To goal of this project is to do a classification for medical tests and occurrence of sepsis and to do a regression for key vital signs including LABEL\_RRate, LABEL\_ABPm, LABEL\_SpO2, LABEL\_Heartrate.

Firstly, we preprocess the patient data, dividing it into 37 patients with 12 hours in the hospital and shuffle the data. Then we divide the dataset into train and validation set with a ratio of 0.8 to 0.2. After that we impute the data for each patient with median of the 12 hours’ states. If there is no data, we just impute 0 so they will not contribute to further classification or regression. The we standardize the data.

For subtask1,2-classification, we use the ANN model, we first construct a flatten input layer and add 1 regular densely-connected NN layer of 128 units, 1 densely-connected NN layer of 32 units and 1 densely-connected NN layer of 8 units, all using activation function with ReLU and initializer for the kernel weights matrix with HE uniform initializer. Between each layer, use BatchNormalization and dropout some data with ratio of 0.4. Finally add a output layer use a sigmoid function to get interval [0,1]. We compile the model with binary\_crossentropy loss. We also account for class imbalance(0/1) in our training data target using a class weight computed from different number of data of each class and scale them with total/2 helps keep the loss to a similar magnitude. To save time, we also use a simple early stopping and stop training when a monitored metric has stopped improving after 3 epochs(patience=3) and ‘min’ mode where training will stop when the quantity monitored has stopped decreasing. After setting up the model, we fit the model with 100 epochs and batch size of 64 individually for the different targets: medical tests and occurrence of sepsis(signed with exp in our ipynb file). Then we use the generated model to predict the target classification and assess the ROC AUC score to evaluate our models. Finally we use our models to predict the test data to get prediction.

For subtask 3, we will train the data with similar ANN but this time we use only 2 dense layer of 256 units and the second with a kernel\_regularizer=l1(0.001), and output layer with linear activation function to get regression predictions. We compile the model with loss function of mean absolute value. We also use a early stopping with patience of 5 and fit the data for 100 epochs and batch size of 64. Then we calculate its R2 score from prediction of validation set to evaluate its performance. Finally we use the model to predict the test set and get results.

Firstly, we preprocess the patient data, dividing it into 37 patients with 12 hours in the hospital and shuffle. Then we divide the dataset into train and validation set. After that we impute the data for each patient with median of the 12 hours’ states. If there is no data, we just impute 0 so they will not contribute. We standardize the data. For subtask1,2-classification, we use the ANN model, we first construct a flatten input layer and add 1 dense NN layer of 128 units, 1 dense NN layer of 32 units and 1 densely-connected NN layer of 8 units, all using activation function ReLU and HE uniform initializer. Between each layer, use BatchNormalization and dropout some data with ratio of 0.4. Finally add a output layer use a sigmoid function to get interval [0,1]. We compile the model with binary\_crossentropy loss. We also account for class imbalance(0/1) in our training data target using a class weight computed from different number of data of each class and scale them with total/2 helps keep the loss to a similar magnitude. To save time, we also use a simple early stopping and stop training when a monitored metric has stopped improving after 3 epochs(patience=3) and ‘min’ mode where training will stop when the quantity monitored has stopped decreasing. After setting up the model, we fit the model with 100 epochs and batch size of 64 individually for the different targets: medical tests and occurrence of sepsis(signed with exp in our ipynb file). Then we use the generated model to predict the target classification and assess the ROC AUC score to evaluate our models. Finally we use our models to predict the the test data. For subtask 3, we will train the data with similar ANN but this time we use only 2 dense layer of 256 units and the second with a kernel\_regularizer=l1(0.001), and output layer with linear activation function, compile the model with loss function of mean absolute value. Finally we use the model to predict the test set.