Data Mining Update

MIMIC - Predicting the survival of patients entering ICU

https://mimic.physionet.org/

Team Members

Austin McElroy, Thomas Plantin, Campbell Saint-Vincent

Tasks Completed

MIMIC Training

Each team member was required to complete a 2 hour CITI program training course as an MIT affiliate in order to gain access to the latest dataset MIMIC-III v1.4. Once approved, further access was requested in order to query data efficiently using MySQL with Google BigQuery. Until the dataset was opened up through the training process, we only had guidance through documentation and were unable to actually handle the data until after the first proposal was submitted, which may be a consideration for future datasets used in the class.

Table Exploration

MIMIC's website contains a description of their tables and the columns present in each table. Each table description was carefully read over and columns of interest were chosen from tables thought to be relevant. In many instances, the columns present in a particular table were not desired as a feature, but we could engineer a feature from it. As an example, one table contains *intime* and *outtime* which were the times logged for when a patient was admitted and released from ICU respectively. These columns were used to calculate a new feature *tot_icu_hours* which is the total time a patient spent in ICU. We were only able to validate this approach once the documents were released after training.

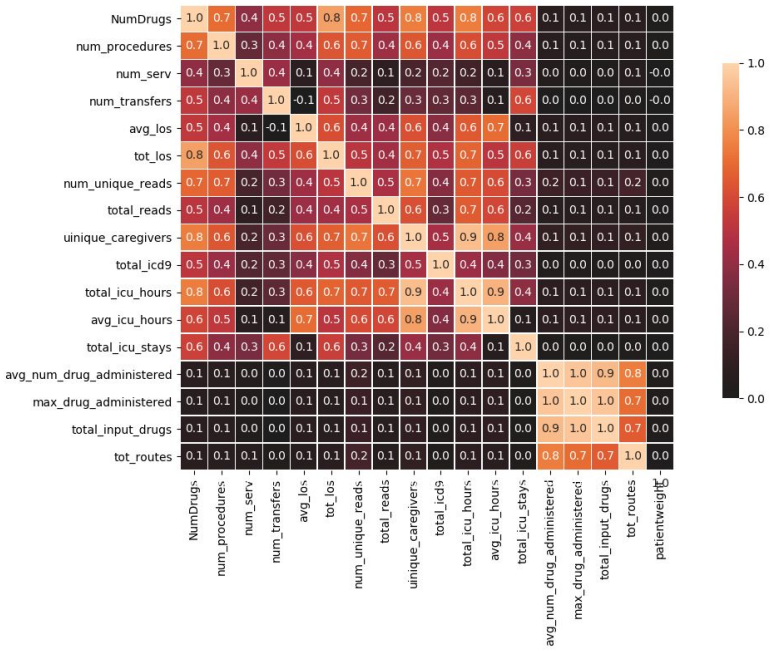
Data and Feature Engineering

Upon access to the Data through Google BigQuery, new tables were generated with SQL queries. These queries created new tables with potentially interesting columns from the original dataset we wanted to use as features as well as newly appended columns of engineered features. In each table subquery, two columns were kept constant: subject_id and hadm_id which were a unique patient's id as well as their unique visit id. Note that each patient has a unique subject_id but may have several hadm_ids if being admitted into the hospital more than once. Finally, all tables were left-joined on hadm_id and saved as a CSV file. The csv file is easily imported into Python using the **pandas** library so each member could experiment in different tool kits.

Feature Selection

Once the final dataset was obtained, features were plotted with a correlation heatmap and domain knowledge was used to trim out unwanted features. Additional patients were dropped based on patient availability. For example, only about 66% of the patients had a recorded weight, but we felt weight was an important factor and dropped patients that did not have a recorded weight. Forward, backward, or mean filling as discussed early in the class did not seem a suitable alternative to dropping due to the number of missing entries.

Initial analysis shows some highly correlated features that we can eliminate, bringing the feature count down to around 15 features over ~24k patients.



Tasks Remaining

Members now will independently evaluate the same pre-processed data with different hypothesis classes. Campbell will explore AdaBoost and Random Forest, Austin will create a Neural Network and KNN, and Thomas will explore Support Vector Machines. Upon completion, a comparison of methods will be presented.

Challenges

Campbell's previous experience with SQL was using Cloudera Impala. By using Google Big Query, some SQL code needed to be re-written as the syntax is different in order to get the same desired result.

Once the data was formed, there was a lot of missing data and a substantial class imbalance between patients who survived ICU and those who died in the hospital. There were many more people who survived, causing our models to under-predict mortality. This was solved by scaling back the training and test split to contain a random sample of living patients the same size as the deceased patients.

We have also been having trouble with feed forward neural networks. We are using a death flag in a one hot encoding method, but the accuracy has been substandard compared to KNN (around 93% accuracy, k=3, 91% accuracy k=5 with testing data). We may need to apply PCA to reduce dimensionality for the feed forward NN.

Timeline

1. Finish CITI training on HIPPA compliance and data sharing (COMPLETE)
2. Gain access to the data through the application process (COMPLETE)
3. Read through the MIMIC Database Documentation (COMPLETE)
4. Aggregate feature data by patient SUBJECT_ID. Join tables (via SQL) to create final dataset for learning consisting of: (COMPLETE)
5. Filter the data and ensure no missing data (COMPLETE)
6. Preprocess data, if needed, and engineer additional features. Explore existing features. (COMPLETE)
7. Apply machine learning models of different types. Each member will propose and work a machine learning model. (Late April, Early May)
8. New Task - Visualize results using Matplotlib (Late April)
9. Work on presentation (Early May)