**COVID-19 Prediction Model Write-Up**

**David Lawler - A00075945**

## **Model Building and Method**

### **Dependencies**

The dependencies for the project include the following python packages and libraries: Datetime, Numpy, Pandas, SciPy, Scikit Learn, and Matplotlib. The project was implemented using Google Colab(Iwendi et al, 2019).

### **The Dataset**

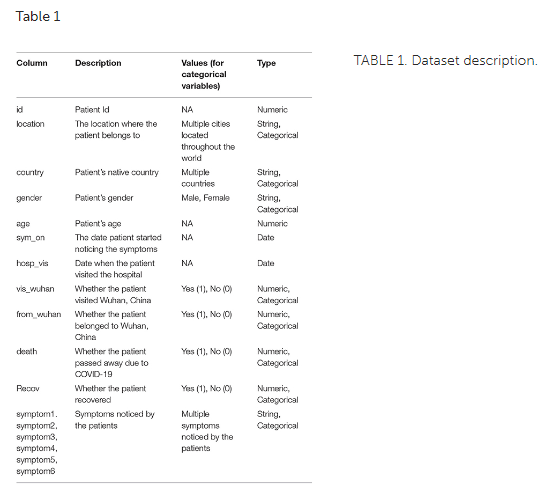
The dataset used in this study was accessed from Kaggle as “Novel CoronaVirus 2019 Dataset”.

<https://www.kaggle.com/datasets/sudalairajkumar/novel-corona-virus-2019-dataset>

The dataset has been compiled from various sources including the World Health Organization and John Hopkins University. However, this dataset was pre-processed further to meet the needs of the study Iwendi et al, 2019.

This dataset has daily level information on the number of affected cases, deaths and recovery from 2019 novel coronavirus/ It is composed of time series data - the number of cases on any given day is the cumulative number.

The dataset has the following features:



(Iwendi et al,. 2019)

### Data Pre-processing

* The dataset consists of columns with the data being the Date, String, and Numeric type.
* Categorical variables are label-encoded.
* Missing values are labeled “NA”
* Patient data records with missing values for both “death” and “recov” columns are separated from the main dataset and compiled into the test dataset.
* Feature engineering has been applied to create a new column corresponding (hosp\_vis—sym\_on) value, representing the number of days that have passed between symptoms being noticed and the patient's first visit to the hospital.

(Iwendi et al,. 2019)

### Evaluation Metrics

The purpose of the Iwendi et al,. 2019 study was to accurately predict the outcome of a particular patient depending on multiple factors, they considered the following evaluation metrics; accuracy, precision, recall, and F1 Score..

The following terms are used in the equations: TP, True Positive; TN, True Negative; FP, False Positive; and FN, False Negative.

**Accuracy** - Accuracy is used to assess the performance of the classification model. Given a dataset consisting of (TP + TN) data points, the accuracy is equal to the ratio of total correct predictions (TP + TN + FP + FN) by the classifier to the total data points.



**Precision** - Precision is a key metric to identify the number of correctly classified patients in an imbalanced class dataset. Precision is equal to the ratio of the True Positive (TP) samples to the sum of True Positive (TP) and False Positive (FP) samples.



**Recall** - Recall is a significant metric to identify the number of correctly classified patients. Recall is equal to the ratio of the True Positive (TP) samples to the sum of True Positive (TP) and False Negative (FN) samples.



**F1 Score** - The F1 Score strikes the perfect balance between Precision and Recall providing a correct evaluation of the model's performance in classifying COVID-19 patients. F1 Score is equal to the harmonic mean of Recall and Precision value.



## Results

The preprocessed dataset was used to train multiple machine learning classification models; the decision tree classifier, the support vector classifier, the gaussian naive bayes classifier, and the boosted random forest classifier.

|  | *The Decision Tree classifier has fairly good precision and Accuracy scores, but the lowest Recall score.* |
| --- | --- |

|  | *The Support Vector Machine also had a high Accuracy - the flaws of using this metric on this dataset will be discussed later.* |
| --- | --- |

|  | *The Naive Bayes Model had the lowest Precision score and equal first Recall score.* |
| --- | --- |

|  | *The Boosted Random Forest Model is the clear winner based on all metrics.* |
| --- | --- |

|  | *The decision tree has a depth of 2 and the Gini index of every node is <0.5, which indicates an imbalance in the training data.(Iwendi et al,. 2019)* |
| --- | --- |

## Issues or limitations of the prediction models and Comparison

**Considering performance metrics**

Accuracy measures the number of predictions that are correct as a percentage of the total number of predictions that are made. An issue with accuracy is that it is only useful when we have an equal distribution of classes on your classification. This means with an unbalanced dataset like ours accuracy becomes less useful.

Precision can be summarized as follows. Within everything that has been predicted as a positive, precision counts the percentage that is correct(*Joos Korstanje 2021)*. The tradeoff being that a not precise model may find a lot of positives - with a higher percentage incorrect, while an precise model might not find all the positives, but they are very likely to be correct.

Recall can be interpreted as; within everything that actually is positive, how many did the model succeed to find. A high recall model will find all positive cases in the data, though it may also identify some incorrectly, while a low recall model will not be able to find a large number of positive cases.

The goal of the F1 score is to combine the precision and recall metrics into a single metric(*Joos Korstanje 2021)*. Ideally, we would want both: a model that identifies all of our positive cases and at the same time identifies only positive cases(there is a trade off in reality). The F1 score has been designed to work well on imbalanced data; these two reasons are why it is the best metric for this dataset.

|  | *We see on the comparative performance figure below how Boosted Random Forest had the highest scores across all metrics - F1 Score, Accuracy, Precision, and Recall.* |
| --- | --- |

**Considering Classification Models**

The Decision Tree Model uses training data to create rules that can be represented by a tree structure. It has a root node, internal nodes, and leaf nodes. The internal node represents condition on attributes, the branches represent the results of the condition and the leaf node represents the class label. The decision tree model benefits from fast learning speed, missing value tolerance, it works well with interdependent attributes, has high explainability, and deals with overfitting. It can be inaccurate and intolerant of redundant attributes.

The Support Vector Machine is a common classification model that benefits from being highly accurate, having fast classification speed, being tolerant of irrelevant data, and being tolerant of redundant and interdependent attributes. It can also model linear and non-linear problems and deal with binary and continuous attributes. Drawbacks using this model include having a relatively slow learning speed, it can be overfit, and can have low explainability.

The Naive Bayes Classifier is based on Bayes rule that assumes conditional independence. Simple to Implement. The conditional probabilities are easy to evaluate. Very fast – no iterations since the probabilities can be directly computed. So this technique is useful where speed of training is important. In most situations, the features show some form of dependency.

The Boosted Random Forest algorithm performed best on all metrics. It consists of two parts; AdaBoost and the Random Forest classifier algorithm(Iwendi et al,. 2019). Random forest models create decision trees on randomly selected data samples, then it gets a prediction from each tree and selects the best solution by voting. Boosting increases the predictive power of classical decision and regression tree models. The Boosting algorithm is called a "meta algorithm" as it can’t learn nor predict anything since it's built on top of some other algorithm(Ajitesh Kumar 2020). The specific boosting algorithm used is called AdaBoost - where the data for the training is resampled and combined in an adaptive manner so the weights in the resampling are increased for those data points which got misclassified.

|  | *Overall, based on F1 Score, Boosted Random Forest performed best, followed by Logistic Regression, Decision Tree and SVM models. Gaussian NB was last.* |
| --- | --- |

## References

Celestine Iwendi, Ali Kashif Bashir, Atharva Peshkar, R. Sujatha, Jyotir Moy Chatterjee, Swetha Pasupuleti, Rishita Mishra, Sofia Pillai and Ohyun Jo. (2020, 07, 03). *COVID-19 Patient Health Prediction Using Boosted Random Forest Algorithm*

Source: https://www.frontiersin.org/articles/10.3389/fpubh.2020.00357/full

Joos Korstanje. (2021, 08, 31). The F1 score: *All you need to know about the F1 score in machine learning. With an example applying the F1 score in Python.*

Source: https://towardsdatascience.com/the-f1-score-bec2bbc38aa6

Ajitesh Kumar. (2020, 09, 09). *Adaboost Algorithm Explained with Python Example*

Source: https://vitalflux.com/adaboost-algorithm-explained-with-python-example/

GitHub - Atharva Peshkar. (2022). *Covid-19-Patient-Health-Analytics*

Source: https://github.com/Atharva-Peshkar/Covid-19-Patient-Health-Analytics/blob/master/LICENSE

Sudalaira Kumar. (2021). *Novel Corona Virus 2019 Dataset*

Source: https://www.kaggle.com/datasets/sudalairajkumar/novel-corona-virus-2019-dataset