**CMSE 492: FINAL PROJECT**

**Kritika Saini**

1. **INTRODUCTION**
   1. **Background**

Individuals who display symptoms that are consistent with COVID-19 and those who have had high-risk exposure to SARS-CoV-2 undergo testing for SARS-CoV-2 infection. Such testing employs either a nucleic acid amplification test or an antigen test to detect SARS-CoV-2. Testing may also be used for screening and determining the length of a patient's isolation period.

Several diagnostic tests for SARS-CoV-2 infection, including antigen tests like RT-PCR, have received Emergency Use Authorizations from the Food and Drug Administration (FDA). However, **none of these diagnostic tests have been approved by the FDA**. The diagnostic tests have been authorized for use in a variety of settings, including laboratory facilities, and point-of-care settings such as physician offices, pharmacies, and school clinics. These tests can be performed by trained personnel at or near the location where the specimen was collected.

The COVID-19 pandemic caused by the SARS-CoV-2 has significantly impacted the global healthcare system, overwhelming ICU bed capacities and resulting in high demand for testing. In an effort to slow the virus spread, the state of Sao Paulo, **Brazil**, implemented quarantine and social distancing measures like many other places. This dataset aims to predict the test result for SARS-Cov-2 (positive or negative) among suspected cases using anonymized data from patients seen at the **Hospital Israelita Albert Einstein** in Sao Paulo, Brazil. The dataset contains results of laboratory tests commonly collected during emergency room visits. The prediction model hopefully provides a practical and more efficient solution to the limitations of testing during an overwhelming healthcare system scenario. Another thing to remember is that, according to many sources, the accuracy of this test is not a 100%, so will we be able to look at other vitals of the body and say or classify the case as positive or negative.

* 1. **The Problem**

The SARS-CoV-2 test, like any other diagnostic test, has a possibility of turning out to be wrong. False negatives and false positives are both possible outcomes. A false negative means that the test result shows that the individual does not have the virus when in fact they do, while a false positive indicates that the individual is diagnosed with the virus when in fact they do not have it.

The accuracy of the test depends on various factors such as the type of test used, the timing of the test, and the severity of the infection. For example, the accuracy of the SARS-CoV-2 test may be lower during the early stages of the infection, as the virus may not have replicated enough to be detected by the test. Similarly, the accuracy of the test may vary based on the type of test used.

It is crucial to check the accuracy of the SARS-CoV-2 test because an incorrect diagnosis can have serious consequences. **False negatives** can lead to the **spread of the virus to others**, and **false positives** can lead to **unnecessary isolation, increased stress and anxiety**, and diversion of resources away from other patients who may need it more urgently. Hence, it is crucial to consider the limitations and potential errors of the test when interpreting the results.

* 1. **Possible Solution & Overview**

To achieve the objective of predicting SARS-Cov-2 test results, we will be utilizing various machine learning models, including Logistic Regression, Random Forest, K-nearest neighbor (KNN), Neural Networks (NN), Support Vector Machines (SVM), and Decision Trees. These models have been widely used in medical data analysis and have shown promising results. So, keeping this in mind, we will try to get the best possible outcome.

**Logistic Regression** is a widely used statistical method that predicts the probability of an event by fitting data to a logistic function. **Random Forest** is an ensemble learning method that constructs a multitude of decision trees at training time and outputs the mode of the classes as the prediction. **KNN** is a non-parametric algorithm that uses the similarity of feature vectors to classify a new data point. **NN** is a popular machine learning model that is composed of multiple layers of interconnected nodes and is trained to recognize patterns in data. **SVM** is a discriminative model that separates data into classes using a hyperplane in a high-dimensional space. Finally, **Decision Trees** are a popular algorithm that uses a tree-like model of decisions and their possible consequences to predict an outcome.

I will be comparing the performance of these models based on their **accuracy, precision, recall, and F1-score.** I will also be performing feature selection to identify the most important features in the dataset that contribute to predicting the SARS-Cov-2 test result. The results of this analysis could provide valuable insights into predicting the test result for SARS-Cov-2 and help in managing the pandemic by identifying suspected cases more accurately and efficiently.

1. **PRELIMINARY INVESTIGATION**

According to some very basic preliminary investigation, this was a pretty big dataset with its shape being (5644, 111). Even though there were so many columns, very few of the main ones were, Patient ID, Patient Age quantile, SARS-Cov-2 exam result, Patient admitted to regular ward, Patient admitted to semi-intensive unit, Patient admitted to intensive care unit, Haematocrit, Haemoglobin, Platelets, Mean platelet volume, and the list goes on.

Here are some of the basic graphs I had plotted to understand the overview of the dataset, this does not necessarily indicate anything, but just to get an overview. I plotted many graphs, but there is no point to cover that many, so here are just a few.

Graphical user interface

Description automatically generated

Figure 1: Correlation Matrix

Chart, histogram

Description automatically generated

Figure 2: Age quantile of Patients

These plots give us a very generic idea of the dataset. This does not prepare us for how he Machine learning models will do, so this is just a part of the procedure to get a basic understanding.

Based on preliminary investigation, I have found some interesting results also. The negative cases are predominantly more than the positive cases as we can see in the plot below which was plotted with the raw data initially.

Chart, bar chart

Description automatically generated

Figure 3: Old situation

The difference in margin between the results will be an issue for the ML project, so in order to prevent the bias this can possibly cause, I had to get the data in a more reliable format by either **generating synthetic data** or by **oversampling the minority data class** or **undersampling the majority data class**. In the method, we first separate the positive and negative cases in the dataset based on the target variable "SARS-Cov-2 exam result". Then, we up sample the minority class (positive cases) to match the number of instances in the majority class (negative cases) using the **resample()** function from the Scikit-learn library.

**Upsampling** involves randomly duplicating instances of the minority class until the number of instances in that class is equal to the number of instances in the majority class. By doing this, we balance the number of instances in each class, which can help improve the performance of machine learning models. Finally, we combine the upsampled positive cases with the negative cases using the **pd.concat()** function from the Pandas library to create a new balanced dataset. In summary, doing this helped us remove the class imbalance between the positive and negative cases by creating a new dataset with an equal number of instances in each class.

Chart, bar chart

Description automatically generated

Figure 4: Up sampled data

1. **MACHINE LEARNING MODELS**

Cleaning of data was done in a complicated manner because this is a huge dataset and almost all the rows had 1 or more missing values. This was a big challenge I faced. Another thing to get the dataset ready was to change the different datatypes of the entries, they were pretty scattered since this dataset was completely raw with no furnishing.

As mentioned before, I went about analyzing 6 different types of Machine Learning models, starting with Logistic Regression, Decision Trees, Random Forest, SVM, KNN and finally NN in the hope of finding the best fitting model that would solve our problem.

**3.1 Logistic Regression**

The output of the logistic regression is shown in the figure below.

Table

Description automatically generated

Figure 5: Logistic Regression Results

Chart, line chart

Description automatically generated

Figure 6: ROC curve plot

In the context of a binary classification problem, the logistic regression model has an **accuracy score of 54%**. This means that the model was able to correctly predict the outcome of 54% of the cases in the test dataset. The **precision score** for predicting negative cases is **53%.** This means that out of all the cases that the model predicted to be negative, only 53% were actually negative. The **recall score** for predicting negative cases is **68%**. This means that out of all the actual negative cases, the model was able to correctly identify 68% of them.

Similarly, the precision score for predicting positive cases is 56%. This means that out of all the cases that the model predicted to be positive, only 56% were actually positive. The recall score for predicting positive cases is 40%. This means that out of all the actual positive cases, the model was able to correctly identify only 40% of them. The F1-score is a harmonic mean of precision and recall scores and is used to measure the overall performance of the model. For this logistic regression model, the **F1-score** for predicting negative cases is **60%,** and for predicting positive cases, it is 47%.

The confusion matrix provides a tabular summary of the performance of a classification model on a set of test data.

Chart, treemap chart

Description automatically generated

Figure 7: Confusion Matrix

As we can see, the confusion matrix has two rows and two columns. The **rows** represent the **true classes** of the test set, while the **columns** represent the **predicted classes** of the model. The elements of the matrix count the number of samples that were predicted to belong to a certain class and actually belong to that class. In this confusion matrix, the top-left element (872) represents the number of **true negatives** (samples that were correctly predicted as negative by the model), the top-right element (401) represents the number of **false positives** (samples that were predicted as positive by the model, but actually belong to the negative class), the bottom-left element (759) represents the number of **false negatives** (samples that were predicted as negative by the model, but actually belong to the positive class), and the bottom-right element (511) represents the number of **true positives** (samples that were correctly predicted as positive by the model).

Summarizing this, while the model's overall accuracy is slightly better than random guessing, the precision and recall scores show that it is **not very good** at correctly predicting positive cases. This means that the model may incorrectly classify actual positive cases as negative, leading to a **false sense of security** and possibly contributing to the spread of the virus. Therefore, further improvements in the model's performance are needed to make it more reliable and accurate in predicting COVID-19 cases.

**3.2 Decision Trees**

This is the output of the Decision Tree ML model:

Table

Description automatically generated

Figure 8: DT Results

The results from the decision tree model show the performance of the model on the dataset. The **precision** for class 0 is **1.0**, meaning that out of all the instances that the model predicted as class 0, all were actually of class 0. The precision for class 1 is 89%, meaning that out of all the instances that the model predicted as class 1, **89% were actually of class 1**. **The recall** for class 0 is 88%, meaning that out of all the instances of class 0 in the dataset, the model was able to **correctly identify 88% of them**. The recall for class 1 is 1.0, meaning that out of all the instances of class 1 in the dataset, the model was able to correctly identify **100% of them**. The F1-score gives an overall measure of the model's performance. **The F1-score for class 0 is 0.93**, and for **class 1 it is 0.94**. The **accuracy** of the model on the test set is 94% meaning that it correctly classified **94%** of the instances in the test set.

As we know, it is not always easy to plot a decision tree, I tried to get something for a visual for this model, but you will be surprised to see the output since it is elaborated for every branch, but looking at it over here will give you absolutely no clarity. I am inserting an image here, just look at it, do not try to interpret it.

A picture containing map, text, light

Description automatically generated

Figure 9: Decision Tree Output

Overall, the results indicate that the decision tree model performed well on the dataset, with high precision, recall, and F1-score, and a high accuracy of 0.94 on the test set.

**3.3 Random Forest**

The output of the Random Forest Model has given below:

Table

Description automatically generated

Figure 10: RF Results

The **precision** for class 0 is **1.0**, meaning that out of all the instances that the model predicted as class 0, all were actually of class 0. The precision for class 1 is 90%, meaning that out of all the instances that the model predicted as class 1, **90% were actually of class 1**. **The recall** for class 0 is 90%, meaning that out of all the instances of class 0 in the dataset, the model was able to **correctly identify 90% of them**. The recall for class 1 is 1.0, meaning that out of all the instances of class 1 in the dataset, the model was able to correctly identify **100% of them**. The F1-score gives an overall measure of the model's performance. **The F1-score for class 0 is 0.95**, and for **class 1 it is 0.95.** The **accuracy** of the model on the test set is 94% meaning that it correctly classified **95%** of the instances in the test set.

Graphical user interface, application

Description automatically generatedChart, histogram

Description automatically generated

Figure 11: RF Diagram

The diagram is a **plot of the feature importance** of a Random Forest model. In Random Forest, feature importance is determined by how much the tree nodes that use a particular feature **reduce impurity** on average across all decision trees in the forest. Therefore, higher feature importance means that the feature is more important in making predictions. We can ignore percent of age quartile for this model. From the given diagram, we can see that the most important feature for making predictions is **"Leukocytes".** It has the highest feature importance score. Other important features include **"Platelets", "Monocytes", "Haemoglobin", and "Red blood Cells”.** Based on this, we can say that the Random Forest model gives a lot of importance to the levels of these blood components in predicting COVID-19 cases.

We can also see that the model is able to differentiate between important and not very important features as the feature importance scores decrease gradually from the most important feature to the least important feature. Overall, this suggests that the model is doing a good job of identifying the most important features for predicting COVID-19 cases.

**3.4 SVM**

The SVM model results are as follows:

Table

Description automatically generated

Figure 12: SVM Results

As much as I would hope this model gave us a better output, it did not, **the accuracy** of this model **is 51.98%** and I believe that is not worth looking into as we have other models that are offering a much better accuracy and result.

**3.5 KNN**

The result of KNN is provided below:

Table

Description automatically generated

Figure 13: KNN Results

For when the model predicts an instance to be in class 0, it is correct **98%** of the time. For class 1, the precision is **74%,** therefore, the model is **less precise** in predicting instances in **class 1**. On the other hand, recall measures the proportion of true positive predictions out of all actual positive instances (true positive & false negative). It tells us how many of the positive instances the model can correctly identify. For class 0, the recall identifies 65% of all instances correctly in class 0. For class 1, the recall is 99%, implying that the model is better at identifying instances in class 1. The F1-score is an overall measure of the model's accuracy. It ranges from 0 to 1, with 1 being the best. For class 0, the F1-score is 0.78, and for class 1, it is 0.85. The accuracy of the model is the proportion of correct predictions over all predictions, regardless of the class. The **accuracy is 82%**.

The macro avg is the average of the metrics for all classes, while the weighted avg is the weighted average of the metrics, weighted by the number of instances for each class. Here, the macro avg for precision, recall, and F1-score is 0.86, indicating that, on average, the model performs well across both classes. The weighted avg for these metrics is also 0.86, which means that the model's overall performance is not significantly affected by the class imbalance in the dataset.

A picture containing chart

Description automatically generated

Figure 14: Confusion Matrix for KNN

Based on the confusion matrix, it appears that the KNN model has predicted **1194 instances correctly** as class 0 (true negatives), 141 instances incorrectly as class 1 (false positives), 4 instances incorrectly as class 0 (false negatives), and 1204 instances correctly as class 1 (true positives). We can calculate the overall accuracy of the model by summing up the diagonal elements of the matrix and dividing it by the total number of instances. In this case, the total number of instances is **1194 + 141 + 4 + 1204 = 2543**, and the sum of the diagonal elements is 1194 + 1204 = 2398. Thus, the overall accuracy of the model is **2398/2543 ≈ 0.943 or 94.3%.**

While the overall accuracy seems relatively high, it is also important to evaluate the model's performance for each class. For example, we can calculate the precision and recall for each class. Precision measures the proportion of correctly predicted instances among all instances predicted as positive, while recall measures the proportion of correctly predicted instances among all actual positive instances.

For class 0, the precision can be calculated as 1194 / (1194 + 141) ≈ 89.4%, and the recall can be calculated as 1194 / (1194 + 4) ≈ 99.7%. This means that the model correctly predicted the majority of the instances in class 0, but it also misclassified some instances as class 1. For class 1, the precision can be calculated as 1204 / (1204 + 4) ≈ 99.7%, and the recall can be calculated as 1204 / (1204 + 141) ≈ 89.5%. This means that the model correctly predicted most of the instances in class 1, but it also misclassified some instances as class 0. Overall, the confusion matrix suggests that the **KNN model is relatively accurate** in predicting the class labels, but there may be some misclassifications, particularly for class 0.

**3.6 Neural Networks**

The result for NN is provided in the image below:

Table

Description automatically generated  
Figure 15: NN Result

The result shows the performance of a neural network model over **10 epochs**. Each epoch represents one complete iteration of the entire dataset through the neural network. From the output, we can see that **the loss is decreasing**, which indicates that the **model is learning from the data**. We can also see that the accuracy is improving over time, from 58.63% in the first epoch to 65.24% in the last epoch. However, the overall accuracy is still relatively low, at only 60.87%. This suggests that the model **may not be able to effectively** capture the patterns in the data, or that the dataset may not contain enough information for the model to accurately predict the target variable. Overall, this result provides insight into the performance of a neural network model on the given dataset, but further analysis and improvements may be needed to increase accuracy and obtain more reliable predictions.

1. **RESULTS**

The machine learning models were applied to the dataset to predict the SARS-Cov-2 test results and their accuracies were recorded. Based on the results, this is a summary of what the final rounded accuracies looked like:

1. Logistic Regression: 56%
2. Decision Tree: 94%
3. Random Forest: 95%
4. SVM: 52%
5. KNN: 82%
6. Neural Network: 61%

It can be observed that the Random Forest model had the highest accuracy of 0.95, followed by Decision tree at 0.94 and KNN with the accuracy of 0.82, then Neural Network at 61%. Logistic Regression had an accuracy of 0.56 and the SVM model had the lowest accuracy of 0.52.

The high accuracies obtained indicate that the dataset contains relevant features that can be used to accurately predict the SARS-Cov-2 test results. The results also suggest that the Random Forest model is the most effective in predicting the test results for SARS-Cov-2 among the models tested. Feature selection techniques could also be used to identify the most important features in the dataset that contribute to predicting the SARS-Cov-2 test results specifically to bring up the accuracy of the Random Forest Model. Overall, the results obtained provide valuable insights that could aid in managing the pandemic by identifying suspected cases more accurately and efficiently.

This model can be used in a real-life situation to help healthcare providers quickly identify COVID-19 cases among suspected cases based on laboratory test results. Early identification of COVID-19 cases can help with timely intervention and management, which is crucial in controlling the spread of the virus.

In emergency rooms, a commonly used SARS-CoV-2 test is the RT-PCR test, which detects viral RNA in respiratory specimens. Studies have shown that **false negatives can occur in up to 30% of cases**, leading to delayed diagnosis and potentially contributing to the spread of the virus. False positives can also occur due to contamination or other factors.

Therefore, having a model that can predict the accuracy of these tests based on laboratory results can be valuable in avoiding false results and ensuring that appropriate measures are taken in a timely manner. In addition, this model can be updated with new data and used to continuously monitor and improve the accuracy of COVID-19 testing in emergency rooms. Overall, this model can be used as a tool to support healthcare providers in making informed decisions and managing the COVID-19 pandemic more effectively.

1. **CONCLUSION**

The project aimed to predict confirmed COVID-19 cases among suspected cases based on laboratory tests commonly collected for a suspected COVID-19 case during a visit to the emergency room. The dataset consisted of 5644 observations and 35 features. We used six machine learning algorithms to build models and evaluate their performance in terms of accuracy, precision, recall, and F1-score.

Among the six models, Random Forest performed the best with an accuracy of 95%, followed by Decision Tree with an accuracy of 94%. The results indicate that the Random Forest and Decision Tree models could be promising for predicting COVID-19 cases among suspected cases using laboratory tests. These models could be used to improve the accuracy of COVID-19 diagnoses, which is crucial in curbing the spread of the virus.

Future work can involve improving the performance of the models further by feature engineering and selection, hyperparameter tuning, and ensemble methods. Additionally, collecting more data, especially from different locations and demographics, can help build more robust and generalizable models.

Overall, the project demonstrated the potential of machine learning in improving COVID-19 diagnoses using laboratory tests, and it provides a foundation for further research in this field. With further development and refinement, these models could aid healthcare professionals in detecting COVID-19 cases early and implementing necessary measures to prevent further spread of the virus.

1. **REFERENCES:**

*SARS-CoV-2 Testing*. (n.d.). COVID-19 Treatment Guidelines. https://www.covid19treatmentguidelines.nih.gov/overview/sars-cov-2-testing/

Sethuraman, N., Jeremiah, S. S., & Ryo, A. (2020). Interpreting Diagnostic Tests for SARS-CoV-2. *JAMA*, *323*(22). https://doi.org/10.1001/jama.2020.8259

Jegerlehner, S., Suter-Riniker, F., Jent, P., Bittel, P., & Nagler, M. (2021). Diagnostic accuracy of a SARS-CoV-2 rapid antigen test in real-life clinical settings. *International Journal of Infectious Diseases*, *109*, 118–122. https://doi.org/10.1016/j.ijid.2021.07.010

CDC. (2020, February 11). *Information for Laboratories about Coronavirus (COVID-19)*. Centers for Disease Control and Prevention. https://www.cdc.gov/coronavirus/2019-ncov/lab/resources/antigen-tests-guidelines.html

*Handling Imbalanced Data- Machine Learning, Computer Vision, NLP*. (2020, November 7). Analytics Vidhya. https://www.analyticsvidhya.com/blog/2020/11/handling-imbalanced-data-machine-learning-computer-vision-and-nlp/

1. **APPENDIX**

* This dataset was taken from the official WHO website for datasets: <https://www.who.int/data/collections>
* Github link (PUBLIC): <https://github.com/sainikri/CMSE492.git>