All the relevant code, and datasets can be found in the .zip file submitted with the proposal (Project.zip)

Recap: We decided to use this dataset from Kaggle to use non-medical data to try and estimate patient wait times at Hospitals. This is a part of a larger initiative, as it can be used to help overworked nurses in a larger scale application, as they already have a lot of unnecessary tasks, taking up extra time, and overworking them.

Current Standing: As of now, the group has completed the research phase of this project. Not only have we looked around for other datasets since submitting this one, but we also have done a deep-dive into this current dataset to extract some information, here are our findings:

- Our Dataset consists of 4 CSV files, a sample, a training set, a testing set, and a dictionary
    - The Testing Set is not actually relevant to us; we determined our target value to be the "Stay" column, and the test set provided does NOT include this. The reason for the testing set is for us to run a model on and submit outputs to Kaggle for "Grading". This set won't be useful to us until later.
    - The Training Set will primarily be used for our training, as well as some validation as well, so we can work on building a good model before final training. It contains over 318,000 entries for us to use. It also comes with a Dictionary, which gives us detailed information about each column, all displayed in the .ipynb file
- The data is imbalanced to an extent, most features do not have an even split of variability, as seen in the graphs in our .ipynb file
- Our features also do not follow any linear trends; our Correlation Matrix shows that only one feature has a 53% correlation with Stay, the second highest being only 18%. Linear Regression may not be a good option.
- We also tested importance scores for Random Forest, which has a relatively workable split, where the highest importance score is 14%

Next Steps: Overall, it is clear our data is not completely ready for training yet, although we did train a temporary RF model, just to see how importance would be split, it is clear there are many features that are irrelevant. Some features have an incredibly low correlation, and importance to different types of training, and some features pose a threat for overfitting, or are just completely going to throw the model off, such as ID numbers. Regardless, it is clear that we need to go through feature selection very carefully, while also using Non-Linear models. Whether it be a non-linear regression model, a Random

Forest, or another model that helps against multicollinearity. Regardless, the main things for us to focus on are removing unhelpful features and then training a model (likely with Random Forest training) after that.

Action Plan:

- Use selective reasoning to decide which features to use. We ideally want to:
    - Minimize Unrelated Data (low correlation data)
    - Minimize Non-Important Data (in a Random Forest Context0
    - Remove Data that will throw off the model (Anything that can cause overfitting, underfitting, or is completely unrelated to the target value)
        - Mainly we have to remove any ID features
    - Only have features that will be beneficial to training
    - **We will prioritize removing low importance features over low correlation ones, since we are going with Random Forest**
- After having a good set of features to train with, we start training our model with Logistic Regression as a ***BASELINE*** model to give us a general idea, as well as something to compare other potential models with
- We then can train our Random Forest Model, and compare the results to the original Logistic Regression model, and evaluate its performance on validation data.
- If necessary, we can train other kinds of models as well, such as KNN, XGBoost, or LightGBM.
- After training any relevant models, we can choose the best one after evaluating all of them. We choose a final model and then begin the fine-tuning process until we reach a model that is favorable.