30 Days Hospital Readmission Prediction

Kris Ghimire, Rajesh Satluri, Suchismita Moharana

May 23, 2021

**Abstract**

In this case study, researchers are going to build a classification model to predict if the patients will readmit with 30 days of discharge from the hospital.

**1. Introduction:**

According to study conducted by CMS (Centers for Medicare and Medicaid Services) US healthcare spends more than $15 billion per year on readmission cost. One in five elderly patients are readmitted to the hospital with 30 days of discharge. Readmission not only bring financial burden to the institution it also indicates a breakdown in caregiving and also impacts patient emotionally.

Leveraging the power of data can help in reducing many preventable hospital readmissions which is a major priority of payer, care provider, and policymaker seeking to lower the healthcare cost and improve healthcare. We don’t want patient going back to hospital. We want them to be well. Keep this thing in mind, in this paper we want to point out some driving factor that are responsible for readmission. And build Machine Learning (ML) model to predict if the patient will be readmitted within 30 days of discharge from the hospital.

## 2. Methods

**Dataset:**

Table

Description automatically generatedTable 1

**Train Test Split:** Dataset was split into training and test set (80/20 split) to keep test data separate and to perform modeling and tuning only on training set.

Figure 1

**Multicollinearity:** One of the key assumptions of linear model is to make sure that features are not highly correlated. High correlation between features can be a problem while explaining which features contributes most to the target feature and to the validity of model in general. We plotted correlation heat map to find all correlated features. Going through each correlated feature and picking one among two is a time-consuming manual process. We tried correlation heatmap. However, since the features are many correlation heatmap is not showing all at once properly.

Table 2

***L1 regularization:*** LASSO stands for Least Absolute Shrinkage and Selection Operator. For L1 regularization, we use a penalty term where the function of the penalty is just the absolute value of the coefficients as it is shown in figure formula below:

**Scaling:** Another key assumption is that all independent features should be normalized or scaled. As algorithm makes decision based on distance metrics, feature scaling is very important. We used *StandardScaler* to scale all the independent features. It is not a good practice to scale categorical data and our dataset does not have any categorical features. However, there are some features with many zeros and some values here and there. For example, feature 'Ca' has max values 24 while there are also many 0s, similarly feature 'Mn' has max value 14 and many 0 values.

Mathematically scaling is done using formula below:

Text

Description automatically generated

Subtract the mean (called centering) and divide by the standard deviation to shift the distribution to have a mean of zero and a standard deviation of one.

Just to verify the importance of scaling, initially we used linear model on the given dataset without scaling. And we used linear model with scaled data. We saw some change in feature importance when we fit our model using scaled data.

We will use the scaled dataset and features that we get from L1 regularization to fit our final model using Ridge or L2 regularization.

**Method of choosing C for L1:** We selected 100 evenly spaced numbers from an interval of 1 and 0 and we used this array as alpha in LassoCV function to get the best alpha value. Here we got best alpha value as 0.23232 as shown in the (figure 3)

Figure 2

Using above alpha values, our L1 model selected the below features (figure 3) as most important. We decided to keep the number of features to 15 so that the model is a generic model.

Figure 3

Here we see that features related to Thermal Conductivity is having importance in the model. It does make sense based on our basic science knowledge.

**L2 regularization:** L2 or RIDGE regularization uses a penalty term that is based on the size of the coefficient squared. Hence, it’s called second order regularization. Unlike L1, L2 provides a general method to prevent overfitting. With L2 the coefficients are not driven to 0. Thus, weak contributions while being suppressed are not suppressed completely to 0s before strong interactions. All the interactions are inhibited or suppressed together, which makes L2 an excellent tool for overfitting for this linear regression model and as well as other more advanced models.

**C selection:** We performed 5-fold cross validation to get the best value of alpha to use for final model using Ridge regularization. Alpha value of 0.1 obtained from cross-validation was used to build ridge model.

Final ridge model was built using important features obtained from L1 and best alpha value.

**Cross-Validation**: We will use K-Fold cross-validator with shuffle = true, to shuffle the data before splitting into batches to validate our model. With K-Folds, our full dataset is first randomly partitioned into a user-specified k number of subsets of data, in our case we will specify k=5.

Dataset of 21263 instances was divided into five separate subsets of data, and then run through five successive cross-validation implementations, in which four of the five subsets represent the training set, and the remaining set the test. Cross-validation provides an iterative look at our model’s performance. So, implementing 5-fold cross-validation will produce the five metrics representing the negative mean of the errors of Ridge model. These values are manipulated to find MAE, MSE and RSME.

**3. Result:**

*Table 3*

Figure 4

Figure 5

4. Conclusion: