Anthony Paolillo
STATS 503 Data Challenge

The Data Challenge for this course gave us the opportunity to apply the skills we have learned this semester to predict if a patient will develop sepsis. We were given a set of 21,634 patients, and for each patient, 38 measurements were recorded. Our goal was to predict the labels of the 6,490 patients in the test set.

First, it was essential to process the data in a way that I could have a final dataframe with one row for each patient. To simplify this, I worked with one patient file first. I was able to group together common measurements, and summarize these by taking the median value of each measurement that was present. If there was no measurement, that variable would be assigned “NA”. Through various literary works, it was clear that time in ICU was essential to account for. Therefore, I added hours as one of my predictor variables. Additionally, for each variable, two more columns were added: A minimum and maximum measurement value.

After researching various works, it became clear that I had to better account for some of the longitudinal variables in my dataset. Instead of using rolling windows and moving averages, I decided to use a simpler approach. For each variable that a patient had a measurement for, I calculated the difference between the last reading and first reading. I then divided this difference by the amount of readings of that variable. Therefore, each patient had 4 columns for each variable: The aforementioned average difference, the median, the minimum, and the maximum measurement. A for-loop created this for each patient and then combined them into one dataframe corresponding to a row per patient.

Once I had the dataframe of all the patients with my summary statistics, I had to clean it up a bit. This involved removing unnecessary columns (ex. maximum of demographic variables). I was then faced with my next decision: How to deal with the missing values in my data for each patient.

Exploratory data analysis allowed me to see that there were a few variables that had a significantly low amount of measurements. After consulting literary works (and even Junting), I settled on a 20% boundary. This resulted in me dropping the following variables from my dataset: *TroponinI, Fibrinogen, EtCO2, and Bilirubin_direct.* 4 different quantities of each variable meant 16 columns were dropped leaving me with 130 columns. To input the missing values, I first tried MICE imputation, but with a large number of missing measurements across all variables, this was a poor choice. Rather than use forward fill, I decided to use two methods, and compare their accuracy when testing my model: mean and median imputation. Once these missing values were imputed, I was able to finally move on to my model.

With all the differences in variances across my measures, and with the sheer amount of predictors variables that I had, it seemed best that I would use some sort of classification tree or boosting method, or potentially, a deeper learning algorithm. Once again, my references had me focus on: XGBoost, DecisionTree and RandomForest.

Circling back to earlier, in order to see if mean or median imputation performed better, I fit each of my models to the two different dataframes. After splitting my data, I performed 10-Fold Cross Validation in order to see my model's performance. Since our project is being graded on the average of BER and AUC of the Receiving Operating Characteristic curve, I used 'roc_auc' and 'balanced_accuracy' as my scoring criterion. I compared the average of the roc_auc and balanced_accuracy for each of the 10 folds for each model to each other.

After evaluating my different models with different datasets, I was able to conclude that my XGBoost Classifier gave me the best score. The results for each of the two datasets are as follows:

- Median Imputation:
    - AUC: .919
    - BER: .2083
- Mean Imputation: Area under Receiving Operating Characteristics Curve: .921
    - AUC: .921
    - BER: .2056

Due to the fact that our goal is to maximize the AUC and minimize the BER, I have concluded that using Mean Imputation to fill the data, and fitting a XGBoost classifier will perform best on the testing set. Therefore, I fit my model using the mean imputation data and made predictions on my test data. The resulting predictions on tse test set were:

- Number of Patients that have sepsis: 860
- Number of Patients that do not have sepsis: 5,630

The labels are located in the test.csv file under the column "Model_Outcome", and the score is located in the "Model_Score" column for each of the patients.

Overall, this data challenge provided me with the opportunity to apply some of the skills that I have learned into the course into a relevant topic. With the success of my model on validation sets using 10-Fold CV, I feel confident that my model will perform well on the true test. I look forward to determining how my predictions compare the true labels and to learning more about different approaches to use for problems like these.