DS7333 Quantifying the World: Case Study 2

Rick Fontenot, Feby Thomas Cheruvathoor

1. **Introduction**

The goal for this case study is to build a logistic regression model predict which hospital patients are likely to be readmitted within 30 days or longer term and evaluate which variables carry the most importance for determining readmission.

The insights generated could potentially help hospitals plan resources such as staffing and rooms, medical insurance companies in determining risk of increased costs, or patients and doctors plan for what is likely for future health risks.

Monetary data elements such as price details, cost of the medicine and procedures, the fee for certain procedures have not been included. Based on the type of information provided we are approaching this analysis with the intent of helping hospitals plan resources. While some information such as the patients race and gender have been provided and could present ethical issues if used to set conditional pricing or denial of treatment, we are utilizing this information since it does not bias decisions on hospital resource planning.

1. **Methods**

**Data description**

The raw data set contains 101,766 observations of patients with 47 features, two identifiers, plus the target variable including readmission outcomes. The features include age, race, gender, diagnoses, medications, limited test results, and information on length of stay, and the number of inpatients, outpatient and emergency room stays.

For a more detailed description of each feature, see appendix section 1.

**Missing Values**

Missing values in many features has been presented as “?” through out in the data set as the non-available data point. This has been replaced with “Nan” in python to summarize the issue and determine how best to impute missing values.

|  |  |  |
| --- | --- | --- |
| **Feature** | **Count** | **% Missing** |
| weight | 98,569 | 97% |
| medical\_specialty | 49,949 | 49% |
| payer\_code | 40,256 | 40% |
| race | 2,273 | 2% |
| diag\_3 | 1,423 | 1 % |
| diag\_2 | 358 | <1% |
| diag\_1 | 21 | <1 % |

Figure 1: Summary of features with missing values

We determined the best method of imputation for these variables prior to modeling were:

weight: With such a large percentage missing we dropped this feature. We explored imputing by using age but this would cause a multicollinearity issue and also observed that the small amount of weight records included seemed unreliable, for instance a large percentage of patients in the 60-80 year range had weight listed as less than 25 pounds

medical\_specialty: With 73 unique values and no other features to help impute missing records, we filled the NA's with "missing" to preserve the rows from being dropped from modeling. Without know the importance of this feature we could still gain prediction possibilities from other features on these records. Including the “missing” category will also help make predictions on future unseen data that has this field missing

payer\_code: Similar to medical specialty with 18 unique values and other obvious feature to assist in filling the blanks, we filled the NA's with "missing" to preserve the rows from being dropped.

race: while we considered replacing missing values with the mode or some sampling based on the distribution, in the end we handled the same as other features and filled with “missing” This could be an important feature indicating it is missing due to patient privacy, race other than what’s available on the check boxes or other reasons that could cause readmission outcomes to be different for this subset.

diag\_1, diag\_2, diag\_3: With >800 unique values and no other obvious feature to assist in filling the blanks, we filled the NA's with "missing" to preserve the rows from being dropped.

Duplicate Observations

There were no duplicates rows are observed in the data set.

**Variable Type and Formatting**

The raw data contains multiple categorical features, some of which are in numeric formats, and some which could benefit from level reductions through regrouping. We re-coded the following variables prior to modeling:

age: Originally binned in the format '[0-10)' Since age is ordered information, rather than one-hot encode this feature we used the upper bound of the bins such as 10, 20, 30, etc. and modeled this feature as numeric

diag\_1 , diag\_2, diag\_3: with >800 unique categories including both numeric and text formats, we grouped these into their broader IC9 categories ([International Statistical Classification of Diseases and Related Health Problems](https://en.wikipedia.org/wiki/ICD)). In addition to the broad 18 IC9 categories we broke out diabetes which is of particular interest from the other diagnoses within the immunity disorder group. We also preserved the “missing” values rather than include them in another category. For more information on these groupings, see appendix section 2.

max\_glu\_serum: This is a test with feature values: “>200,” “>300,” “normal,” and “none” if not measured. Rather than one-hot encode we treated this feature as numeric with the ordering 0=None, 1=Normal, 2=>200, and 3=>300 with the idea that severity of results should be ordered.

A1Cresult: This is a test with feature values: “>200,” “>300,” “normal,” and “none” if not measured. Rather than one-hot encode we treated this feature as numeric with the ordering 0=None, 1=Normal, 2=>7, and 3=>8 with the idea that severity of results should be ordered.

Medications: All 24 of these features treated this feature as numeric with the ordering 0=Not taken, 1=Dosage Down, 2= Dosage Steady, and 3= Dosage Up, with the idea that change of dosage should be ordered.

encounter\_id & patient\_nbr: These identifier features were dropped from modeling set as they are unique and not predictors of health status

One-hot encoding: After all processing summarized above the following categorical variables were one-hot encoded for modeling: 'race', 'gender', 'admission\_type\_id', 'discharge\_disposition\_id', 'admission\_source\_id', 'payer\_code',' medical\_specialty', 'change',' diabetesMed',' diag\_1\_group', 'diag\_2\_group', 'diag\_3\_group'

**Target Variable & Class Imbalance**

The target variable “readmission” has the categories: patients readmitted in less than 30 days, patients readmitted in more than 30 days, and patients not readmitted. The time frame of data was not provided, so it is unclear if the non-readmissions or the more than 30 days are bound on the upper end by 60 days, 90 days, or multiple years etc.

There is significant class imbalance in the target with just 11% of the observations in the readmission less than 30 days category.

|  |  |  |
| --- | --- | --- |
| **Readmission** | **Count** | **Percentage** |
| NO | 54,864 | 54% |
| > 30 days | 35,545 | 35% |
| < 30 days | 11,357 | 11% |

Figure 2: Summary of observations in each readmission category

In order to deal with effects of this imbalance we used stratified splits so that the training and validation sets have approximately equal proportions of the three readmission classes. See our cross-validation method description below for more details.

In addition to using stratified splits, we evaluated additional models using SMOTE from the python imblearn.over\_sampling package to resampling and balance the observations in each class of a training set.

Models of the original training set and the resampled training set will be compared for both performance and feature importance with particular attention to the validation set to ensure the resampled model is not overfitting

**Expired Patients**

Exploratory data analysis shows that 3 categories (11,19,20) had zero patients readmitted. These discharge categories represent “expired” patients. Since death is a 100% certainty of not being readmitted, these rows were deleted from our modeling data set so that the weights and importance of other features are not affected.

Upon this realization we discussed also removing records who were discharged to hospice as this is typically an end-of-life setting. (discharge codes 13,14). However the figure below shows that some hospice patients to get readmitted to the hospital, so the records were included in our models.

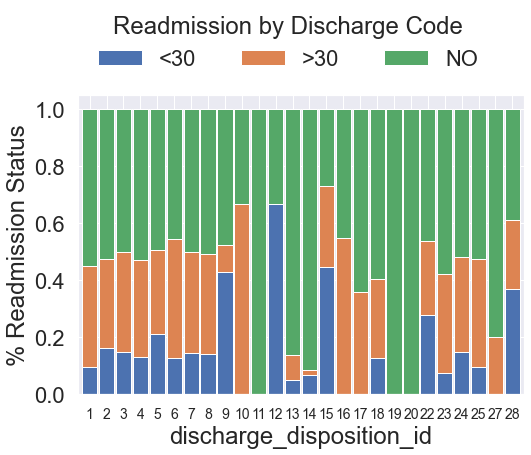


Figure 3: Readmission Rates by discharge type – expired patients 0%

**Standardizing data**

In order to determine which features are most important to predicting the critical temperature, we scaled the data so that the regression weights can be compared. We scaled all features to a mean of zero scaled to unit variance. Scikit learn StandardScaler function has been used to scale the data points.

**Cross Validation**

In order to compare the models for overfitting, we first set aside 10% of the data into a validation set (stratified shuffled data to get random observations with similar proportion of records in the target classes). After our model hyperparameters are tuned on the training set, we will compare performance metrics of the model on this unseen data.

For the training data used to build models, we used StratifiedKFold cross validation with 10-folds for cross validation and the mean accuracy across the folds for model tuning. In addition to accuracy, we will also summarize the precision and recall for each class, for comparison.

Particular attention will be focused on the validation set results to check for overfitting on the re-sampled data model

**Logistic Regression modeling**

For prediction of classifications we used the sklearn LogisticRegression package with the “saga” solver so that elastic net regularization could be used to minimize risks of overfitting.

Grid searches were used to tune hyperparameters individually for models using the original records versus the SMOTE resampled records.

1. **Results**

The optimized base model using original observations for training was able to classify the 3 classes with an overall accuracy of 58%. As expected, the majority class had higher precision and recall than the minority classes. The readmitted less than 30 days class particularly had extremely poor recall at 2% and concerningly low 4% f1-score.

The second model with resampling to balance the observation counts by class, significantly improved the minority class recall to 42% and f-1 score to 26% however it came at the expense of overall accuracy which dropped to 49%.

Classification Performance Summary

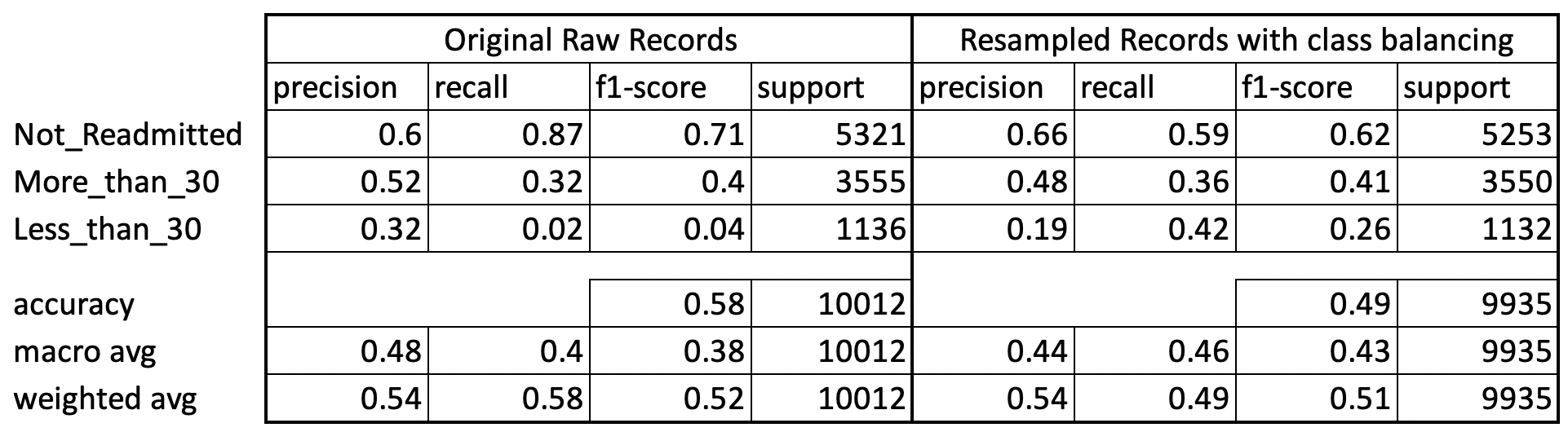


Figure 4: Classification Performance metrics for original and resampled training data models

Original observations Resampled observations

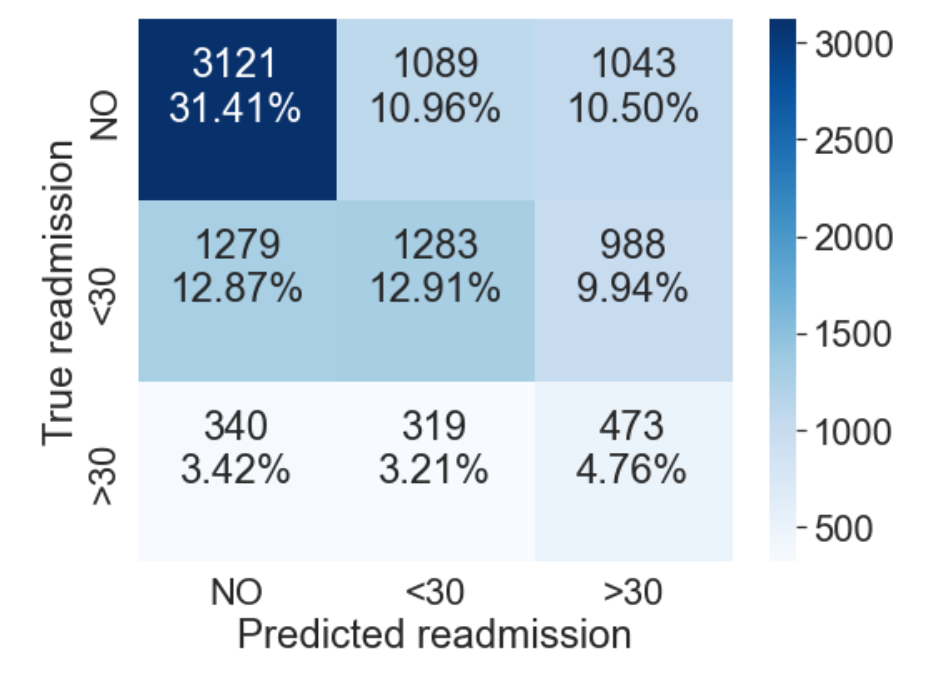
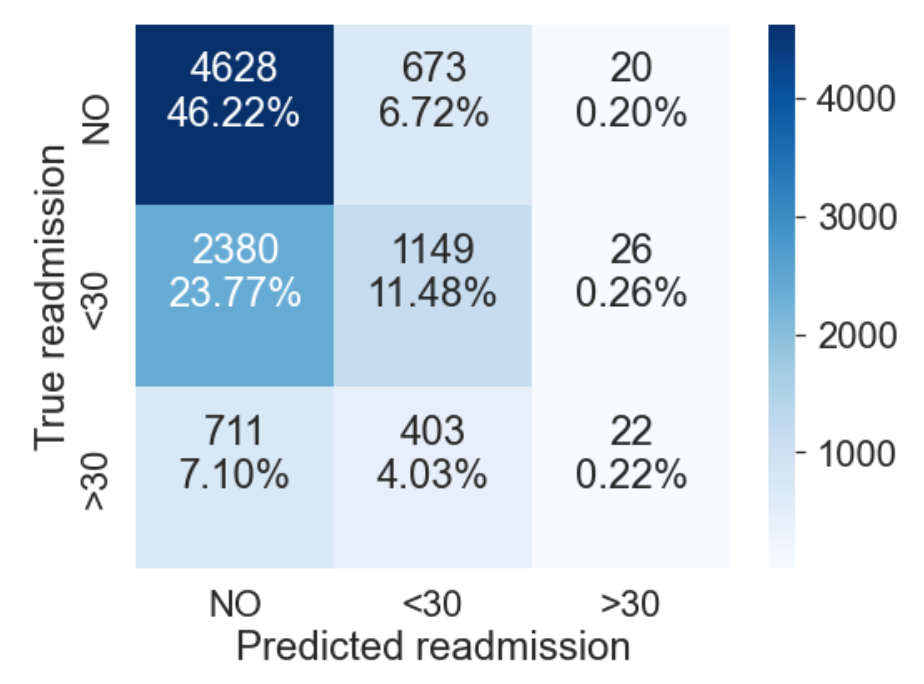
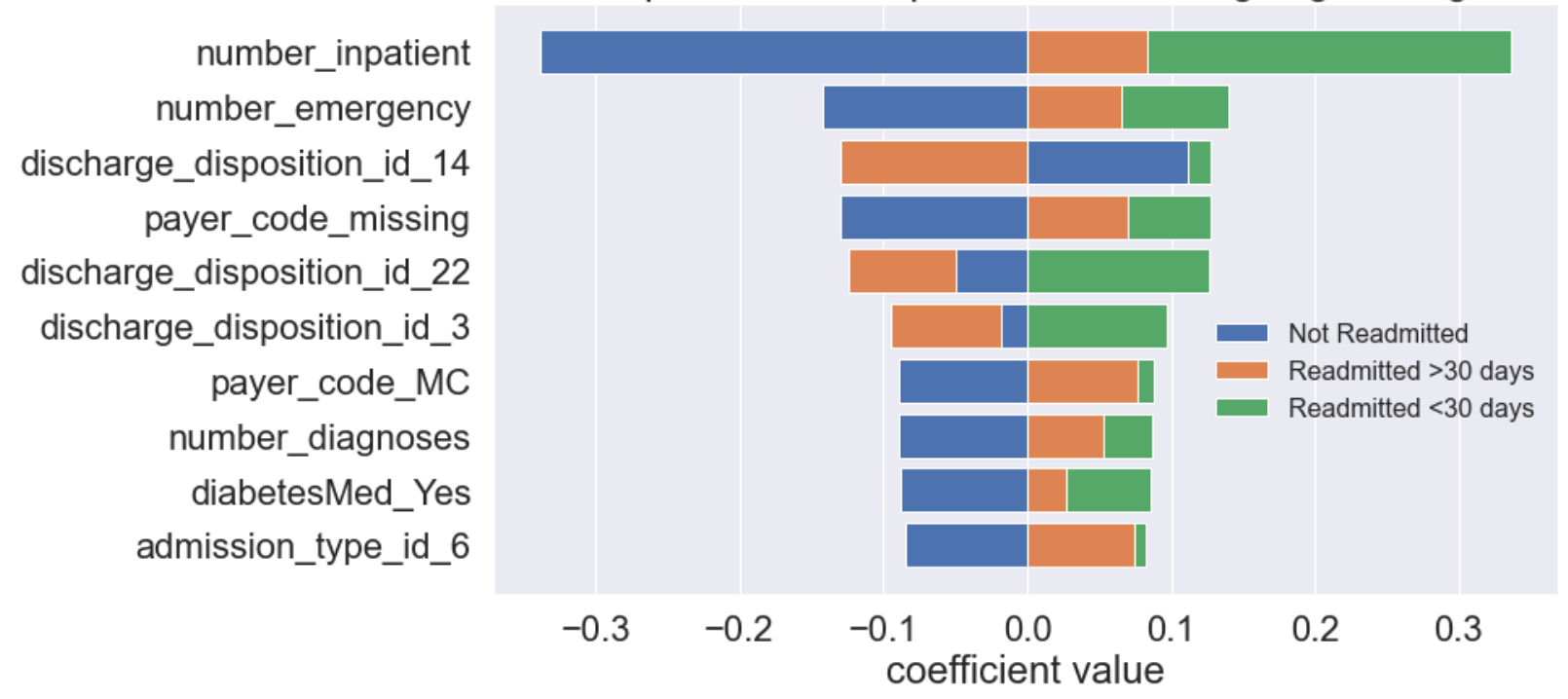


Figure 5: Confusion Matrix for original and resampled training data models

Based on the feedback from the hospitals resource planning team, we could choose the model more appropriate based on their priority for overall accuracy vs. the precision and recall for predicting the amount of patients expected to be readmitted in less than 30 days.

Both models have similar rankings of the most important features, so the driving factors can be understood with either model:

Feature Importance: Modeling Original Data



Feature Importance: Modeling Resampled Data

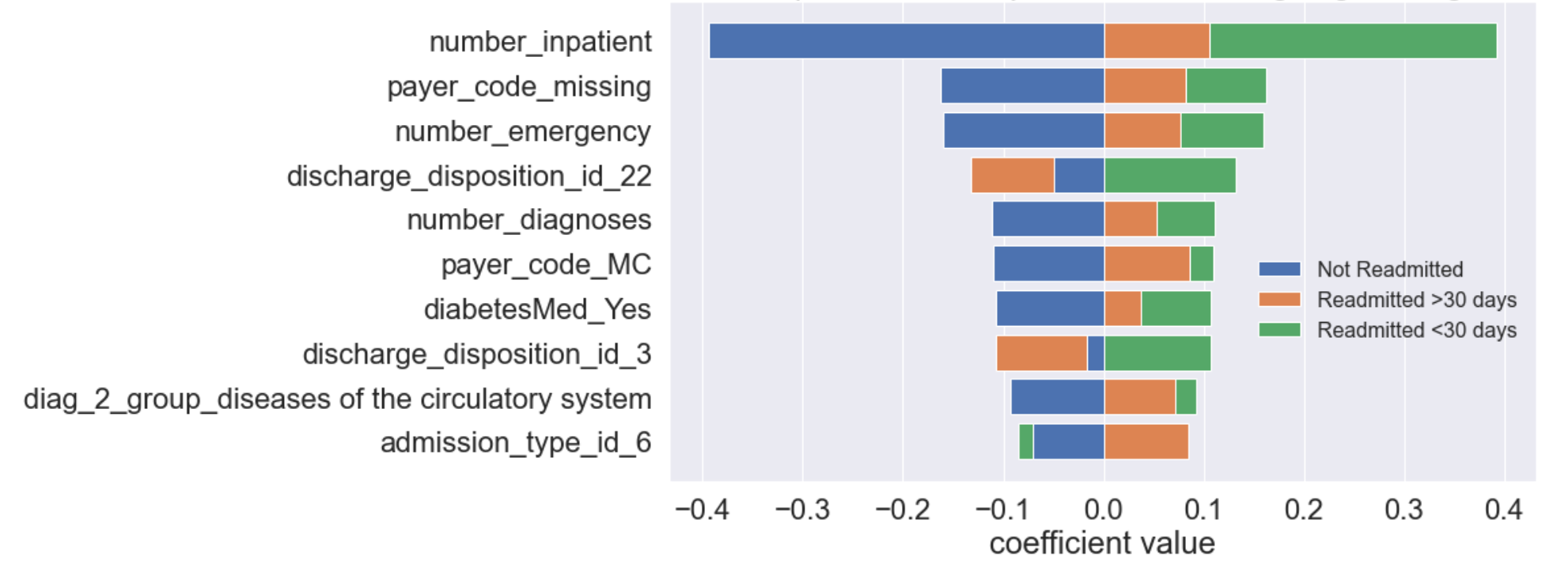


Figure 6: Ranking of most important features

The number of inpatient visits of the patient in the year preceding the encounter are the most important factor determining the likelihood of readmission, followed by the number of emergency visits of the patient in the year preceding the encounter

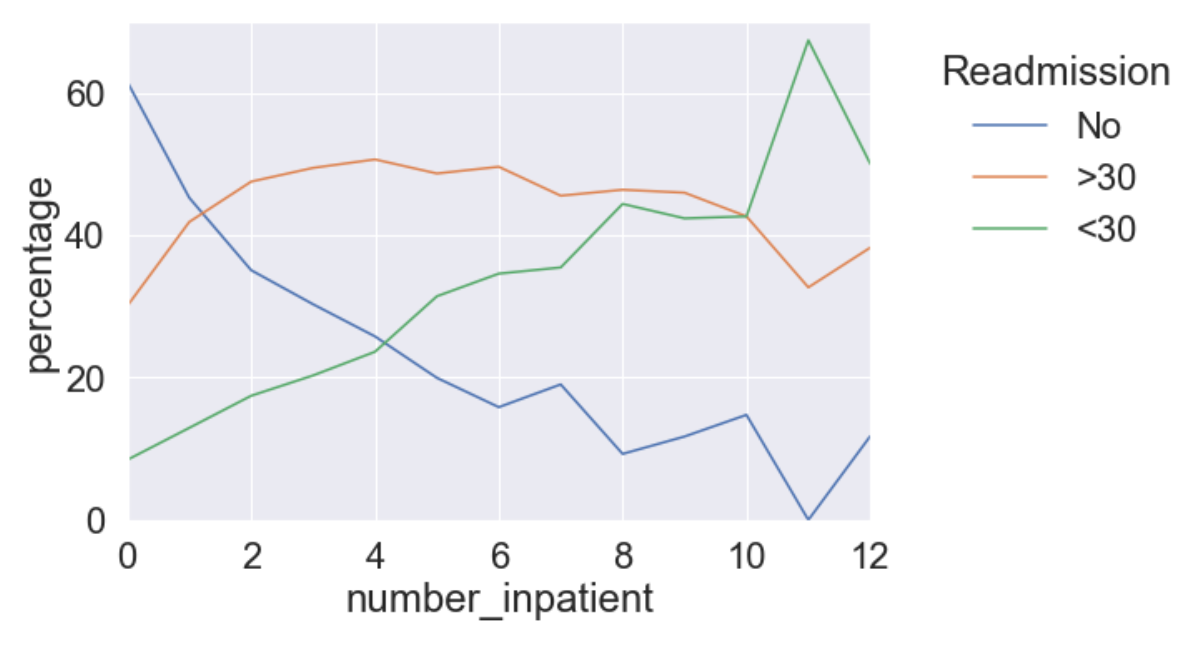
1. **Conclusions**

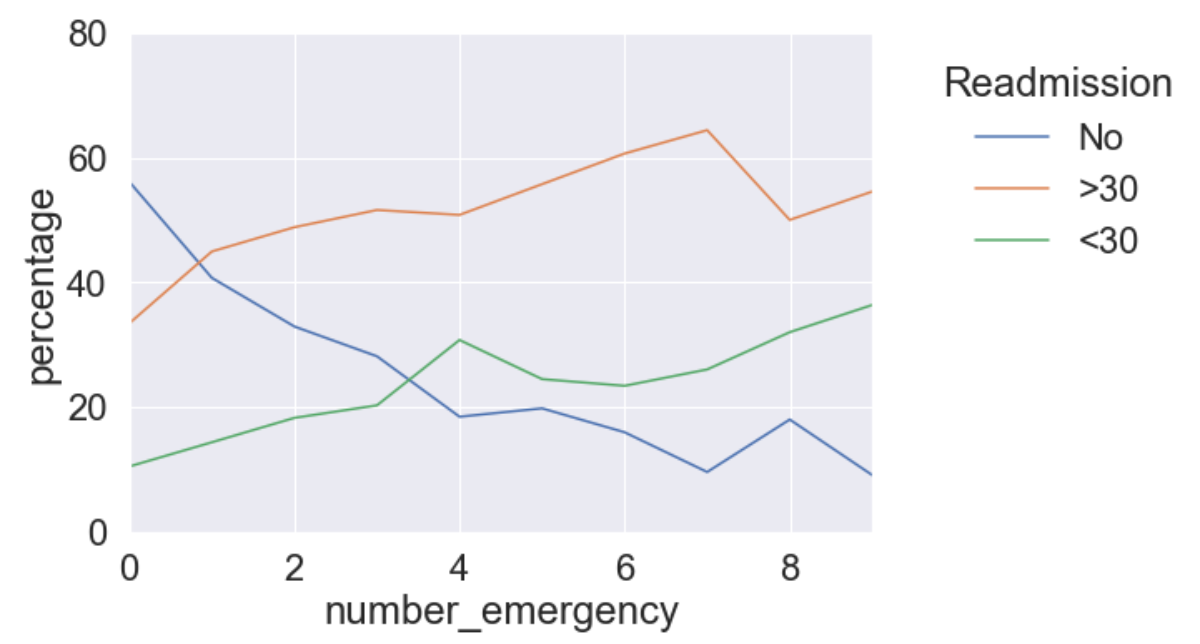
Show summary table of metrics for training and validation to compare original model to re-sampled (SMOTE) model.

Important to discuss that while overall accuracy dropped with SMOTE resampling, the poor recall for Readmitted<30 days drastically improved with resampling

Most important feature ranking is very similar for original and resampled models

Discuss reasoning on why specific features may be most important





**Appendix**

1. **Variable descriptions**

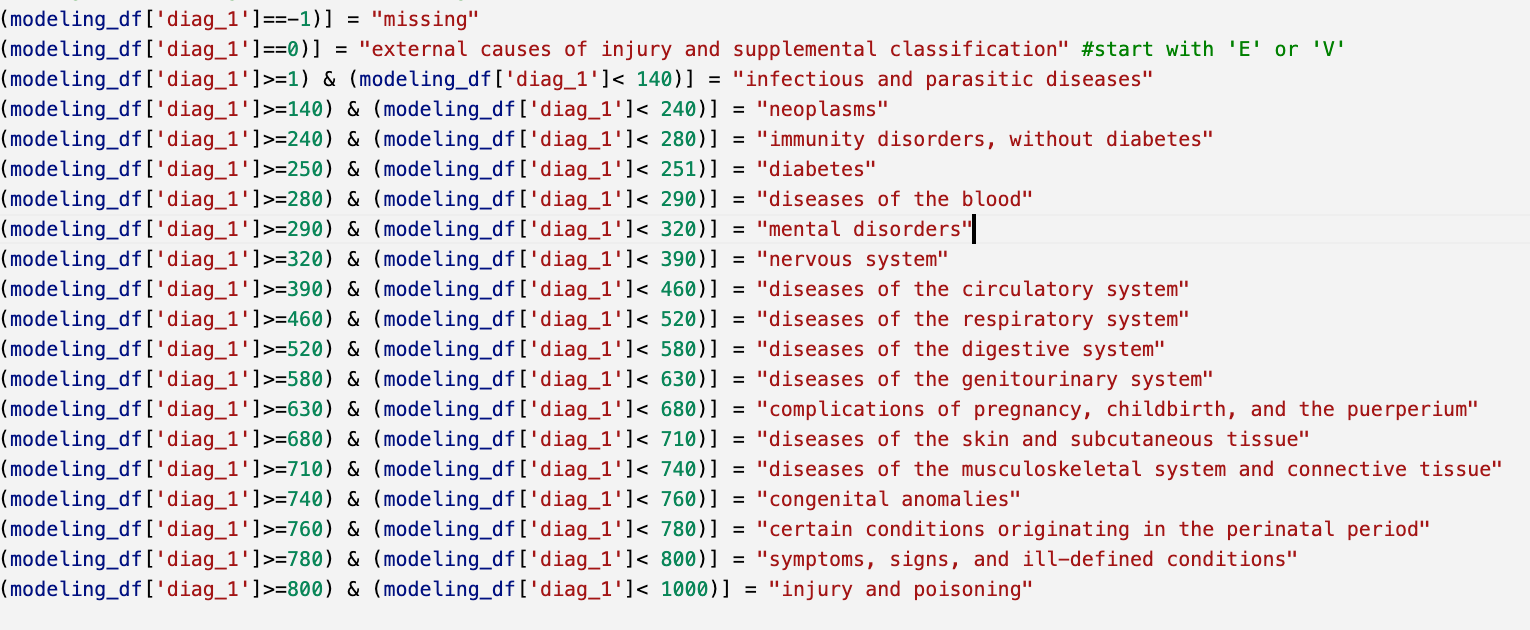






1. **Diagnosis categories and descriptions**

We utilized the diagnosis code and descriptions here <https://en.wikipedia.org/wiki/List_of_ICD-9_codes> to create the rules below for grouping and reducing the categories for features diag\_1 , diag\_2, and diag\_3:



1. **Code**

A rendered notebook containing code for this analysis can be accessed at: