Classification Models for COVID-19 Severity

Rebecca Quay Dragon **Shiln Max Zhou** **Tom Wiseman**

Abstract

The recent spread of the COVID-19 virus has placed an incredible strain on the worldwide healthcare system. Intelligent allocation of limited medical resources during a crisis is critical for preventing a collapse of the healthcare system and treating the greatest number of people possible. This project aims to assist hospitals allocate limited beds, personal protective equipment, and staff during the early stages of a developing outbreak. Given a dataset of COVID-19 patients that includes demographic data, medical history and a time series of vitals/blood test results, we build several predictors that predict whether or not that patient will need ICU care in the near future. The classifiers we built were based on Multi-tiered Logistic Regression, Multi-tiered Ridge Regression, k-nearest neighbors, and multi-layer perceptron. The results of these predictors can be used to triage individual COVID cases, even before severe symptoms may become apparent.

1 Data pre-processing

1.1 Dataset Overview

- The dataset is an analysis of 384 patients admitted to the hospital after a positive COVID-19 test. For each patient, basic demographic information and medical history was recorded (i.e. Age, gender, and 'Disease Grouping'). Additionally, blood tests and vital signs were recorded over a 10-12 hour time series, every two hours.
- We aim to predict if at any point, the patient will need to go to the ICU for advanced care. We also aim to predict this as early as possible (i.e. with as little time series data as possible) so that in a clinical setting hospital staff can make decisions as soon as possible
- Source: Kaggle.com. See References for full URL.

1.2 Data Preparation and Exploratory Data Analysis

The original dataset was prescaled between -1 and 1, and the data was balanced between target groups. We encoded several features with a one-hot encoding scheme. Age percentiles were mapped to whole numbers and scaled, since we expected age to have a linear impact on the target variable. Patients with large chunks of missing data were discarded. Missing datapoints in timeseries data were filled in with approximations using the mean of related data. Both the population mean and mean of other measurements of that individual patient were considered. Through early visualizations of data, we were able to confirm our hypothesis that certain features were indeed predictive of COVID severity. Data was randomly separated into training and tests sets before the training of any models.

2 Model Training and Validation

2.1 Logistic Regression and Ridge Regression

Classifying individuals likelihood of needing ICU care is done by monitoring these patients over a period of time. The data set provides measurements in two hour time increments, a 6 hour time block,

35th Conference on Neural Information Processing Systems (NeurIPS 2021).

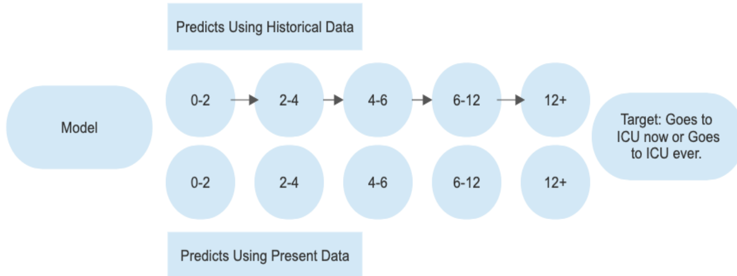

Figure 1: Model Flow Example

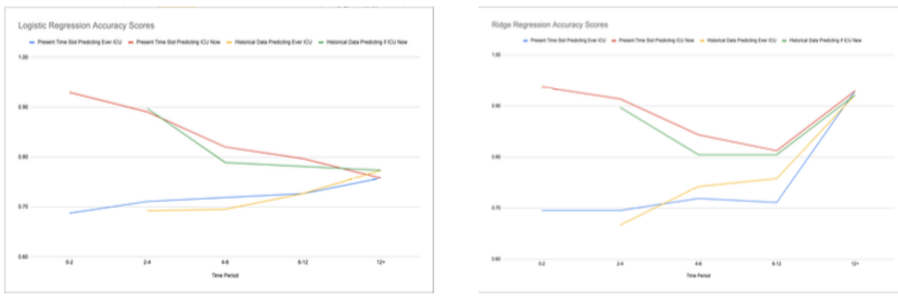

Figure 2: Ridge and Logistic Regression Accuracy

and finally an ambiguous amount of time for 12+ hours. The target data indicates when the patient is admitted into the ICU. It is important to model a prediction for both when the patient needs ICU care, and if the patient will ever need ICU care.

Logistic regression and ridge regression were used, with two separate data sets to model both historical patient data as well as present data for the time slot. The historical data encompassed the past medical measurements for that patient while the present data only included the measurement for that time period. If that patient had been previously admitted to the ICU they were excluded from the data set. This is because doctors are trying to preemptively predict whether or not the patient will need the ICU. An initial ridge regression was performed using all of the data for the targets of 'if ever' going to the ICU and going to ICU in that time block. The 'if ever' ICU performed at 74% accuracy, 69% precision, and 72% recall. The ICU for that time block performed at 84% accuracy, 28% precision, and 79% recall. Thus creating a higher false positive rate. This is better for the safety of the patient but could cause overflow in the ICU. Below are the explanation and results of our tiered model.

In theory, this results in four separate models. Two tiered ridge regression models one of which predicts whether the patient will be admitted into the ICU during the allotted time slot or ever be admitted to the ICU and two tiered linear regression models which predict for the allotted time slot and if the patient will ever be sent to the ICU.

Results:

Ridge Regression changes the features used in the later stages and therefore is a more dynamic and accurate model. The logistic regression features were decided in EDA and remained constant.Present data predicting whether or not a person will go to the ICU in the current time slot performs the best. Predicting 'if ever' admitted to the ICU remains consistent in the 70s for accuracy for both logistic regression and ridge regression until the 12+ time slot.Using the historical data to predict the present ICU status is not better than the present data. However, the precision and recall/precision scores trail or are 0 for the present data in the 2-4 time period because all points are assigned to 0 and true values are mostly 0. This means that this subset of the larger model has not been properly tested.

Unlike the models below, examining coefficient values ridge regression was able to elucidate which features had the biggest impact on predictive power. Across all four models, the most relevant features turned out to be respiratory rate, blood pressure, blood lactate mean (a indirect measure of how much

Ridge Regression - Accuracy

Time Period	Present Predict Ever	Present Predict Now	Historical Predict Ever	Historical Predict Now
0-2	0.70	0.94		
2-4	0.70	0.91	0.67	0.90
4-6	0.72	0.84	0.74	0.80
6-12	0.71	0.81	0.76	0.80
12+	0.93	0.93	0.92	0.92

Logistic Regression - Accuracy

Time Period	Present Predict Ever	Present Predict Now	Historical Predict Ever	Historic Predict Now
0-2	0.69	0.93		
2-4	0.71	0.89	0.69	0.90
4-6	0.72	0.82	0.70	0.79
6-12	0.73	0.80	0.73	0.78
12+	0.76	0.76	0.77	0.77

Figure 3: Accuracy

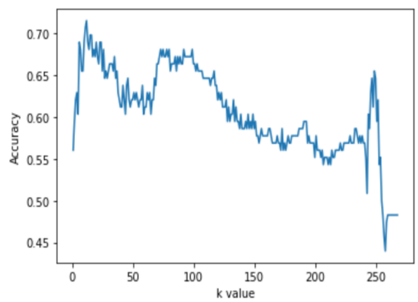

Figure 4: Feature: Relationship between k value and the prediction Accuracy

oxygen cells are getting), temperature, lymphocyte count (a measure of immune system activity), PCR mean (a measure of the amount of COVID in a blood test), blood oxygen levels, sodium levels, and calcium levels. In future models, it would be interesting to examine how far one could get with these features alone.

2.2 K-nearest Neighbors Classification and Principal Component Analysis

Since our dataset has 86 columns or features, we must first reduce our dataset dimension by using PCA (Principal Component Analysis), because the kNN classifier makes the assumption that similar points share similar labels, but in high dimensional spaces, points that are drawn from a probability distribution, tend to never be close together. So, we imported PCA from sklearn.decomposition and use pca = PCA(nComponents=5) to reduce our dataset dimension to only keep five features for later training process, nComponents can be from 1 to the number of the features, but after we tried different values, we found 5 features can be best to get the high accuracy prediction result.

After we have our new reduced dataset, we can implement our kNN model. We used the kNN classifier KNeighborsClassifier from sklearn.neighbors. We created a method which can get different accuracy results by testing different k values. In this method, getKnn(neighbor, trainData, testData, trainLabel, testLabel), neighbor can be set from 1 to the length of the training dataset which is 269 in our case. We generated a plot comparing the relationship between number of neighbors (k value) and prediction accuracy.

2.3 Multilayer Perceptron Classifier

We trained a multilayer perceptron to predict if patients would ever need ICU care. After much tuning, we determined that a 5 layer classifier would provide the best results. Our classifier had 5 layers total: The input layer took in 84 features, there were three hidden layers of 10, 8, and 2 neurons each, and the output layer output either 0 or 1 (the prediction). Due to the fact that we wished to predict the probability that a patient would be in a particular group, the sigmoid activation function was the clear winner. The MLPClassifier from the sklearn.neuralnetwork package was used. The model was trained on the entire dataset with every entry in the time series considered as an individual datapoint. If entries in a particular patient's time series were missing, those entries were filled in with

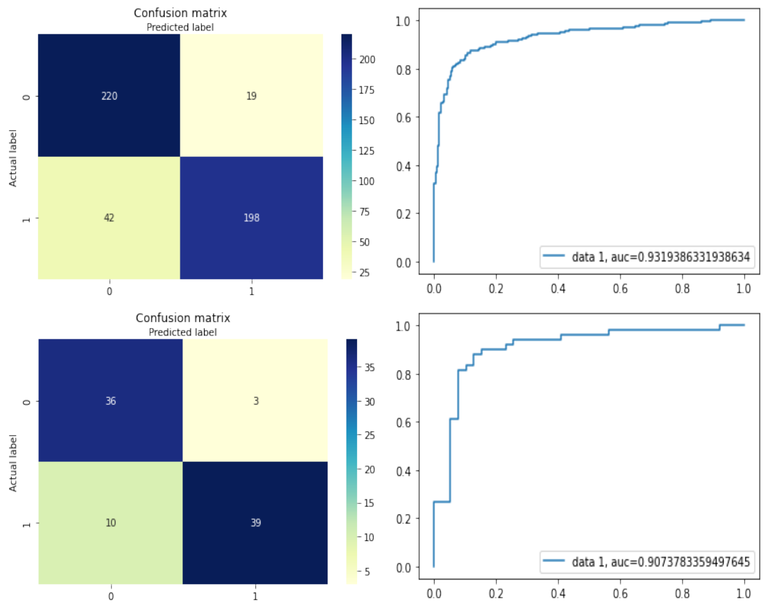

Figure 5: Top: Confusion Matrix and ROC Curve for network tested on all test data (Accuracy 87%, Precision 91%, Recall 83%). Bottom: Confusion Matrix and ROC Curve for network tested on first-entry timeseries data (Accuracy 85%, Precision 93%, Recall 80%).

the mean of other entries for that same patient. Because we wish this tool to be useful in a clinical setting, it was evaluated on the test data as a whole, and a subset of the test data that only included time entries from the first time window immediately after patients entered the hospital. The first window measurement is much more useful in practice, since hospital staff can get immediate results rather than having to take several measurements over several hours. Thankfully, the first time-series entry proved to be about as accurate (85%) as the average time series entry (87%), which implies that data taken immediately upon arrival to the hospital is not drastically different from data collected hours after admission. One interesting extension of this project would be to encode the time-series data in a way so that the network could consider each patient as a whole and also incorporate how an individual patient's measurements changed over time as a prediction parameter.

3 Comparison of Models

Both regression models performed about as well as one another, with ridge regression coming out ahead in certain situations. By penalizing large coefficients, ridge regression was able to create a more generalize-able/less overfit model than logistic regression alone. Ridge regression had an overall accuracy of about 75% when predicting if a patient would ever need ICU care. K-nearest neighbors performed the worst of all the models, with an overall average of 72% when using 14 neighbors in the classifier. It makes sense that kNN performs the worst, because the dataset had a large number of dimensions, even after being reduced through principal component analysis. The multilayer-perceptron was the clear winner in terms of accuracy with a score of about 85%. It was able to capture non-linear responses in the data that the other three models were not.

References

[1] Dataset obtained from Kaggle.com `https://www.kaggle.com/S%C3%ADrio-Libanes/covid19?select=Kaggle_Sirio_Libanes_ICU_Prediction.xlsx`

[2] Special thanks to Sírio-Libanês data for AI and Analytics by Data Intelligence Team for collecting and publishing the data.