This project mainly consists of data manipulation and model selection and training. All the testing codes are included in the latter part of the submitted code, indicated by the comment cell '''Production Code during testing'''. Run the cells until this cell can reproduce the prediction.

I averaged the twelve tests for one patient and used KNN for data imputation. This algorithm tries to impute the missing data based on k other samples with a value for the feature and the nearest Euclidean distance. The lost data is computed by averaging the neighbor's existing data. Our final solution also has built-in support for missing data.

The second part is to find appropriate models for the classification and regression tasks. I first tried the models introduced in the lectures, including the Logistic Regression, SGD method, SVM with RBF or polynomial kernels, and KNN. For the Logistic Regression, the model is too simple to capture the data's features. The SGD Classifier has the same issue. The runtime was very high for the SVM with kernels. It needs about ten minutes to train one model. Since I used 5-fold cross-validation to choose the hyperparameters, the runtime was very long. The performance was not that good either. The KNN classifier is fast, but the performance is not good.

I then turned to the ensemble method suitable for both classification and regression. I tried the Gradient Boosting method, a decision tree, and a regression tree type method. Even with the default parameters, this algorithm runs fast and outperforms the methods above. I used x-validation to choose k=1000 for the KNN imputation step.

The histogram-based methods provided by Sklearn also have built-in support for the missing values. I tried without imputation, and the performance during x-validation was even better. I decided to use the Histogram Gradient Boosting Classifier and Regressor for the non-imputed averaged data. The prediction result is submitted.