Data Challenge Summary

Section 1. Data preprocessing and exploratory data analysis

I. Feature engineering

For each variable in each txt file, we **1)** extracted the last record **2)** extracted the min and max records **3)** Calculated the mean of records **4)** Engineered changes and change rate for the first and last record, with regard to the time feature

II. handling missing values

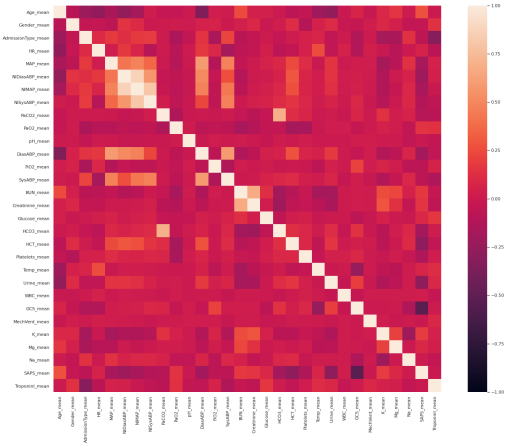
First, we dropped variables that had missingness greater than 50 percent, except the variable “respiration”.

Then, we tried three different strategies: mean, median and nearest neighbor. We have found out that mean is the most effective way, since it has the best accuracy after fitting a random forest model.

III. EDA

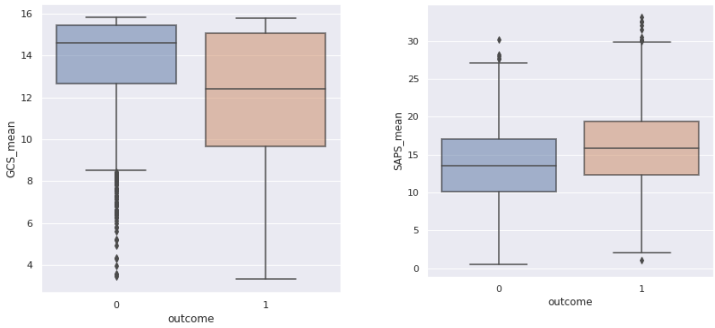
We do exploration on the data imputed missing value with mean.

1) Correlation, heatmap
   The whole correlation between variables:



From the heatmap, we can observe that NIMAP and NIDiasABP (corr=0.8713), NISysABP and NIMAP (corr=0.8068), BUN and Creatinine (corr=0.6674), SAPS and GCS (-0.5284) are the most correlated variables.

2) Boxplot: GCS and SAPS have the most separated boxplot among all variables.



Section 2. Modeling

I.Different models and performance

The data used for model comparison is the variables calculated from mean and last records, since we assumed that mean and last record have the most effect on the prediction result. We used the rest as add-on variables to improve the selected model.

Similar to cross validation to reduce variance on the training set, we used averaged scores of ten fits on random train-validation split (ratio 7:3)[1]. Then, we perform hyper-parameter tuning for each model. The result is as follows. Consequently, we chose the **Extreme Gradient Boosting classifier** as our model.[2]

1) Random Forest Classifier with 100 trees

a) Averaged AUC: 0.760
b) Averaged BER: 0.306

2) Gradient Boosting Classifier with tuned parameters (50 trees)

a) Averaged AUC: 0.766 (original) → 0.77390 (with additional variable)

b) Averaged BER: 0.309 (original) → 0.30384 (with additional variables)

3) Extreme Gradient Boost Classifier with tuned parameters (1200 trees)[3]

a) Averaged AUC: 0.786 (with additional variables)
b) Averaged BER: 0.297 (with additional variables)

4) Logistic Regression (the number of variables are shrunk to the best number by LR with L1 penalty)

a) Averaged AUC: 0.748
b) Averaged BER: 0.334

5) KNN with K = 20 (Discarded due to score lower than baseline) AUC score: 0.6202

6) MLP (Discarded due to score lower than the baseline) AUC score: 0.5022

II. Feature selection:

In order to feed our model more data to increase performance, we used 2 (min, max) and 4 (change, change_rate) from the above engineered features. We developed **a greedy forward selection algorithm** to select variables.

First, we created candidate variables by the top features from the feature importance of the Gradient Boost classifier. Note the reason we did not use all variables is that too many variables in forward selection will take huge computational resources. Then, we added the variable from the candidates one by one to see if there's a drop in BER and an up in AUC. To measure performance accurately, we used averaged BER and AUC same as above. Consequently, we have improved performance of GB after adding *maximum of **HR** and **SysABP**, change rate of **PaO2***.

[1] We were not sure how to use the cross validation package with AUC and BER scoring.

[2] Initially we used GB and selected features for it, and later we found that XGboost performs better on the same selected data. Therefore, we held the features and then tuned the model.

[3] The submitted performance on the test set (around 3000 samples) was **AUC: 0.791** and **BER: 0.321**.