Prediction of ICU Admission Among Hospitalized Patients with COVID-19
(Data-Scientist Track)

Ansel Lim, Gordon Wong, Felicia Khor
April 2021

1) Introduction

a) Motivation

The COVID-19 pandemic continues to impose a huge amount of burden on healthcare systems across the world. Although the disease is self-limited and mild in the vast majority of patients, a minority of patients can become very sick and require intensive care management. It is of interest to physicians to triage and prognosticate the course of disease so as to allocate the right amount of manpower to care for patients who require intensive care admission.

b) Dataset summary

Hospital Sírio-Libanês, São Paulo and Brasilia, 3 hospitals in Brazil, have publicly released an anonymized dataset on Kaggle[1], containing data on 385 hospitalized patients with COVID-19.

Each of the 385 patients was observed over 5 time windows (time windows: 0-2 hours, 2-4 hours, 4-6 hours, 6-12 hours, and after 12 hours). In each time window, clinical, physiological, and laboratory parameters are available, making up a total of 54 unique parameters (variables). The 54 feature variables are divided into 12 categorical feature variables and 42 numeric feature variables. For the numeric feature variables, since a patient might have had multiple measurements in any time window, each numeric feature variable is further represented by different summary statistics, including the mean, median, maximum, and minimum values. Therefore, for the numeric feature variables, there are a total of 216 columns of data which we will call ‘feature columns’.

Categorical feature variables include gender, age group/percentile, presence or absence of various groups of disease, presence or absence of hypertension, and whether the patient was immunocompromised. Numeric data included vital signs such as blood pressure and respiratory rate, as well as laboratory parameters such as individual components of the full blood count, such as white blood cell count, lymphocyte count, and neutrophil count.

Numeric data were already scaled by column according to Min Max Scaler, from -1.0 to 1.0 units. The dataset owners mentioned on the dataset’s Kaggle webpage that this was performed in the interest of data privacy, so that the patients cannot be identified. In each time window, the patient’s ICU admission status is tracked.

c) Problem Statement

We wanted to put on a doctor’s hat and ask this question: “If my patient has been admitted to hospital with COVID-19, what is the likelihood that he’ll eventually be admitted to ICU?”

Thus, our aim is to shortlist a selection of predictor variables to construct a machine learning model. This model will predict intensive care unit admission among patients hospitalized with COVID-19 who are NOT currently admitted to ICU. The implication is that identification of parameters that predict ICU admission may help physicians to construct

[1] COVID-19 - Clinical Data to assess diagnosis (Dataset): https://www.kaggle.com/S%C3%ADrio-Libanes/covid19

clinical scores that triage patients into risk categories. This will facilitate resource management and early clinical intervention for patients whose clinical trajectory is likely to involve ICU admission.

2) Data Cleaning

a) Data Preprocessing

We created a new variable 'EverAdmitted' that tracks whether a person will be admitted at any point in time. This value is 1 if the patient ever gets admitted in any window, and 0 otherwise. This engineered variable is our target variable in our clinical question.

The data consists of time series data for many patients, and we are interested to predict, among patients in the general ward, which patients will be admitted to ICU. However, some patients in the dataset were already in the ICU at time window 0-2. These patients, numbering a total of 32, were removed. The remainder of patients, namely 353 patients, meant that we still had more than 90% of the original number of patients that could be analyzed to answer our clinical question.

Furthermore, since we are interested to predict ICU admission in advance of the actual event of ICU admission, for any particular patient who gets admitted to ICU, it is necessary to distinguish between data prior to his admission to ICU, and data after his admission. Clearly, we should only consider data prior to admission if we are to answer the problem statement. Therefore, for patients who get admitted to ICU, we discarded the time windows that occurred after ICU admission.

b) Outliers Detection and Handling

Outlier detection was performed for every potential feature variable by consistent application of the '1.5 IQR rule'. However we did not remove outliers for feature variables which outlier removal would result in an unacceptably narrow range of values (the threshold used was less than 10% of the maximum possible range of values). We thought that this conditional approach of outlier removal would prevent inadvertent normalization of non-normal data, or the removal of physiologically aberrant data points that could have true clinical relevance.

c) Imputation of missing data

From inspection of each feature variable's distribution by plotting their density plots, we realized that most of the potential feature variables have non-normal distributions, therefore we decided to perform imputation using the median value of a feature column on a patient-by-patient basis, with a custom function that we created.

Because there is a very large number of columns, many of which were clustered missing data for many patients, it was not practical to implement a regression-based methodology for imputation. We decided to impute using the median, a measure of central tendency, since most data was non-normal, and we had already flattened the time dimension into two time windows, namely, prior to admission (for all patients), and the point of time of admission (for admitted patients). Note that median imputation did not eradicate all missing values because missing data for a patient was imputed from the same patient's other time windows, NOT other patients.

3) Exploratory Data Analysis

a) Principal Component Analysis - Shortlisting features for further exploratory analysis

We performed PCA, a dimensionality reduction technique, on our 42 numeric feature variables, focusing only on the median summary statistic to avoid multicollinearity between the various summary statistics of each feature variable. Figure 1 below presents the top-ranking features with outlier removal:

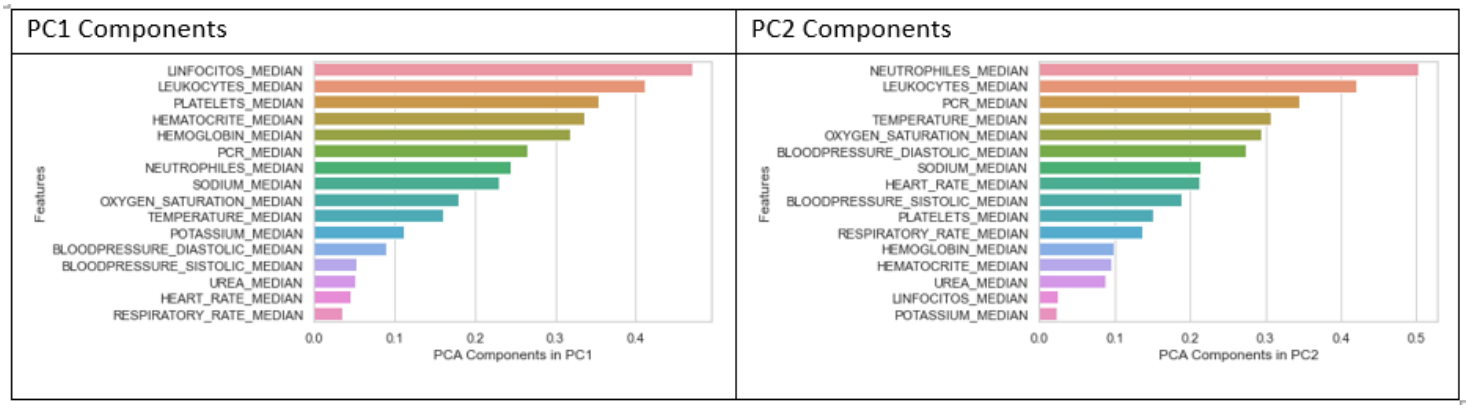

Figure 1: PCA bar plots with the top-ranking features

PCA identified 6 numeric feature variables which also demonstrated a potential relationship with ICU admission status on exploratory data analysis. For example, we observe that patients who are EverAdmitted will have a lower LINFOCITOS_MEDIAN, PLATELETS_MEDIAN, and higher PCR_MEDIAN as shown from PC1 components. The boxplots of these shortlisted feature variables can be seen in Figure 2.

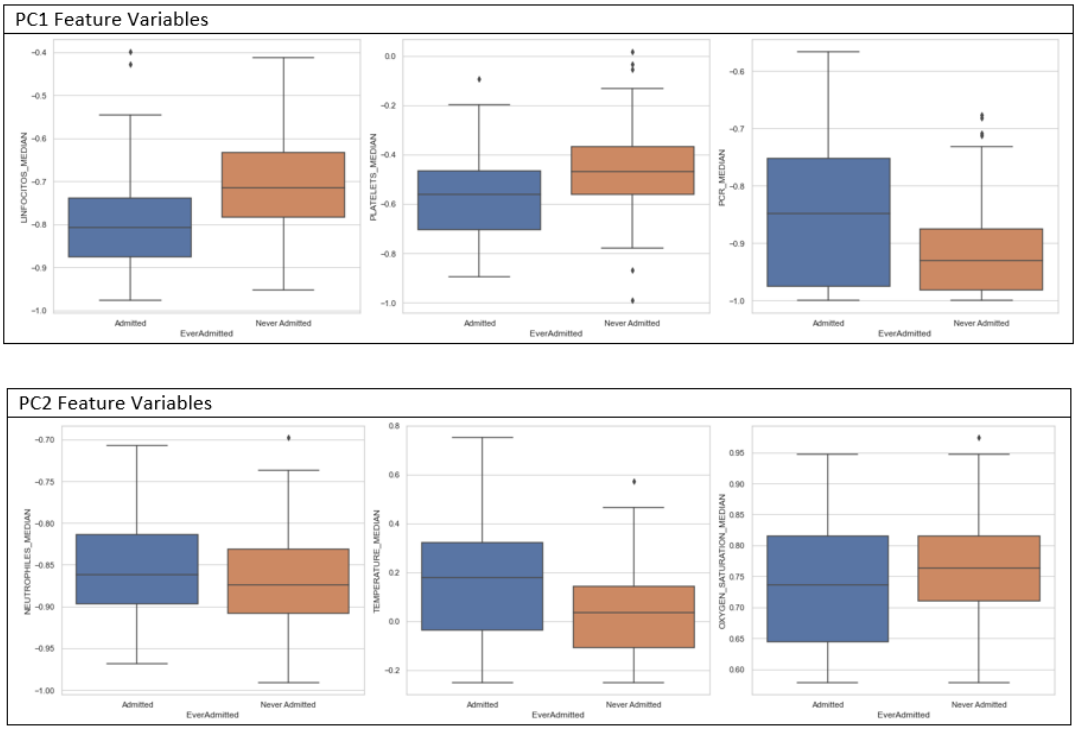

Figure 2: Boxplots of shortlisted feature variables from PCA

To further improve the predictive power of the machine learning model that we will build, we performed data visualization for the remaining variables, which is explained in the next section.

b) Data Visualization of Features - Shortlisting a total of 23 potential numeric feature variables

We have further identified additional feature variables based on visual differences observed in the distributions between the EverAdmitted and NeverAdmitted groups after plotting boxplots, kernel density estimation (KDE) plots, and bar graphs. We have showcased an example of the boxplot and KDE plot of a potential feature variable, SODIUM_MEDIAN, in Figure 3. The 18 shortlisted numeric feature variables are shown in Figure 4.

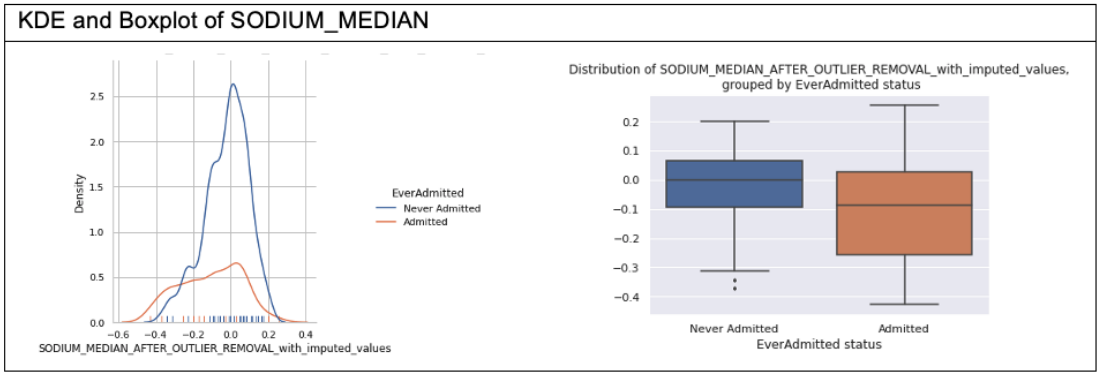

Figure 3: KDE and boxplot of SODIUM_MEDIAN

Shortlisted numeric feature variables	
1. CALCIUM_MEDIAN	10. POTASSIUM_MEDIUM
2. GLUCOSE_MEDIAN	11. SODIUM_MEDIUM
3. HEMATOCRITE_MEDIAN	12. UREA_MEDIAN
4. LACTATE_MEDIAN	13. BLOODPRESSURE_DIASTOLIC_MAX
5. NEUTROPHILES_MEDIAN	14. BLOODPRESSURE_SISTOLIC_MIN
6. LINFOCITOS_MEDIAN	15. RESPIRATORY_RATE_MEDIAN
7. PC02_VENOUS_MEDIAN	16. TEMPERATURE_MIN
8. PCR_MEDIAN	17. OXYGEN_SATURATION_MEDIAN
9. PLATELETS_MEDIAN	18. HEART_RATE_MAX

Figure 4: Shortlisted numeric feature variables

Amongst the categorical feature variables, we have identified 3 potential variables which demonstrated differences in the proportion of patients who were EverAdmitted across the different categorical bins. The shortlisted categorical feature variables which are GENDER, HTN (Hypertension) and IMMUNOCOMPROMISED are shown in Figure 5.

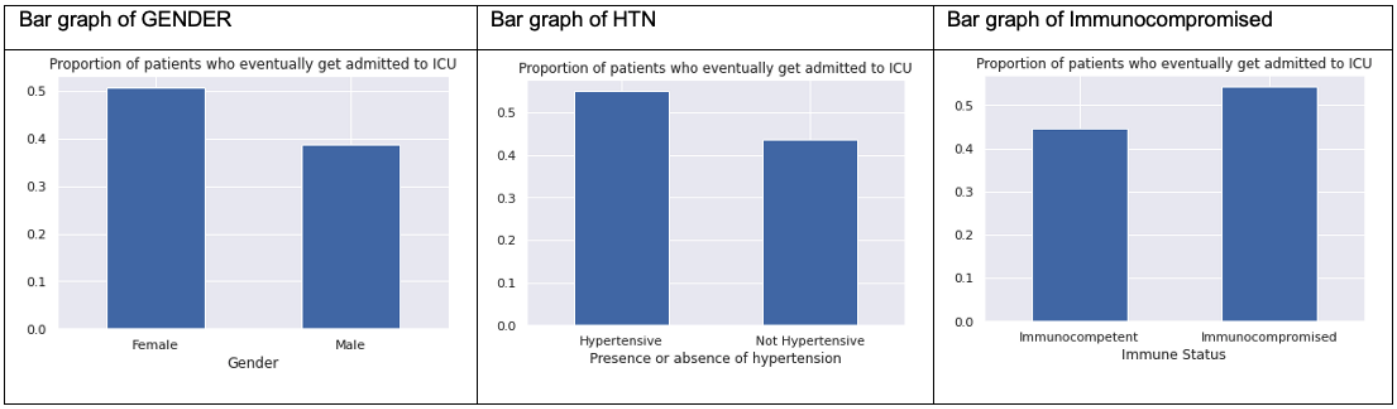

Figure 5: Bar graphs of the shortlisted categorical feature variables

4) Methodology

a) Overall approach and explanation

We have included a diagram outlining our overall approach to the problem statement in Figure 6. The first priority was feature selection, followed by application of classification models. Secondly, we used base classification models known to perform well with binary classification. Lastly, we improved the base models' performance with optimization techniques such as oversampling of the minority target class, hyperparameter tuning with stratified cross-validation, and use of ensemble machine learning techniques.

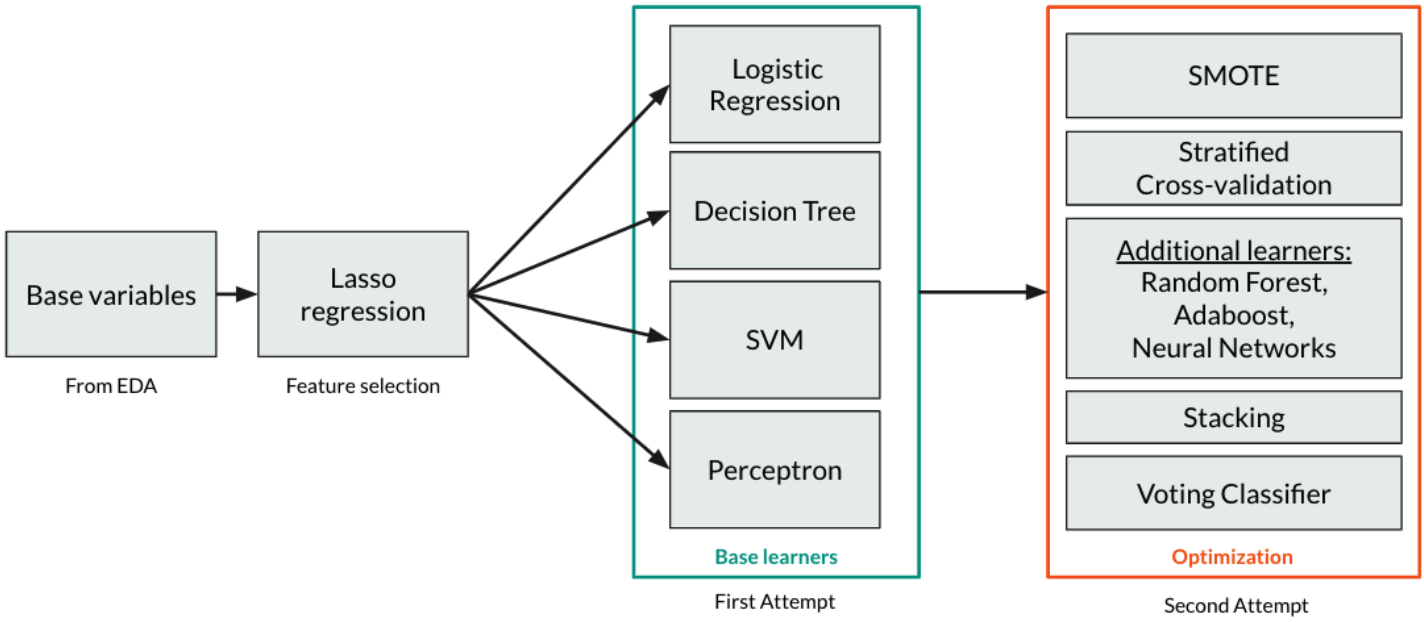

Figure 6: Overview of approach

Feature selection:

Feature selection reduces the computational cost of modeling and improves predictive performance in an attempt to provide physicians with predictions with the least amount of input data. From our exploratory data analysis, we have identified a total of 21 potential feature variables. We proceeded to conduct lasso regression on these potential feature variables. Lasso regression implements an L1 regularization term that severely penalizes nonessential or correlated features by coercing their corresponding coefficients toward zero. Based on our Lasso regression results, we did not drop any of the identified 21 potential feature variables as dropping any feature would lead to inferior predictive performance as measured by Area Under the Curve (AUC) of the receiver operating characteristic curve.

First Attempt - 4 base learners:

We started with 4 base machine learning models in our first attempt, which were chosen for their suitability with binary classification tasks such as our problem statement.

i) Logistic Regression: Logistic regression provides a probabilistic interpretation of clinical risk. It identifies the contribution of each feature variable, while providing for an explainable model. Logistic regression finds the parameters and the weights to assign to the parameters based on the principle of the maximum likelihood estimation.[2]

[2] Essentials of Business Analytics, B. Pochiraju, S. Seshadri (eds.). 2019

ii) Decision Tree: Decision trees are explainable, greedy classification models. A decision tree works on the basis of reducing misclassification, or impurity. Decision trees are popular among physicians as they are familiar tools, and a decision tree can be directly translated into a clinical algorithm and visual tool that may be relied upon for clinical decision-making on the ground.

iii) Support Vector Machines(SVM): SVM are especially suitable for binary classification. We applied SVM on kernel-transformed feature space data, and we trained the model to find an optimal separating hyperplane between the two target classes.

iv) Perceptron: A perceptron may be considered to be a simplest form of a neural network, with just a single layer. It is a biologically inspired, discriminative binary classifier. Through an iterative process of updates of weights, it minimizes misclassification error.

We evaluated the 4 model performance on a test subset of the data generated using train-test split. The initial performance metrics of our 4 base models can be found in Figure 7 - the lowest and highest scores for each metric is color-coded for convenient viewing. Model hyperparameters were tuned to achieve the best AUC scores. We have identified a few patterns from these initial metrics.

The perceptron model has a high recall, which means that it is great at identifying, or picking up, admitted patients. But, among the patients that it predicts to be admitted, a minority of them are actually admitted. The perceptron model also has poor specificity, which means it does poorly at identifying NeverAdmitted patients.

The SVM and logistic regression models perform differently, demonstrating poorer recall scores as compared to the perceptron model, but among the positive predictions, a larger proportion are actually admitted. With higher specificity scores, SVM and logistic regression perform better with identifying NeverAdmitted patients. Thus, they might be great at identifying candidate patients for whom the acuity of care may be de-escalated. Because SVM and logistic regression do a great job with the majority class, they have a superior overall accuracy compared to perceptron.

The decision tree model has a different pattern of performance. It has a recall that is similar to logistic regression, but it has specificity which is intermediate in performance, compared to the other 3 models. Additionally, it's precision score is similar to perceptron. In terms of metrics, the decision tree model does not perform as well, but where it does well is in the explainability of the decision tree modeling process.

Models	AUC	F1	Accuracy	Precision	Recall	Specificity
Perceptron	0.786	0.545	0.659	0.375	1.000	0.571
SVM	0.776	0.632	0.841	0.600	0.667	0.886
Logistic Regression	0.721	0.556	0.818	0.556	0.556	0.886
Decision Tree	0.649	0.435	0.705	0.357	0.555	0.743

Figure 7: Model test performance of first attempt - 4 base learners

Second Attempt - Optimization:
Building on our first attempt, we identified a few areas to improve the performance of our predictions.

- Addressing the problem of unbalanced classes with SMOTE: 81% of patients were admitted to ICU, whereas 19% were not admitted to ICU. Our problem statement focused on the target class EverAdmitted, however this was the minority class. We hypothesized that applying a synthetic oversampling technique would improve the model's performance in identifying Admitted patients (recall) while maintaining specificity. In our implementation of SMOTE, we combined undersampling and oversampling such that the proportion of EverAdmitted, including synthetic data, in the dataset, was 40% rather than the 20% in the raw data. We chose not to artificially balance out the two classes to achieve a 50-50 split, as this may introduce overcorrection during synthetic data creation.

- Stratified Cross-validation: Model evaluation on a single train-test split instance is prone to random variations in the splitting process. We applied stratified cross-validation to tune the hyperparameters of our models. In addition, we also extended the search space of our models and included other hyperparameters that we did not consider previously. A stratified approach to cross-validation improves the chances that the 2 classes are fairly represented across all test and training folds.

- Additional learners: Building on our base learners, we have also employed 3 additional models.

  i) Random Forest: The CART algorithm used by the decision tree classifier is a heuristic greedy algorithm which aims to make locally optimal decisions at each node, and is not guaranteed to return a globally optimal decision tree. To mitigate this, we decided to employ feature- and sample-based ensemble learning with Random Forest, together with bootstrap sampling as supported by the API of sklearn's Random Forest implementation.

  ii) Adaboost: In contrast to Random Forest, Adaboost attempts to improve the performance of the model sequentially, by iteratively generating weak models on the dataset, with the misclassified data points given a higher weightage.

  iii) Neural Network: Extending on our earlier efforts for the perceptron model, we decided to build a fully connected, feedforward neural network with two hidden layers. The output layer has a single node which takes on a value of either 0 or 1 and is computed by applying the sigmoid activation function on the second hidden layer. We tuned the number of nodes in both hidden layers, and also optimized the learning rate, finding the best multi-layer perceptron that maximizes AUC.

- Combining predictive power of learners with meta-classifiers:

  - Stacking: We applied this ensemble learning technique to combine 6 classification models via a logistic regression-based meta-classifier that is trained on the outputs of the individual models (meta-features).

  - Voting Classification: We applied a soft voting meta-classifier on our ensemble of 6 well-calibrated models, with a weighted sum approach that gave more emphasis to the best performing base models, specifically SVM and Adaboost-ed decision trees.

The performances of our models for the second attempt can be found in Figure 8, and our analysis is explained in the following page.

Models	AUC	F1	Accuracy	Precision	Recall	Specificity
Perceptron	0.906	0.773	0.809	0.771	0.798	0.820
SVM	0.974	0.892	0.920	0.976	0.830	0.984
Logistic Regression	0.923	0.799	0.836	0.820	0.797	0.865
Decision Tree	0.848	0.726	0.784	0.770	0.701	0.843
RandomForest	0.909	0.745	0.814	0.844	0.683	0.907
AdaBoost	0.957	0.845	0.898	0.870	0.840	0.926
NeuralNetwork	0.862	0.842	0.873	0.96	0.75	0.974
Stacking	0.987	0.938	0.950	0.942	0.939	0.957
VotingClassifier	0.878	0.862	0.887	0.962	0.781	0.974

Figure 8: Model test performance of second attempt - optimization

After optimization of perceptron, the recall decreased from 1 to 0.798, however the specificity improved and most remarkably, the precision doubled to 0.771 (higher positive predictive value). Although the recall decreased, it was still at a respectable level and the other parameters improved which was a worthwhile trade-off.

SVM, logistic regression and decision tree showed improvement of all metrics across the board. The biggest improvement was seen in the precision of the decision tree. This may be because of the models' greater exposure to EverAdmitted class.

The performance of random forest was superior to the decision tree model in all aspects except recall. Perhaps this is because a random forest averages better performing decision trees with poorer performing ones in an effort to maximize AUC, which might be driven more by specificity for NeverAdmitted class, rather than sensitivity for EverAdmitted class. In contrast to random forest, Adaboost was superior to the decision tree in every aspect.

Stacking is superior compared to the base models. Voting classifier does not perform as well as stacking, perhaps because the user-specified weights in the process of final aggregation have not been optimized whereas the logistic regression based meta-classifier in the stacking classifier optimizes the weights for each model based on maximum likelihood estimation.

Predictor variables implicated in ICU Admission

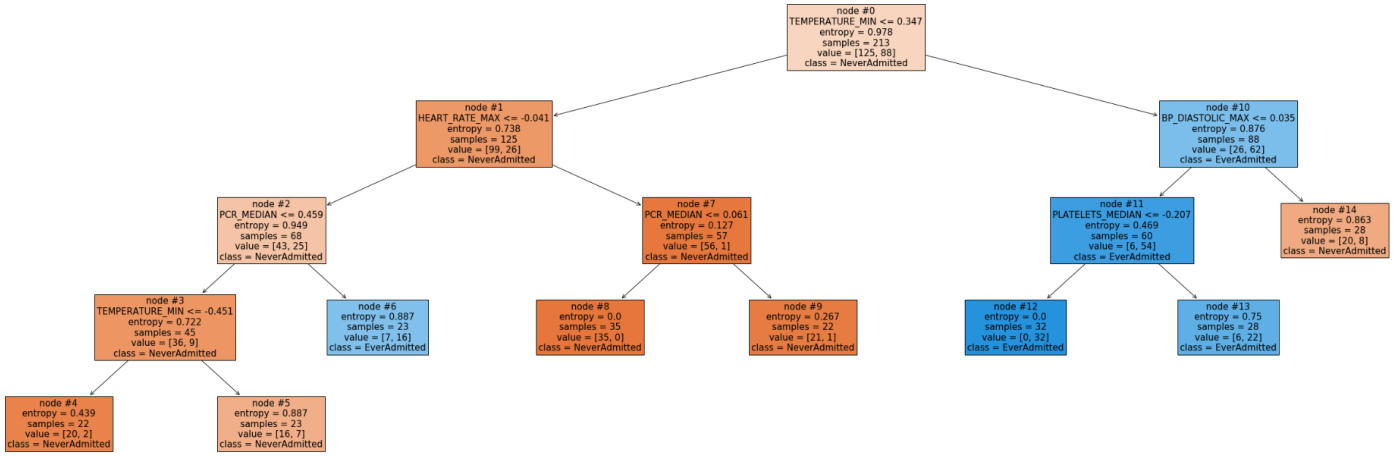

Figure 9: Decision Tree Plot

We show the plot of the decision tree from the second attempt in Figure 9. As mentioned, the strengths of the decision tree model lie in its interpretability. To a physician, this tree structure suggests that Hyperthermia, Low Diastolic Blood Pressure, Thrombocytopenia, and Albuminuria may be potential predictors of ICU admission.

The limitations of the decision tree model, as explained in earlier sections, are its lower test performance scores as compared to other models, and its greedy approach, which might result in a globally suboptimal decision tree. The results of the logistic regression model may be mined for additional predictor variables to complement those already found by the decision tree model. Figure 10 displays key predictor variables that have a positive (odds ratio >1) and negative (odds ratio <1) association with the target variable EverAdmitted, accompanied by supporting scientific literature.

Positive predictor variables	**Odds Ratio**	**P-value**	**Plausible biological explanation / Literature review**
PCR_MEDIAN	2.17	<0.001	Acute kidney injury due to sepsis predicts poor outcomes[3].
HEMATOCRITE_MEDIAN	1.64	<0.001	Polycythemia contributes to the thromboembolic and tissue perfusion-related complications of COVID-19[4]
TEMPERATURE_MIN	1.59	<0.001	Hyperthermia due to loss of autoregulation and homeostasis occurs in overwhelming sepsis in COVID-19[5]

Negative predictor variables	**Odds Ratio**	**P-value**	**Plausible biological explanation / Literature review**
BP_DIASTOLIC_MAX	0.22	<0.001	Hemodynamic complications of COVID-19, including hypotension, portend poorer outcomes[6]

[3] Crit Care 24, 346 (2020), BMC Nephrol 22, 92 (2021)
[4] BMJ 2020;369:m2058, Lancet. 2020 6-12 June; 395(10239): 1758–1759.
[5] J Med Virol. 2020 Jun 10 : 10.1002/jmv.26154.
[6] Clin Cardiol 2020 Oct;43(10):1054., *Intensive Care Medicine* 47 254–255(2021)

HEART_RATE_MAX	0.49	<0.001	Inappropriate/relative bradycardia due to loss of cardiac autoregulation contribute to arrhythmias and cardiac failure[7]
CALCIUM_MEDIAN	0.54	<0.001	Hypocalcemia is associated with higher in-hospital mortality[8]
LINFOCITOS_MEDIAN	0.65	<0.001	Lymphopenia predicts severe clinical outcomes in COVID-19[9]

Figure 10: Key predictor variables and supporting literature

5) Conclusion

Our problem statement was to shortlist a selection of predictor variables, to predict intensive care unit (ICU) admission among patients hospitalized with COVID-19. A key observation that we made after performing exploratory data analysis is the identification of the 21 feature variables which we have used for our model building.

We have employed both white-box (Logistic regression, Decision Tree) and black-box models (SVM, Perceptron, Random Forests etc.) and evaluated their performance. We observed that although the white-box models were interpretable and showed the relationships between predictor variables and our target class, black-box models in general had performed better compared to the white-box models. We find that the choice of model would depend on the trade-off between interpretability and accuracy. A very accurate model that uses a large amount of features and complex decision rules may not be feasible to apply in a dynamic clinical environment, may be very costly to operationalize, difficult to understand for clinicians, and may be overfit to the study population, as well as the natural history of the disease, which may change with mutations of the virus.

In combining the learners into 2 meta-classifiers, namely, the stacking and voting classifiers, we observed that stacking gives us the best performance amongst all models. The voting classifier did not manage to give us the performance that the stacking classifier demonstrated. We could look at tuning the user specified weights in the voting classifier by comparing it quantitatively with the stacking classifier.

In terms of implementation of our code, we found that in the model building stage, there was a significant reuse of code. We could organize the machine learning flow into a pipeline to improve modularity and workflow generalizability to new datasets.We could also apply our learnings from this workflow and validate the variables identified on other similar datasets, preferably those which data has not been artificially normalized.

We were interested in probabilistic interpretation of ICU admission and that drove a focus on logistic regression. We could expand on this approach and consider other probabilistic classifiers such as naive bayesian classifier. Additionally, we could expand on the problem statement and instead of flattening the time dimension for our problem statement to predict if a patient will be admitted to ICU during the course of the patient's stay, we could predict which time window the patient will be admitted, perhaps by applying a time series approach.

---

[7] Clin Microbiol Infect. 2021 Feb; 27(2): 295–296.
J Clin Med 2021 Mar 23;10(6):1317.
[8] Endocr Res 2018 May;43(2):116-123., Endocrine. 2020 Jun 12 : 1–4.
[9] Int J Infect Dis. 2020 Jul; 96: 131–135.