AUCs: Train = 0.97; Eval = 0.945

Developing a Probabilistic Model of Length of Hospital Stay

# Corey Robinson

To produce a model capable of accurately predicting the probability that a given patient will stay for a “long” length of time post-surgery (i.e. more than 5 days of hospitalization post-surgery), we first made the decision to use LightGBM due to its versatility, power, and efficiency. HyperOpt was used following the baseline model in order to optimize the hyperparameters of LightGBM. The target variable, “length of stay” (LOS), was binarized since our interest was solely in whether or not the patients’ outcomes were greater or less than 5 days.

After binarizing the target (such that LOS > 5 equaled 1, and LOS < 5 equaled 0) to prepare the data for LightGBM as a classification algorithm, the feature data was cleaned and prepared. After noticing that there were many columns containing very few observations (i.e. containing many NaN values), we decided to develop models based on 1) the complete dataset, in which no columns were dropped during cleaning 2) a reduced dataset, in which columns containing more than 80% NaN values were removed. For both DataFrames, new features were engineered from columns containing date/time information. For example, new features were generated that provided information regarding the month, day, and weekday of patients’ procedures. After engineering new features, categorical features (i.e. features containing string variables) were encoded using the CatBoostEncoder included in the package called *category\_encoders*. CatBoost encoding was chosen over One-Hot Encoding in order to mitigate the issue of high-dimensionality that could occur by one-hot encoding features with many levels. Next, we checked the target for imbalanced classes using a bar chart of frequency counts for each class. The bar chart indicated an imbalance to be addressed.

The first model that we ran was, more or less, used as a baseline. Using the complete DataSet, we applied LightGBM to the complete dataset *without* addressing the imbalanced target or the large number of NaN values contained within the feature set. Further, no tuning of hyperparameters was performed. LightGBM is designed to handle NaNs internally, but target imbalances and poor tuning of hyperparameters can have significant negative impacts on its performance. The model performed well on the training set, attaining an AUC score of 0.98. However, because the model was extremely overfit to the training data, the AUC score obtained for the evaluation set was only 0.80.

The second model attempted to address the issue of overfitting that plagued the first model by performing hyperparameter optimization using HyperOpt in order to determine the ideal values of LightGBMs hyperparameters. Similar to the first model, however, LightGBM and HyperOpt were performed on the complete dataset without addressing the imbalanced target or NaN values. The resultant AUC scores were significantly better for the second model than the first, suggesting the importance of hyperparameter optimization. The AUC score for the training data was 0.857, and the AUC for the evaluation data was 0.820. This model performed reasonably well, and did not suffer from the severe overfitting of the first model. We considered stopping here, as these results indicated that the model possessed adequate predictive power and the ability to generalize to new, unseen data.

However, we decided to find out what the impact would be on the model’s effectiveness if we were to finally address the imbalanced classes of the target. First, though, we implemented a strategy for handling the many missing values within the dataset. We used Scikit-Learn’s IterativeImputer (with ExtraTreesRegressor as the estimator) in order to predict and impute the missing values. Then, following imputation, we used SMOTE (Synthetic Minority Oversampling Technique) to resolve the target imbalance by synthesizing balanced target classes. Finally, we performed LightGBM with HyperOpt one last time. This time, the AUC score for the training data and evaluation data, respectively, were 0.97 and 0.945—a gigantic improvement over models #1 and #2. This model displays very strong predictive ability, and does not significantly suffer when exposed to unseen data (i.e. is not overfit).

In order to develop a deployment strategy, we took into account both the generated ROC-AUC scores and the results of a Threshold Plot displaying precision, recall, and f1-score. Based on this Threshold Plot, we suggest that the discrimination threshold be set at 0.43. The implication of this threshold is that patients whose predicted probability scores are greater than 0.43 should be considered to be high-risk (for requiring a post-surgery hospital stay of more than 5 days), while patients whose predicted probability scores are less than 0.43 should be considered low-risk (for requiring a post-surgery hospital stay of more than 5 days). However, if the potential consequences of the discrimination between high-risk and low-risk are significant or severe, we suggest that for any patient whose predicted probability falls within a 5-point range of the threshold, the patient’s condition should be inspected by an expert in the field. Provided that these protocols are followed, we strongly support the deployment and implementation of this model.