STATISTICAL LEARNING FOR HEALTHCARE DATA

A.Y. 2023/24

Project Report:

Suspected Pneumonia Database

Authors: Francesco Benedetto, Carlotta Pagani, Riccardo Sena

21 June 2024

1 Introduction

The primary objective of this study is to predict 24-hour mortality in patients with respiratory conditions, such as COVID-19 or other forms of pneumonia, such as viral or bacterial.

2 Material and Methods

2.1 Dataset description

The Successful Clinical Response in Pneumonia Therapy (SCRIPT) CarpeDiem Dataset features 12,495 patient-ICU-days from 585 patients enrolled in the SCRIPT study between June 2018 and March 2022, all of whom were critically ill patients on mechanical ventilation suspected of having pneumonia who underwent a bronchoalveolar lavage as part of routine clinical care. Each patient has demographics, pneumonia episode information as adjudicated by a panel of critical care physicians, outcomes, and 44 clinical parameters for which data are present including vital signs, laboratory parameters, and mechanical support devices. Data have been deidentified per Health Insurance Portability and Accountability Act (HIPAA) Safe Harbor rules. This dataset combines expert clinician adjudication with per-day granular information to provide a unique tool to examine the clinical courses of critically ill patients with pneumonia.

2.2 Memory Saving

Given the substantial size of the dataset, we optimized memory usage by converting variables to their appropriate types (e.g., some discrete variables were stored as $float64$). This conversion saved approximately 8 MB of memory, reducing the original dataset size from 14.1 MB.

2.3 Variable cleaning

We initially checked for duplicates; variables containing redundant information, a high number of 'unknowns,' or deemed irrelevant to our study were discarded, with justification provided for each removal.

2.4 Missing Data

Managing missing values is a crucial aspect of machine learning. Our dataset contained a significant number of NaNs across various variables. The figure below shows a barplot of the missing values for each column.

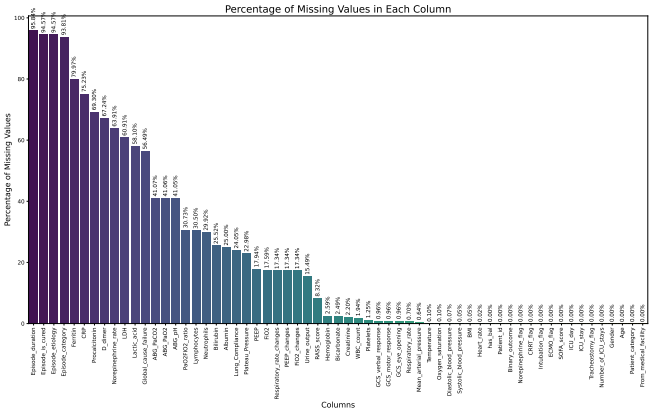

Figure 1: Percentage of Missing Values in Each Column

As shown, the variables with the highest scores are episode-related and contain 'fake' missing values, indeed they do no hold any information if the patient did not experience a pneumonia episode. For these variables we just trivially filled the NaNs. The same happened with the vairbale 'Norepinephrinerate', whose missing values were filled with zeros, meaning that the patient was not recieving norepinephrine.

For 'real' missing values in numerical variables, we used different approaches based on the percentage of NaNs in each column. First, we examined correlations among high-NaN variables, discarding those with correlations exceeding 60%. Variables with more than 30% missing values (11 variables) were dropped. For the remaining variables, we used interpolation: missing values for each Patient_id - ICU_stay group were replaced with the mean of the two closest data points. Rows that still contained NaNs due to a lack of measurements during the entire ICU_stay were dropped, resulting in the removal of 203 groups (1246 rows).

2.5 Outliers

Since our dataset is still very large, we decide to implement some outliers detection methods to both identify anomalies and to reduce the number of data points. In particular we applied the to identify the 10% of data furthest away from the rest. Before dropping these data points, we decided to only drop the ones with Binary outcome = 0, since we did not want to further increase the class imbalance.
In this occasion we performed a PCA reduction to the first two PCA scores for each instance present in the dataset, to visually identify differences in a R^2 space between normal data (in blue) and the points identified as outliers (in red). Moreover, the PCA scores will prove to be useful to the models.

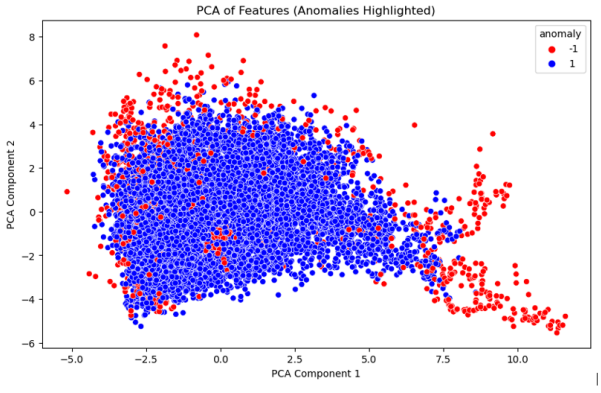

Figure 2: PCA

2.6 Feature engineering

2.6.1 Episodes

Since the episodes variables had a lot of NaNs we took a close look on how to deal with them. First, we noticed that there was a big difference between the category of the episode and to whether the episode was cured or not. To deal with this last one, we decided to count both the number of the episodes that were cured and the number of episodes that instead were not (standardized by the number of ICU days) for each ICU stay (in the case it was 'Indeterminate' we treated it as a not cured episode). With this method we can have an overall idea of how many episodes a patient has experienced and also specifically if the cure was successful or not. Now we have to deal with the episode duration variable. Originally in the dataset this was saved as the number of days for which the episode lasted, but for our scope (predicting in the next 24 hours) we wanted the actual cumulative value (we adopted the same procedure also for 'Cumulative_intubation_days'), hence we computed it. Finally, we examined the episode etiology variable. This variable did not hold a particular important information but we had to handle it very carefully. In fact, when the etiology is 'Viral', no information on the duration and on the success of the cure was displayed. We decided to consider this case as a episode that was not cured and 'neverending' (except for the case in which a patient had another episode), considering it as the worst case scenario.

2.6.2 BAL

We payed close attention to the 'has_bal' variable, which has value 'True' if the patient had a Broncho-Alveolar Lavage that day. This score was created by the need to distinguish the case where the patient never had a BAL from the case in which the patient did not have a BAL today. The more time since the last bal, the better the patient should feel. We wanted to create a score that increases when a BAL is performed and decreases over time, of course grouping for Patient id and ICU stay. The higher the score, the worse the patient's condition. We came up with this idea:

$$score(x,t) = \frac{x}{(t+1)^a}$$

where:

- x is the number of BAL, that increases when the variable 'has_bal' is true
- t is the time from last BAL in days, normalized over the number of icu days
- $a = 0.25$, we wanted the score to decrease nor too slowly nor too rapidly

We needed to use $t+1$ in the denominator to deal with the case where $t = 0$, otherwise our score function would assume an indeterminate value.

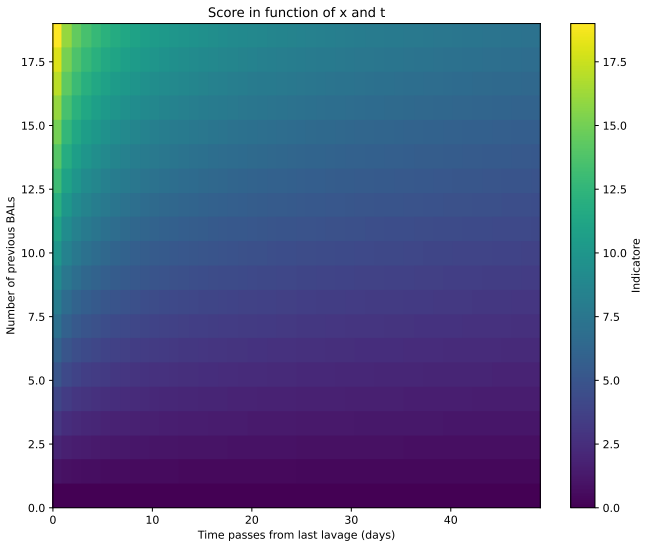

Figure 3: BAL score

For example, the patient identified by the number 7027 had a BAL only on the fifth, indeed the BAL_score is zero before that day, one on the fifth day and then decreases over time.

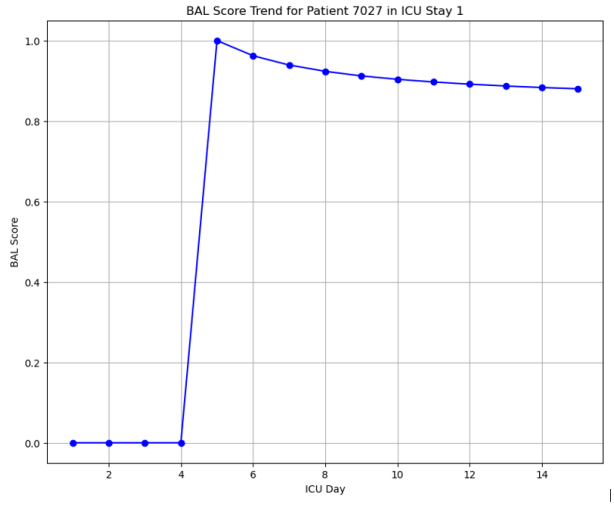

Figure 4: BAL score

2.7 Models

The dataset was split according to a 70:30 train-test ratio.

2.7.1 Preprocessing

Before applying any classification model we had to perform some adjustments. The most important one regards the *Binary_outcome* variable, the focus of our study. In the original dataset, this column assumed values 0 or 1 depending on if the patient was alive or died on the last day of their stay. But the aim of our project is to predict patient mortality in the following day, meaning we had to leave the value 1 only for the last day of a passing patient. After this correction, the dataset showed a very significant class imbalance, with only the 2.5% belonging to class 1.
Next, we had to transform into dummy the categorical variables there were left in the final dataset. Finally, we scaled the numerical variables using a standardized scaler.

2.7.2 Feature selection

We performed a feature selection through a random forest method, which selected 23 features:

- 'Age',
- 'ICU_day',
- 'SOFA_score',
- 'Temperature',
- 'Heart_rate',
- 'Mean_arterial_pressure',
- 'Norepinephrine_rate',
- 'Respiratory_rate',
- 'Oxygen_saturation',
- 'Urine_output',
- 'FiO2',
- 'Hemoglobin',
- 'Platelets',
- 'Bicarbonate',
- 'Creatinine',
- 'Bilirubin',
- 'bal_score',
- 'Scaled_not_cured_episodes',
- 'Episode_duration_cumulative',
- 'Cumulative_intubation',
- 'GCS_score',
- 'pca_0',
- 'pca_1'

2.7.3 Model selection

We trained 7 models, performing for each one of those a grid search for hyperparameters tuning for maximizing the recall. We also applied some ensemble methods for simple decision trees, but they didn't achieve a very high accuracy. The worst performing models were the ones that did not include the possibility of adding class weights, which it's quite crucial in the case of very imbalanced classes (as in ours).

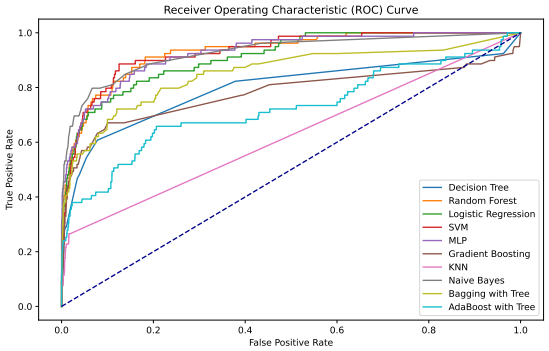

Figure 5: Roc curve

To assess which model to choose, we focused our attention on both the AUC and the recall. In particular, we tried to improve recall at the expenses of the precision, since we would prefer to fit a 1 even if the the actual observation was 0 instead of the contrary (we don't want to miss any deaths).

3 Results

3.1 Model choice

The 3 best models were SVM, logistic regression and random forest. The random forest had a low recall values so we decided to discard it.

Models	AUC	Recall
SVM	0.930	0.835
Log Regression	0.912	0.823
Random Forest	0.930	0.671
Decision Tree	0.796	0.608
Bagging	0.853	0.557
MLP	0.917	0.519
Naive Bayes	0.925	0.329
AdaBoost	0.723	0.241
KNN	0.624	0.215
Grad Boosting	0.579	0.177

Between the other two, since they had quite similar values we ended up adopting the logistic regression model since it would be easier to interpret and it overfits less than SVM. In the table below, we report the coefficients for each variable in the model. An increase of one unit in the variable x_i will result in an increase in the probability of dying by e^{β_i}.
It is possible to notice the the Bal score has an effective and logic effect on the probability of dying, however some coefficients, such as the ones related to the SOFA score and VAP episodes, were not expected to be negative.

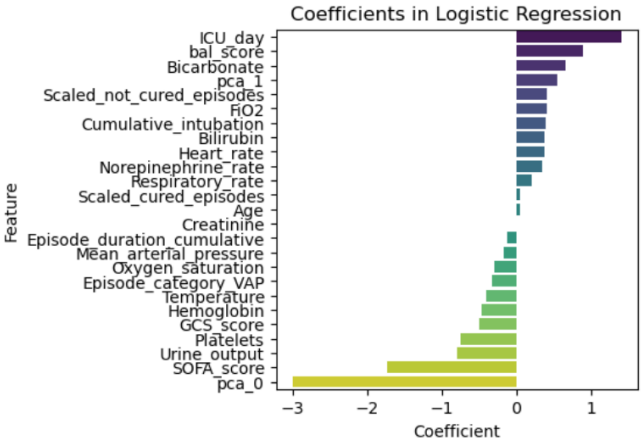

Figure 6: Coefficients of the logistic regression

Moreover,the confusion matrix for the logistic model shows good achievement in the prediction of both classes.

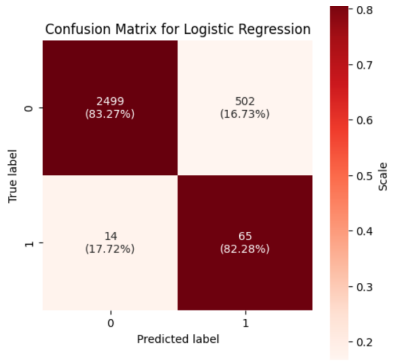

Figure 7: confusion matrix

4 Discussion and Conclusions

4.1 Conclusion

The primary goal of this study was to develop a predictive model for 24-hour mortality in ICU patients with suspected pneumonia. The logistic regression model emerged as the best-performing model. The decision to prioritize recall over precision was driven by the clinical imperative to avoid missing potential deaths, even at the cost of some false positives. Feature engineering played a pivotal role in enhancing model performance. The creation of the BAL score, which quantifies the patient's condition based on the frequency and timing of Broncho-Alveolar Lavage procedures, provided a novel and effective predictor. Similarly, the cumulative measures for episode duration and intubation days offered insights into the patient's clinical trajectory, which were essential for accurate mortality prediction.

4.2 Limitations and further development

The confusion matrix for the logistic regression model demonstrated good predictive power for both classes.

The model correctly identified a significant number of true positives (patients who died within 24 hours), which is critical for timely clinical interventions. However, some features, such as the SOFA score and VAP episodes, exhibited unexpected coefficient signs, indicating potential areas for further investigation and model refinement.

Despite the strengths of our approach, several limitations should be acknowledged. The significant class imbalance, with only 2.5% of observations in the positive class, posed challenges in training models that accurately predict minority class events. Although class weights and recall prioritization mitigated some of these issues, further techniques such as SMOTE (Synthetic Minority Over-sampling Technique) could be explored to enhance model performance.