Predictive Model for Survival In Ill Patients

Guillermo Jose' Gallucci
Politecnico di Torino
Student id: s331589
email@s331589.polito.it

***Abstract*—In this report, I present a potential solution for predicting the mortality of critically ill patients. The goal is to build a model that accurately predicts whether a patient lives or dies, based on features representing health conditions, demographics, and economic factors. Prior to building the learning model, I performed a comprehensive analysis of the dataset's features to determine which could be excluded. Following the identification through statistical analysis and subsequent hyperparameter tuning, the model achieved results that exceeded those of a naive baseline.**

I. Problem overview

By anticipating the potential outcomes of severe medical conditions, healthcare providers can prioritize care, and assist patients and their families in making decisions about treatment options. To accomplish this, I analyzed a dataset comprising information collected from five different cities in the United States and from two distinct time periods, 1989-1991 and 1992-1994, encompassing a total of 9,105 critically ill patients that fit into 9 categories of delicate diseases: acute respiratory failure, chronic obstructive pulmonary disease, congestive heart failure, liver disease, coma, colon cancer, lung cancer, multiple organ system failure with malignancy, and multiple organ system failure with sepsis.

The dataset is divided into two parts:

- The development set, containing 7,285 patients and the target
- The evaluation set, containing 1,280 patients and does not include the target, that must be predicted

In total, there are 43 features to describe each patient, representing various categories:

- Demographic data of the patient, including features like 'sex' and 'age'.
- Health condition descriptors such as 'dzgroup', 'ph', and 'dementia'.
- Economic data relating to both the patient and the hospital, some are 'income', 'charges', and 'totmcst'.
- Prediction from a model about the patient, including features like 'surv2m', 'sps', and 'scoma'.

Not all features are quantitative, some are qualitative: 'sex', 'dzgroup', 'dzclass', 'income', 'race', 'ca', 'dnr'. These must be encoded.

It is important to note that many patients were withdrawn from the study before its completion, resulting in incomplete data for some features, as illustrated in Fig. 1. Addressing this issue is crucial to achieve optimal results in predicting whether a patient survived or died.

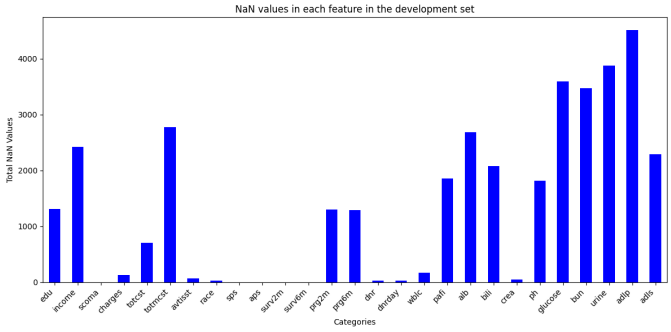

Fig. 1: Distibution of NaN values for each feature of the development set

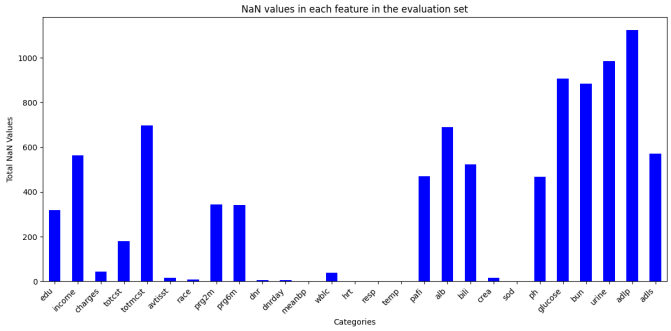

Fig. 2: Distibution of NaN values for each feature of the evaluation set

Fig. 1. illustrates that more than half of the features in the dataset contain missing values, with some features showing particularly high quantities of missing data. For example, the feature 'adlp' has more than 4,000 missing values. This situation is also present in the evaluation set, as shown in Fig. 2.

II. Proposed approach

A. Preprocessing

Due to the large quantity of missing data in both datasets, it was not feasible to remove patients with missing values. Therefore, I imputed data using a Simple Imputer with mean strategy.

- Simple Imputer with mean strategy: replaces missing values with the mean of the non-missing values in that column

Then, I performed one-hot encoding on the following remaining qualitative features: 'dzgroup', 'ca', 'dnr', 'sex', and 'race'. Now I can proceed with the removal of less important features

First, I will analyze the following features: 'edu', 'income', 'scoma', 'hday', 'dnrday', 'pafi', and 'dzclass' to determine if they should be considered in the final model.

- 'edu': A personal hypothesis suggests that there is no correlation between a patient's level of education and their survival outcome (whether they survive or die). To test this hypothesis, I initially calculated the biserial-point correlation between 'edu' and 'death', which yielded a correlation coefficient of 0.0123 with a p-value of 0.8453. These results indicated a very weak and statistically non-significant association between education level and survival. Furthermore, it is conducted a logistic regression that considered all relevant features. The logistic regression model returned a p-value of 0.9952 for the variable 'edu'. This high p-value reinforces the conclusion drawn from the correlation analysis, indicating that education level does not significantly predict survival outcomes in our model. I proceed with the removal of this feature.
  - Point-Biserial Correlation: Is a correlation coefficeint used when one variable (Y) is dichotomous. It is mathematically equivalent to the Person's Correlation Coefficient [1].

$$r_{pb} = \frac{\bar{X}_1 - \bar{X}_0}{s_X} \sqrt{\frac{n_1 n_0}{n(n-1)}}$$

    Where:
    - $\bar{X}_1$ and $\bar{X}_0$: mean of X for Y = 1 and Y = 0 respectively.
    - s_X: Standard Deviation of X.
    - n_1 and n_0: number of observations where Y = 1 and Y = 0 respectively.
    - n: number of total observations
  - Logistic regression is a supervised machine learning algorithm used for binary classification tasks by predicting the probability of a specific outcome [2]
- 'income': One plausible hypothesis is that higher patient income correlates with increased hospital charges. This could mean that those who spend more on healthcare might have better survival odds. Therefore, higher income levels could potentially signify improved chances of survival. However, Fig. 3. does not display any perceptible pattern supporting this hypothesis. Additionally, income does not appear to significantly influence either hospital charges or survival outcomes. Consequently, this feature can safely be removed from the analysis.
- 'scoma': This feature is a prediction generated by a model. The Coma Score, based on the Glasgow Coma Scale, ranges from a minimum of 3 to a maximum of 15 [3]. Fig. 4. illustrates that only the value 9 falls within this interval, indicating that the other values are incorrect. It is highly likely that the model made erroneous predictions. Therefore, this feature should be removed.

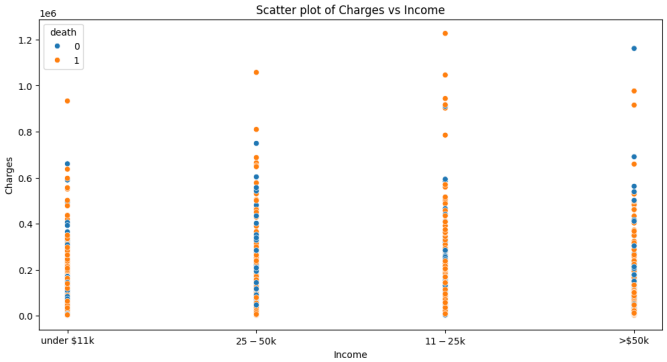

Fig. 3: Relation between 'income', 'charges' and 'death'

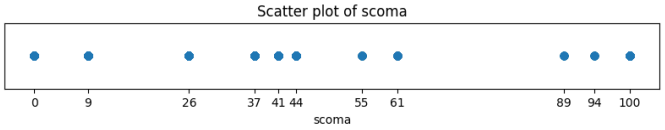

Fig. 4: Distribution of the predicted Coma Scores

- 'hday', 'dnrday': Both features represent points in time: one indicates the day when the patient entered the study at the hospital, and the other marks the day of the do-not-resuscitate order. Since the data is collected from different time periods and specific dates are not provided, and given that the goal of the project is not time series analysis, it would be more appropriate to delete both features.
- 'dzclass': This feature is deleted due to the existence of a more specific one, 'dzgroup'.
- 'pafi': Problem specification states that it diagnostic utility is controversial, therefore, I have excluded it from consideration.

Following these preprocessing steps, both the development and evaluation sets now contain 51 features (excluding 'Id' and 'death').

Besides feature selection, another approach to dimensionality reduction is Principal Component Analysis (PCA), as illustrated in Fig. 5. Using 30 Principal Components there is an achievement of more than 80% cumulative explained variance.

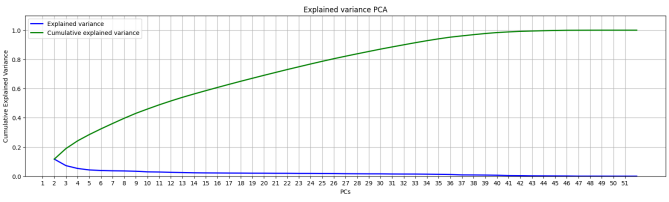

Fig. 5: Principal Component Analysis of the development set: Cumulative Explained Variance and Explained Variance of each Principal Component

B. Model selection

The following models have been considered:

- Random Forest Classifier: A random forest is an ensemble learning method used for classification and other tasks. It operates by constructing numerous decision trees during training. In classification tasks, the output of the random forest is determined by the majority class selected across all trees. It can be highly accurate and efficient for large datasets
- K-Nearest Neighbors Classifier (KNN): is a non-parametric supervised learning method where the output is a class membership. An object is classified based on a majority vote of its k nearest neighbors, assigning it to the class most common among them. KNN is straightforward to implement and can be robust to noisy training data and outliers due to its reliance on the entire training dataset rather than individual data points.

C. Hyperparameters tuning

A grid search was conducted for both the Random Forest Classifier and the K-Nearest Neighbors Classifier, employing 10-fold cross-validation for model validation. The hyperparameters explored during the search are detailed in TABLE I.

Model	Hyperparameters	Values
RF	n_estimators	{200, 225, 250}
	max_features	sqrt, log2
	criterion	{gini, log_loss, entropy}
KNN	n_neighbors	{3, 5, 7, 9, 11}
	metric	{manhattan, euclidean, minkowski}
	weights	{uniform, distance}

TABLE I: Hyperparameters

The best model returned by the grid search for Random Forest is : {'criterion': 'gini', 'max_features': 'log2', 'n_estimators': 250 } and for K-Nearest Neighbors is : {'metric': 'manhattan', 'n_neighbors': 3, 'weights': 'uniform'}

III. Results

A. Naive approach results

For a naive approach without hyperparameter tuning and utilizing all features, the Random Forest Classifier achieved a private score of 0.692, while the K-Nearest Neighbors Classifier attained a public score of 0.470; These will serve as internal baselines to analyze the validation scores.

B. Principal Component Analysis results

The public score for the Random Forest Classifier after applying PCA to the dataset is 0.663, while K-Nearest Neighbors (K-NN) achieved a score of 0.665. Random Forest with PCA performs worse than the naive Random Forest, whereas KNN performs better with PCA, though the improvement is not substantial. Therefore, this approach is discarded as it did not yield satisfactory results.

C. K-Nearest Neighbors

The best private 10-fold cross-validation score that resulted in our best model for K-Nearest Neighbors is 0.529. It does not improve the KNN-PCA approach, therefore, it falls short of being a satisfactory result.

D. Random Forest Classifier

The selected model from the grid search achieved an F1 macro score of 0.754 in 10-fold cross-validation on the private validation. Nevertheless, the prediction of the evaluation dataset for the public test achieved a higher F1 macro score of 0.755, which is the best score obtained so far. Despite this improvement, it does not surpass the best score currently displayed on the leaderboard.

Through a brute-force approach, I managed to achieve a private F1 macro score of 0.756 and public F1 macro score of 0.758 with the following model:

- criterion = entropy
- max_features = log2
- n_estimators = 248

This result improves upon both the private and public scores of the best model selected by the grid search. Hence, to conclude this section on results, this is the best model achieved.

IV. Discussion

It's noteworthy from the results of both the Random Forest Classifier and K-Nearest Neighbors that the public score is higher than the private validation score, which is typically expected to be the opposite. The following could be several reasons for this unusual behavior.

- The test set might have a distribution that is more favorable to the training set than the validation set.
- Cross-validation estimates the model's performance by averaging results across multiple folds. The variance in these folds can cause the performance to vary.
- The test set might be of different size with respect to the validation set, which can affect the performance metrics

The choice of imputation method could significantly enhance model performance. More complex approaches such as MICE (Multiple Imputation by Chained Equations), KNN Imputation, or iterative imputation methods are often better suited for handling missing values in medical datasets.

The utilization of more sophisticated models could also improve accuracy, models such as Gradient Boosting Machine (GBM): Known for their exceptional accuracy and versatility, these build an ensemble of trees sequentially, where each new tree helps correct errors made by previously trained trees, with the goal to construct a robust and predictive model [4].

Another strategy to enhance performance could involve conducting a more exhaustive grid search.

However, the achieved F1 macro score of 0.758 is highly satisfactory when compared to the public and private baselines.

REFERENCES

[1] Wikipedia, "Point-biserial correlation coefficient," 2024. [Online]. Available at https://en.wikipedia.org/wiki/Point-biserial_correlation_coefficient.

[2] V. Kanade, "What is logistic regression? equation, assumptions, types, and best practices," 2022. [Online]. Available at https://www.spiceworks.com/tech/artificial-intelligence/articles/what-is-logistic-regression/.

[3] C. Clinic, "Glasgow coma scale (gcs)," 2023. [Online]. Available at: https://my.clevelandclinic.org/health/diagnostics/24848-glasgow-coma-scale-gcs.

[4] D. Dillu, "Understading gradient boosting machines (gbm)," 2023. [Online]. Available at: https://shorturl.at/HkKJv.