Mohammed Khan
Dr Peter Salemi
Due July 2, 2021
DSE6211

Final Report

Executive Summary

The incidence of short-term readmissions have an adverse effect on the status and impression of a care facility. Based on Medicare reimbursements, the hospital industry lost an estimated $566 million in FY 2019. These penalties are leveraged on hospitals based on their rate of readmissions within a 30 day period. Centers for Medicare and Medicaid (CMM) services withhold reimbursements from hospitals at a maximum rate of 3% per Medicare patient readmitted. This situation opens an avenue of business improvement to save costs and therefore increase revenue. By analyzing the incidence of readmissions and the features that can become predictive of this target through machine learning, hospital governance and clinical consultants can legislate best practices to keep costs for the hospital low and quality of patient care high.

The model for this exercise is able to predict the negative classes, but is unable to identify patterns and features in the data which would increase the classification probability above 50%. There is only one observation for which the model predicted a positive target level out of the 53,673 total observations. We would expect the model predict closer to the split of negative/ positive classes present in the original dataset. This means the model is unable to reliably predict the positive class of <30 day readmission. The <30 day readmission level made up less than 9% of the total observations in the dataset. When the number of observations were taken within both the training and the test set, both subsets of data indicated about a 91/9 split in the negative/ positive class proportion.

The findings from this model indicate the most important features from this exercise to help predict the classification of the target levels are ‘lab_procedures_per_day’, ‘medications_per_day’, and ‘num_lab_procedures’. Based on the features used and the random forest algorithm, the treatment provided during the encounter proved to be most predictive in terms of this model (Figure 2). Procedures and medications are ordered at the direction of the providers, who are informed by their deep medical knowledge, but also display subjectivity in their decision-making. The model doubles down on this conjecture as the next most important features are similarly numeric attributes in direct control of care providers. Diagnosis codes are less subjective by definition, but these factors are also decided by the medical providers. The fact that the model assigns more significance to the treatment provided by the hospital rather than the demographic traits of a patient brings up the possibility that short-term readmissions can truly be affected by the performance and quality of care within a hospital. This plot emphasizes the importance of care provided by the hospital, however there may be bias in play when analyzing the results of this plot. The features used in the model are the features aimed to represent a direct influence of a care provider’s actions. Attributes such as race and payer codes are taken out from the analysis, so this model already heavily skews towards aspects of patient care in which the hospital is directly responsible for. Another quirk of this model is how the most important features end up being the numerical values. The most important categorical factor attribute based

on the plot is ‘insulin’, which has a direct connection to the patient population this dataset focuses on. Patients with a diagnosis of diabetes are likely to be treated with insulin at some stage of their encounter. The rest of the categorical variables are fairly weak compared to the numerical features, so an improvement to this model would be to either collect more relevant categorical data, or to strengthen the impact of the categorical variables already present through further feature engineering.

Data and Approach

The data provided includes patient demographics, hospital intake information, procedure information, and diagnosis and lab information. While not a comprehensive reservoir of data, this is a reasonable amount of data to be available at the point of service in an inpatient setting. The expectation is that this data and the features derived from it can be used to adequately predict the classification of the target variable, which is the status of readmission. The ideal value for the target variable for all patients would be any value that does not indicate that the patient had to be readmitted after 30 days from discharge. In this situation, expecting only the ideal outcome is unrealistic from a real world perspective, so we try to leverage relevant features to predict which features are correlated with the target outcomes.

` The objective of this exercise is to create a model which gives indications on which patients are at risk of readmissions within 30 days. Patients being readmitted would be the target for this project. The target will be split into two categorical variables (else, <30 days). According to the data, <30 day admissions only make up approximately 8.79% of the total number of encounters given in this dataset. Since the <30 day readmission encounters make up a small chunk of our dataset, there might be a better opportunity to find features that correlate strongly with the target, however since there is a low prevalence of the target level in question, smaller training samples may not provide the requisite information to consistently classify patients correctly.

This dataset is taken from 130 hospitals based in the USA over a timespan of 10 years between 1999 and 2008. There are 50 features, including the target feature of readmission. The other features outline patient and hospital data at the time of the encounter. These features include lab tests and medications, encoded logically. The encounters are inpatient, which all include a diagnosis of diabetes. The nature of the encounter is a stay of at least 1 day and up to 14 days.

Specific variables of interest range from age, weight, admission type, discharge disposition, admission source, time in hospital, number of procedures, and diagnosis codes. These variables are significant because they are hospital based variables. Demographic variables can be important, however demographics and traits unique to patients cannot be altered. Attributes such as race and payer code are removed from the data. It would not be realistic to expect improvement in readmission by focusing on factors out of the hospital's control. Age is an interesting patient demographic because it is already binned in the data and will serve as a way to break down the hospital variable metrics by age, to see if the hospital's procedures affect certain age groups more than others. The weight variable is structured similarly to the age variable,

albeit with a significant number of missing data. This data will have to be imputed taking into account the rest of the dataset to provide reasonable estimates. Feature engineering of variables could revolve around the number of days spent in the hospital. Number of procedures and medications given per day in the hospital may give us pertinent data points to compare to our target variable. The numerical variables will be scaled with a fixed mean and standard deviation. Isolating diagnosis codes and seeing which codes show up the most in our population of interest could also be illuminating.

How missing values will be treated will depend on the column. As we see in the 'weight' column, most of these inputs are missing. Imputing these values without having a larger sample size would leave most of these missing values to be inaccurate. Removing these values outright might introduce a layer of bias in the analysis. The solution here will be to utilize the MICE package to impute the missing data using the data in the other columns. The imputed data may not be fully accurate since the vast majority of the observations in this attribute would be from imputation, but along with age, weight is a useful feature regarding strictly patient demographics.

There looks to be missing data within the diagnosis groups as well. This set of missing values may be by design because not all patients have multiple diagnoses. In this case, it is not that the data his missing, it is more likely that is does not exist in reality. This missing data has been imputed with a factor level of 0.

Race, payer code and medical speciality are also attributes that are missing a significant number of data. These attributes do not have direct relation to other columns in the dataset. For the weight example, it can be argued that weight is near linearly correlated with age. This relation gives us grounds to calculate missing values with at least a modicum of estimation. Race, payer code and medical speciality cannot be related to other attributes and do not seem like helpful features for this exercise. If this information were to be randomized based on the levels already the respective columns, that would be satisfactory for our analysis. Otherwise, this data is tidy as a whole and the features we are interested in are all intact.

The target of prediction for this assignment is the readmission variable. This will tell us whether the patient had been needed to be admitted again into the hospital. The categorical levels are (NO, <30, >30). This target will be split into two categories, eventually being dubbed 0 for the NO and >30 levels, and 1 for the <30 level. This problem requires supervised learning because we are feeding the model with a target along with features to act as predictors for said target. This model is a classification model because we are sorting observations into groups rather than producing a numerical value.

Random forest is an appropriate machine learning technique for this exercise because its strength is supervised classification. A random forest algorithm is made up of multiple decision trees. These decision trees have nodes that set a criteria for any dataset put through it. Through multiple nodes, the dataset is eventually filtered and a classification is assigned to the remaining observations. The nodes in they case will be the features we engineer and employ to classify the model. The challenge will be to make sure the features are crafted in such a way that the observations that fit under a certain feature are as similar as possible, however the results filtered out using that feature should be as completely different. Based on the features the decision tree is

programmed with, along with the random subset of the data is will process, the tree should be able to produce a classification relevant to our target variable namely either NO, <30 days, or >30 days in terms of readmission. The random forest is made when there are a multiple number of random decision trees. The trees all take in the same inputs from the randomly spliced data and each tree gives a classification at the end for each given observation. The classification which is supported by most of the trees in the random forest analysis is the final classification to which an observation is assigned. Ideally, there will be only a minimal number of decision trees that choose an incorrect classifier, while most of the trees in the random forest will end up choosing the correct level of target, based on the actual targets in the dataset. For this process to be as streamlined as possible, utilizing features that are explicitly defined is a necessity. If each node is relegated to random guessing of an observation, we will not produce a reliable and methodic model. The decision trees in the random forest differ by which observations they analyze and which features they use to classify those observations. This gives us trees within the larger forest that are nearly independent, so we can be assured that the predicted classification is as unbiased as possible. This will produce a high variance, but with only three possible target levels in this classification experiment, that would not be a concern.

Neural networks are another machine learning model we can use to output classifications. Neural networks are comprised of the input layers, hidden layer and the output layer. Training neural networks involves the target variable for each observation, which is a trait of a supervised learning model. The outputs are in the form of a value which are to be compared to assigned values of the target outcomes. After the predicted value is made, an error value is calculated between the predicted and actual values. Neural networks rely on backpropagation to adjust the weights of each layer to make sure the output corresponds to a certain input. Once the error is calculated, we can utilize backpropagation to adjust the weights of each layer to minimize error. This method is appropriate for this analysis because of the large number of observations available for training as neural networks are likely more accurate due to being tuned with a greater number of inputs. There are multiple features involved in this analysis that should not all be weighed the same way. Through neural networks, we can assign an individual weight to features which will result in the inception of an accurate and precise model.

If the target variable is encoded two ways, then the probability of an observation regarding the target variable will tell us which outcome is most likely while also giving us a hint as to how the other outcomes stack up in terms of probability. A classification probability of below 50% would predict a negative target case, while a probability of above 50% would predict a positive target case.

Since we already have the target included in our dataset, we should be able to utilize ROC curves and AUC to find error based on our predicted values. The challenge will be to blend the true positive/false positive values for three different levels. The ROC curve is first generated using the confusion matrix. This matrix will be able to give us a numerical value of how accurate the predicted output are compared to the actual target values. Since the actual target values are all correct in a supervised classification problem, these values will serve as the backdrop to the predicted target values. We assign a threshold value to compare to the confidence value of the

target level. Based on this value we can create a curve to classify each target and plot it with the true positive rate as a function of the false positive rate. The area of the curve can be calculated from the resulting plot which will give us a measure of how the model performs. The perfect classifier has an area under the curve value of 1. We cannot expect a perfect classifier and the model will be better than a random guessing model, so we will expect a value between 0.5 and 1.

Evaluation

The random forest model starts at the hyperparameters for the model. The hyperparameter value of 12 is the output for the model, and values of 10 and 14 are tested as well. The model will determine which of these values will produce the most accurate prediction. The non-encoded categorical attributes were exempted from model training, along with the target variable. The model performed under expectations based on the evaluations plots used to critique the model. The ROC curve produced the signs of a barely functioning model, with an AUC value of 0.628. This value indicates the model is poor with determining true positive rates while performing below expectations with the false positive rate, especially at the beginning of the curve, where false positive rates should start off low as true positive rates rise. Further analysis with the calibration curve exposes the model to be either extremely overfitted. The calibration curve presents as a horizontal line along 0, when it should be more in line with the 45 degree angle outlined on the plot. Based on this curve, the model is severely overconfident for all predicted probabilities. The model only predicted one positive classification within 53,637 observations. The feature importance plot places a heavy emphasis on the effectiveness of numerical features that contribute greatly to the model. This could be framed as a negative trait, as the model does not perform well enough to have any confidence in the true value of these features. With a more accurate model, the importance plot would hold more weight.

Mohammed Khan
Dr Peter Salemi
Due July 2, 2021
DSE6211

Appendix

Figure 1 Classification Curve

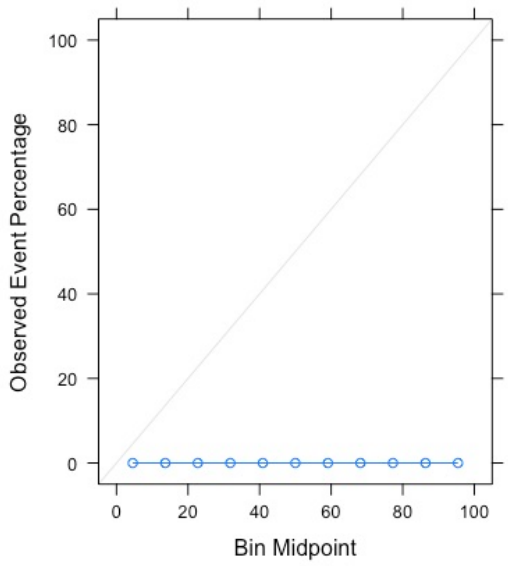

Figure 2 Importance Plot

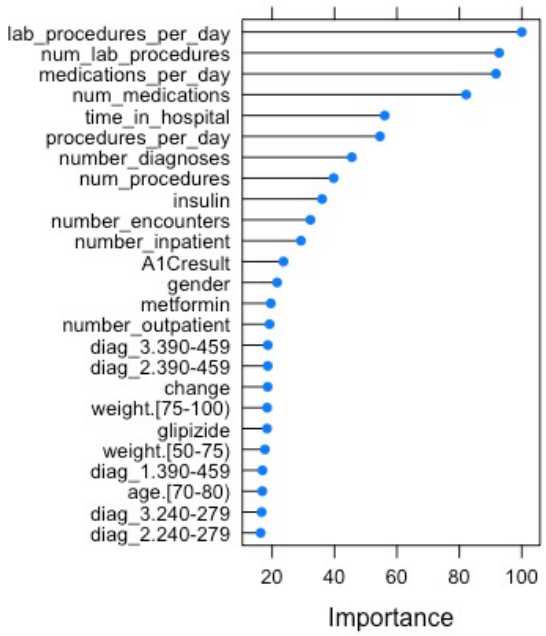

Figure 2 ROC Curve and AUC