Vrije Universiteit Brussel

DAHCCC Project Report

Mortality Prediction Model of Patients in ICU with HF

Siyan Luo, Weiyi Wang

Mehran Khodadadzadeh Gojeh, Saeid Shahriari

2023-5

Table of content

Introduction

Heart failure is a syndrome due to a heart function disorder, it's the final phase of cardiovascular disease. Heart failure has posed a huge threat to human health, such that mortality prediction modeling is worth a dive in.

This research targets the mortality prediction modeling of intensive care units (ICUs)-admitted heart failure (HF) patients.

We started this project with data extraction through Google big query. Then we processed the raw data to a clear dataset using methods including one-hot encoding and missing data supplementation. To reduce the dimensions, we selected the most important 11 features using eXtreme Gradient Boosting (XGBoost). For the modeling part, we applied 4 classification models, Decision Tree (DT), Random Forest (RF), Gradient Boosting (GB), and XGBoost, with the combination of hyperparameter tuning and cross-validation. The result shows RF has the best performance with the AUC reaching 0.83 and the accuracy reaching 0.75.

Data and methods

Data extraction

Taking the literature[1] as a reference, we extracted 48 features including 5 demographic characteristics, 7 vital signs, 9 comorbidities, and 27 laboratory variables.

In the queries, we filtered data with the condition of patients in the CPU with heart failures.

Data processing

For demographic characteristics, we calculated ages based on the date of birth and admit time as a feature. As we noticed there were some abnormal data in ages such as 300, we filtered out data with the age larger than 100.

For comorbidities, we took each disease as a feature and one-hot encoded it.

For vital signs and lab variables, since the data is in numbers with units, firstly, we checked all if the measurements of units are the same for each feature and unified

them. We noticed that for the same HADM_ID, there are multiple records, so we computed the mean value as the representatives.

After combining all the features, we found there are lots of missing data which would result in biased output if we just simply discard or fill them all. Also, the dataset we got so far consists of 15277 records which is too large for modeling so we also had to narrow down the dataset in a proper way.

Before missing data supplementation, we felt like there was no point to keep the rows and columns which have reached a specific data missing tolerance. Firstly, we checked the missing data horizontally, we dropped the records which has more than 25% missing data. The dataset shrank to 5906.

As we noticed even though we had filtered some erroneous data during data extraction, there were still some abnormal data. To filter them out, we calculated the Interquartile Range (IQR) of each feature, which is the difference between the 75th percentile and the 25th percentile of the data. We dropped data that is out of the range of 1.5 times the IQR. Till then, the dataset was decreased to 4134.

At last, we checked the missing data vertically, we dropped the columns with more than 10% missing data. For the rest missing data, we made a fillup using the Shapiro-Wilk test. For normally distributed data, we took the mean value as a supplement, for skewed distributed data, we took median values.

In the end, we got the dataset of the shape (3720, 36).

Data visualization

In the data analysis and visualization section, we combined individual analysis with the XGBoost model mentioned in the reference for feature selection.

Using preprocessed data for statistical description, we can observe the overall situation of the current data set. First, we present the distribution of death. The difference between the number of survivors and deaths was not significant, showing the validity of this data set. Second, we plot feature heatmaps, revealing some highly correlated features. We need to do further processing on these features.

Next, we perform classification analysis on different features. At this stage, we mainly use the histogram to realize the visual presentation of our analysis results, because it can provide a more intuitive visual perception effect.

First, we examined the effect of gender characteristics on death conditions. According to the results of the visualized histogram, we found no significant difference in the death rate between males and females. (Figure1) Therefore, we speculated that gender has no effect on mortality, so this feature will be discarded in subsequent data processing.

The second characteristic is ethnicity. Through the observation of the histogram, we found that the vast majority of samples came from the white population. Therefore, racial characteristics did not provide any informative value for mortality prediction. Therefore, in subsequent data processing, we will also discard this feature.

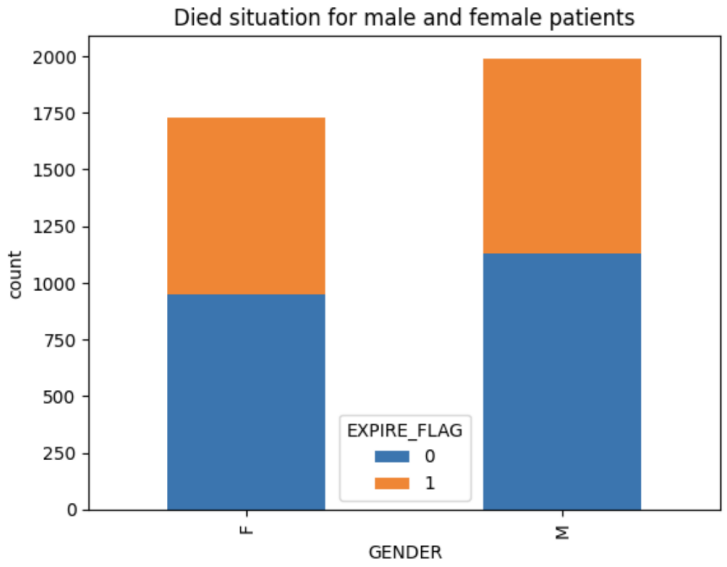

Figure 1. Label distribution based on gender

The third characteristic is age. According to the results of the histogram, we can clearly observe that the death rate is higher in the older population. Therefore, we speculated that age will be one of the features that will be of great significance in subsequent machine learning.

For the analysis of comorbidities, we performed a correlation matrix and linear regression coefficient. In addition, we show the proportion of patients who died with different complications to assess the possible impact of certain complications on patient mortality. Through the analysis of the histogram, we concluded that almost all the patients who died suffered from the disease hypoferric_anaemia.

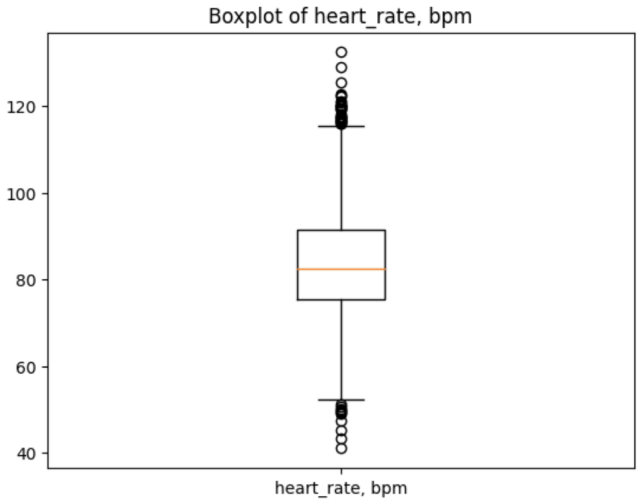

Figure 2. Boxplot of heart rate

For the analysis of vital signs and laboratory indicators, we drew boxplots for each feature. (Figure2) By observing the boxplot, we found that, for example, the prothrombin_time feature has too many outliers. Therefore, in subsequent processing, we choose to discard this feature.

All of the above analyzes were performed manually. We discard the gender and ethnicity which obviously are useless features, and get our new dataset. Then according to the guidance of the reference, we chose to use the XGBoost model to output the top eleven feature importances to construct the final data set and perform subsequent modeling and learning operations.

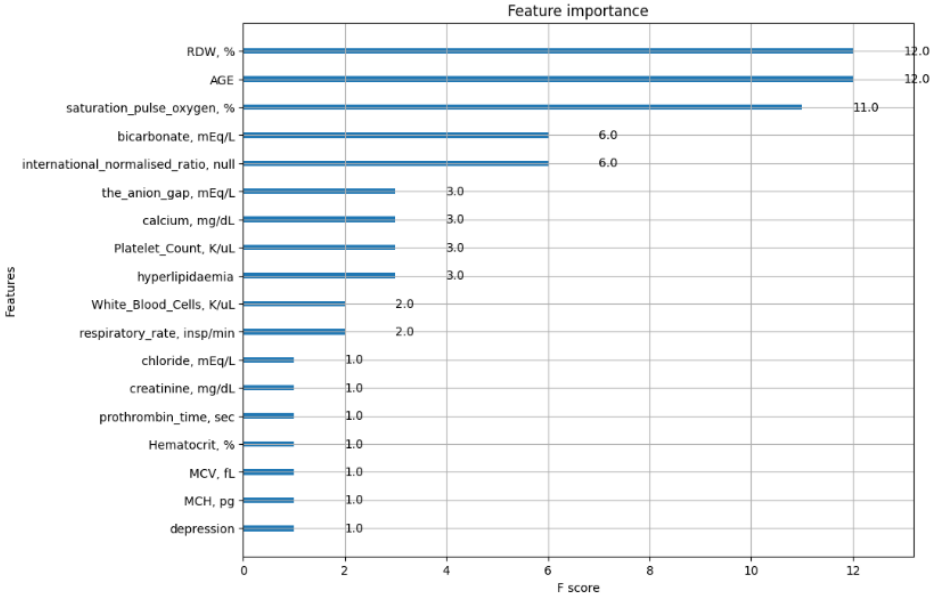

Figure 3. Feature selection result of XGBoost

Modeling

For model training, we used 4 models, Decision Tree (DT), Random Forest (RF), Extreme Gradient Boosting (XGBoost), and Gradient Boosting (GB). Among them, the former two have been covered in class.

XGBoost is a popular technique due to its efficacy and effectiveness in predictive modeling tasks. It sequentially creates an ensemble of weak prediction models, with each new model fixing the errors generated by the prior models. XGBoost is well-known for its ability to handle complex patterns with high accuracy.

Like XGBoost, Gradient Boosting is a strategy that combines prediction models. It operates in a similar sequential design, improving model performance by iteratively minimizing a loss function. Gradient Boosting models are efficient at collecting complicated patterns while reducing bias.

At first, we also tried Logistic Regression, but we got an error since our dataset is too large for this algorithm. Then we chose Decision Tree because it has interpretability and feature importance analysis. Then we chose Random Forest, XGBoost, and Gradient Boosting because they are often more suited for large data sets than Logistic Regression.

Each model was performed with the combination of GridSearchCV to find the best hyperparameters. We evaluated the performance of each model using the confusion metrics, ROC curve, and 10-fold cross-validation.

Results

The result showed that RF has the best performance with AUC reaching 0.83, accuracy reaching 0.75, sensitivity reaching 0.64, and specificity reaching 0.84. While DT has the worst performance, this could due to its limited ability to recognize complicated patterns and the potential of overfitting. The results of XGBoost and GB are quite close and slightly worse than RF. While as an optimized and enhanced implementation of the gradient boosting algorithm, XGBoost didn't outperform GB much. The reason behind this might be the dataset is not complicated enough, both XGBoost and GB are capable of effectively handling it.

	DT	**RF**	XGBoost	GB
AUC	0.72	**0.83**	0.80	0.81
Accuracy	0.67	**0.75**	0.73	0.73
Sensitivity	0.48	**0.64**	0.65	0.63
Specificity	0.83	**0.84**	0.80	0.81
10-fold cv AUC	0.70	**0.80**	0.78	0.80
10-fold cv Accuracy	0.67	**0.73**	0.70	0.72
10-fold cv Balanced Accuracy	0.65	**0.72**	0.69	0.71
10-fold cv Recall	0.54	**0.64**	0.60	0.63
10-fold cv Precision	0.64	**0.71**	0.67	0.70
10-fold cv F1 score	0.58	**0.67**	0.63	0.66

Table 1. Results of model training

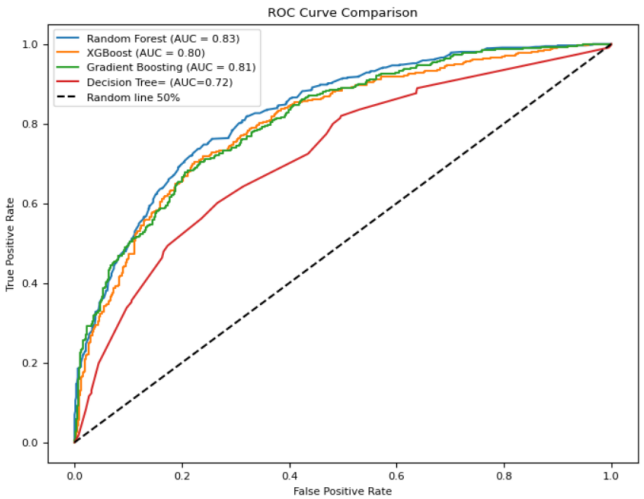

Figure 4. ROC curves of all models

Discussion

Using data derived from the MIMIC-III database, we developed the in-hospital mortality prediction nomogram with variables selected by the XGBoost model using multiple models and got relatively good results, but there are still some limitations worth discussing.

Firstly, the data were collected from patient medical records and we relied on the accuracy of the records. Later we can monitor the model's performance on new data on a regular basis and change it as needed. Because medical data might change over time, it is critical to retrain and validate the model on a regular basis.

Secondly, in this project, we cannot use the logistic regression analysis because the data set is too large. Afterward, we should further screen the data to formulate standards, obtain more concise and effective data sets, and participate in training

Thirdly, when dividing the data set, it is obvious that the mortality rate of the elderly is higher, so it’s better to consider that the training set and the test set should have the same age hierarchy, or based on other strategies to split the dataset not randomly, may help us construct a more robust model.

The above are some areas where we need to discuss further to make a robust prediction model.

Conclusion

From data extraction to processing and feature selection, we constructed a dataset consisting of 3720 records and 11 features. After mode training using 4 different models, we found out that Random Forest has decent performance at predicting in-hospital mortality in intensive care unit patients with heart failure.

Workload Division

Data extraction	Each group member was in charge of some features.
Data processing	Following the templates and methods provided by Siyan, each group member processed the data they extracted. The integration of code and datasets were done by Weiyi.
Data visualization	Completed by Weiyi
Modeling	Constructed by Mehran with the help of Siyan
Report Composition	Written by Siyan, Weiyi, and Mehran

References

[1] Li, Fuhai, et al. "Prediction model of in-hospital mortality in intensive care unit patients with heart failure: machine learning-based, retrospective analysis of the MIMIC-III database." *BMJ open* 11.7 (2021): e044779.

[2]https://machinelearningmastery.com/feature-importance-and-feature-selection-with-xgboost-in-python/