Predicting COVID-19 Patients At Risk of Mortality or ICU

Kyren Whatmore
University of Canberra
Canberra, Australia
u3228127@uni.canberra.edu.au

Myeisha Foo
University of Canberra
Canberra, Australia
u3251507@uni.canberra.edu.au

***Abstract*— The COVID-19 pandemic caused immense pressure on healthcare systems worldwide. In this study, our goal is to predict patients with COVID-19 potentially at risk of mortality or admission into the ICU can strongly improve patient management and resource allocation within hospitals, reducing mortality rates. We use several machine learning models, logistic regression, linear regression and naïve bayes to find the most appropriate machine learning model for our goal. We utilised the 'COVID-19 Dataset' by Meir Nizri published on Kaggle to train and test against these models. We performed hyperparameter tuning and ensemble methods to further improve the accuracy of the models. Our results found that Logistic Regression with AdaBoost is the most balanced model for use, with 94% accuracy. Further work can be focused on improving model performance in the future.**

Keywords*— *Machine Learning, PRML, COVID-19, Logistic Regression, AdaBoost, Naïve Bayes, Linear Regression.

I. Introduction

Coronavirus disease (COVID-19) is an infectious disease caused by the SARS-CoV-2 virus that has changed the world [1]. With over 776,000,000 reported cases [2] and over 7,000,000 deaths worldwide, it has become one of the deadliest pandemics of all time. Many countries introduced enforced lockdowns as a response to the COVID-19 pandemic as an attempt to reduce the rapid spread of the virus. With no cure to the disease and despite these efforts, the unpredictability of high-risk cases has put immense pressure on healthcare systems worldwide. Timely identification of COVID-19 patients at elevated risk of mortality can significantly improve patient management and resource allocation within hospitals, reducing mortality rates [3]. The application of machine learning in health care has grown significantly in recent years. Machine learning can aid in prompt identification, potentially playing a crucial role and strongly aiding the healthcare industry.

In this study, we compare several machine learning models using a large COVID-19 dataset. We explore three different machine learning models, logistic regression, linear regression and naive bayes. Each model has its own distinct advantages- logistic regression for its efficiency, linear regression as a good baseline model and naive bayes for its efficiency with large datasets. On top of these model selections, we introduce hyperparameter tuning and ensemble methods to look to maximise the performance of these models. We aim to find the most effective model for use. The goal is to predict patients with COVID-19 potentially at risk of mortality or admission into the ICU, reducing mortality rates.

II. Reasoning

A. Methodology

For classification of high-risk patients at risk of mortality or admission into the ICU, we chose to use the following three models:

- Logistic Regression
- Linear Regression
- Naive Bayes

- Logistic Regression

Logistic regression is a supervised learning algorithm that predicts a dependent variable, given a set of independent variables.

$$log\ Y1-Y = b0+ b1X1 + b2X2 + ... + bnXn$$

Logistic regression utilises maximum like-hood for its estimation method, performs well on low-dimensional data and is efficient when the dataset has features that are linearly separable. It is a suitable model for this dataset, as the problem is

categorical and is a simple model to implement for this study.

- Linear regression

Linear regression is a model that predicts a dependent variable based upon the values of independent variables [4]. It is used when the dependent variable is categorical, which applies to this study.

$$Y = b0 + b1X + e$$

Using a least square estimation method, linear regression is a good simple baseline model to create a comparison with other algorithms.

- Naïve bayes

Naive Bayes is a conditional probability model based upon Bayes theorem that uses maximum likelihood to build the model. It ensures that the features are independent and makes equal contribution.

$$p(Ck|x) = p(Ck)p(x|Ck)p(x)$$

It is a suitable algorithm to apply to this dataset as it is efficient and one of the simplest supervised algorithms when working with large datasets and it is accurate and fast.

B. Dataset

The dataset used is the 'COVID-19 Dataset' by Meir Nizri [5], published on November 14th, 2022. We chose to use this dataset due to the high datapoint and feature count as it has 1,048,576 data points with 21 features each. This dataset was provided by the Mexican government and has an enormous number of anonymized patient-related information including binary pre-conditions, where 1 means 'yes' and 2 means no. Some pre-conditions included in this dataset includes 'Sex' (1 is female, 2 is male), 'Pregnant', 'Diabetes', 'COPD' (Chronic obstructive pulmonary disease), 'Asthma', 'Inmsupr' (Immunosuppressed), 'Hypertension', 'Other_Disease', 'Obesity', 'Tobacco' and others. These features are independent with the dependent variable being 'Mortality', a custom feature made from another feature, 'DATE_DIED'.

C. Data Visualisation

For visualisation of our data, we have chosen to use a variety of different plots to gain a wider understanding of our goal and the model. The following data visualisation techniques used are:

- Correlation Heatmap
- Bar plot
- Confusion matrix
- Scatterplot

D. Dataset Cleaning, Processing & Modelling

To be able to use this dataset, we must ensure it is ready to use. As the dataset already has values 97 and 99 as missing data, we will first convert all values containing 97 or 99 to null data. After this we check the whole dataset for null data, there are five features with null data, 'INTUBED', 'PNEUMONIA', 'AGE', 'PREGNANT' and 'ICU'. It is standard to assume all null data means no. Therefore, our next step is to convert all null data to '2' (no). Now the dataset has been cleaned.

When performing the data modelling, the main priority is on the 'MORTALITY' feature that we will create from the 'DATE_DIED' feature. The 'DATE_DIED' feature is useless for our problem, as it is the only feature that's data type is an object instead of an integer. When trying to convert the 'DATE_DIED' feature into an integer feature called 'MORTALITY', we first take note that if the patient has died, the entry will be their date of death (for example 21/06/2020). However, if the patient did not die, the entry will be '9999-99-99'. Then we will convert 'DATE_DIED' to 'MORTALITY' by converting all '9999-99-99' to '2' and all other entries to '1'.

We then define the features 'x', being all features besides 'DATE_DIED' and 'MORTALITY'. Then we define the target 'y', 'MORTALITY'. We can now create our base models.

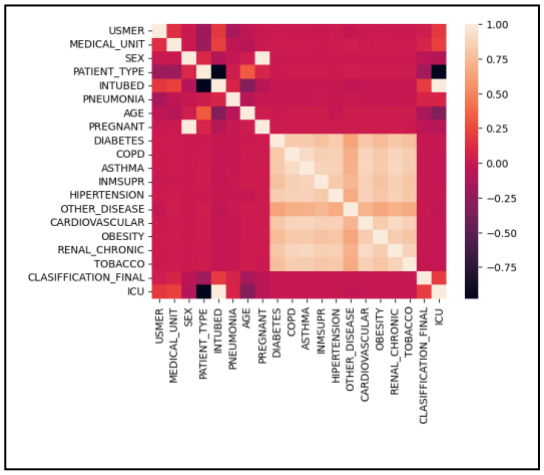

Fig 1: Correlation among all figures

III. Evolution Harness

To select the model, we first created all models both with an 80-20 training split (80% training data, 30% testing) and a 70-30 split to find the most accurate and appropriate model to move forward with. After analysis of the models, we decided that the 70-30 split is the better split in terms of accuracy, precision, recall, and ROC AUC. Based on the main aim of predicting mortality in COVID-19 patients, it seems that Logistic Regression as well as Naive Bayes are the best models to further analyse. Logistic regression showed a good balance between accuracy, precision, and AUC. On the other hand, Naive Bayes showed the highest recall and AUC, illustrating that it is good at finding patients at risk. After performing cross-validation for the Logistic Regression and the Naive Bayes models, we can conclude that Logistic Regression is well balanced in all metrics, especially performing well in precision and ROC AUC, displaying it is better suited if the main goal is to prioritise overall model performance along with reducing false positives. Naive Bayes excels in recall, making it an excellent choice if the priority is identifying as many at-risk patients as possible and that it doesn't matter that it will have a lot of false positives. The results from cross-validation shows that both models do not overfit.

IV. Model Analysis

To further optimise our models to improve results, we used hyperparameter tuning (GridSearchCV) and an ensemble method (AdaBoost). When conducting hyperparameter tuning (using GridSearchCV), the model was optimised for each specific metric (accuracy, precision, recall, ROC AUC) based on cross-validation. The hyperparameter tuning results provided show that Logistic Regression performed very well across different metrics, especially in terms of accuracy (94.58%) and ROC AUC (95.73%). Similarly, Naive Bayes had strong results for recall (73.54%) and ROC AUC (93.06%).

With the ensemble method, the goal is different. AdaBoost focuses on improving model performance by iteratively correcting errors from earlier models. It doesn't directly optimise the same hyperparameters as GridSearchCV does. In this case, AdaBoost didn't outperform the hyperparameter-tuned models in some areas (e.g., precision, recall, and F1-score). While it showed strong results for ROC AUC and accuracy, it didn't significantly improve over the hyperparameter-tuned results.

We can take away that Logistic Regression with Hyperparameter Tuning is performing very well, especially with accuracy (94.58%) and ROC AUC (95.73%). If AdaBoost doesn't significantly improve upon these results, it suggests that the hyperparameter-tuned model is already optimal. Naive Bayes with Hyperparameter Tuning also did well in terms of recall (73.54%) and ROC AUC (93.06%), which aligns with the fact that Naive Bayes typically focuses on maximising recall at the expense of precision. Again, AdaBoost didn't outperform it because the tuned Naive Bayes was already strong.
We will stick with hyperparameter tuned models and choose to use AdaBoost selectively in the future.

V. Classification Report

To finalise the outcomes of the chosen learning models, we evaluated each one using the classification reports after both original training, hyperparameter tuning, and applying ensemble methods like AdaBoost. The results for the different models are as follows:

- Logistic Regression:
  Original Model:
  - Precision: 0.69

- Recall: 0.48
- F1-Score: 0.57
- Accuracy: 0.95

Tuned Model:

- Precision: 0.69
- Recall: 0.48
- F1-Score: 0.57
- Accuracy: 0.95

AdaBoost + Logistic Regression:

- Precision: 0.64
- Recall: 0.50
- F1-Score: 0.56
- Accuracy: 0.94

The original model showed strong precision (0.69) meaning it was good at avoiding false positives. The tuned model shows that after hyperparameter tuning, the results remained the same, suggesting that the original hyperparameters were already well-suited for the data. When combined with AdaBoost, Logistic Regression's performance slightly decreased in terms of precision (0.64) but improved recall (0.50). This trade-off implies that AdaBoost helped identify more at-risk patients (higher recall), but at the cost of slightly more false positives (lower precision).

- Naive Bayes:

  Original Model:
  - Precision: 0.38
  - Recall: 0.73
  - F1-Score: 0.50
  - Accuracy: 0.89

  Tuned Model
  - Precision: 0.38
  - Recall: 0.73
  - F1-Score: 0.50
  - Accuracy: 0.89

  AdaBoost + Naive Bayes
  - Precision: 0.25
  - Recall: 0.96
  - F1-Score: 0.40
  - Accuracy: 0.79

The original Naive Bayes model was very strong in terms of recall (0.73), meaning it was better at identifying at-risk patients. Similar to Logistic Regression, hyperparameter tuning did not significantly alter Naive Bayes' results, indicating that the original configuration was already well-optimised for the dataset. When combined with AdaBoost, Naive Bayes' recall skyrocketed to 0.96, meaning it caught almost all at-risk patients. However, precision dropped further to 0.25, resulting in a significant increase in false positives.

The best performing model and the recommended model for this project is Logistic Regression with AdaBoost as it provides the best balance between precision and recall, while also maintaining a high accuracy (0.94). Although Naive Bayes with AdaBoost excels in recall, it suffers from significantly lower precision and accuracy, making it more prone to overfitting and producing too many false positives.
Moreover, Logistic Regression with AdaBoost is likely to perform better on unseen data, as it doesn't show signs of overfitting like Naive Bayes with AdaBoost. This ensures the model can generalise better to new cases, making it more reliable for real-world applications, where both false positives and false negatives have real consequences.

- Confusion Matrix

Confusion matrices provide insight into how well the model performed by showing the number of true positives, true negatives, false positives, and false negatives. These values help in understanding the balance between precision and recall.

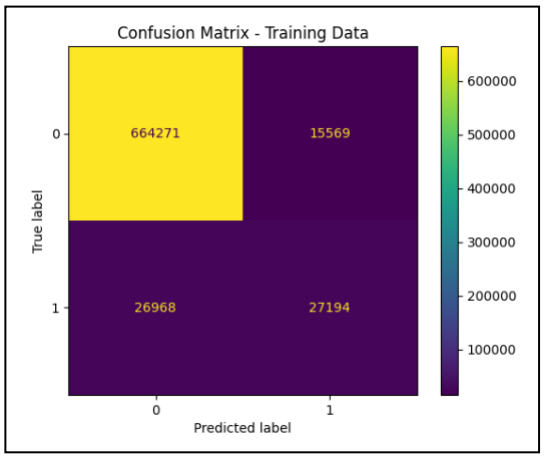

Fig 2: Confusion Matrix for Training Data

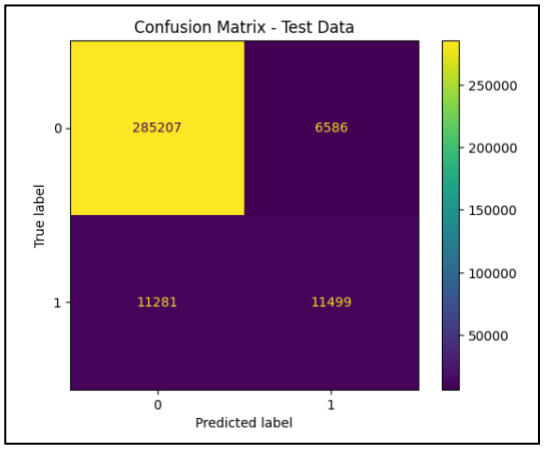

Fig 3: Confusion Matrix for Test Data

Positive Insights: The confusion matrices reveal that the model supports a strong level of accuracy in both training and test datasets. The number of true positives and true negatives shows that the model performs well across both classes. Although there are false positives and false negatives, these are relatively minor compared to the total cases, meaning the model effectively balances identifying both at-risk and non-risk patients.

- Scatterplot

PCA (Principal Component Analysis) reduces the dataset's dimensions, making it easier to visualise high-dimensional data in a 2D or 3D space. The scatter plots below visualise the model's predictions in the PCA-reduced feature space.

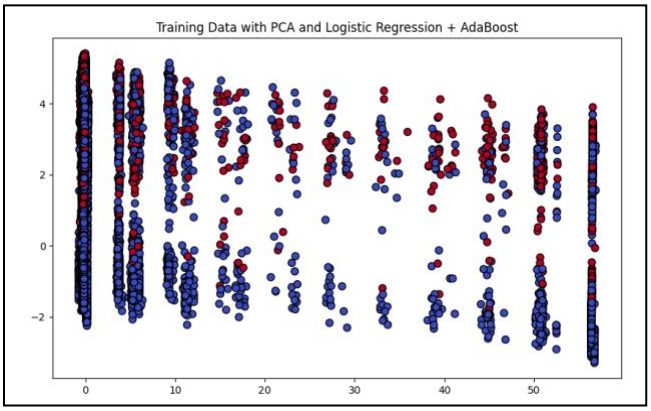

Fig 4. Scatterplot with PCA, Logistic Regression + AdaBoost for Training Data

In these plots, red dots stand for the at-risk patients (1), while blue dots stand for non-risk patients (0). The scatterplot shows that the model is generally able to separate the at-risk patients from non-risk patients quite well. While there is a slight overlap in certain areas, this is expected in high-dimensional data. This shows that the model has learned useful decision boundaries between the two classes, especially in the training data.

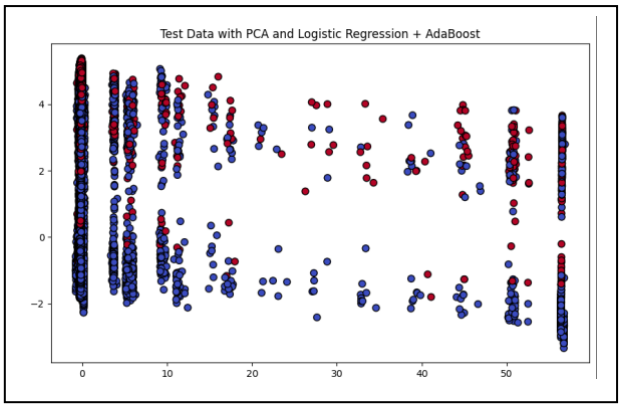

Fig 5. Scatterplot with PCA, Logistic Regression + AdaBoost for Test Data

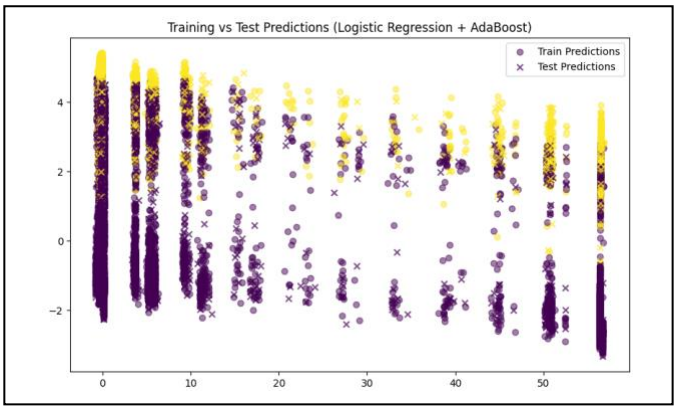

Fig 6. Scatterplot of Training vs Test Predictions for Logistic Regression + Adaboost

The final plot overlays the training and test predictions using PCA. Training predictions are represented as yellow circles, and test predictions are purple crosses. The overlap of yellow and purple points shows consistency between training and test predictions, meaning the model generalises well. Areas where the points align show impressive performance on both seen and unseen data. The visual separation in many regions confirms that the model is learning patterns that work for both datasets, suggesting it is robust against overfitting and generalises well to new examples.

These visualisations affirm that the model is performing well overall. While some minor misclassifications remain, the potential for

improvement through further tuning is promising, particularly in enhancing its precision.

VI. CHANGE

We chose to use the optimised AdaBoost + Logistic Regression model to make predictions on the test dataset (which serves as unseen data). The test dataset was scaled and transformed using PCA (Principal Component Analysis) to match the input requirements of the model. Predictions were then made on this unseen data, and several key evaluation metrics were calculated, including Accuracy, Precision, Recall, F1-Score, and the ROC AUC Score.

```
Performance on Unseen Data (Test Set):
Accuracy: 0.9233373493592902
Precision: 0.9082749958095288
Recall: 0.9233373493592902
F1-score: 0.9137529332236494
ROC AUC Score: 0.930663091359397
```

Fig 7. Performance scores on unseen data

TABLE I. CLASSIFICATION REPORT

	Precision	***Recall***	***F1-Score***	***Support***
0	0.94	0.98	0.96	291793
1	0.45	0.26	0.33	22780
Accuracy			0.92	314573
Macro Avg	0.70	0.62	0.64	314573
Weighted Avg	0.91	0.92	0.91	314573

The results from the unseen test dataset provide insights into how well the optimised model generalises to new data. The Accuracy of 92.33% shows that the model performs well overall. However, there is room for improvement in the Recall and F1-Score for predicting at-risk patients. While the optimised model performs well on unseen data, there is potential to improve recall for at-risk patients, which is crucial for healthcare-related predictions. Future steps could involve further tuning or balancing the trade-off between precision and recall.

VII. SAVING MODEL

By using the pickle module, we can save the model through the function pickle.dump. The file then gets saved to the local computer, where it can be used in the future. As we are using Colab, the file is saved to the local file system on Colab. To have the model on the local computer, we import the relevant Colab module to then download the model to the local computer.

To use the model to predict future unseen data, we first will upload the model by using pickle.load. Now the model is accessible and can be used to predict unseen data.

We can further tune this model to produce higher accuracy rates by including new data rather than creating a new model from scratch. By using further hyperparameter tuning and added ensemble methods with the new data, it will provide greater accuracy and performance from the model for future use.

VIII. ETHICS

When working with extremely sensitive information, ethical and privacy issues will appear that have to be evaluated. Recognising patient privacy, the potential of harm and following The Privacy Act 1988 will ensure we address these concerns.

It is the top priority to protect the privacy of the patients. By utilising anonymisation, encryption and other techniques to improve data security, it will prevent unwanted breaches of data. The model is valuable to aid in morality detection, but it is important to recognise to reduce harm that it is only a tool to help. Even with its high accuracy, the model shouldn't be used as the primary reason to determine a result, as the model can present false positives.

Following The Privacy Act 1988 ensures patient data is correctly and appropriately managed, protecting patient's information.

IX. CONCLUSION

This study compared three models in finding COVID-19 patients potentially at risk of mortality or admission into the ICU. By using the 'COVID-19 Dataset', we have shown the capability of

machine learning to significantly improve patient management and aid in resource allocation. We found that Logistic Regression with Adaboost appeared as the best choice model, achieving a balance between precision, recall, accuracy and ROC AUC. Future work can be focused on improving model performance by performing further parameter tuning.

REFERENCES

[1] World Health Organization, "Coronavirus Disease (COVID-19)," World Health Organization, 2024. https://www.who.int/health-topics/coronavirus#tab=tab_1

[2] World Health Organization, "COVID-19 cases | WHO COVID-19 dashboard," *datadot*, 2024. https://data.who.int/dashboards/covid19/cases

[3] D. Bertsimas *et al.*, "COVID-19 mortality risk assessment: An international multi-center study," *PLOS ONE*, vol. 15, no. 12, p. e0243262, Dec. 2020, doi: https://doi.org/10.1371/journal.pone.0243262.

[4] Dr. M. Wang, "PRML_lecture_week4_regression and evaluation.pdf."

[5] "COVID-19 Dataset," *www.kaggle.com*. https://www.kaggle.com/datasets/meirnizri/covid19-dataset

[6] scikit-learn: AdaBoostClassifier, scikit-learn.org, https://scikit-learn.org/dev/modules/generated/sklearn.ensemble.AdaBoostClassifier.html

[7] scikit-learn: Logistic Regression, scikit-learn.org, https://scikit-learn.org/1.5/modules/generated/sklearn.linear_model.LogisticRegression.html

[8] Gaussian Naive Bayes, Builtin.com, https://builtin.com/artificial-intelligence/gaussian-naive-bayes#:~:text=Gaussian%20Naive%20Bayes%20is%20a%20machine%20learning%20classification%20technique%20based,of%20predicting%20the%20output%20variable

[9] scikit-learn: Naive Bayes, scikit-learn.org, https://scikit-learn.org/1.5/modules/naive_bayes.html

[10] Pickle: Python Object Serialization, docs.python.org, https://docs.python.org/3/library/pickle.html

[11] Min Wang, PRML Lecture Week 3: Data and Visualization, 2024.

[12] Min Wang, PRML Lecture Week 5: Estimation and Regularization, 2024.

[13] Min Wang, PRML Lecture Week 6: Bayesian Classifiers, 2024

[14] Min Wang, PRML Lecture Week 7: kNN and Decision Tree, 2024.

[15] Min Wang, PRML Lecture Week 9: Support Vector Machine, 2024.

[16] Min Wang, PRML Lecture Week 10: Neural Networks, 2024.