STAT3612 Proposal

1 Group member

CHAN Yuk Wai Xtra; DING Zhui; YUNG Yiu Yin; WANG Xizhuo; YU Xinyao

2 Data exploration

We are going to design machine learning models for the Electronic Health Records (EHRs), which mainly includes 2 parts.

1. Task 1: Mortality Prediction - use clinically grouped time-varying labs and vitals features to estimate the patients' mortality status which is binary result.

   We explored predictors used for training. **hours_in** denotes the duration of the period window (unit:per hour). However, most are marked as **mask-0**, indicating missing measurement. Also, there should be correlation between **mask** and **time_since_measurement** because each time the data is obtained, the time since the last measurement is recalculated from 0.

   In terms of the object variable, we found that it highly unbalanced with 92.69% of $y_i's$ being 0. It can lead to poor classification performance as we may ignore or misclassify the minority samples, which rarely occurs but matters. We may use some techniques such as over-sampling in the minority group to reuse the data to deal with this issue. AdaBoost will also be explored to improve the prediction accuracy of the minority.

2. Task2: Length-of-stay (LOS) Prediction - use clinically grouped time-varying labs and vitals features to forecast the patients' length of staying in ICU by regression to enhance better treatments to the patients and help hospital operations.

   We use the same explanatory variables as task 1, but may use different models for prediction. Figure 1 shows the distribution of **LOS-ICU**. It is also unbalanced with the skewed distribution. Most of the patients spent less than 5 hours in ICU. We may use tree-based models that are more suitable for unbalanced datasets.

3 Feature Selection

1. Step 1: Only "mean" of the features are considered while the mask and time since measurement will not.

2. Step 2: For each predictors at different timestamp of the period window, depending on the properties (biological meaning) of the predictors, we will take use an aggregate function (e.g. mean/ max/ min) to merge them into a single value. For example, for "blood pressure", we will pick the maximum value among the 24 hours.

3. Step 3: While there 104 predictors, we will eliminate those containing a lot of missing values (e.g. If we eliminate the features with more than 90 percent of value are missing, 30 features will be eliminated). For predictors with lower smaller amount of missing value, we will impute the missing value by the mean.
4. Step 4: We will plot the remaining features with response and calculate correlation coefficient, features that are weakly correlated with response will be eliminated. We will also compute the pairwise correlation among predictors, if two variables are highly correlated, one of them will be eliminated.
5. Step 5: Forward selection and LASSO regression may be implemented to further remove insignificant predictors.

4 Proposed model

1. For task 1, we will use classification algorithms such as Logistic Regression, Discriminate Analysis, Naive Bayesian, Classification Tree, K Nearest Neighbours, Support Vector Machine, XGBoost and Adaboost. For task 2, random forest, regression tree and generalised additive models will be adopted. We will also apply deep learning techniques such as convolutional neural network in both tasks.
2. To further enhance the performance of our model, we will implement adjustments to each model's parameters. We will adopt parameter tuning measures including random search, grid search, Bayes optimization and gradient-based optimization. We will also improve the generalization capacity of the model with L1/L2 regularization, neural network layer drop out, decision tree pruning and ensemble methods of bagging and bootstrap aggregation.

5 Performance analysis

1. In Task 1 Mortality Prediction, we would prefer AUC as the data is imbalanced and AUC utilizes the probability of prediction. Area Under the Curve (AUC) calculates the area under the Receiver Operating Characteristics (ROC) curve. If the model has good classification accuracy, it should have both sensitivity and specificity close to 1, indicating the ROC curve is close to the upper left corner of the graph with a high AUC. Also, AUC must be greater than 0.5 (the AUC for the random classification reference line).
   In our model, we would use the **roc_auc_score** from sklearn.metrics to calculate the AUC for train and validation sets, with AUC greater than 0.8 regarded as acceptable.
2. In Task 2, we would use Root Mean Square Error (RMSE), with formula below, to assess the prediction accuracy of out model. We would utilize the sklearn.metrics MSE with squared variable be False to calculate the RMSE. A smaller RMSE would indicate better performance.

$$RMSE = \sqrt{\frac{\sum_{t=1}^{T}(\hat{y}_t - y_t)^2}{T}}$$

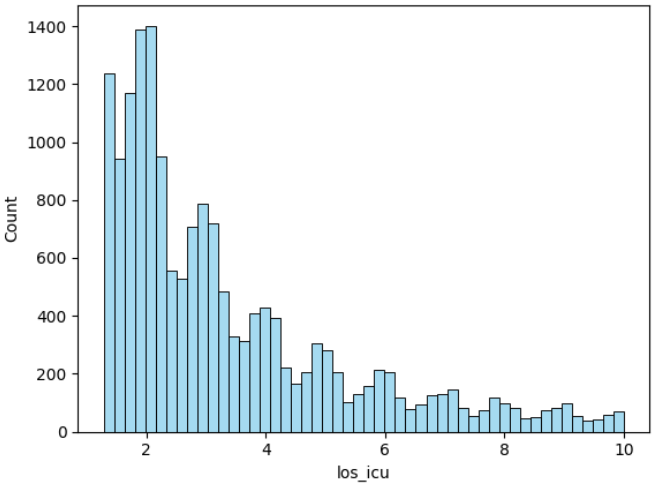

Figure 1: Distribution of LOS_ICU

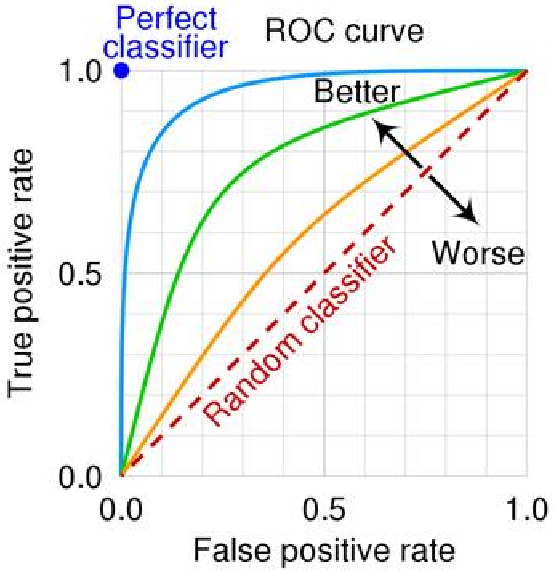

Figure 2: ROC Curve (Source from Wikipedia)