Predicting Intubation within 5 days of Hospitalization for COVID-19 Patients

Yael Yossefy

Introduction

The goal of this retrospective analysis is to predict whether a COVID-19 patient will require intubation or mechanical ventilation within the first five days of hospitalization. Such a prediction model is critical for hospital resource management, as surges in COVID-19 cases have strained the availability of ventilators and other critical care equipment.

The study population consists of 1,345 COVID-19 positive patients hospitalized at New York-Presbyterian Hospital. The outcome of interest is whether a patient was intubated within five days of admission (indicated by "Yes" in the event variable of the baseline table). Patient data are stored in two datasets: baseline and vitals_and_labs.

The baseline dataset contains demographic, clinical history, and diagnostic test variables recorded at admission, such as age, BMI, and other clinical measures. The vitals_and_labs dataset includes repeated measurements of five vital signs—systolic blood pressure, diastolic blood pressure, heart rate, blood oxygen level (SpO_2), and respiratory rate—collected over the course of hospitalization.

Data Cleaning and Feature Engineering

To derive meaningful features from the vitals and labs data, the following preprocessing steps were performed:

1. Renaming and formatting:
    - Vital sign names were simplified for readability:
        - vs_bp_noninvasive (s) → sbp
        - s_bp_noninvasive (d) → dbp
        - vs_hr_hr → hr
        - xp_resp_rate_pt → resp_rate
        - xp_resp_spo2 → spo2
    - The time_stamp column was converted to POSIXct format to correctly represent the measurement times.
    - Missing data rows were removed using drop_na().

2. Summary statistics:
    - For each patient and each vital sign, the following metrics were calculated over the observation window: mean, standard deviation, interquartile range, and skewness. These were recorded as:
        - mean_sbp, sd_sbp, iqr_sbp, skewness_sbp
        - mean_dbp, sd_dbp, iqr_dbp, skewness_dbp

        - mean_hr, sd_hr, iqr_hr, skewness_hr
        - mean_resp_rate, sd_resp_rate, iqr_resp_rate, skewness_resp_rate
        - mean_spo2, sd_spo2, iqr_spo2, skewness_spo2

3. Clinically Anchored Metrics:
    - Even though I explored the thresholds and abnormalities of all the vital signs within our population, I decided not to include them all as features to avoid overcomplicating the models with too many features and possibly creating correlated features.
    - A study of adult patients hospitalized with COVID-19 in the UK[1] indicated that patients who deteriorate typically experience rapidly-worsening respiratory failure, with low SpO_2 and high FiO_2, but only minor abnormalities in other vital signs. For this reason, I focused on SpO_2 abnormalities. For each patient, the proportion of SpO_2 measurements below 95% was calculated as abn_prop_spo2. This approach accounts for the varying number of SpO_2 measurements per patient (ranging from 8 to 209).

4. Temporal dynamics:
    - To capture how vital signs changed over time, the linear slope and net shift were calculated to quantify whether the measurements increased or decreased over time and the overall shift across the window. In addition, volatility of vitals was captured by rolling standard deviation. The following temporal features were created for each patient:
        - slope_sbp, slope_dbp, slope_hr, slope_resp_rate, slope_spo2
        - net_shift_sbp, net_shift_dbp, net_shift_hr, net_shift_resp_rate, net_shift_spo2
        - vol_mean_sbp, vol_mean_dbp, vol_mean_hr, vol_mean_resp_rate, vol_mean_spo2

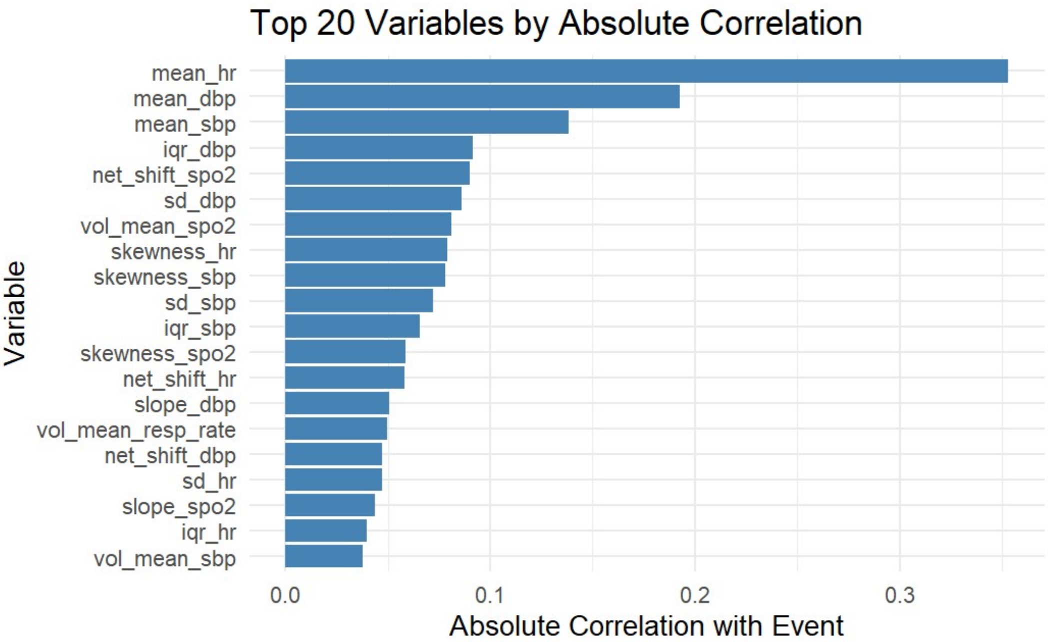
Overall, 36 features were engineered from the vitals and labs data. When combined with the 25 baseline features, the final dataset included 61 predictors. Patient identifiers (mrn and subject) were removed, and character variables were factorized.

Finally, the dataset was randomly split into 70% training data and 30% testing data.

Statistical Methods

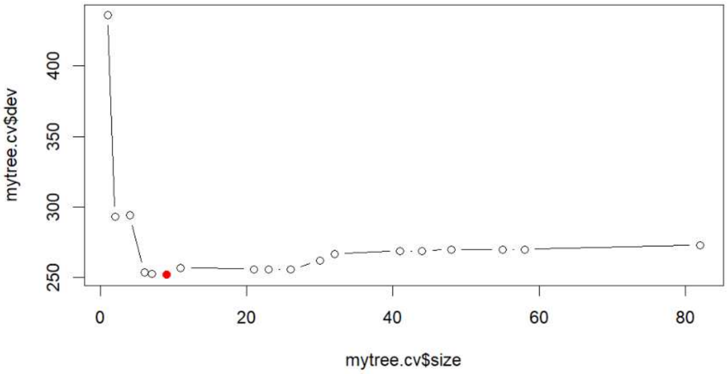
Logistic Regression

I first fit a logistic regression model on the training data using glm() with family = binomial, including all 61 predictors. Given the large number of covariates, I then refit a reduced logistic regression model that included only the 25 baseline features and the “best 20” engineered numerical features. These 20 features were selected based on their strength of correlation with the binary outcome (intubation within 5 days).

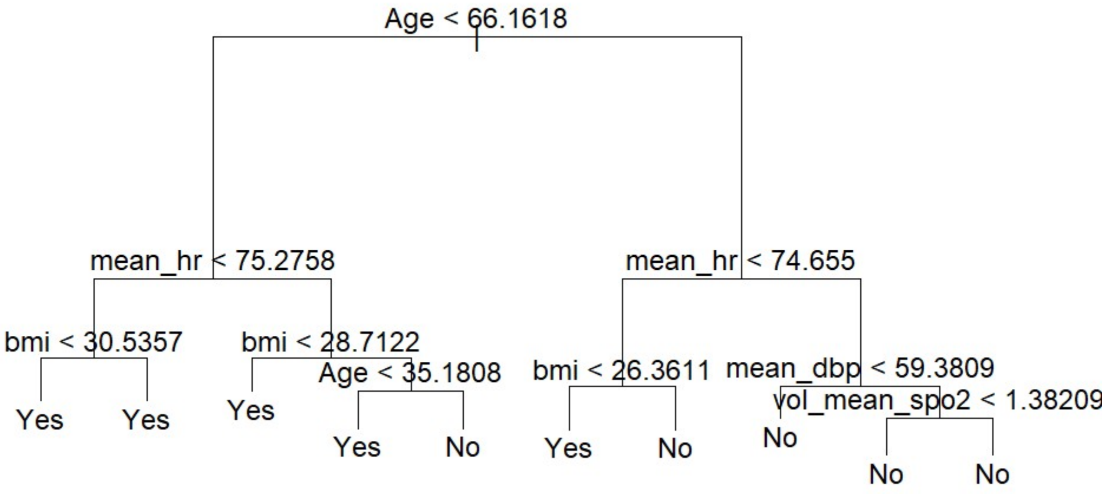


Classification Tree

Next, I fit a classification tree to the training set using the Gini impurity criterion. Gini uses a measure of total variance across the two classes using class purity and is thus more sensitive than misclassification rate as a measure of error for tree growing. I performed 10-fold cross-validation to identify the optimal level of tree complexity, using pruning and plotting the cross-validated deviance as a function of tree size. The tree sizes achieving the minimum cross-validated deviance are highlighted in red on the plot.



Using these results, I pruned the original full tree to the optimal size (10), producing the final pruned tree shown below:

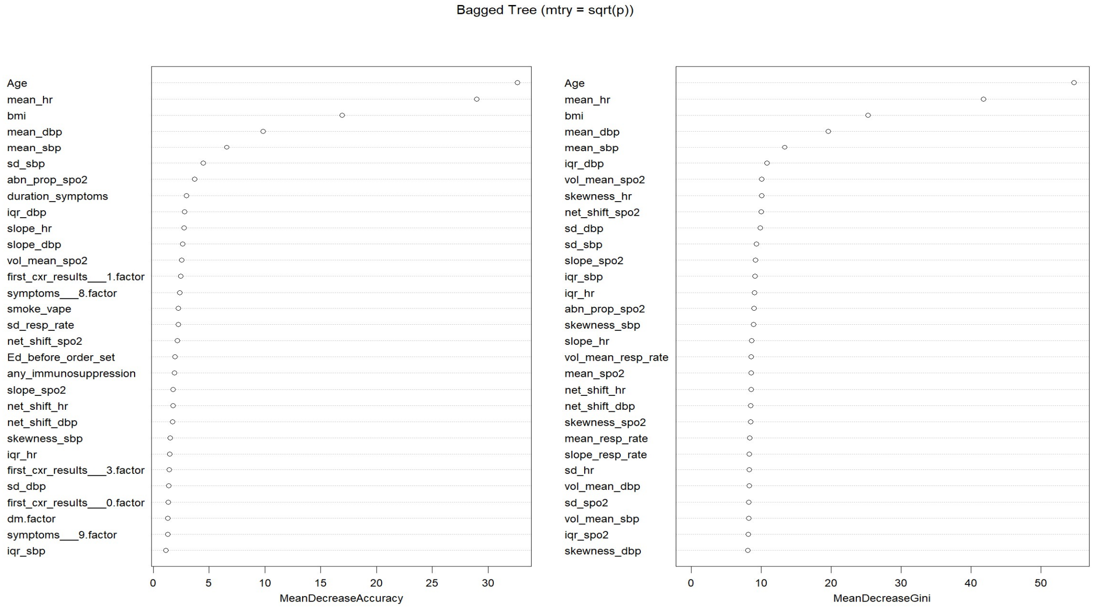


Random Forest Classification

Because ensemble methods often improve the predictive accuracy of decision trees, I fit a random forest model to the training data. For each split, a random subset of predictors of size √61 was considered as candidate split variables, which decorrelates the trees and reduces variance when averaging.

Using the randomForest() function, I set: mtry = sqrt(61), ntree = 500, importance = TRUE.

This allowed for evaluation of variable importance across the ensemble:



Support Vector Machine

To tune the cost parameter, I fit an SVM with 10-fold cross-validation over the following grid of cost values: c(0.001, 0.01, 0.1, 1, 5, 10, 100).

I selected the model with the lowest cross-validated error, which used **cost = 0.01**, and proceeded with this tuned model for further analysis.

Lasso Regression

Since we have a lot of predictors, I used Lasso to shrink some of those down to 0. Using cv.glmnet(), I ran a 10-fold cross validation (with alpha=1 to represent L-1 penalty) and determined the minimum lambda = 0.007036965. Using this regularization parameter, I refit the model.

Elastic Net

For the elastic net model, I ran a 10-fold cross validation to optimize alpha and lambda, using a range of 0 to 1 in intervals of 0.1 for alpha. The alpha for which the CV error was minimized was 0.2 and the lambda was 0.00703696. Finally, I used these parameters to the fit the final Elastic Net Model.

Results

To compare prediction errors across the statistical methods, I will compare the misclassification error when applying those models to the test data, or the 30% hold-out set of the original dataset that is new, unseen data for the model.

The first logistic regression model, with all 61 predictors included, had a misclassification rate of 21.0396%. The logistic regression model that included only 25 predictors (20 of the most correlated engineered features and 25 of the baseline features) had a misclassification rate of 20.29703%.

The pruned classification tree of size 10 had a misclassification rate of 32.67327%. The Random Forest had a misclassification rate of 24.50495%.

The Support Vector Machine resulted in a misclassification rate of 20.79208%.

The Lasso Model had a misclassification rate of 19.30693% and the Elastic Net Model had a misclassification rate of 20.79208%.

Conclusion

After comparing multiple statistical learning methods, the ones that resulted in the lowest prediction error on the test set were Support Vector Machines, Logistic Regression model (with only 45 predictors), the Lasso Regression, and the Elastic Net Model. In our results, we notice that filtering to only the most correlated features only slightly improved the accuracy of the Logistic Regression model, reducing the misclassification rate from about 21% to 20%. Meanwhile, Random Forest greatly improved the accuracy of the classification tree, as the prediction error decreased from about 32% to 24.5%.

I would recommend the Lasso Regression Model or the Logistic Regression (limited predictors) model to the hospital for use in predicting intubation. The benefits of logistic regression is the ease of interpretation and the ability to interpret coefficients of the mathematical equation that represents the model. We were also able to inspect the 20 numerical vital features that are most highly correlated with intubation within five days. This may give us additional information about the relationships worth inspecting between vital measurements and the event of intubation. Lasso Regression is also useful as it includes a penalty term. In shrinking some coefficients down to zero, it performs a version of feature selection, which is crucial to a dataset with so many predictors (p=61).

Bibliography

1. Pimentel MAF, Redfern OC, Hatch R, Young JD, Tarassenko L, Watkinson PJ. Trajectories of vital signs in patients with COVID-19. Resuscitation. 2020 Nov;156:99-106. doi: 10.1016/j.resuscitation.2020.09.002. Epub 2020 Sep 10. Erratum in: Resuscitation. 2021 May;162:91-92. doi: 10.1016/j.resuscitation.2021.02.021. PMID: 32918984; PMCID: PMC7481128.