Classification Models for Acute Kidney Injury Based on Machine Learning Algorithms: An Report for SPH6004 Assignment 1

SHI Yile

Saw Swee Hock School of Public Health, NUS

March 7, 2024

1 Introduction

Acute kidney injury (AKI), previously known as acute renal failure, refers to a sudden and often reversible decline in kidney function, marked by a rapid increase in serum creatinine levels, reduced urine output, or both [1] [2]. It is a serious medical condition that can occur due to various factors such as dehydration, severe infections, medication toxicity, or kidney damage from trauma [3], which shows association with increased morbidity, mortality, duration of hospital stay, and healthcare cost [4]. Although potentially results in high public health burden, the diagnosis of acute kidney injury remains difficulties due to the lack of forecasting models with high accuracy, which often leads to delayed diagnosis in practice [5]. It is important to propose early and accurate detection of acute kidney injury in clinical scenarios, as recognizing earlier stages of renal impairment allows for early appropriate action that may interrupt the process of functional decline in kidney.

This study aims to construct models for acute kidney injury classification with high accuracy, based on a pre-extracted subset of MIMIC-IV (Medical Information Mart for Intensive Care IV) [6] database and state-of-art machine learning algorithms. Potential risk factors important to the risk of acute kidney injury will be determined, and further models performance will be compared. The results are expected to shed a light on early detection of acute kidney injury.

2 Materials & Methods

2.1 Dataset

This study used a pre-extracted dataset with 162 columns and 50920 observations, from MIMIC-IV (Medical Information Mart for Intensive Care IV) database, a large, freely-available database comprising of de-identified health-related data of over forty thousand patients who stayed in critical care units at the Beth Israel Deaconess Medical Center. The extracted subset is a snapshot of patients' health status at admission that comprises of demographics, vital sign measurements made at bedside, laboratory test results, etc.

The outcome, `aki`, is patients' status of acute kidney injury in 4 levels, where 0 stands for no renal impairment and 1 to 3 stands for increasing severity in renal injury. Here, we converted it to a binary outcome with 1 for kidney injury and 0 for no injury.

Before any further selection and modelling, we recoded categorical variables into dummy variables where 1 represents the presence of the corresponding level of the variable, while 0 represents its absence. In particular, over 20 levels in the column for patient's ethnicity was observed, while the numbers of observations among levels were severely imbalanced, which may lead to biased and unstable model performance. As a remedy, we re-categorized patient's race into 6 levels, minimizing the imbalance issue in this variable to some extend.

2.2 Feature Selection

Feature selection shows its great importance as a proper selection on variables can not only reduce model complexity and increase its efficiency, but also enhance model performance, by dropping redundant or problematical columns and keeping relevant variables for outcome forecasting. In this study, feature selection consisted of two main steps: initial selection before model construction and further selection during model fitting.

2.2.1 Selection before model fitting

Two main criteria were considered during the feature selection before model fitting:

- **Missing values**

  Issue of missing values was detected in the dataset, which may significantly affect the model performance. By computing the proportion of missing values, we observed that numbers of columns included over 50% missing values, indicating a great loss of statistical information in these variables. Thus, we manually dropped columns with over 50% missing values, and as for the rest of variables, we only kept observations complete in each column because some algorithms, such as Logistic regression, requires datasets with no missing values.

- **Feature importance (mutual information)**

  After dealing with the issue of missing values, we measured the importance to AKI of each factor by calculating its mutual information with AKI. Mutual information (MI) is a measure of mutual dependence between two random variables X and Y, with the following definition:

  $$I(X;Y) = D_{KL}\big(P_{(X,Y)}||P_X \otimes P_Y\big)$$

  where D_{KL} is the Kullback–Leibler (KL) divergence and $P_X \otimes P_Y$ is the outer product distribution of X and Y.

  Mutual information ranges in $[0, +\infty)$, with 0 indicating independence between variables and larger value suggesting stronger dependence. Comparing to Pearson's correlation which only measures the strength of linear relationship between variables, MI shows its superiority for capturing both linear and non-linear associations.

  Here, columns with 0 mutual information were considered to be dropped, as they showed no association with AKI. Inclusion of them in models would increase model complexity and running time but have insignificant contribution for classification.

2.2.2 Selection during model fitting

Some state-of-art algorithms can be applied to select important features when fitting method is specified. Here, Genetic Algorithm (GA) was considered for further feature selection during model fitting.

GA is a metaheuristic optimization technique inspired by the process of natural selection and evolution. Initially, a population of potential feature subsets and fitness function were determined. In each generation, the performance of fitness function in each subset candidate was evaluated and well-performed ones were selected for reproduction in the next generation. The selected candidates undergo genetic operations to produce offspring, which replace some subsets in the current population. The algorithm iterates through previous steps until convergence and an optimal feature subset can be determined.

2.3 Machine Learning Algorithms for Classification

(Contents in this section is referred to *Probabilistic Machine Learning: an Introduction* by Murphy [7].)

The objective of this study is to fit classification models for AKI, which is converted to a binary outcome in the dataset. Therefore, we consider several supervised learning algorithms for binary classification, including Logistic regression, Decision tree classifier, eXtreme gradient boosting (XGBoost) and Support vector classifier (SVC).

2.3.1 Logistic regression

Logistic regression is a widely-used technique for binary classification problems, where the goal is to predict the probability of an instance belonging to one of the two classes.

Logistic regression assumes a linear relationship between features and the log odds of the outcome. The input features are linearly combined using weights and summed up to produce a score. The score is then transformed using the sigmoid function, defined as the following, to obtain a probability value between 0 and 1.

$$P(Y=1|X) = \frac{1}{1+exp(\omega^T X + b)}$$

where $P(Y=1|X)$ is the probability of the outcome being 1 given input features X, ω is the weights associated with each feature, and b is the bias term.

2.3.2 Tree-based models

- **Decision tree classifier**

  A decision tree is another popular algorithm for classification problems, which works by recursively partitioning the input space into regions, with each partition representing a decision or a split based on the values of input features.

  The decision tree classifier is characterized by its intuitive representation as a tree structure. Starting from the root, the classifier makes a decision (a node), based on a feature value to split the data into subsets that are as homogeneous as possible with respect to the outcome. During training, the optimal feature and split point at each node are searched to minimize impurity and to maximize information gain. The algorithm continues splitting the data until it reaches maximum purity (all samples in a node belong to the same class) or until it reaches the predefined stopping point.

- **Ensemble model: eXtreme gradient boosting**

  Decision tree classifiers are prone to the issue of overfitting when the tree depth is not appropriately controlled, due to its nature that it keeps splitting data until the maximum purity is reached. Ensemble models such as Random forests and Gradient boosting are often considered to mitigate overfitting, which can also improve generlization performance on classification. Here, we applied eXtreme gradient boosting (XGBoost), an efficient and scalable implementation of gradient boosting algorithm, by introducing regularization terms and approximating the gradient of the loss function with second-order Taylor expansion instead of first-order approximations.

The main inspiration behind gradient boosting algorithm is to combine multiple weak learners, typically decision trees, to create a strong learner. Staring with an initial base learner, the algorithm computes the residuals between the actual outcomes and predictions made from the current model at each iteration. A new learner is fitted to reduce the error using gradient descent method to find the optimal direction. Predictions of all base learners are then combined, and those perform better are given more weights. The algorithm repeats the process until a pre-defined maximum iteration or a certain threshold of model performance is reached.

2.3.3 Support vector classifier

Support vector classifier (SVC) is a specific type of support vector machine (SVM), which is designed for classification tasks, primarily in binary scenarios.

SVC aims to find the hyperplane that best separates the classes in the feature space. This hyperplane is chosen to maximize the margin, which is the distance between the hyperplane and the nearest data points (support vectors) from each class. In a linear SVC, the hyperplane can be written as a set of points x satisfying:

$$\omega^T x - b = 0$$

where ω is the normal vector vertical to the hyperplane and $\frac{b}{\|w\|}$ determines the offset of the hyperplane from the origin along w.

Besides, SVC can also handle non-linear cases using kernel functions, which implicitly map input features into a higher-dimensional space where classes become separable, but do not explicitly computing the transformed feature vectors. A common choice is the Gaussian radial basis function, defined as follows:

$$k(x_i, x_j) = exp\big(-\gamma\|x_i - x_j\|^2\big)$$

where x_i, x_j are different input features, and $\gamma > 0$. Sometimes $\gamma = \frac{1}{2\sigma^2}$ as default.

2.4 Model evaluation and comparison: F1-score

In this study, we considered using F1-score as the metric for classification performance evaluation and further model comparison.

After training, we can apply the fitted model to predict and compare the predictions with the actual outcomes. We start with computing the confusion matrix as follows:

		Predicted condition	
		Predicted positive	Predicted negative
Actual condition	Positive	True positive (TP)	False negative (FN)
	Negative	False positive (FP)	True negative (TN)

Table 1: Confusion matrix

We can then define F1-score as the harmonic mean of precision and recall:

$$F_1 = \frac{2}{recall^{-1} + precision^{-1}} = 2 \times \frac{precison \cdot recall}{precision + recall}$$

where *precision* $= \frac{TP}{TP+FP}$ measures the proportion of true positives among all predicted positives, and *recall* $= \frac{TP}{TP+FN}$, also known as sensitivity, measures the proportion of true positives among all actual positives.

F1-score ranges in [0, 1], where higher values suggest both high precision and recall of the model predictions, and further better classification performances. Specifically, a score of 1 indicates a perfect fit, if both precision and recall are 1, while a score of 0 indicates a bad fit, if either precision or recall is 0.

F1-score was chosen rather than area under the ROC curve (AUROC), which is another commonly-used metric for model evaluation for two reasons:

- First, the outcome, `aki`, is heavily imbalanced, and F1-score is more reliable than AUROC when handling imbalanced data, by focusing on more positive cases and minimizing false positives and false negatives.
- Second, the objective of our study is to identity whether a patient has AKI or not. We would care more about correct forecasting on positive cases as we hope not to miss patient with AKI as much as possible. Thus, the nature of F1-scores perfectly suits our request.

3 Results

After the initial feature selection based on missing values and mutual information, 76 features were selected as potential influential factors for AKI classification in further model fitting steps. Table 2 displays the 10 variables most relevant to AKI, as well as the corresponding mutual information with AKI:

Feature	MI	Feature	MI
`bun_min`	0.028636	`gcs_verbal`	0.011545
`bun_max`	0.023402	`admission_age`	0.011167
`weight_admit`	0.019158	`pt_min`	0.010761
`pt_max`	0.014665	`gcs_motor`	0.009525
`sbp_min`	0.013868	`gcs_eyes`	0.009267

Table 2: 10 features most relevant to AKI and corresponding mutual information

A training-testing split was then conducted with a training-testing ratio 7:3. Standardization was applied to ensure the contribution of each factor on AKI forecasting in a common scale and to avoid features in large magnitudes from dominating the learning algorithm. Moreover, some algorithms require scaled features for faster convergence and more effective regularization process.

Aforementioned machine learning algorithms were then applied to the training set. Note that the numbers of observations in two classes of the outcome `aki` were severely imbalanced, with a 0-1 ratio approximate to 1:5. Imbalanced outcomes can lead to biased model performance, where the model tends to predict the majority class more frequently, ignoring the minority class. This bias occurs because the model minimizes the overall error rate, which can be achieved by simply predicting the majority class for most instances, especially when the majority class dominates the dataset. To minimize the influence of imbalanced outcomes, we adjusted class weights inversely proportional to class frequencies in the outcome during model fitting. The adjusted weight in each class is defined as follows:

$$adjusted\ weight\ of\ class\ i = \frac{\#\ samples\ in\ all\ classes}{\#\ classes \times \#\ samples\ in\ class\ i}$$

During training, Genetic algorithm with default settings was used for the secondary feature selection, based on the specified estimator. Grid search was later conducted to search for the best key hyperparameters in the algorithm, such as the learning rate, in terms of F1-score. A 3-fold cross validation (CV) was applied together with grid search, in order to avoid the issue of overfitting, where the chosen hyperparameters may be tuned to perform perfectly on the specific training set, while have a poor performance on the testing set and poor generalization to new data.

After identifying model hyperparameters with the best performance, here the highest F1-score, based on grid search with cross validation, we applied the tuned model on the testing data and evaluated the model performance using the same metrics. Furthermore, we compared classification performances across different machine learning algorithms with optimal feature subsets and hyperparameters.

Model evaluation and comparison

With optimal feature subset determined by Genetic algorithm and tuned hyperparameters using gird search and cross validation, we tested the classification performance of models on the testing data. Table 3 displays the F1-scores of each model with optimal parameters on both training and testing sets.

	Logistic regression	Decision tree	XGBoost	SVC
Training	0.7562	0.7544	0.8716	0.8329
Testing	0.7675	0.7997	0.8713	0.8308

Table 3: F1-scores of models with optimal hyperparamters on training and testing data

For each model, F1-scores in both training and testing sets did not show significant difference, indicating that there did not exist the issue of overfitting. Furthermore, F1-scores among all model candidates were all over 0.75, which suggests good selection of feature subsets, good tuning on key hyperparamters, and good classification performance in all models.

Model performance in AKI classification in Logistic regression tended to be the worst among 4 candidates, with an F1-score 0.7675 in the testing data. Decision tree, whose F1-score in testing data was 0.7997, performed slightly better than Logistic regression. SVC out-performed the previous 2 models, but showed worse classification performance than XGBoost, with an F1-score 0.8308. As the best-performed algorithms among all candidates, XGBoost, with an F1-score 0.8713, showed a significant improved power for AKI classification, given its significant increase in F1-score by at least 0.04 comparing to SVC, and by at most over 0.1 comparing to Logistic regression.

Thus, XGBoost classifier was considered to the best model for AKI classification. Further, we listed 10 features in the optimal feature subset for XGBoost classifier based on Genetic algorithm, which were most important to AKI classification, shown in Table 4:

Feature	Importance	Feature	Importance
`bun_min`	0.105153	`sbp_min`	0.027391
`bun_max`	0.085666	`pt_max`	0.025182
`gcs_verbal`	0.034930	`gcs_eyes`	0.023119
`weight_admit`	0.030211	`dbp_min`	0.022858
`ptt_max`	0.030112	`gcs_motor`	0.022816

Table 4: 10 features most important to AKI in the optimal XGBoost classifier and corresponding feature importance

Comparing to Table 2 where features were ranked based on its mutual information with AKI, although `pt_min` and `admission_age` became less relevant than `dbp_min` and `ptt_max`, the rest of features remained most important to AKI classification after applying Genetic algorithm for feature selection during model fitting. The high consistence in two feature selection steps suggested that `bun_min`, `bun_max`, `gcs_verbal`, `weight_admit`, `sbp_min`, `pt_max`, `gcs_eyes` and `gcs_motor` were crucial for AKI detection.

4 Discussion

Results based on F1-scores and important features tended to show the superiority of XGBoost classifier on AKI forecasting, while a bunch of concerns needs further discussing before we reach a final conclusion.

4.1 Alternative feature selection strategies

The current strategy for initial feature selection before model fitting involves dropping variables with over 50% missing values in this column. An alternative strategy could be setting a lower tolerance for missing values, i.e. a stricter selection in terms of missing values. Doing this will drop more features, but keep more complete observations in the dataset. Hence, a 'trade-off' in the loss of statistical information between rows and columns has to be considered. Current strategy aims to include more information among different features in trade of more information loss across observations, while the alternative strategy focuses more on statistical information contained among observations and sacrifices information in columns.

To compare two different feature selection strategies, we shrank the tolerance for missing values to 10% and obtained another feature matrix with 61 columns and 38361 observations before splitting. We repeated the same modelling process, and compared F1-scores of optimal models on testing data, shown in Table 5.

	Logistic regression	Decision tree	XGBoost	SVC
NA prop. $\leq$ 10%	0.7365	0.7571	0.8020	0.7556
NA prop. $\leq$ 50%	0.7675	0.7997	0.8713	0.8308

Table 5: F1-scores of models with optimal hyperparameters on testing data, based on two feature selection strategies

According to comparison in classification performance, models based on the current feature selection strategy generally showed better forecasting power, particularly for XGBoost and SVC where significant differences were observed. Hence we kept using the current strategy with a selection criteria that columns including more than 50% missing values were dropped.

4.2 Considerations in model comparison

4.2.1 Linear assumption

Classifiers such as Logistic regressions and SVC with a linear kernel assume a linear relationship between features and the (log odds of) outcome, while other models used in this study assume a more complex, non-linear association. Comparison based on F1-scores on testing data apparently showed that Logistic regression performed the worst. An additional comparison between SVC using linear kernel and Gaussian radius basis function (rbf) was then conducted and the results also suggested a worse AKI classification performance of SVC with linear kernel. Both comparisons indicated that the linear assumption did not perfectly held, and the relationship between AKI and risk factors was more complex than expected.

4.2.2 Ensemble models

As mentioned before, ensemble models are always considered to be more superior than single Decision tree classifier, as they can not only mitigate overfitting issues, but also improve classification performance, due to the superiority of boosting where each tree corrects the errors made by the previous ones. Although we had shown that overfitting issue did not occurred, a higher F1-score of XGBoost on testing data still implied its greater classification power, comparing to the single Decision tree.

4.2.3 Computation issues

Comparing to Logistic regression and single Decision Tree, algorithms of XGBoost and SVC are more complex, particularly for SVC with rbf kernels which projects input features to a higher-dimensional space using Gaussian radial basis function. Therefore, computation issue arises when a dataset with large sample size is used.

Moreover, grid search with cross validation was applied to search for optimal hyperparameters in this study, which further increases the computation requirement of the model fitting. Consider a grid with m hyperparameters waiting for tuning and each hyperparameter has n candidates. Then a grid search with k-fold cross validation will require $m \times n \times k$ times computation of a single fit with one determined hyperparameter.

In this study, although F1-scores of XGBoost and SVC with rbf kernel tended to be similar, with SVC slightly performing worse than XGBoost, computation requirement and running time of SVC for hyperparameters tuning is multiple times those of XGBoost. Note that the hyperparameter grid of XGBoost was more complicated than that of SVC. Thus, XGBoost was considered to be more efficient.

On the other hand, it is hard to measure the importance of features for SVC using rbf kernel, as the kernel trick does not explicitly compute the transformed feature vectors. Also, determination of the support vector could not be easy and intuitive in high dimensions.

Therefore, the XGBoost classifier with a learning rate of 0.25, maximum depth for base learners of 6, and number of boosting rounds of 1500, was considered to be the best model for AKI classification, based on previous considerations.

4.3 Limitations

One of the main limitations of this study was the lack of clinical interpretation. Although forecasting models we constructed showed good classification performance, important features determined based on statistical methods may not be clinical meaningful for AKI. On the contray, features with strong clinical significance may be dropped due to poor statistical significance. Future work could be more detailed feature selection based on more background knowledge and clinical interpretations of features.

Another main limitation of this study is actually related to the previous one. Without sufficient background knowledge and detailed explanation of each column, we failed to detect the missing mechanisms among features with missing values. As a result, we did not apply any imputation methods to columns with missing values in case improper imputations lead to more severe bias in models. However, current strategy that observations with missing values were all dropped can also introduce bias. Therefore, detailed background research may help address this issue.

References

[1] Charles Hobson et al. "Cost and mortality associated with postoperative acute kidney injury". In: *Annals of surgery* 261.6 (2015), p. 1207.

[2] Azra Bihorac et al. "Incidence, clinical predictors, genomics, and outcome of acute kidney injury among trauma patients". In: *Annals of surgery* 252.1 (2010), pp. 158–165.

[3] Azra Bihorac et al. "National surgical quality improvement program underestimates the risk associated with mild and moderate postoperative acute kidney injury". In: *Critical care medicine* 41.11 (2013), pp. 2570–2583.

[4] Emma Borthwick and Andrew Ferguson. "Perioperative acute kidney injury: risk factors, recognition, management, and outcomes". In: *Bmj* 341 (2010).

[5] J Stewart et al. *Adding Insult to Injury. A review of patients who died in Hospital with a Primary Diagnosis of Acute Kidney Injury.* 2009.

[6] Alistair Johnson et al. *MIMIC-IV*. 2023. DOI: 10.13026/6mm1-ek67. URL: https://https://physionet.org/content/mimiciv/2.2/.

[7] Kevin P Murphy. *Probabilistic machine learning: an introduction.* MIT press, 2022.

Appendix

Programming in this study is completed with Python.

Please refer to this Github repository for complete code and other materials.