Comparative Analysis of Machine Learning Models for Predicting Bacteremia

Yueting Zhang, Melin Li

2024-04-29

Abstract

This project aims to predict the blood culture result for patients who are suspected to have bloodstream infection due to Bateremia, using machine learning classification methods. We tested four major models to see their performance on a real hospital record of blood test statistics and blood culture results. The four models are: logistic regression, Lasso, Random Forest and Boosting. Various methods are applied to these models to improve their capability of classifying minority groups in the imbalanced dataset. Among the four models, we discovered that logistic regression and Lasso performed the best, while Random Forest and Boosting have their imperfection.

Introduction

Bacteremia is a clinical situation under which there's presence of bacteria in blood. It can be either dangerous or mild, depending on whether the person's immune system successfully responds and eliminates the bacteria. If the immune system works perfectly, bacteremia will be self-cured, and no medical treatment is needed. However, if the immune system fails to react, bacteremia may develop into acute and deadly consequences such as sepsis, systemic inflammatory response syndrome (SIRS), and multiple organ dysfunction syndrome (MODS). Once the bloodstream infection occurs, the patient can have symptoms like fever, chills and shaking, which will be suspected of having the infection and blood culture will be required for diagnosis. Deadly consequences can happen in a very short period of time so prompt antibiotic treatment is required. For precise antibiotics prescription to be given, the exact type of bacteria needs to be identified. However, blood culture can take as short as 2 days, or can also be over 7 days for organisms that are hard to culture in laboratory environment. Although very likely to be cured once treated correctly, the mortality of bacteremia in emergency department (ED) can be as high as 37% due to untimely treatment.

Such high-risk disease is paid high attention in clinical scenarios. The doctors usually decide to proceed with blood culture for most of the patients with suspected symptoms. However, the result of blood culture usually have a low positive rate, and even a high false positive rate due to contamination or other incidents in lab environment. Thus, suspected bloodstream infection can lead to high hospitalization rate, economic burden, and workload for hospital faculty.

We want to figure out a machine learning method which fastens the diagnosis process and save resources. However, disease detection has long been a discussed topic in machine learning which is known for imbalanced data set and difficulty in training. In this report, several models were tested using patients' blood test data from Vienna General Hospital, Austria. In the data set, the blood culture results has only a 10% positive rate. To deal with the imbalanced and high dimensional data set, resampling, class weight adjustment, and principle component analysis (PCA) were tested. During our investigation, we paid extra attention to true positive rate (TPR) and true negative rate (TNR) due to the specific setting. Among all of the models we tested, Lasso and Random Forest are proven to be especially effective with over 50% true positive rate.

In this report, we are going to discuss about details of our data set, methods we use, and visualization of our results. Our investigation provides new insights about model selection for disease detection, and also some practical solutions to classification models when imbalanced outcome variables are present.

Data and Methods

Codebook

The original data comes from the Vienna General Hospital, Austria Between January 2006 and December 2010. After excluding the ommiting values, it includes 3979 obs. of 52 variables. **BloodCulture** is variable we are expected to predict. However, we find it unbalanced:

```
## [1] "Number of positive blood culture tests: 321"
```

```
## [1] "Number of negative blood culture tests: 3658"
```

The details of each variable present in the following table.

Table 1: Variable Descriptions

Variable	Label	Units	Variable	Label	Units
SEX	Patient sex (male=0; female=1)	binary	MCV	Mean corpuscular volume	pg
AGE	Patient Age	years	HGB	Haemoglobin	G/L
PLT	Blood platelets	G/L	HCT	Haematocrit	%
MCH	Mean corpuscular hemoglobin	fl	MCHC	Mean corpuscular hemoglobin concentration	g/dl
RDW	Red blood cell distribution width	%	MPV	Mean platelet volume	fl
LYM	Lymphocytes	G/L	MONO	Monocytes	G/L
EOS	Eosinophils	G/L	BASO	Basophiles	G/L
NT	Normotest	%	APTT	Activated partial thromboplastin time	sec
FIB	Fibrinogen	mg/dl	SODIUM	Sodium	mmol/L
POTASS	Potassium	mmol/L	CA	Calcium	mmol/L
PHOS	Phosphate	mmol/L	MG	Magnesium	mmol/L
CREA	Creatinine	mg/dl	BUN	Blood urea nitrogen	mg/dl
HS	Uric acid	mg/dl	GBIL	Bilirubin	mg/dl
TP	Total protein	G/L	ALB	Albumin	G/L
AMY	Amylase	U/L	PAMY	Pancreas amylase	U/L
LIP	Lipases	U/L	CHE	Cholinesterase	kU/L
AP	Alkaline phosphatase	U/L	ASAT	Aspartate transaminase	U/L
ALAT	Alanin transaminase	U/L	GGT	Gamma-glutamyl transpeptidase	G/L
LDH	Lactate dehydrogenase	U/L	CK	Creatinine kinases	U/L
GLU	Glucoses	mg/dl	TRIG	Triclyceride	mg/dl
CHOL	Cholesterol	mg/dl	CRP	C-reactive protein	mg/dl
BASOR	Basophile ratio	%	EOSR	Eosinophil ratio	%
LYMR	Lymphocyte ratio	% (mg/dl)	MONOR	Monocyte ratio	%
NEU	Neutrophiles	G/L	NEUR	Neutrophile ratio	%
PDW	Platelet distribution width	%	RBC	Red blood count	T/L
WBC	White blood count	G/L	BloodCulture	Blood culture result for bacteremia	no, yes

Classification

Lasso

We first consider the Lasso regression model to predict and analyze the BloodCulture response variable in the training data and validate the model's performance on testing data. Lasso regression increases the sparsity of the model by imposing an absolute value penalty on the coefficients, causing the coefficients of less important variables to tend toward zero, thus achieving automatic selection of variables and reducing complexity.

A model matrix is first created and then standardized. Subsequently, a binary classification Lasso model is fitted using the cv.glmnet function with cross-validation.

Additionally, to address potential class imbalance issues, the weights parameter is used to assign weights to the classification outcomes, giving higher weights to the minority class (BloodCulture == yes). This is because in medical and other fields, the identification of minority classes is often more important and more challenging. Increasing their weight helps the model pay more attention to these categories, potentially improving the model's prediction accuracy for minority classes.

Next, the fitted Lasso model is used to make predictions on the test dataset, and the probabilities generated are used to determine the classification of each sample (threshold set at 0.5).

Logistic regression

Secondly, we adopted a logistic regression model for prediction. We set the parameters and introduced weights similar to what we do in the LASSO model.

Random forest

Next, we use the Random Forest algorithm to classify and predict bacteremia data. Random Forest builds on bagging by creating B bootstrapped samples from the original data and constructing a tree for each. It introduces additional randomness by selecting a subset of features for building each tree, rather than using all features. This approach simplifies each tree, reduces variance, and diversifies the ensemble of trees, decreasing the correlation between their predictions.

We employ the SMOTE (Synthetic Minority Over-sampling Technique) to oversample the training data, enhancing the number of minority class samples with parameters perc.over = 300 and k = 10. This helps address class imbalance issues, where perc.over = 300 increases the minority class samples to 300% of their original number, and k = 10 refers to the number of nearest neighbors considered while generating synthetic samples.

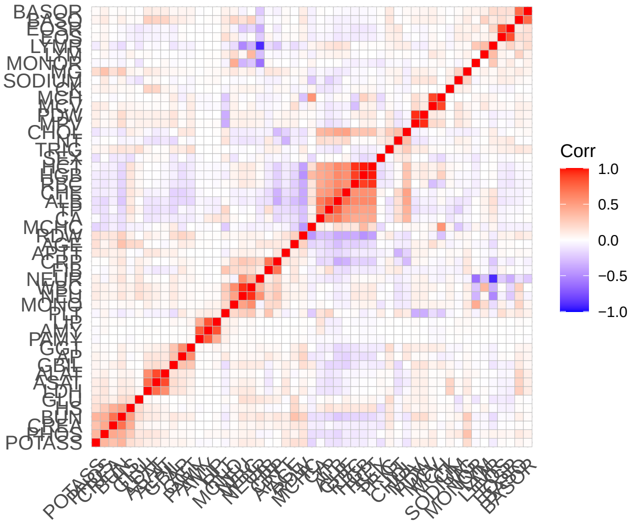
In configuring the Random Forest model, we specifically set up 500 trees, which we find that it give the best performance. We use Lasso regression to select features before inputting them into the Random Forest to reduce the complexity of the model and improves the accuracy of predictions. By this method, the Random Forest model can focus on the most predictive variables, thereby optimizing the overall performance of the model.

Boosting

We implement a boosting algorithm using the gbm function with a multinomial distribution to predict **BloodCulture** outcomes from the taining data. The model is configured with 1000 trees, an interaction depth of 4, a learning rate (shrinkage) of 0.05, and weighted instances to address class imbalance. Despite these efforts, the resulting analysis indicates that the boosting model's performance is not satisfactory.

Classification with PCA

Given that the data contains 51 numerical features, we would apply a principal component analysis (**PCA**) method called for dimension reduction. PCA, by extracting the main sources of variation, helps to uncover the hidden and most informative structures within the data.



```
## Importance of components:
##                           PC1     PC2     PC3     PC4     PC5     PC6     PC7
## Standard deviation     2.4545 1.99004 1.80961 1.69706 1.60876 1.55383 1.53719
## Proportion of Variance 0.1181 0.07765 0.06421 0.05647 0.05075 0.04734 0.04633
## Cumulative Proportion  0.1181 0.19578 0.25999 0.31646 0.36721 0.41455 0.46088
##                           PC8     PC9    PC10    PC11    PC12    PC13    PC14
## Standard deviation     1.4229 1.36551 1.30793 1.25634 1.21941 1.16520 1.11798
## Proportion of Variance 0.0397 0.03656 0.03354 0.03095 0.02916 0.02662 0.02451
## Cumulative Proportion  0.5006 0.53714 0.57069 0.60164 0.63079 0.65741 0.68192
##                           PC15   PC16    PC17    PC18    PC19    PC20    PC21
## Standard deviation     1.08370 1.0568 1.03226 0.96819 0.95410 0.93268 0.91127
## Proportion of Variance 0.02303 0.0219 0.02089 0.01838 0.01785 0.01706 0.01628
## Cumulative Proportion  0.70495 0.7268 0.74774 0.76612 0.78397 0.80103 0.81731
##                           PC22    PC23    PC24    PC25    PC26    PC27    PC28
## Standard deviation     0.87588 0.85500 0.82277 0.80095 0.76831 0.75087 0.73120
## Proportion of Variance 0.01504 0.01433 0.01327 0.01258 0.01157 0.01105 0.01048
## Cumulative Proportion  0.83235 0.84669 0.85996 0.87254 0.88411 0.89517 0.90565
##                           PC29    PC30    PC31    PC32    PC33    PC34   PC35
## Standard deviation     0.70030 0.68186 0.66566 0.64778 0.62495 0.60275 0.5976
## Proportion of Variance 0.00962 0.00912 0.00869 0.00823 0.00766 0.00712 0.0070
## Cumulative Proportion  0.91527 0.92438 0.93307 0.94130 0.94896 0.95608 0.9631
##                           PC36   PC37    PC38    PC39    PC40    PC41   PC42
## Standard deviation     0.58345 0.5247 0.47836 0.45684 0.43703 0.41160 0.3641
## Proportion of Variance 0.00667 0.0054 0.00449 0.00409 0.00374 0.00332 0.0026
## Cumulative Proportion  0.96976 0.9752 0.97964 0.98374 0.98748 0.99080 0.9934
##                           PC43    PC44   PC45    PC46    PC47    PC48    PC49
## Standard deviation     0.33885 0.31757 0.2370 0.20615 0.12842 0.05541 0.04384
## Proportion of Variance 0.00225 0.00198 0.0011 0.00083 0.00032 0.00006 0.00004
## Cumulative Proportion  0.99565 0.99763 0.9987 0.99956 0.99989 0.99995 0.99999
##                           PC50      PC51
## Standard deviation     0.02669 2.287e-15
## Proportion of Variance 0.00001 0.000e+00
## Cumulative Proportion  1.00000 1.000e+00
```

The confusion matrix illustrates the relative low corrlation between features. As observed, 51 principal components (PCs) were generated, with the first two PCs explaining only 19.6% of the variance, which might indicate insufficient explanatory power. However, to enhance our model's effectiveness when applying above models , we opt to use the first 15 principal components that cumulatively explain over 70% of the variance. In all the PCA training dataset we also address class imbalance by over-sampling the minority class. And predictions are made on the PCA-reduced test dataset.

We will see if integrating **PCA** into the classificaiton process before running **Logistic**, **Lasso**, **Random Forest**, and **Boosting** model can improves prediction performance. Performance results and discussion will be illustrated in next section.

Results

Table 2: Comparison of all models

Model	Accuracy	TPR	TNR	AUC
Logistic	0.76	0.59	0.77	0.74
Lasso	0.75	0.56	0.77	0.72
Rforest	0.70	0.51	0.73	0.67
Boosting	0.89	0.21	0.96	NA
PCA-Logistic	0.74	0.56	0.76	0.72
PCA-Lasso	0.75	0.56	0.76	0.72
PCA-Rforest	0.76	0.34	0.81	0.63
PCA-Boosting	0.89	0.11	0.97	NA

After making our best effort to balance overall accuracy and sensitivity, our evaluation statistics are shown in Table 2. There are four main statistics we focus on: accuracy, true positive rate (TPR), true negative rate (TNR), and AUC value. The calculation for TPR is: TPR = True Positive/(True Positive + False Negative). The calculation for TNR is: TNR = True Negative/(True Negative + False Positive). Note that in R Caret package, TPR is noted in "Specificity" while TNR is noted in "Sensitivity", which is different from what we usually interpret. Besides accuracy, these two statistics are especially important because in disease detection, true positive and true negative are determinants of life-saving treatments. False positive and false negative not only waste time and resources, but also can let the patient miss the best treatment time. The AUC is the area under ROC curve, which is a visualization of the classification model performance using TPR and FPR (false positive rate). Closer to 1 the AUC is, better the model performs.

Among all 8 models which have been tested, the regular logistic regression model did the best in TPR, and shows a stable performance but no improvement with PCA. PCA-boosting did the worst in TPR, which only successfully identified 11% of the true positive blood culture results. However, Boosting had a better overall accuracy, which is mostly contributed by high TNR. The performance of Boosting under this specific setting indicates that it doesn't have a good ability identifying the minorities in a dataset, even though weighted classes are defined.

Besides logistic regression, Lasso also had a stable and high performance in identifying the minority group when weighted classes are defined. They also return satisfying results for the majority group. In real life setting, although we need a good overall accuracy, sometimes they are contributed by TNR and will return biased results for positive groups. That is to say, if the TNR of a classifier is too high, while the TPR is low, the model would probably return most of the results as "negative" and still keep a good overall accuracy, because there are only 10% of positive results present. However, for life-saving purposes, such models cannot be applied because it would miss the people who need prompt treatment. Thus, it is important that logistic and Lasso can do a good job in identifying minorities, because it approaches the major purpose of developing such model.

Random forest had a mediocre performance among these models. Although the overall statistics are satisfying, it contributes the two lowest AUC values in this comparison. That is to say, although it has a relatively higher TPR, its combined performance is not high. The valuable information is that the random forest can be greatly improved in terms of learning minority group after resampling and class weights. However, the random forest lost some minority classification power under PCA.

Conclusion

In conclusion, we received evident result which differentiates the performance on blood stream infection detection of 4 major classifiers: logistic regression, Lasso, Random Forest and Boosting. Under the setting of fatal disease detection, logistic regression performed the best, while Boosting had its strength but is too unbalanced. Random Forest can be greatly improved by resampling and class weight methods, and can have a relatively higher TPR after improvement. For this specific dataset containing 52 variables, PCA doesn't seem to be helpful in terms of assisting other models to learn better about the minority group. Since model performance may vary under different setting and datasets, we cannot conclude that the logistic regression and Lasso are just better classifiers. Provided with some insights about how different models may perform with imbalanced dataset, we can have a better knowledge and expectation regarding model selection.

Reference

Julián-Jiménez A, González Del Castillo J, García-Lamberechts EJ, Huarte Sanz I, Navarro Bustos C, Rubio Díaz R, Guardiola Tey JM, Llopis-Roca F, Piñera Salmerón P, de Martín-Ortiz de Zarate M, Álvarez-Manzanares J, Gamazo-Del Rio JJ, Álvarez Alonso M, Mora Ordoñez B, Álvarez López O, Ortega Romero MDM, Sousa Reviriego MDM, Perales Pardo R, Villena García Del Real H, Marchena González MJ, Ferreras Amez JM, González Martínez F, Martín-Sánchez FJ, Beneyto Martín P, Candel González FJ, Díaz-Honrubia AJ; INFURG-SEMES investigators. A bacteraemia risk prediction model: development and validation in an emergency medicine population. Infection. 2022 Feb;50(1):203-221. doi: 10.1007/s15010-021-01686-7. Epub 2021 Sep 6. PMID: 34487306.

Lee KH, Dong JJ, Kim S, Kim D, Hyun JH, Chae MH, Lee BS, Song YG. Prediction of Bacteremia Based on 12-Year Medical Data Using a Machine Learning Approach: Effect of Medical Data by Extraction Time. Diagnostics (Basel). 2022 Jan 3;12(1):102. doi: 10.3390/diagnostics12010102. PMID: 35054269; PMCID: PMC8774637.

National Library of Medicine, https://www.ncbi.nlm.nih.gov/books/NBK441979/

Ratzinger F, Dedeyan M, Rammerstorfer M, Perkmann T, Burgmann H, Makristathis A, Dorffner G, Lötsch F, Blacky A, Ramharter M. A risk prediction model for screening bacteremic patients: a cross sectional study. PLoS One. 2014 Sep 3;9(9):e106765. doi: 10.1371/journal.pone.0106765. PMID: 25184209; PMCID: PMC4153716.