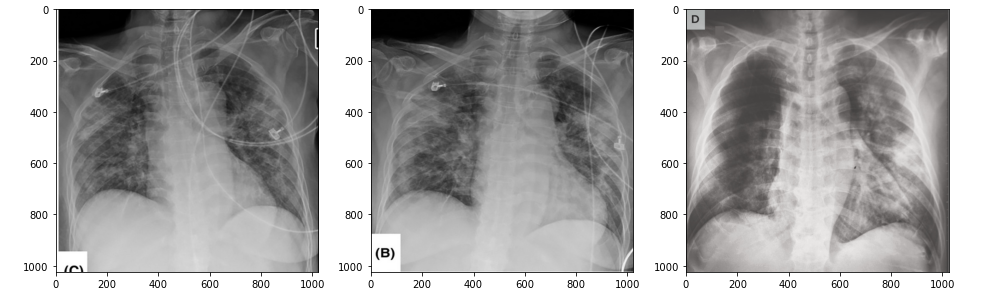
Due to these reasons, I believe there is a role for artificial intelligence approaches for the detection and characterization of the virus on medical imaging chest CT scans. My goal was to create an application consisting of four classic Machine Learning Algorithms that will evaluate and detect the present of the COVID-19 virus on medical imaging using chest CT scans from a globally diverse dataset provided by the Cancer Imaging Archive. A program like this could be utilized by doctors and radiologists to significantly shorten the time of detection and diagnosis of COVID-19.

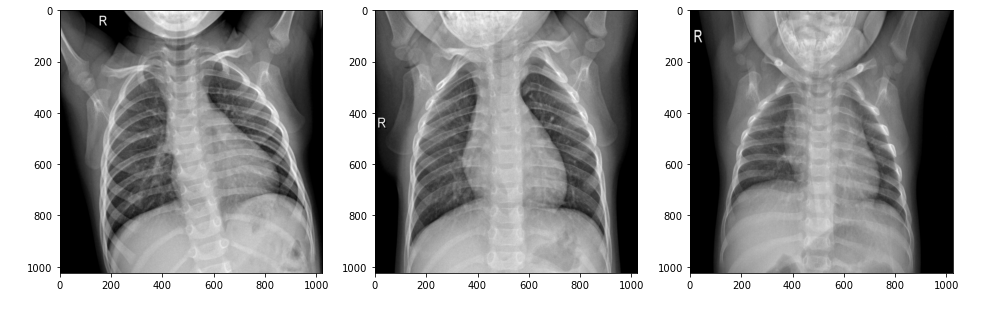
**Methodology**

**4.1 Tools/Dataset**

This project utilizes Python 3.8 using Jupyter Notebook via Anaconda. Four machine learning algorithms: Logistical Regression, K-Nearest Neighbors, Decision Tree, and Random Forest were imported as packages and utilized as supervised learning algorithms. Input data is tagged. Supervised learning establishes a learning process, compares the predicted results with the actual results of the “training data” (ie, input data), and continuously adjusts the predictive model until the predicted results of the model reach an expected accuracy, such as classification and regression problems. [6]

The COVID-19 CT scan image dataset we’ll be using for this project was curated by The Cancer Imaging Archive. In total 199 COVID Chest CT Scans for Training, 199 Normal Chest CT Scans for Training, 20 COVID Chest CT Scans for Testing, 20 Normal Chest CT Scans for Testing were used. All images were resized to 224x224 pixels and plotted . Each image from the training folders is labeled and both are appended into individual arrays. The shape of the image array was resized from a 3D array to 2D array (combining the image dimensions) due to the implemented scikit-learn’s specifications, while the label array was encoded.





**4.2 Logistical Regression**

Logistic Regression is a Machine Learning algorithm which is used for the classification problems, it is a predictive analysis algorithm and based on the concept of probability. We can call a Logistic Regression a Linear Regression model but the Logistic Regression uses a more complex cost function, this cost function can be defined as the **‘Sigmoid function’** or also known as the ‘logistic function’ instead of a linear function. The hypothesis of logistic regression tends it to limit the cost function between 0 and 1. Therefore linear functions fail to represent it as it can have a value greater than 1 or less than 0 which is not possible as per the hypothesis of logistic regression. [1]

[1]

In order to map predicted values to probabilities, we use the Sigmoid function. The function maps any real value into another value between 0 and 1. In machine learning, we use sigmoid to map predictions to probabilities.

[1]

[3]

For logistic regression we are going to modify the formula of the hypothesis:

[1]

We expect our classifier to give us a set of outputs or classes based on probability when we pass the inputs through a prediction function and returns a probability score between 0 and 1.

For logistic regression, the Cost function is defined as:

[1]

Advantages:

* Logistic regression is easier to implement, interpret, and very efficient to train.
* It can easily extend to multiple classes(multinomial regression) and a natural probabilistic view of class predictions.
* It is very fast at classifying unknown records.
* Good accuracy for many simple data sets and it performs well when the dataset is linearly separable.

Disadvantages:

|  |
| --- |
| * The major limitation of Logistic Regression is the assumption of linearity between the dependent variable and the independent variables. * Non-linear problems can’t be solved with logistic regression because it has a linear decision surface. Linearly separable data is rarely found in real-world scenarios. * Logistic Regression requires average or no multicollinearity between independent variables. * It is tough to obtain complex relationships using logistic regression. More powerful and compact algorithms such as Neural Networks can easily outperform this algorithm. |
|  |

**4.3 K-Nearest Neighbor**

The k-nearest neighbors (KNN) algorithm is a simple, easy-to-implement supervised machine learning algorithm that can be used to solve both classification and regression problems. The KNN algorithm assumes that similar things exist in close proximity. In other words, similar things are near to each other. KNN captures the idea of similarity (sometimes called distance, proximity, or closeness) with some mathematics we might have learned in our childhood— calculating the distance between points on a graph. KNN’s main disadvantage of becoming significantly slower as the volume of data increases makes it an impractical choice in environments where predictions need to be made rapidly. Moreover, there are faster algorithms that can produce more accurate classification and regression results. [2]

Just like almost everything else, KNN works because of the deeply rooted mathematical theories it uses. When implementing KNN, the first step is to transform data points into feature vectors, or their mathematical value. The algorithm then works by finding the distance between the mathematical values of these points. The most common way to find this distance is the Euclidean distance, as shown below.

[2]

KNN runs this formula to compute the distance between each data point and the test data. It then finds the probability of these points being similar to the test data and classifies it based on which points share the highest probabilities.

[3]

This is a fundamental machine learning algorithm that is dependable for many reasons like ease of use and quick calculation time. It is a good algorithm to use when beginning to explore the world of machine learning, but it still has room for improvement and modification.

Advantages:

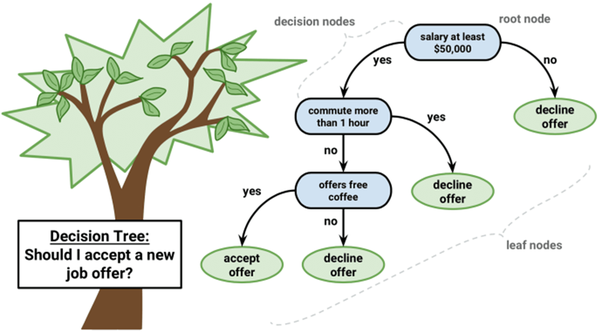
* K-NN algorithm is very simple to understand and equally easy to implement. To classify the new data point K-NN algorithm reads through whole dataset to find out K nearest neighbors.
* K-NN is a non-parametric algorithm which means there are assumptions to be met to implement K-NN. Parametric models like linear regression has lots of assumptions to be met by data before it can be implemented which is not the case with K-NN.
* K-NN does not explicitly build any model, it simply tags the new data entry based learning from historical data. New data entry would be tagged with majority class in the nearest neighbor.
* Very easy to implement for multi-class problem:Most of the classifier algorithms are easy to implement for binary problems and needs effort to implement for multi class whereas K-NN adjust to multi class without any extra efforts.
* One Hyper Parameter:K-NN might take some time while selecting the first hyper parameter but after that rest of the parameters are aligned to it.
* Variety of distance criteria to be choose from:K-NN algorithm gives user the flexibility to choose distance while building K-NN model.
  + Euclidean Distance
  + Hamming Distance
  + Manhattan Distance

Disadvantages:

* K-NN slow algorithm: K-NN might be very easy to implement but as dataset grows efficiency or speed of algorithm declines very fast.
* Curse of Dimensionality:KNN works well with small number of input variables but as the numbers of variables grow K-NN algorithm struggles to predict the output of new data point.
* K-NN needs homogeneous features: If you decide to build k-NN using a common distance, like Euclidean or Manhattan distances, it is completely necessary that features have the same scale, since absolute differences in features weight the same, i.e., a given distance in feature 1 must means the same for feature 2.
* Optimal number of neighbors: One of the biggest issues with K-NN is to choose the optimal number of neighbors to be consider while classifying the new data entry.
* Outlier sensitivity:K-NN algorithm is very sensitive to outliers as it simply chose the neighbors based on distance criteria.
* Missing Value treatment: K-NN inherently has no capability of dealing with missing value problem.

**4.4 Decision Tree**

In decision analysis, a decision tree can be used to visually and explicitly represent decisions and decision making. As the name goes, it uses a tree-like model of decisions. Though a commonly used tool in data mining for deriving a strategy to reach a particular goal, it’s also widely used in machine learning. A decision tree is drawn upside down with its root at the top.In the image on the left, the bold text in black represents a condition/**internal node,** based on which the tree splits into branches/**edges**. The end of the branch that doesn’t split anymore is the decision/**leaf**, in this case, whether the passenger died or survived, represented as red and green text respectively. [4]

[3]

Although, a real dataset will have a lot more features and this will just be a branch in a much bigger tree, but you can’t ignore the simplicity of this algorithm. The **feature importance is clear** and relations can be viewed easily. This methodology is more commonly known **as learning decision tree from data** and above tree is called **Classification tree** as the target is to classify passenger as survived or died. **Regression trees** are represented in the same manner, just they predict continuous values like price of a house. This algorithm is recursive in natureas the groups formed can be sub-divided using same strategy. Due to this procedure, this algorithm is also known as the **greedy algorithm**, as we have an excessive desire of lowering the cost. **This makes the root node as best predictor/classifier. [4]**

Classification : G = sum(pk \* (1 — pk))

A Gini score gives an idea of how good a split is by how mixed the response classes are in the groups created by the split. Here, pk is proportion of same class inputs present in a particular group. A perfect class purity occurs when a group contains all inputs from the same class, in which case pk is either 1 or 0 and G = 0, where as a node having a 50–50 split of classes in a group has the worst purity, so for a binary classification it will have pk = 0.5 and G = 0.5. [4]

Advantages:

* Simple to understand, interpret, visualize.
* Decision trees implicitly perform variable screening or feature selection.
* Can handle both numerical and categorical data. Can also handle multi-output problems.
* Decision trees require relatively little effort from users for data preparation.
* Nonlinear relationships between parameters do not affect tree performance.

Disadvantages:

* Decision-tree learners can create over-complex trees that do not generalize the data well. This is called overfitting.
* Decision trees can be unstable because small variations in the data might result in a completely different tree being generated. This is called variance, which needs to be lowered by methods like bagging and boosting.
* Greedy algorithms cannot guarantee to return the globally optimal decision tree. This can be mitigated by training multiple trees, where the features and samples are randomly sampled with replacement.
* Decision tree learners create biased trees if some classes dominate. It is therefore recommended to balance the data set prior to fitting with the decision tree.

**4.5 Random Forest**

The random forest is a classification algorithm consisting of many decisions trees. **It uses bagging and feature randomness when building each individual tree to try to create an uncorrelated forest of trees**whose prediction by committee is more accurate than that of any individual tree. Random forest, like its name implies, consists of a large number of individual decision trees that operate as an ensemble. Each individual tree in the random forest spits out a class prediction and the class with the most votes becomes our model’s prediction (see figure below).

[3]

The fundamental concept behind random forest is a simple but powerful one — the wisdom of crowds. In data science speak, the reason that the random forest model works so well is: A large number of relatively uncorrelated models (trees) operating as a committee will outperform any of the individual constituent models. The low correlation between models is the key. Just like how investments with low correlations (like stocks and bonds) come together to form a portfolio that is greater than the sum of its parts, uncorrelated models can produce ensemble predictions that are more accurate than any of the individual predictions.  The reason for this wonderful effect is that the trees protect each other from their individual errors (as long as they don’t constantly all err in the same direction). While some trees may be wrong, many other trees will be right, so as a group the trees are able to move in the correct direction. So the prerequisites for random forest to perform well are: There needs to be some actual signal in our features so that models built using those features do better than random guessing. The predictions (and therefore the errors) made by the individual trees need to have low correlations with each other. With a random forest model, our chances of making correct predictions increase with the number of uncorrelated trees in our model. So how does random forest ensure that the behavior of each individual tree is not too correlated with the behavior of any of the other trees in the model? It uses the following two methods: [5]

**1. Bagging (Bootstrap Aggregation) — Decisions trees are very sensitive to the data they are trained on — small changes to the training set can result in significantly different tree structures.**Random forest takes advantage of this by allowing each individual tree to randomly sample from the dataset with replacement, resulting in different trees. This process is known as bagging. Notice that with bagging we are not subsetting the training data into smaller chunks and training each tree on a different chunk. Rather, if we have a sample of size N, we are still feeding each tree a training set of size N (unless specified otherwise). But instead of the original training data, we take a random sample of size N with replacement. For example, if our training data was [1, 2, 3, 4, 5, 6] then we might give one of our trees the following list [1, 2, 2, 3, 6, 6]. Notice that both lists are of length six and that “2” and “6” are both repeated in the randomly selected training data we give to our tree (because we sample with replacement). [5]

**2. Feature Randomness —**In a normal decision tree, when it is time to split a node, we consider every possible feature and pick the one that produces the most separation between the observations in the left node vs. those in the right node. In contrast, each tree in a random forest can pick only from a random subset of features. This forces even more variation amongst the trees in the model and ultimately results in lower correlation across trees and more diversification. [5]

Advantages:

* Random Forests work well with both categorical and numerical data. No scaling or transformation of variables is usually necessary.
* Random Forests implicitly perform feature selection and generate uncorrelated decision trees. It does this by choosing a random set of features to build each decision tree. This also makes it a great model when you have to work with a high number of features in the data.
* Random Forests are not influenced by outliers to a fair degree. It does this by binning the variables.
* Random Forests can handle linear and non-linear relationships well.
* Random Forests generally provide high accuracy and balance the bias-variance trade-off well. Since the model’s principle is to average the results across the multiple decision trees it builds, it averages the variance as well.

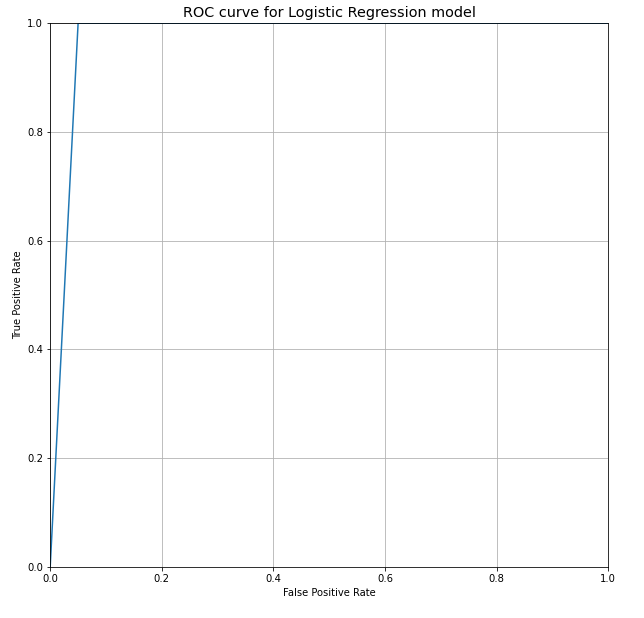
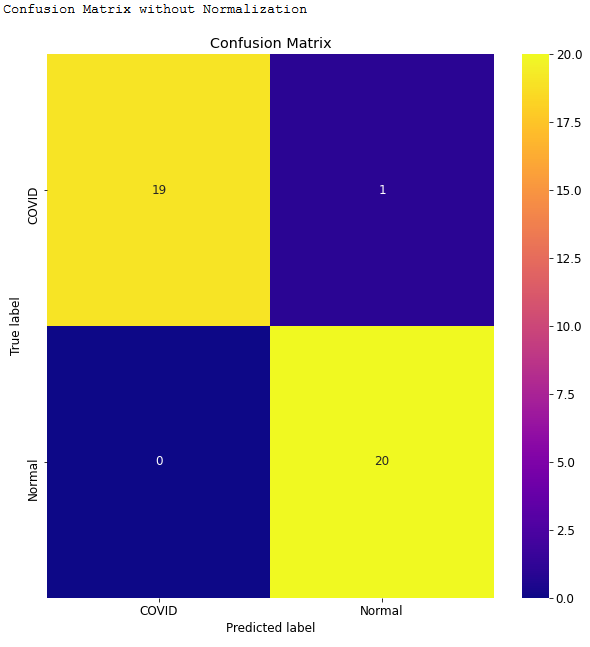
Disadvantages:

* Random Forests are not easily interpretable. They provide feature importance but it does not provide complete visibility into the coefficients as linear regression.
* Random Forests can be computationally intensive for large datasets.
* Random forest is like a black box algorithm, you have very little control over what the model does.

**Results**

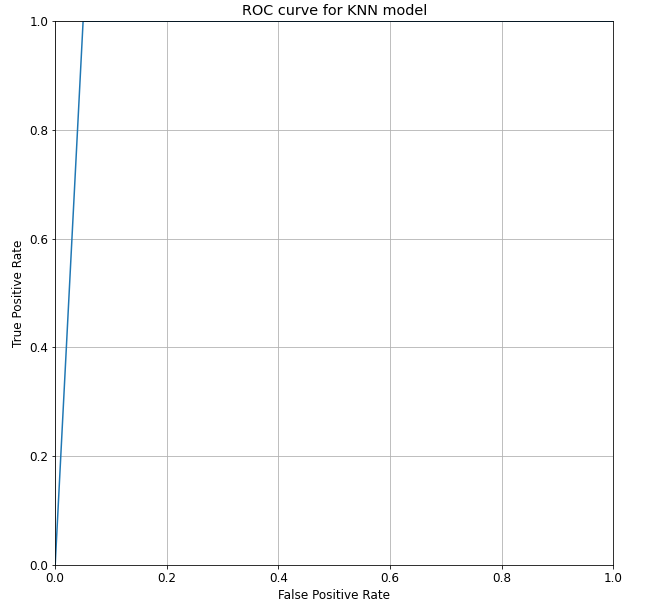
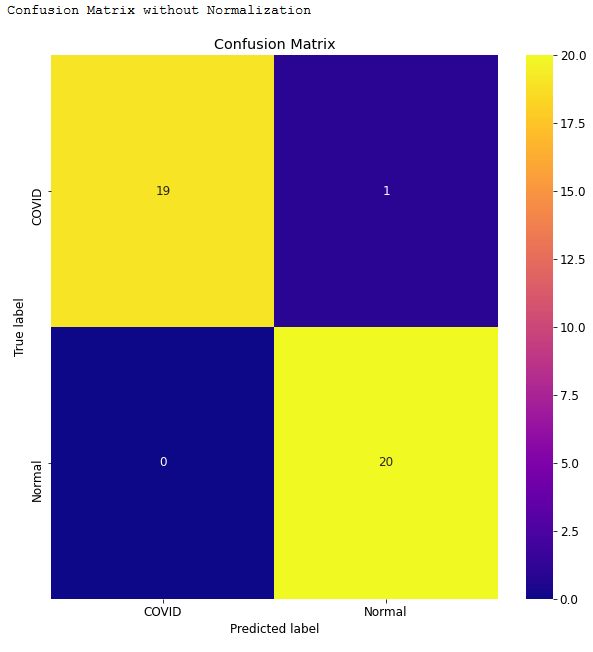
**5.1 Logistic Regression Model Results:**

* Training Time: ~13.85 s
  1. Total time was 1 min 37 s, however due to optimization seven hyper parameters were tested. The best hyper parameter being ‘C’:10.
* Prediction Time: 244 ms
* When the logistic regression model fit both the X\_train and y\_train (the 398 COVID/Normal training images and data labels), it was able to predict up to 97% accuracy when given the 20 COVID test image and 20 Normal test images.
* We are also obtaining 95% sensitivity and 100% specificity implying that:
  1. Of patients that do have COVID-19 (i.e., true positives), we could accurately identify them as “COVID-19 positive” 95% of the time using our model.
  2. Of patients that do not have COVID-19 (i.e., true negatives), we could accurately identify them as “COVID-19 negative” 100% of the time using our model.



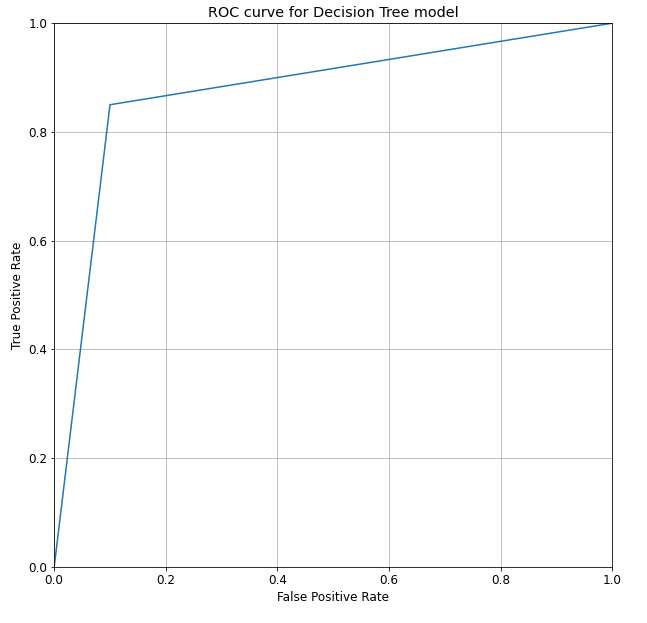
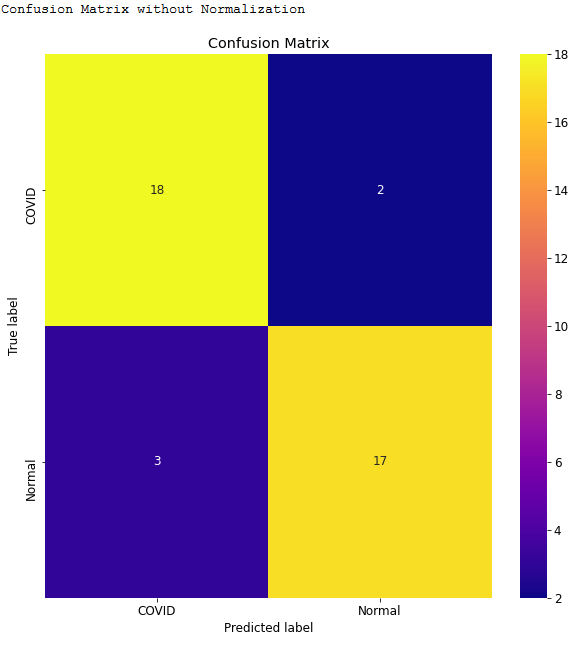
**5.2 K-neighbors Model Results:**

* Training Time: ~10.5 s
  1. Total time was 52.5 s, however due to optimization 5 neighbors and 2 metrics were tested. The best parameters being {'metric': 'manhattan', 'n\_neighbors': 3}.
* Prediction Time: 1.63 s
* When the optimized KNN model fit both the X\_train and y\_train (the 398 COVID/Normal training images and data labels), it was able to predict up to 97% accuracy when given the 20 COVID test image and 20 Normal test images..
* We are also obtaining 95% sensitivity and 100% specificity implying that:
  1. Of patients that do have COVID-19 (i.e., true positives), we could accurately identify them as “COVID-19 positive” 95% of the time using our model.
  2. Of patients that do not have COVID-19 (i.e., true negatives), we could accurately identify them as “COVID-19 negative” 100% of the time using our model.



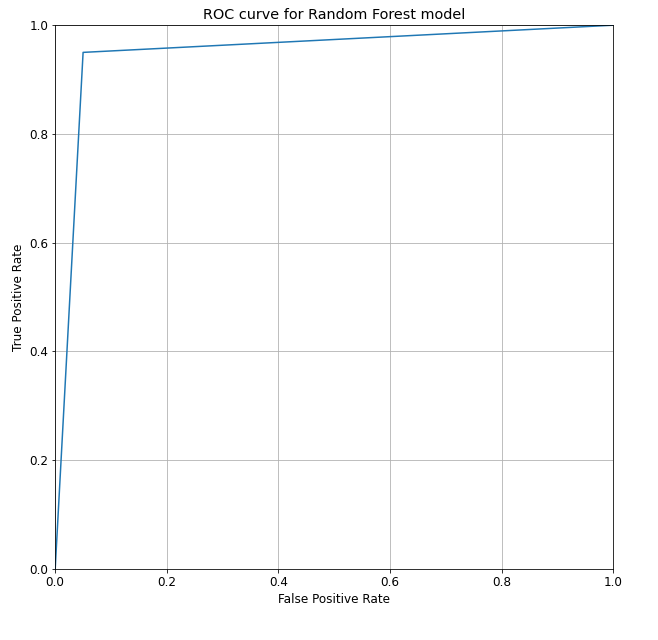
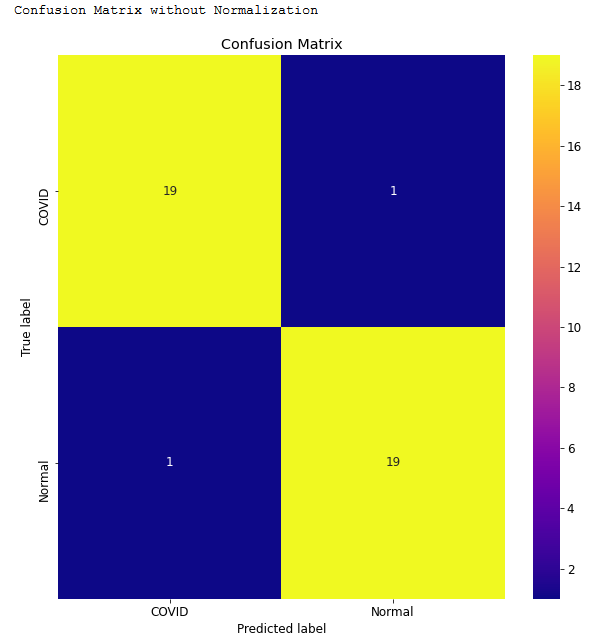
**5.3 Decision Tree Model Results:**

* Training Time: 5.65 s
  1. Optimization was not utilized for this model, due to time and hardware constraints.
* Prediction Time: 20.9 ms
* When the Decision Tree Model fit both the X\_train and y\_train (the 398 COVID/Normal training images and data labels), it was able to predict up to 85% accuracy when given the 20 COVID test image and 20 Normal test images.
* We are also obtaining 90% sensitivity and 85% specificity implying that:
  1. Of patients that do have COVID-19 (i.e., true positives), we could accurately identify them as “COVID-19 positive” 90% of the time using our model.
  2. Of patients that do not have COVID-19 (i.e., true negatives), we could accurately identify them as “COVID-19 negative” 85% of the time using our model.



**5.4 Random Forest Model Results:**

* Training Time: 1.49 s
  1. Optimization was not utilized for this model, due to time and hardware constraints.
* Prediction Time: 25.9 ms
* When the Decision Tree Model fit both the X\_train and y\_train (the 398 COVID/Normal training images and data labels), it was able to predict up to 95% accuracy when given the 20 COVID test image and 20 Normal test images.
* We are also obtaining 95% sensitivity and 95% specificity implying that:
  1. Of patients that do have COVID-19 (i.e., true positives), we could accurately identify them as “COVID-19 positive” 95% of the time using our model.
  2. Of patients that do not have COVID-19 (i.e., true negatives), we could accurately identify them as “COVID-19 negative” 95% of the time using our model.



**5.5 Utilized Machine Learning Algorithms’ Results Table**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **Logistic Regression (optimized)** | **K Nearest Neighbor (optimized)** | **Decision Tree** | **Random Forest** |
| Training Time | ~13.85 s | ~10.5 s | 5.65 s | 1.49 s |
| Prediction Time | 0.244 s | 1.63 s | 0.0209 s | 0.0259 s |
| Accuracy | 97.5% | 97.5% | 87.5% | 95% |
| Sensitivity | 95% | 95% | 90% | 95% |
| Specificity | 100% | 100% | 85% | 95% |

**Conclusion**

Based off the final results for each supervised learning machine learning algorithm, we can see that each model relatively did well; given the circumstances of the strict and concise data set provided. The optimized Logistic Regression model has been one of the two best models in accuracy, sensitivity, and specificity categories. Where it falls short is timing: the training time testing ‘C’: hyper parameters was averaged to around 14 seconds each, total of a minute and 37 seconds, the longest time taken compared to all models. Its prediction time was second to last with 244 milliseconds as well. The optimized K-Nearest Neighbor was the other best model in the accuracy, sensitivity, and specificity categories. As expected for optimization implemented models, the timing for training and predicting was slower in comparison. Due to hardware and timing constraints, I was unable to implement optimization for the remaining models like I had planned to. The Decision Tree model’s issue is lacking overall accuracy, sensitivity, and specificity. Its bright spot is possessing the quickest prediction time out of all the models. Lastly, the Random Forest model contains the best average for each of the categories. Quick training and prediction time, and overall 95% for accuracy, sensitivity, and specificity. I believe if the latter two models were optimized with proper hardware, better results would be produced for overall accuracy.

This project was an excellent learning device that enabled me to understand important concepts of Machine Learning algorithms, and how they can be utilized for image recognition among many other things. Hopefully someday a program like this can be expanded upon to recognize other diseases/viruses, and be utilized by medical professionals. For now, I

believe I can expand on this project to include Deep Learning algorithms or Convolutional Neural Networks to further my understanding on those concepts, and expand my knowledge as begin my career in the field of Software Engineering.

**References**

[1] “Chapter 5: Logistic Regression.” *Speech and Language Processing*, by Dan Jurafsky and James H. Martin, Prentice Hall, Pearson Education International, 2014.

[2] Raschka, Sebastian. “Nearest Neighbor Methods.” STAT 479: Machine Learning Lecture Notes. Department of Statistics, 2018, University of Wisconsin–Madison, Wisconsin.

[3] Varghese, Danny. “Comparative Study on Classic Machine Learning Algorithms.” *Medium*, Towards Data Science, 10 May 2019, towardsdatascience.com/comparative-study-on-classic-machine-learning-algorithms-24f9ff6ab222.

[4] “Decision Tree Learning.” *Machine Learning*, by Tom M. Mitchell, McGraw Hill, 2017.

[5] Louppe, Gilles. “Understanding Random Forests: From Theory to Practice.” *University of Liège*, Faculty of Applied Sciences Department of Electrical Engineering & Computer Science, 2014.

[6] Nazrul, Syed Sadat. “Basics of Image Recognition Using Python's Scikit-Learn Implementation.” *Medium*, Towards Data Science, 6 Apr. 2018, towardsdatascience.com/basics-of-image-recognition-using-pythons-scikit-learn-implementation-5efcff465b64.

[7] Pal, Mahendra, et al. “Severe Acute Respiratory Syndrome Coronavirus-2 (SARS-CoV-2): An Update.” *Cureus*, 2020, doi:10.7759/cureus.7423.

**Other References:**

Pant, Ayush. “Introduction to Logistic Regression.” *Medium*, Towards Data Science, 22 Jan. 2019, towardsdatascience.com/introduction-to-logistic-regression-66248243c148.

Harrison, Onel. “Machine Learning Basics with the K-Nearest Neighbors Algorithm.” *Towards Data Science*, towardsdatascience.com/machine-learning-basics-with-the-towardsdatascience.com/machine-learning-basics-with-the-k-nearest-neighbors-algorithm-6a6e71d01761-nearest-neighbors-algorithm-6a6e71d01761.

Gupta, Prashant. “Decision Trees in Machine Learning.” *Medium*, Towards Data Science, 12 Nov. 2017, towardsdatascience.com/decision-trees-in-machine-learning-641b9c4e8052.

Yiu, Tony. “Understanding Random Forest.” *Medium*, Towards Data Science, 14 Aug. 2019, towardsdatascience.com/understanding-random-forest-58381e0602d2.