Data Challenge

Yizhou Zhang

April 5, 2023

1 Introduction

In this data challenge, I aim to solve a medical problem in which we need to predict whether the patient will develop sepsis based on patients' information including vital signs, laboratory values, and demographics. I proposed a gradient boosting model which takes the summary statistics of patients information as input. The result on validation set shows a strong result.

2 Dataset

The dataset is composed of 21634 samples, with 15144 samples in training set and 6490 samples in test set. In training set, we have the binary response variable which indicates whether this patient has sepsis or not. There are approximately 40 prediction variables. In the training set, each patient's information is stored in a hourly time sequence. Since samples only have values in a very limited number of variables in each hour, the raw data is highly sparse and then further processing is needed before modelling.

3 Data Preprocessing and Exploratory Data Analysis

3.1 Data Preprocessing

By analyzing the properties of prediction variables. I decided to treat summary statistics of prediction variables as new prediction variables. The summary statistics includes mean, min, and max. In this way, each sample's vairables are stored in a vector. Therefore, the size and complexity of data are greatly reduced, and the design matrix becomes much less sparse, which makes imputation more accurate.
Apart from all summary statistics variables, I added one more variable: the time length of patient's stay in ICU. I found the time in ICU plays a significant role in prediction. I will elaborate this in the next section.

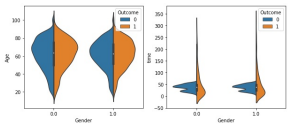

Figure 1: The violin plots indicate time length in ICU shows huge differences between two classes.

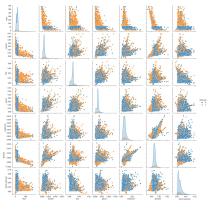

Figure 2: The pairplot of several important features of patients

3.2 Exploratory Data Analysis

I conducted exploratory data analysis on some demographics variables and some other variables that may be important for the diagnosis of sepsis. I have several fundamental findings:

- The time length of staying in ICU will be a significant feature in prediction, as shown in Figure 1 and Figure 2.
- Other features do not show big difference in density functions between two classes, as shown in the diagonal plots in Figure 2.
- Combining two or more features may be helpful to predict whether the patient will develop sepsis.

Considering the exploratory data analysis results, I concluded that applying tree models would work well given these conditions. XGBoost is a gradient boosting model that works excellently on classification. The XGBoost model has a variety of hyperparameters that help to reduce overfitting and improve accuracy on imbalanced data. More details are introduced in the following section.

4 Models and Experiments

4.1 Modelling

Boosting is an ensemble algorithm that constructs a collection of learners. In this algorithm, the model will first make a weak learner, then use the loss from previous model to fit the next learner. Then the final model is a weighted sum of all learners.
XGBoost is a gradient boosting decision tree (GBDT) model. In this model, there are lots of regularization methods to reduce the overfitting which will greatly improve the accuracy of prediction. As a result, I applied XGBoost as my classfication model.
The given data is highly imbalanced with class 0 consisting of approximately 86% of all training data. More importantly, in the medical field, false negative is generally more serious than false positive. As a result, it's very important to address the imbalance problem to diminish the possibility of false negative. As a result, I raised the negative weight by setting the hyperparameter 'scale_pos_weight' as the ratio of number of class 0 and number of class 1. There are some hyperparameters that are quite useful in preventing overfitting problems, among them are:

- 'max_depth' This hyperparameter limits the max depth of each tree.
- 'eda' which is the learning rate.
- 'min_child_weight' This hyperparameter sets a threshold of tree partitioning. If the leaf node has a sum of hessian less than this value, then the tree stops further partitioning.

4.2 Experiments

4.2.1 Hyperparameter Tuning by Cross Validation

By applying grid search, I obtained the hyperparameters values by 10-fold cross validation. The result from cross validation is that:

max_depth	eda	min_child_weight	auc	auc_sd	ber
3	0.267	5	0.9276	0.006246	0.8576

The feature importance shown by XGBoost model shows that the time length of ICU is the most important. Some moderately important features includes: minimum of PaCO2 and maximum of temperature.

5 Conclusion

In this challenge, I summarized the mean, min, and max value of a variety of features recorded during patients' stay in ICU and time length in ICU as the features. I applied XGBoost, a gradient boosting method to predict whether the patient developed sepsis. I also applied several ways to reduce overfitting. The method generates a very good result. It can solve imbalanced data problem and accurately predict in both classes.