Predicting Length of Stay from Patient Summary Data

(Proof of Concept)

As a proof of concept, a short time window (24 hours) was set aside to understand what might be possible while avoiding resource intensive dead-ends. The results of this show that further elaboration on this approach and a large data sample could be worthwhile.

Methodology

The first half hour was spent identifying important questions and major challenges/risks. (See Initial_Half_Hour_Assessment.txt). Questions 1 and 3 in this file turned out to be critical to the subsequent work done: Length of Stay is measured in whole numbers because of the daily mostly time-invariant process hospitals use to determine when a patient is usually discharged, and the data appears to have some duplication which might have been created by a sampling with replacement method that could skew the training model.

The data provided had over 200 variables with no data dictionary. But since the number of data samples was more than 100 times the number of variables, the entire feature set might not overfit. So it was judged worthwhile to spend a significant portion of the time window on exploratory data analysis (EDA), heavily using automated methods to reduce that time. Writing those automated methods from scratch meant heavily using SQL as the primary language of EDA. The SQL and python programs in the EDA folder describe almost entirely the EDA done during this process (very little ad hoc, unrecorded analysis was done).

What would turn out to be a critical result from the EDA (see Column_Corellation_LOS.sql for method) is that **every predictor explains by itself very little** (R^2<.03) of the predictability of Length of State (the column actualLOS). Consequently it was felt that a quite large number of predictors would be needed by the prediction model, and it is possible that unsupervised learning (clustering) of successful or failed **predictions to infer likely cohorts may prove impractical**.

Another factor in the 200+ predictors was missing data. Find_Non-Nullable_Columns.sql was used to determine which predictors had missing data in the provided dataset. The large majority of them did.

The column encounterID is not a predictor (nor is it even a primary key in the dataset). The column adjustedLOS and proceduresCount appear to be data created after the Length of Stay was known, so they can't be used as predictors. These three columns were removed from all prediction models.
event

Initial Model Selection

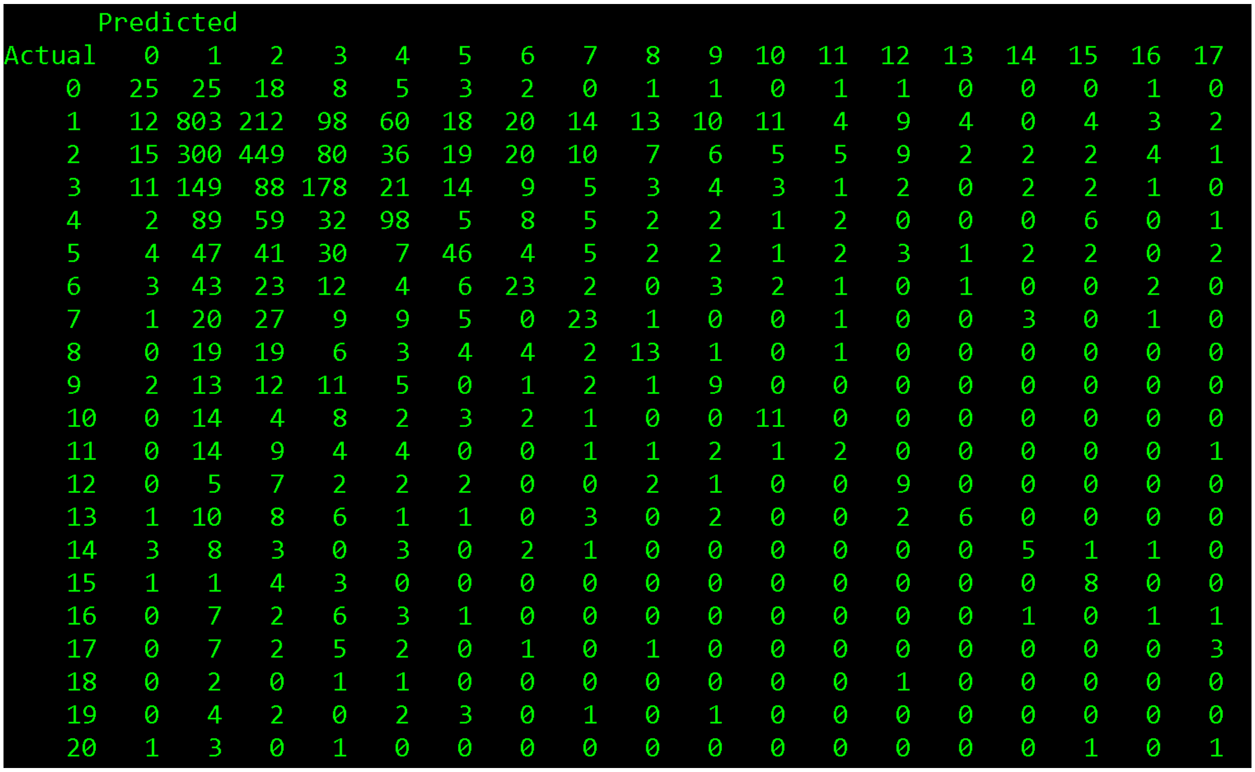
Because of the whole number nature of the outcome variable Length of Stay regression-based models, the large number of predictors with missing data and my unfamiliarity with predicting whole numbers and the very short time window to complete this POC, the project's desire to infer causality/explain themselves, I chose a classification model that handles missing data, C5.0. I have used C5.0 in another project previously. This extremely-highly regarded supervised-learning algorithm is only available open-source in R. So for that part of this project I used R.

It had three problems though.
1) I don't know how (or if it is possible) to change the loss function used its training to recognize for example that predicting a 2-day stay class as a 3-day stay class is nearly not as costly as predicting a 2-day stay class as a 10-day stay class.

2) For a not-yet-known-reason, using data with missing values was causing a C5.0 error message "c50 code called exit with value 1". So I only used a data subset with predictors that had no missing data.

3) Due to the very low correlation in predictors, the tree is so large (over 6000 decisions) that while you can read each decision point in the text (see tree.txt), it is too large to be get an overall feel (inference/self-explainability desire in project requirements).

Regardless of these issues, the confusion matrix is promising. The likelihood of being far away from number of days is low and generally declines with increasing day divergences.



```
      Predicted
Actual   0   1   2   3   4   5   6   7   8   9  10  11  12  13  14  15  16  17
     0  25  25  18   8   5   3   2   0   1   1   0   1   1   0   0   0   1   0
     1  12 803 212  98  60  18  20  14  13  10  11   4   9   4   0   4   3   2
     2  15 300 449  80  36  19  20  10   7   6   5   5   9   2   2   2   4   1
     3  11 149  88 178  21  14   9   5   3   4   3   1   2   0   2   2   1   0
     4   2  89  59  32  98   5   8   5   2   2   1   2   0   0   0   6   0   1
     5   4  47  41  30   7  46   4   5   2   2   1   2   3   1   2   2   0   2
     6   3  43  23  12   4   6  23   2   0   3   2   1   0   1   0   0   2   0
     7   1  20  27   9   9   5   0  23   1   0   0   1   0   0   3   0   1   0
     8   0  19  19   6   3   4   4   2  13   1   0   1   0   0   0   0   0   0
     9   2  13  12  11   5   0   1   2   1   9   0   0   0   0   0   0   0   0
    10   0  14   4   8   2   3   2   1   0   0  11   0   0   0   0   0   0   0
    11   0  14   9   4   4   0   0   1   1   2   1   2   0   0   0   0   0   1
    12   0   5   7   2   2   2   0   0   2   1   0   0   9   0   0   0   0   0
    13   1  10   8   6   1   1   0   3   0   2   0   0   2   6   0   0   0   0
    14   3   8   3   0   3   0   2   1   0   0   0   0   0   0   5   1   1   0
    15   1   1   4   3   0   0   0   0   0   0   0   0   0   0   0   8   0   0
    16   0   7   2   6   3   1   0   0   0   0   0   0   0   0   1   0   1   1
    17   0   7   2   5   2   0   1   0   1   0   0   0   0   0   0   0   0   3
    18   0   2   0   1   1   0   0   0   0   0   0   0   1   0   0   0   0   0
    19   0   4   2   0   2   3   0   1   0   1   0   0   0   0   0   0   0   0
    20   1   3   0   1   0   0   0   0   0   0   0   0   0   0   0   1   0   1
```

But I am not satisfied with a mean absolute length of stay difference of 2.1. So more work is needed.

Also, the out-of-sample accuracy is not great (.4 vs .7 in-sample accuracy). This can be expected when the very low correlation of all predictors will necessarily generate a very large decision tree. That increases the model complexity (its VC dimension) which increases the difference between out-of-sample and in-sample accuracy. (See Probably Approximately Correct theory or more specifically the current research on how it applies to decision trees at https://arxiv.org/pdf/2010.07374.pdf).

The README.md file contains the few steps needed to reproduce the environment, convert the data, and run the prediction model and show its results.

Next Steps

Much more data will be needed to cancel out the increased error due to this model complexity. Alternatively, much better predictors are needed. My general belief is that far too much information is lost when summarizing history by variables such as Admissions in the last 3 months, Admissions in the last 6 months, Minimum Albumin level in the last 12 months, etc. Prediction models that evaluate chains of events could be more fruitful while still providing some inference ability. A further investigation addressing these concerns would be useful.