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 be readmitted within 30 days of initial hospitalization.

**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 do not want patient going back to hospital. We want them to be well. Keeping this thing in mind, in this paper we want to point out some driving factors which are possibly cause 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

The dataset as depicted in table 1 consists of 101766 samples, 50 features. It is a good size data to build classification model. Dataset mostly has categorical features and few numerical features.

We did not find any missing data upon initial check. After checking the distinct values of categorical variables, we noticed that it has lot of '?’. So, we converted those to NaN for systematic imputation for all NaN’s. We did drop the weight column as we found that 97 % of the values were missing. There were other features with nearly 50% missing data, and we decided to keep them instead of dropping for retaining maximum information. We have created custom feature ‘AgeGroup’ by grouping Age into different age groups for the ease of dummy encoding and we eventually dropped Age column.

{'0-10':'child', '10-20':'young adult', '20-30':'twenties', '30-40':'thirties', '40-50':'fourties', '50-60':'fifties', '60-70':'sixties', '70-80':'seventies', '80-90':'eighties', '90-100':'nineties'}

Also, we have dropped encounter\_id and patient\_nbr columns as it is just record identifier and will not contribute to the model. We mapped few other features to its actual name to make better sense of the data. Some of the mapped features includes admission\_type\_id, discharge\_disposition\_id, A1Cresult etc. One of the challenges we had was mapping diagnosis ICD9 to its appropriate name (Ref: https://icd.codes/icd9cm), since their codes were many, we binned all ICD9 codes to few groups to make feature more appropriate to feed to ML model. All the features which need imputation are categorical. So, we have imputation with Mode group by AgeGroup to make it more relevant using sklean simpleImputer library.

In the case of target features, we noticed three categories >30, <30 and NO. We have converted >30 and NO as ‘NO’ and <30 and ‘YES’. Doing so made the target feature unbalanced as it is show in figure 1 below.

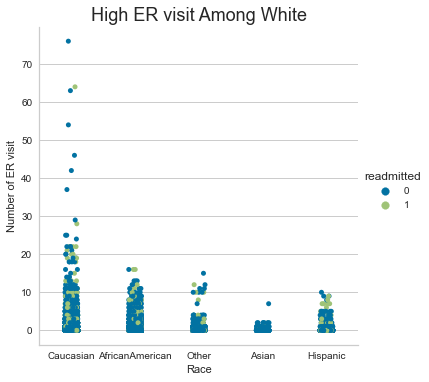
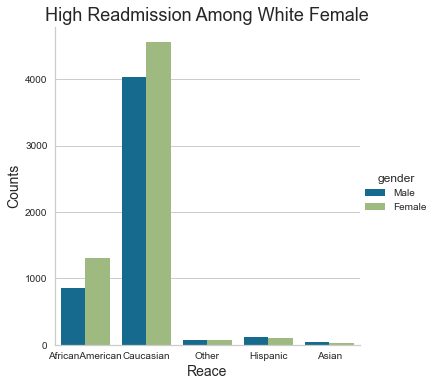
Chart, bar chart

Description automatically generated

Figure 1

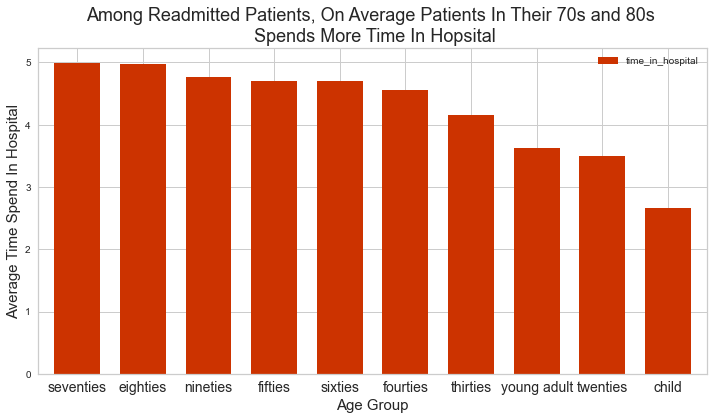
**Data Exploration:**

We have observed that majority of sample were of white race and mostly female (figure 2), which further depicts that, in general female has higher chances of getting readmitted compared to male. The data also shows that patients who visit ER often has the higher likelihood of getting readmitted (figure 3)



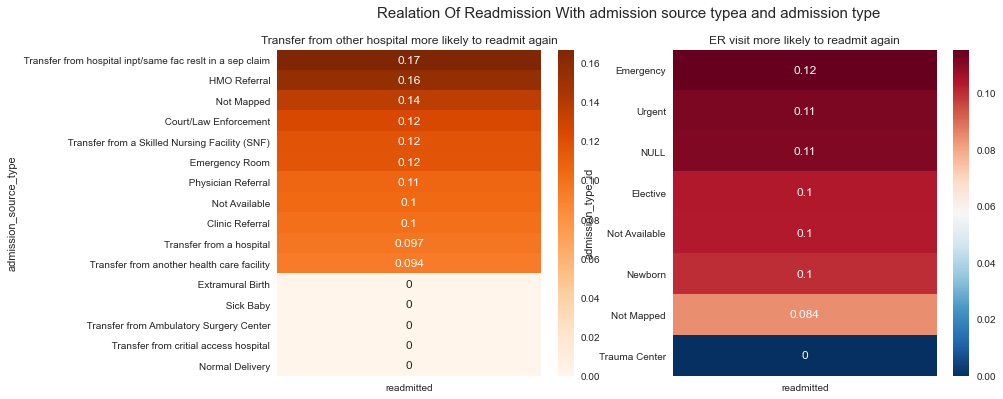
*Figure 2*   *Figure 3*

As expected, figure 4 below shows that among patients who were readmitted are age above 70 years, spends more time in the hospital compared to other age groups.

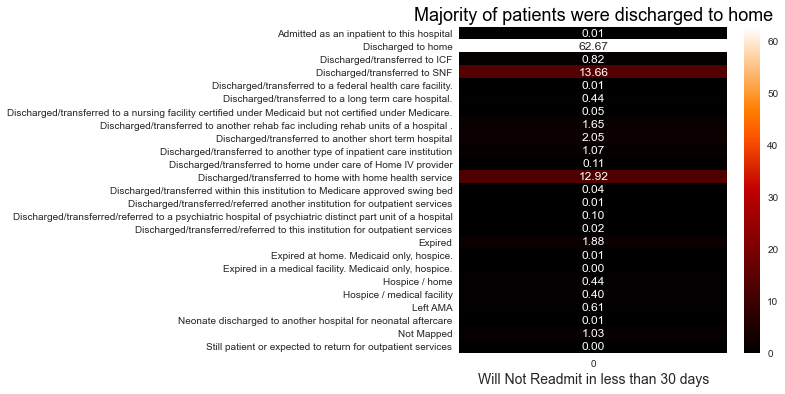


*Figure 4*

While exploring the relation of readmission with admission source type and admission type we noticed that the patients are more likely to get readmitted again in less than 30 days if they are ‘Transfer from hospital inpt/same fac result in a sep claim’ (figure 5) and if a patient is ER patients (figure 6). In contrast, patients are less likely to get readmitted if they were discharged to home after their initial visit. (figure 7)

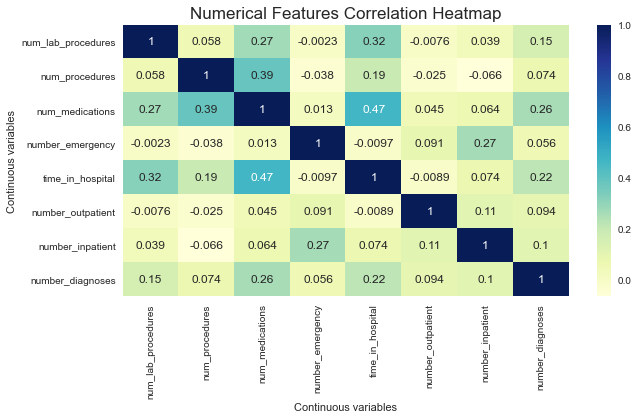


*Figure 5* *Figure 6*



*Figure 7*

**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 of all numerical features to check their correlation strength. (figure 8). There seems to be some correlation between number of medication and time in hospital and between number of procedure and number of medications. However, they are not strong enough to make much difference to final model.

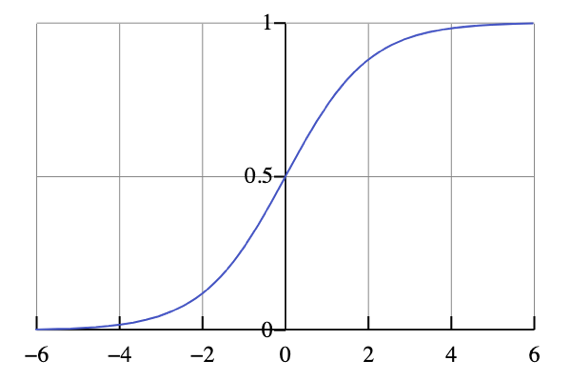


*Figure 8*

**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.

**Logistic Regression:** Logistic regression is like linear model main difference is the sigmoid function, which helps in transforming from a linear regression to a logistic regression.

The sigmoid function takes the input X and it essentially squeezes the output so that it is between 0 and 1. And it gives us the S-shaped curve that is shown in the figure 9 below. The sklearn library provides the default threshold to 0.5. For this study, any probabilities that is greater than 0.5 is considered to the probability that the patient will be readmitted within 30 days and the probability less than 0.5 are considered to the probability that the patient will not be readmitted.



*Figure 9: Sigmoid curve*

**Sigmoid function:**

Where y is the target and the exponential negative sum implies sum, the summary of all m’s (slopes) and all the x’s (independent features).

In the above equation, if sum is extremely large and it's positive, we will have 1 over 1 plus e to the negative large number. An e to negative large number is, in essence, very small number because its 1 over e to that power. So that number becomes 1 plus an extremely small number, that becomes 1 over 1, which means our output is 1 or positive target. But if the sum is large and its negative, which mean we have 1 + large positive exponential number. So, 1 over 1 plus a large number exponential which mean 1 over larger number is roughly a number close to 0 or our negative target.

**Imbalance target features:** Dataset had imbalanced target features. Modeling on imbalanced target features would give evaluation metrics that are not accurate. For example, while modeling on data without balancing we got an accuracy of almost 89%, after we balanced the data, we got an accuracy of about 67 %. There are several techniques to balance such as SMOTE, MSMOTE and advanced boosting methods like Gradient boosting and XG Boost. For simplicity, we have used sklearn class\_weight argument and specified the *class\_weight* as “*balanced*.” which invert the ratio of the class distribution in the training dataset and gives equal importance to the both classes. For example, if the class distribution of the training dataset is a 1:100 ratio for the minority class to the majority class. The inversion of this ratio could be used with 1 for the majority class and 100 for the minority class. We have also built another model with SMOTE technique to compare with sklearn class\_weight argument *class\_weight* as “*balanced*.”. And the SMOTE technique under performed with accuracy of 64% in this case.

**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. 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 Logistic Regression model on the given dataset without scaling. Then we used another model with scaled data. We didn’t see big changes in accuracy but saw some significant changes on feature importance.

**C selection:**  C is the inverse of lambda in linear regression. For small values of C, we increase the regularization strength which will create simple models which underfit the data. For big values of C, we low the power of regularization which implies the model is allowed to increase its complexity, and therefore, overfit the data. Our goal is to choose best C that offers the smallest difference between the training and testing accuracy. The best value of C we got for our model is **c =0.0001**.

Hence, we used scaled data set, **‘L1’ regularization** (L1 pushes lower weight features towards 0 and gives higher weight to important features) and C =0.0001 to get an important feature based on weight that is contributing the most to the target feature ‘readmitted’.

Using scaled data, C values, L1 model selected the features (figure 10) as most important. We decided to keep the number of features to 13 so that the model is a generic model.

**Chart, funnel chart

Description automatically generated**

*Figure 10*

**L2 regularization:** L2 or RIDGE regularization uses a penalty term that is based on the size of the coefficient squared. Unlike L1, L2 provides a general method to prevent overfitting. With L2 weak contributing coefficients 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 logistic linear regression model. We will build final model using l2 regularization.

**Cross-Validation**: We will use StratifiedKFold cross-validation with k=5 because with stratifiedKFold, the class distribution in the dataset is preserved in the training and test splits. Shuffle is set to true so that the splitting will be random.

**3. Result:**

Models were build using several approaches. First, we built model 1 on the data as it is without scaling and balancing. The 5- fold Stratified cross-validation gave mean accuracy of about 88 %. Model did a good job on majority class, but it did terrible job in predicting minority class (Figure-11,12). The roc curve with score of just 51% proves that the model is not a good model.

**Model 1**

A picture containing chart

Description automatically generated Chart, line chart

Description automatically generated

*Figure 11: Confusion metric Figure 12: ROC*

**Model 2**

Next, we scaled our data, performed 5-fold cross-validation to get the best C. Using best hyperparameter C = 0.0007 and L2 regularization we built another model which generated a mean accuracy of 67% on training set. (Figure 13,14). The ROC curve and score shows that model 2 is better at predicting both majority and minority class compared to model 1 above.

A picture containing logo

Description automatically generated Chart, line chart

Description automatically generated

*Figure 13: Confusion metric Figure 14: ROC*

**Model 3**

We also tried SMOTE up sampling technique to balance the data to see if get can better score. However, we did see much change in score. Mode (Model 3) built using SMOTE gave an accuracy of 64% and ROC score of 62 %

**Model 4**

Since we had many features, one might argue that feature reduction technique such as PCA might help get better accuracy. So, we tried PCA (model 4) as with n\_components = 6. One problem with PCA is that it is not interpretable. We cannot tell which features are contributing most to the target feature. With PCA we can 5-fold mean accuracy of 57% and ROC 0.56 worse than the first model.

Receiver Operating Characteristic (ROC) Curve: *ROC* is an excellent method for measuring the performance of a Classification model. More the area under the curve, better is the model at distinguishing between classes. The best model, Model 2 has ROC of 62% not the best but better than the coin flip.

Accuracy is not always the best metric to rely on depending on what we are focused on. Our goal is to build model that will predict readmission of the patient within 30 days of initial hospitalization. Hence, we need other metric such as precision and recall.

Precision score from our best model, Model 2 is not so good (Figure 14,15), which means our model predicted many patients as the patients who will readmit again even though actual test data had them labeled as NO readmissions, a high false positive rate. While a high false positive rate is present in our model, we focused specifically on readmissions, and further research is necessary to find if these false positive cases may be predictive of some other state that justifies additional consideration or intervention. In terms of recall, modal predicted patients who will readmit 55.7% of time out of total actual patient who will readmit.

Model is good at predicting if the patient will not readmit with precision 92.5% and recall 68.6% (Figure 14,15). Precision is defined as the number of predictions made that are correct out of all the predictions based on the positive class. While recall also known as sensitivity is a measure of a model to identify the percentage of relevant data point. It is defined as the number of instances of the positive class that were correctly predicted.

Precision =

Recall =

Chart, treemap chart

Description automatically generated Chart, line chart

Description automatically generated

*Figure 14: Precision, Recall, F1-score Figure 15: Precision-Recall*

There is a famous saying by George Box that all models are wrong, but some are useful. Therefore, out of all the model we have built, the model evaluation metrics, accuracy, ROC, precision and recall suggest that model 2 is statistically useful model.

**4. Conclusion:**

As in any dataset understanding feature relationship with target variable and meaning missing data imputation is cumbersome and iterative process. We spend substantial amount of time in this process. Model building and choosing the best out of the researched models is another challenging task, which needs skill to build meaning models to the context within resources. We have spent remaining most of the time in this process.

We have chosen the best model as the one with scaled and balanced data, best hyperparameter C = 0.0007 and L2 regularization. This model generated a mean accuracy of 67% on training set and ROC curve and score (62%) shows it is better at predicting both majority and minority class compared.

Further, we can create hybrid features and try other classification models to compare the model accuracy and predictability.

Even though this is not the best model, 67% accuracy is considerably decent, and we can deploy into production, while we do further investigation with different classification models.