STAT3612 Group Proposal

PU Rui Ling 3035917989
CHAN Hiu Yu 3035784162
Lyu Zhiheng 3035772432
Wong Wai Chak 3035784186
Jiang Xiao 3035851894

Background:

- Objective: we will make use of EHR data and possibly image data to measure the 30-day readmission rate of each patient.
- Dataset available:
    - EHR tabular dataset: each patient id corresponds to 171 features, where each of these features is a time series with length equal to admission duration.
    - Image dataset: each patient id corresponds to a feature array (1024 values), which is the image features extracted from its original chest x-ray image.

Feature Selection:

For the 171 features, we want to do a screening for the features that really matter to our objective. Thus we proposed varies feature selection techniques outlined as following:

- Correlation analysis:
    - calculate the correlation coefficient between features and remove features with high multicollinearity (eg. when the coefficient > 0.7).
    - select features that have a significant correlation with the target variable as candidate features (eg. only retain the top n features).
- Forward/backward selection: this can determine an optimal feature subset, which is better than manual selection.
- Tree ensembling: As proposed in *stage one,* tree ensembling methods can handle the feature selection process automatically, by changing the tree depth as a hyperparameter.

Note that the final selection will be based on performance of models.

Modeling:

We split our modeling methods into 3 stages. After performing stage one, we will consider further performing stage 2/3 if the performance is unsatisfactory. We may stop trying if there are limitations in time or computational resources.

Stage One: Use EHR Tabular data to train **basic machine learning model**

At this stage, we seek to reduce the time series feature record to one single feature record. We assume that the **latest feature record** of any patient is the most representative of that patients' health situation, and thus most related to the readmission rate.

We start from the simplest machine learning algorithm (LR, LDA) before more complicated ones (Decision Trees).

- **Logistic Regression (LR)** would be a natural choice for a probability prediction, and can handle both numeric and categorical data. It is also simple to interpret. L1/L2 regularization will possibly be used to prevent overfitting.
- **Linear Discriminant Analysis (LDA).** After doing the feature selection in the data pre-process section, we assume that there will be less correlations between the input features. Assumptions of independence and normality might be suitable for aggregated clinical features.
- **Tree ensembling**. We will try the current two most popular tree ensembling methods, random forest and boosting. The advantage of using decision trees is that further feature selection is performed automatically by controlling the depth of each tree. Gradient Boosting, XGB

Stage Two: Use Temporal model and **make use of the time series nature** of EHR data
Compared to our last stage, which only makes use of the final day's features of each patient, we now consider **using time-series data to better understand how a patient's condition evolves**. We're going to use an **LSTM (Long Short-Term Memory)** network for this, which is well-suited for data that changes over time and has different lengths. The input of LSTM will be the 171 features from each day.

Because our dataset is small, we will **use three supervised signals to train our model**: 1. Predicting the condition change score for each patient; 2. Predicting a score for the patient's condition based on our previous model's output; 3. Predicting the actual final outcome. We still need to experiment to find out the relative importance of these three signals.

Here's a bit more detail on each method:

- **Change prediction**: We will predict the next day's patient condition, focusing on how it changes from the current day. This method is similar to how language models predict the next word in a sentence.
- **Score prediction**: We'll use the **scores from our first-stage ensemble model** to teach our LSTM more about patient conditions. It's like giving the LSTM a hint using what we already know.
- **Final outcome prediction:** We'll try to predict the **actual final result** for the patient. We're still deciding whether to do this at every step or just at the end. To help the model understand how close we are to the final day, we'll use a technique called **positional embedding**.

We're planning to use a bigger model with strong dropout (to prevent overfitting) and L2 normalization (to keep the model weights in check), assuming we have the resources for it. This should help our model work well on new data it hasn't seen before.

Stage Three (optional): Use **additional Image dataset** to assist prediction
If in stage two, the temporal model shows better results, then use stage two model as a base model, and modify it to handle image data **along with** tabula data.

- Handling of multimodal input:
    - Simply try extending the input dimension, eg. concatenating the image features to original features, so that each id will correspond to 172 features in each day, while the last one is an array that contains image features extracted from image data.
    - Combine the image features only as the input to selected layers of the base model, eg. attention layers.
- Better image features extraction: Other than MOCO-CXR, explore other deep learning algorithm to extract features from original image dataset (eg. bilinear pooling, autoencoder).

Evaluation:

We will use the AUROC metrics (area under the ROC) to measure the performance of each model, as required in the instruction file.