Machine Learning Health Care Project

1. **Introduction**

In this project we were requested to analyze, using ML methods, medical records of thousands of people that were hospitalized in ICUs in the US. Using these methods, we built 2 classification models that can predict whether:

* 1. A given patient will be infected with Bacteremia (Model A)
  2. A given patient who has Bacteremia is treated with appropriate ABX (Model B).

This paper describes the data we had, the methods and models we used the results of each model.

1. **Cohort description**

We were given 2 datasets, MIMIC and eiCU, each taken from a different hospital. The dataset include all the medical data the hospital had on each patient. The data we used for this assignment is composed mainly from:

* 1. Basic blood count and blood gases
  2. Invasive procedures, I.E catheter, arterial lines, etc.
  3. Input (textual) of the nurse who treated the patient.
  4. Drug types and dosage.

For model A, we used clinical data that was collected before the blood culture collection time. For model B, we used the same data as in model A, in addition to extra 24 hours of data collected after the blood culture was taken.

For model A, we had:

* 1. 3,104 patients from the MIMIC dataset, 2939 of them are labeled negatives, and 165 of them are labeled positives
  2. 128 patients from the eiCU dataset, 118 of them are labeled negatives and 10 labeled positives.

For model B, we had only data from the MIMIC dataset. It contained:

1. 100 Patients who were treated with an appropriate ABX.
2. 32 Patients who were treated with an inappropriate ABX.
3. **Methods**
   1. **Data preprocessing**

At first, we built a list containing all the information we needed on each patient. During the creation process of this list, we removed anomalies based on the possible range of each element in the blood count and vital signs.

After we finished creating the list, we removed patients from the cohort that were missing more then ?% of the features and limited the maximum amount of patients we removed to 4%. We also removed features which appears in less than a ?% of the patients.

Now that we removed all the “noisy” data, we wanted to convert each patient to a corresponding vector, where each entry of the vector represented a feature. The vector is composed of the following features:

1. Various representators of each component in the blood count, including Average of all samples and average of the last 5 samples, Maximum and Minimum values, Value of the most recent sample, Total amount of samples, Standard deviation, The difference between the highest and lowest values, 25th and 75th quantiles and the maximum difference between any 2 adjacent (date wise) values.
2. Binary vector corresponding to whether the patient had an invasive procedure during its staying in the ICU. We used the built-in categories column of the database as the indicator of the procedure.
3. Textual input of the nurse who took care of the patient and searched for keywords such as Fever, Chills, Nausea, Vomit, Diarrhea, Fatigue and Weakness. We then mapped each combination of those keywords into a binary vector using one-hot encoding.
4. Term-frequency measurement of the drug types and invasive procedure types based on the whole dataset.

Finally, we used k-NN algorithm with K=? to impute the missing values on the training data and test data separately and normalized each of the groups by using z-score function.

* 1. **Models**

To choose the hyperparameters to our model, we used a python library named Hyperopt, which receives a score function and minimalize its output. For every hyperparameter, we defined it’s search space and limited the amount of optimization iterations. The score function for the model was the AUROC values we got per each trained optimized model, multiplied by -1 since we wanted the maximum value.

We also used Hyperopt to get an optimized instance of each classifier we used during the training process.

We wanted to choose the most significant features out of our base set of features to train our model. To find these significant features, we used an optimized instance of XGB classifier and selected the top K=? relevant features.

Since the data wasn’t balanced, we also had to choose a balancing method so we can avoid overfitting. We tested a several different methods, and ended up using ? for model A and ? for model B.

After we selected the features and balanced our classes, we used Hyperopt to choose the optimal classifiers for our model. It received a non-optimized classifier and returned an optimized instance of the given classifier.

During the process of choosing the specific model, we tried various classifiers, and we ended up choosing an ensemble of 2 classifiers: Random Forest Classifier and k-NN classifier, combined with weights for each model. Final classification was done using soft voting.

* 1. **Evaluation**

In order to evaluate our model, we used 5-fold cross validation. For each fold, we trained our model on the other 4 folds, and calculated the AUROC and AUPR on the test fold. We repeated this process for each fold separately and calculated the average result and the standard deviation.

1. **Results:**

**Model A (5-fold cross validation):**

Average AUROC: 0.664

STD AUROC: 0.029

Average AUPR: 0.102

**![Chart

Description automatically generated]()**STD AUROC: 0.017

**![Chart, histogram

Description automatically generated]()**

**Model B (5 fold cross validation):**

Average AUROC: 0.6594

STD AUROC: 0.0698

Average AUPR: 0.372

STD AUROC: 0.113****



**Model A (eiCU data as validation set):**

**![Chart, line chart, box and whisker chart

Description automatically generated]()**

**![Chart, box and whisker chart

Description automatically generated]()**