# Evaluating A Mixed-Ensemble Method for Predicting Hospital Readmissions Across Multiple Disease Types

By

Phillip R. Salm

Thesis Project
Submitted in partial fulfillment of the
Requirements for the degree of

MASTER OF SCIENCE IN PREDICTIVE ANALYTICS

June, 2018

Dr. Lawrence Van Fulton, First Reader

Dr. Nathaniel Bastian, Second Reader

## Abstract

*Objective*

Hospitals are under significant financial pressure to reduce the rate of readmissions that occur in less than 30-days from discharge. Traditional predictive models could flag potential patients at-risk for early readmission, but the lack of transparency from these models meant they provided no insight for care teams on where to direct interventions to prevent the readmission. Turgeman and May demonstrated an alternative approach ensembling highly sensitive SVM models with transparent C5.0 decision tree models on data for patients with Congestive Heart Failure. This paper evaluates that approach against data from patients with acute myocardial infarction, chronic obstructive pulmonary disease, coronary artery bypass graft, pneumonia and stroke.

*Materials and Methods*

Data for this study was acquired from the Nationwide Readmission Database, providing national anonymized data on readmissions that occurred in 2014. This data was subset by disease type, and Logistic Regression, Random Tree, Neural Network and Ensemble models were built and evaluated for each.

*Results*

The results ran contrary to those of Turgeman and May, who found ensembles achieved the highest sensitivity of 0.91 against their training data, and 0.25 against their test data. The Random Tree models in this study had better sensitivity scores than all ensemble models, ranging from 0.62 to 0.94, and therefore were better able to flag patients at-risk. Additionally, the Random Tree decision rules provided transparency, but the actual clinical utility of that transparency is suspect and requires further investigation.

Table of Contents

## Introduction

Reducing unplanned hospital readmissions within 30-days of discharge is a high priority for healthcare systems in the United States. This is because the Affordable Care Act uses this as a measure of the quality of care provided by a hospital, though some question whether this measure accurately reflects the quality of care (Kansagara, et al 1688). Regardless, healthcare systems that exceed the maximum level of readmissions for specific diseases are fined by a reduction in the Medicare payments that they receive, and so have a significant financial motivation to address this issue (Hadzikadic, et al. 2922).

To do so, hospitals are seeking effective ways to predict patients that are most at risk for readmission so that they can target limited resources on effective interventions (Kansagara, et al 1688). However, hospital systems not only want models that provide the best predictions, but they are seeking ways to understand what factors are the most significant predictors so that they can design effective clinical interventions.

Turgeman and May have recently shared results of a promising ensemble methodology that leveraged the transparency of C5.0 decision tree models with the higher sensitivity of support vector machine models. Their method was used specifically for predicting readmissions for Congestive Heart Failure (CHF) patients. Pourhomayoun et al. demonstrated similar positive results in their application of a multiple-model-classification system to predict readmissions for heart failure.

This paper examines whether Turgeman and May's ensemble methodology can effectively be extended to other disease types beyond CHF. Specifically, the objective of this study is to determine if their ensembling method can provide benefits over single-model

approaches when applied to predicting 30-day readmissions for other disease types monitored by the Center for Medicare and Medicaid Services (CMS), including:

- Chronic obstructive pulmonary disease (COPD),
- Acute myocardial infarction (AMI),
- Pneumonia (PN),
- Stroke, and
- Coronary artery bypass graft (CABG).

## Literature

Initial efforts to develop predictive models for readmissions focused on leveraging a single model type, such as a logistical regression model, to provide the best predictions. However, the most effective predictive models often require a trade-off in transparency of the factors impacting the predictions. This limits their value to clinical teams seeking insight into why the patients are at risk. Newer efforts have attempted to use ensemble techniques to leverage multiple modeling approaches in conjunction with one another (Pourhomayoun, et al. 106). This has resulted in improved predictions, while at the same time maintaining a level of explanatory transparency important for care providers.

Turgeman and May used data from 20,321 admissions of Congestive Heart Failure patients from 2006 through 2014. They claimed that most readmission models did not provide "clinically useful, interpretable rules that could explain the reasoning process behind their predictions (Turgeman and May 73)." While decision trees could provide actionable insights to care teams, they traditionally were outperformed by more advanced modeling methods, such as Support Vector Machine (SVM) models. They found that combining the C5.0 decision tree

model with the SVM model into and ensemble model provided greater sensitivity than using the C5.0 model alone, yet also provided highly intuitive transparency.

## Methods

### Data Source

The Agency for Healthcare Research and Quality (AHRQ) compiles the Nationwide Readmissions Database (NRD) as part of the Healthcare Cost and Utilization Project (HCUP). The NRD data is assembled by a collection of Federal, State and Industry partners, and the data pertains to the discharge of patients across the United States. This anonymized data is available for purchase by students and researchers that complete required data use training, and the data for 2014 was purchased for this study. Strict data usage requirements limit who the data can be shared with to protect patient confidentiality, and therefore the data files used in this study are not being provided publicly for review.

The NRD includes both clinical and nonclinical variables relevant to readmission analysis. It is separated into four data files—the core file, hospital file, severity measure file, and diagnosis and procedure groups file shown in the table below.

| File | Description | Records | Variables |
|---|---|---|---|
| Core | Primary data set includes demographic, diagnostic and logistical information | 14,894,613 | 148 |
| Hospital | Data about the hospital where care was provided | 2,048 | 12 |
| Severity Measure | Severity measures and comorbidity data on patients | 14,894,613 | 34 |
| Diagnosis & Procedure Groups | Body system, chronic condition and procedure class data | 14,894,613 | 80 |

When combined for analysis, there are 14,894,613 records and 274 total variables. Clinical data is coded in ICD-9 format that was in use in 2014 prior to the industry transition to ICD-10 coding. Patient demographic data, payment data, and information about the healthcare facility where treatment was performed is also captured. A data dictionary is included in Appendix A with an overview of all the data elements.

## Data Preparation

While the NRD data files are compiled for the explicit purpose of studying hospital readmission rates in the United States, there are four primary issues that needed to be addressed for this analysis. First, the data files are broken into four separate files that need to be combined for all variables to be included in the modeling process. Second, the NRD does not specifically document the number of days from the past discharge to the secondary readmission. Third, the NRD does not flag admissions that occur prior to readmissions that are less than 30 days from the time they were discharged. And fourth, analyzing the full combined data set required more computing power than available for this study, so it needed to be subset into separate data files specific to each disease type prior to analysis and modeling. The following process was used to address each of these issues.

Initial data processing was performed on a Microsoft Azure Data Science virtual machine configured with four virtual CPUs and 14 GB of memory. The DataScienceVM comes with Python 2.7.13 installed with Anaconda 4.3.1. PyCharm was added to this environment as a preferred IDE. To begin analysis, first all four of the data files were merged together (See Appendix B for the Python code used). An inner join was used for Core and Severity Measures data sets using the KEY_NRD and HOSP_NRD fields. The KEY_NRD is the NRD record

identified, and the HOSP_NRD is the NRD hospital identifier. Once these data sets were merged, the step was repeated using the new combined Core and Severity Measures data set, and joining it with the Diagnosis data set. Finally, the hospital details for each admission were added to the data set by merging the combined data set with the Hospital data set using an inner join on the HOSP_NRD and NRD_STRATUM fields.

Once the full data set was combined and filtered, an additional variable was created, DaysFromDischarge, to hold the value for the number of days it had been since the previous hospital visit had ended. To calculate this, the data was subset to only records that had non-unique NRD_VisitLink values, and therefore included more than one admission. Records were sorted by descending order based on the values of the NRD_VisitLink and NRD_DaysToEvent fields. This grouped all records for the same patient together, and ordered based on when they occurred. Next, the NRD_VisitLink value between the current record and the subsequent record were compared. If the values were not identical, the value of DaysFromDischarge was set to 0 since it was the original visit for that patient. If the values were identical, then the value of DaysFromDischarge was calculated by taking the NRD_DaysToEvent value and subtracting from it the combined values of the NRD_DaysToEvent and the LOS from the earlier admission. Essentially the NRD_DaysToEvent plus the LOS variable equals the discharge date for that event.

With the days since the previous discharge calculated for all of the records, then the records with the target condition could be flagged. The target records were all records where a subsequent readmission occurred in less than 30 days from the time of discharge. This may seem counter-intuitive. Why flag the records of an admission that occurred prior to a readmission that was less than 30 days from discharge? Shouldn't we just flag the record of

early readmission itself? Since the goal is to predict when a patient is at risk to be readmitted early, we want to identify patterns in the discharges that were before the early readmission. This would allow the information from the models to be shared with the care team prior to the patient being discharged so that they could potentially leverage this information to modify the care provided and reduce the risk.

The last step in the initial data preparation process was to subset the complete data set into separate data sets for each disease type to facilitate faster processing. The values of the DRG field were used to identify the records of the disease types of interest. These DRG values along with the size of the resulting data sets are specified below.

| Disease Type | DRG Codes | Number or Records |
|---|---|---|
| AMI | 280 – 285.x | 118,234 |
| CABG | 233 – 236.x | 68,891 |
| COPD | 190 – 192.x | 414,153 |
| Pneumonia | 193 – 195.x | 346,640 |
| Stroke | 61 – 63 | 14,722 |

The smaller data sets separated out by disease type were then used to create models from.

## General Modeling Approach

Models were built using IBM SPSS Modeler which provided a rapid way of iterating through different modeling approaches. Four different models were built for each disease type: Logistic Regression, C5.0 Decision Tree, SVM, and an Ensemble of the C5.0 and SVM modeling approaches. Logistic Regression models were intended to provide the baseline

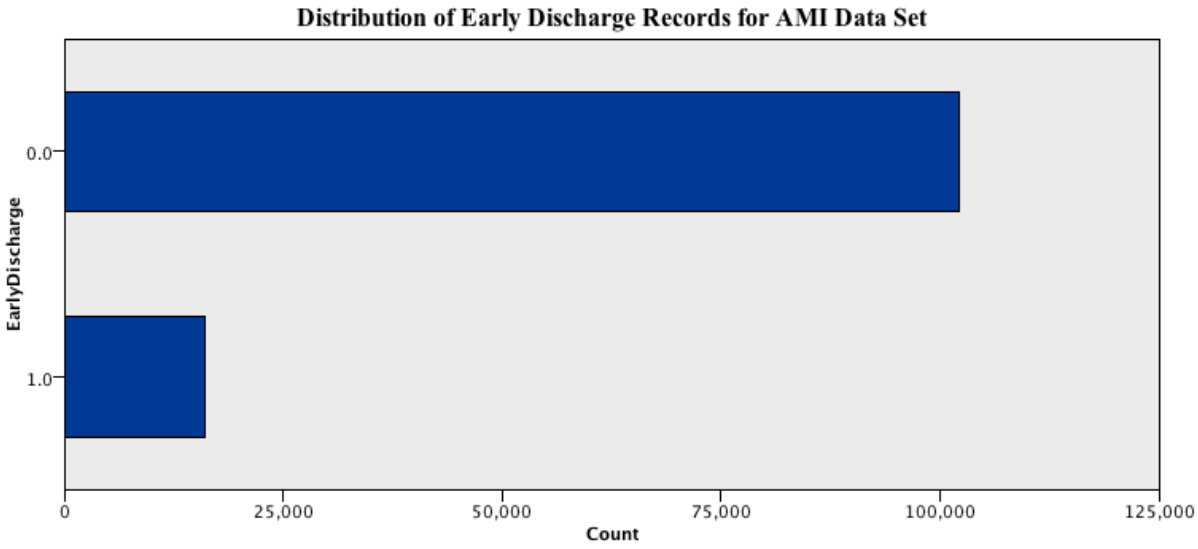
approach that the other models could be compared against. The SVM and C5.0 models, as well as their ensemble, were meant to follow the approach used by Turgeman and May. However, the SVM and C5.0 modeling approaches were not reliable, and these models often froze or simply kept processing for an extended period of time without completing. Only the Stroke data set—the smallest of the five—was able to reliably complete C5.0 and SVM models.

Because of their unreliability, these modeling methods were replaced by similar methods that provided reliable results for each data set. Random Tree models were used instead of C5.0 Decision Trees, and Neural Network models were used as an alternative 'black box' classifier method to replace the SVM models.

The modeling process steps used were identical for five disease types (See Appendix C for a snapshot of the Modeler Canvas). First the data file was loaded and all data types assigned were verified to match those specified by the NRD and corrected where they did not match. Values that represented missing data for each variable were specified following the documentation provided by the NRD. Record identifier variables were not included as predictors. All 'null' values for the EarlyDischarge variable were changed to '0' since they would belong to discharge records that did not have a subsequent readmission—early or otherwise. The initial 'null' assignment was a by-product of the data preparation.

Next, a data audit was performed on the data subset to review the characteristics of the data. Initial models were developed with no additional feature engineering. The data was split into training and test subsets using and 80/20 split. Then the five modeling approaches were applied. These models, however, simply predicted all records not to be early discharges. This is because the occurrences of early discharge records were rare and imbalanced in the data. For example, see the distribution of early discharge values for the AMI records in the graph below.

Early discharges were only 13.64% of the records. Thus, a model could achieve over 86% accuracy simply by predicting all records were not early discharges; however, this would result in 0% Positive Predictive Validity (PPV).




However, models with no sensitivity to true positives—to cases where the patient was being discharged too early—would ultimately have no practical use. The purpose of these models is to provide the best results for identifying patients at-risk for readmission—not to have the models with the highest overall accuracy. Models that failed to identify any at-risk patients could not be used to direct care teams to focus care to those individuals, and therefore could not help reduce early readmission rates. Thus, the best measure of a successful model was its sensitivity, and not it's accuracy.

The following dimensionality reduction rules were applied to improve the sensitivity of the models.

| Variable Drop Condition | Description |
|---|---|
| 01: Record identifiers | Variables that were either used to identify records of the same patient over time, or to relate the records across the multiple data files. |
| 02: Disease identifiers | These variables contained the DRG values that were used to subset the data set for the specific disease type of interest, so were not relevant for modeling. |
| 03: Single value present in all rows | Any variable that had a single value for all records was discarded since it could not be used to distinguish records that were early discharges from those that were not. These were typically comorbidity measures whose values were connected to the presence of the disease type being analyzed. |
| 04: Single category too large | Any categorical variable with more than 90% of records with the same value was removed. |
| 05: More than 50% or records with missing values | Any variable with more than 50% of its records with missing values was dropped. |
| 06: Pearson chi-square value less than .95 | Any variable where the Pearson chi-square value test for independence between it and the target is less than .95 was dropped. |
| 07: Nominal variable with over 100 unique values | Any nominal variable with more than 100 unique values was dropped. |

Applying these drop conditions resulted in a significant reduction in the variables used in the models. The variables dropped from 274 initial variables down to between 31 – 68 variables used in the final models. The specific variables dropped for each model are listed in the analyses for the diseases later in this paper.

For the variables that remained, the following transformations were applied. For continuous variables, outliers that were more than 3 standard deviations from the mean were replaced with a value equal to 3 standard deviations. These variables were then transformed into standard units with a mean of 0 and a final standard deviation of 1. Additionally, missing values

for continuous fields were replaced with the mean. For nominal fields with missing values, these were replaced with the mode. Finally, as was mentioned earlier, the frequency of records where the EarlyDischarge value was True was very small. To address this, the occurrence of records with a True EarlyDischarge value were boosted until they were approximately 25% - 30% of the total data set.

Once the data preparation steps were completed, the following modeling approaches were used:

1. Logistic regression models were created using backwards stepwise method for removing fields based on the Wald statistic.
2. Multilayer Perceptron (MLP) models were generated for the neural network models utilizing bagging (bootstrap aggregation). This ensemble approach across 10 component models was used to get the most reliable predictions from this model type. Voting was used as the combining rule across the multiple models, selecting the result based on the value with the highest probability that occurred most frequently across all base models.
3. Random Tree models were built to provide a decision tree classification method. As with the Neural Network models, bagging was also used in the construction of these models. A maximum of 100 component models could be built in the construction of the final Random Tree model.
4. Finally, an Ensemble model combining both the Random Tree and Neural Network models was constructed. Voting was used to determine the predicted output, and in cases where the models were at odds, the chosen value was randomly selected.

# Results

## AMI Analysis

*AMI Dimensionality Reduction*

The following table details the variables that were dropped from that AMI models.

| **Variable Drop Condition** | **Variables Dropped** |
|---|---|
| 01: Record identifiers | Key_NRD, NRD_DaysToEvent, NRD_VisitLink |
| 02: AMI identifiers | DRG, DRG_NoPOA |
| 03: Single value present in all rows | CM_CHF, CM_PULMCIRC, CM_VALVE, MDC, MC_NoPOA |
| 04: Single category too large | BODYSYSTEM1, CHRON1, CM_AIDS, CM_ALCOHOL, CM_ARTH, CM_BLDLOSS, CM_COAG, CM_DEPRESS, CM_DRUG, CM_LIVER, CM_LYMPH, CM_METS, CM_NEURO, CM_PARA, CM_PSYCH, CM_TUMOR, CM_ULCER, CM_WGHTLOSS, DIED, ELECTIVE, ORPROC, REHABTRANSFER, RESIDENT, SAMEDAYEVENT, SERVICELINE |
| 05: More than 50% or records with missing values | BODYSYSTEM15, BODYSYSTEM16, BODYSYSTEM19, BODYSYSTEM20, BODYSYSTEM21, BODYSYSTEM22, BODYSYSTEM23, BODYSYSTEM24, BODYSYSTEM25, BODYSYSTEM26, BODYSYSTEM27, BODYSYSTEM28, BODYSYSTEM29, BODYSYSTEM30, CHRON19, CHRON20, CHRON21, CHRON22, CHRON23, CHRON24, CHRON25, CHRON26, CHRON27, CHRON28, CHRON29, CHRON30, DX19, DX20, DX21, DX22, DX23, DX24, DX25, DX26, DX27, DX28, DX29, DX30, DXCCS19, DXCCS20, DXCCS21, DXCCS22, DXCCS23, DXCCS24, DXCCS25, DXCCS26, DXCCS27, DXCCS28, DXCCS29, DXCCS30, E_CCS1, E_CCS2, E_CCS3, E_CCS4, E_MCCS1, ECODE1, ECODE2, ECODE3, ECODE4, PCLASS10, PCLASS11, PCLASS12, PCLASS13, PCLASS14, PCLASS15, PCLASS3, PCLASS4, PCLASS5, PCLASS6, PCLASS7, PCLASS8, PCLASS9, PR10, PR11, PR12, PR13, PR14, PR15, PR4, PR5, PR6, PR7, PR8, PR9, PRCCS10, PRCCS11, PRCCS12, PRCCS13, PRCCS14, PRCCS15, PRCCS4, PRCCS5, PRCCS6, PRCCS7, PRCCS8, PRCCS9, PRDAY10, PRDAY11, PRDAY12, PRDAY13, PRDAY14, PRDAY15, PRDAY2, PRDAY3, PRDAY4, PRDAY5, PRDAY6, PRDAY7, PRDAY8, PRDAY9 |

| **Variable Drop Condition** | **Variables Dropped** |
|---|---|
| 06: Pearson chi-square value less than .95 | AWEEKEND, BODYSYSTEM17, BODYSYSTEM18, CHRON10, CHRON11, CHRON12, CHRON14, CHRON15, CHRON16, CHRON17, CHRON18, CHRON3, CHRON7, CHRON9, CM_OBESE, DISCWT, DX10, DX16, DX17, DX18, DXCCS17, FEMALE, HOSP_UR_TEACH, HOSP_URCAT4, N_HOSP_U, S_HOSP_U |
| 07: Nominal variable with over 100 unique values | DX1, DX11, DX12, DX13, DX14, DX15, DX2, DX3, DX4, DX5, DX6, DX7, DX8, DX9, DXCCS10, DXCCS11, DXCCS12, DXCCS13, DXCCS14, DXCCS15, DXCCS16, DXCCS18, DXCCS2, DXCCS3, DXCCS4, DXCCS5, DXCCS6, DXCCS7, DXCCS8, DXCCS9, HOSP_NRD, PR1, PR2, PR3, PRCCS1, PRCCS2, PRCCS3, PRMCCS1 |

*AMI Transformations*

After the dimensionality reduction, 56 predictor variables remained, along with the target variable. The following transformation were performed on these variables, where outliers were defined as any value more than 3 standard deviations.

| **Transformations** | **Variable** |
|---|---|
| Outliers removed, transformed to standard units | AGE, N_DISC_U, N_CHRONIC, NDX, NECODE, NPR, NRD_STRATUM, S_DISC_U, TOTAL_DISC, |
| Missing values replaced with mode. | BODYSYSTEM10, BODYSYSTEM11, BODYSYSTEM12, BODYSYSTEM13, BODYSYSTEM14, BODYSYSTEM2, BODYSYSTEM3, BODYSYSTEM4, BODYSYSTEM5, BODYSYSTEM6, BODYSYSTEM7, BODYSYSTEM8, BODYSYSTEM7, CHRON13, CHRON2, CHRON4, CHRON5, CHRON6, CHRON8, PAY1, PCLASS1, PCLASS2, PL_NCHS, ZIPINC_QRTL |
| Outliers removed, transformed to standard units, missing values replaced with mean. | LOS, PRDAY1, TOTCHG |

When looking at the distribution of the EarlyDischarge target variable, only 14% or the records resulted in subsequent readmissions within 30 days. To improve model performance, these records were boosted by a factor of 2.5 so that records with a positive value for EarlyReadmission then accounted for 29% of the total records.

*AMI Model Results*

The table below provides the accuracy measures for how each model performed against both the training and the test data sets. Unlike the results achieved by Turgeman and May, the Logistic Regression model provided the highest accuracy of all of the models—though the neural network model had nearly identical accuracy.

| | **Training** | | | **Test** | | |
|---|---|---|---|---|---|---|
| **Model** | **Correct** | **Wrong** | **AUC** | **Correct** | **Wrong** | **AUC** |
| Logistic Regression | 70.69% | 29.31% | 0.653 | 70.9% | 29.1% | 0.652 |
| Random Tree | 61.89% | 38.11% | 0.727 | 62.31% | 37.69% | 0.736 |
| Neural Network | 70.63% | 29.37% | 0.648 | 70.77% | 29.23% | 0.65 |
| Ensemble | 66.49% | 33.51% | 0.672 | 66.37% | 33.63% | 0.667 |

However, the Area Under the Curve (AUC) score of the Random Tree model was well ahead of the other models. Further examining the confusion matrixes of the models in the table below shows that the Random Tree model is far better at predicting true positives than any of the other models. As discussed earlier, sensitivity is the best success metric for evaluating these models since ultimately we want models that are best able to identify patients that are at-risk for early readmission.

For Turgeman and May, the intent of ensembling the decision tree with a black box model was to blend the better sensitivity of the black box model with the transparency of the decision tree model. But while the Ensemble model in this case was more accurate than Random

Tree model, since the Neural Network model was far less sensitive than the Random Tree model, the Ensemble was less sensitive than the Random Tree model.

| Model | True Positive | False Positive | True Negative | False Negative | Sensitivity | Specificity |
|---|---|---|---|---|---|---|
| Logistic Regression Train | 2,379 | 2,206 | 71,765 | 28,537 | 0.08 | 0.97 |
| Logistic Regression Test | 579 | 536 | 18,048 | 7,109 | 0.08 | 0.97 |
| Random Tree Train | 24,269 | 32,940 | 41,045 | 6,668 | 0.78 | 0.55 |
| Random Tree Test | 6,118 | 8,205 | 10,366 | 1,590 | 0.79 | 0.56 |
| NN Train | 99 | 68 | 73,926 | 30,718 | 0.00 | 1.00 |
| NN Test | 35 | 12 | 18,548 | 7,664 | 0.00 | 1.00 |
| Ensemble Train | 12,206 | 16,369 | 57,520 | 18,776 | 0.39 | 0.78 |
| Ensemble Test | 3,026 | 4,258 | 14,409 | 4,577 | 0.40 | 0.77 |

Thus, the results for the AMI models based on the NRD data indicate that using the Random Tree model on its own provided both the desired transparency and the best sensitivity over and above an Ensemble model. But let's take a closer look at the transparency provided by the Random Tree model. The following are the top decision rules from the AMI Random Tree model.

1. (LOS_x_transformed > -0.054865708959565385) and (CM_ANEMDEF_transformed = {0}) and (CM_RENLFAIL_transformed = {0}) and (APRDRG_Risk_Mortalit_transformed = {1,3,4})
2. (CM_RENLFAIL_transformed = {0}) and (LOS_x_transformed > -0.054865708959565385) and (NCHRONIC_transformed > -0.7616871035500913) and (DQTR <= 3) and (APRDRG_Severity_transformed = {2,4})

3. (BODYSYSTEM2_transformed = {0,3,4,5,6,7,9,10,11,13,14,16}) and (DQTR <= 1) and (LOS_x_transformed > -0.7245607725979278) and (BODYSYSTEM12_transformed = {0,1,2,3,4,6,7,8,9,10,11,12,13,14}) and (DQTR <= 3)
4. (DMONTH <= 4) and (DQTR <= 3) and (APRDRG_Risk_Mortalit_transformed = {1,3,4}) and (PAY1_transformed = {0,1,2,4,5}) and (CM_ANEMDEF_transformed = {0})
5. (BODYSYSTEM2_transformed = {0,1,2,6,7,8,10,11,13,14,15,16}) and (CM_CHRNLUNG_transformed = {0}) and (CM_RENLFAIL_transformed = {0}) and (DMONTH <= 9) and (NDX_transformed > -0.5508266257365806)

There are a few things to note upon review of the decision rules output from the Random Tree model. For example, there are several fields that are repeated across the different rules, including: APRDRG_Risk_Mortalit_transformed, CM_ANEMDEF_transformed and CM_RENLFAIL_transformed. So there might be some potential utility in sharing the influence of renal failure, anemia and mortality risk levels on predicting early readmission.

However, there are still unanswered questions as to whether this approach could provide useful information to care teams. Because some of the continuous variables in the rules have been transformed to improve model results, the values are not readily interpretable. Second, the time-related variables, such as DMONTH and DQTR, are not likely as relevant here since the data set the models were built on only included data for a single year (2014). Thus, any early month or early quarter is going to be more likely to result in a subsequent early readmission than a discharge later in the year since readmissions for those discharges might not occur until 2015.

Finally, any information provided to care teams must come with the strong caution that the model rules do not provide a *causal* claim that these factors *cause* early readmissions. This may

be difficult for someone not drilled on the mantra that correlation does not imply causation to consider. And for those that do grasp the mantra, the question may arise, then what value does this transparency ultimately provide? That is a fair question and beyond the scope of this project.

## CABG Analysis

*CABG Dimensionality Reduction*

The following table details the variables that were dropped from that CABG models.

| **Variable Drop Condition** | **Variables Dropped** |
|---|---|
| 01: Record identifiers | Key_NRD, NRD_DaysToEvent, NRD_VisitLink, HOSP_NRD |
| 02: AMI identifiers | DRG, DRG_NoPOA, DRGVER |
| 03: Single value present in all rows | CM_CHF, CM_PULMCIRC, CM_VALVE, MDC, MC_NoPOA, ORPROC |
| 04: Single category too large | BODYSYSTEM1, CHRON1, CM_AIDS, CM_ALCOHOL, CM_ARTH, CM_BLDLOSS, CM_DEPRESS, CM_DRUG, CM_LIVER, CM_LYMPH, CM_METS, CM_NEURO, CM_PARA, CM_PSYCH, CM_TUMOR, CM_ULCER, CM_WGHTLOSS, DIED, REHABTRANSFER, RESIDENT, SERVICELINE |

| **Variable Drop Condition** | **Variables Dropped** |
|---|---|
| 05: More than 50% or records with missing values | BODYSYSTEM15, BODYSYSTEM16, BODYSYSTEM17, BODYSYSTEM18, BODYSYSTEM19, BODYSYSTEM20, BODYSYSTEM21, BODYSYSTEM22, BODYSYSTEM23, BODYSYSTEM24, BODYSYSTEM25, BODYSYSTEM26, BODYSYSTEM27, BODYSYSTEM28, BODYSYSTEM29, BODYSYSTEM30, CHRON15, CHRON16, CHRON17, CHRON18, CHRON19, CHRON20, CHRON21, CHRON22, CHRON23, CHRON24, CHRON25, CHRON26, CHRON27, CHRON28, CHRON29, CHRON30, DX15, DX16, DX17, DX18, DX19, DX20, DX21, DX22, DX23, DX24, DX25, DX26, DX27, DX28, DX29, DX30, DXCCS15, DXCCS16, DXCCS17, DXCCS18, DXCCS19, DXCCS20, DXCCS21, DXCCS22, DXCCS23, DXCCS24, DXCCS25, DXCCS26, DXCCS27, DXCCS28, DXCCS29, DXCCS30, E_CCS1, E_CCS2, E_CCS3, E_CCS4, E_MCCS1, ECODE1, ECODE2, ECODE3, ECODE4, LOS_y, PCLASS10, PCLASS11, PCLASS12, PCLASS13, PCLASS14, PCLASS15, PCLASS7, PCLASS8, PCLASS9, PR10, PR11, PR12, PR13, PR14, PR15, PR7, PR8, PR9, PRCCS10, PRCCS11, PRCCS12, PRCCS13, PRCCS14, PRCCS15, PRCCS7, PRCCS8, PRCCS9, PRDAY10, PRDAY11, PRDAY12, PRDAY13, PRDAY14, PRDAY15, PRDAY7, PRDAY8, PRDAY9 |
| 06: Pearson chi-square value less than .95 | AWEEKEND, BODYSYSTEM17, BODYSYSTEM18, CHRON10, CHRON11, CHRON12, CHRON14, CHRON15, CHRON16, CHRON17, CHRON18, CHRON3, CHRON7, CHRON9, CM_OBESE, DISCWT, DX10, DX16, DX17, DX18, DXCCS17, FEMALE, HOSP_UR_TEACH, HOSP_URCAT4, N_HOSP_U, S_HOSP_U |
| 07: Nominal variable with over 100 unique values | DX1, DX2, DX3, DX4, DX5, DX6, DX7, DX8, DX9, DX10, DX11, DX12, DX13, DX14, DXCCS2, DXCCS3, DXCCS4, DXCCS5, DXCCS6, DXCCS7, DXCCS8, DXCCS9, DXCCS10, DXCCS11, DXCCS12, DXCCS13, DXCCS14, PR1, PR2, PR3, PR4, PR5, PR6, PRCCS1, PRCCS2, PRCCS3, PRCCS4, PRCCS5, PRCCS6, PRMCCS1 |

*CABG Transformations*

After the dimensionality reduction, 57 predictor variables remained, plus the target variable. The following transformation were performed on these variables, where outliers were defined as any value more than 3 standard deviations.

| **Transformations** | **Variable** |
|---|---|
| Outliers removed, transformed to standard units | AGE, DMONTH, LOS_X, NCHRONIC, NDX, N_HOSP_U, NCHRONIC, NDX, NECODE, NPR, TOTAL_DISC |
| Missing values replaced with mode. | BODYSYSTEM10, BODYSYSTEM12, BODYSYSTEM2, BODYSYSTEM3, BODYSYSTEM4, BODYSYSTEM5, BODYSYSTEM6, BODYSYSTEM7, BODYSYSTEM8, BODYSYSTEM9, CHRON8, DISPUNIFORM, ELECTIVE, PAY1, PCLASS1, PCLASS2, PCLASS3, PCLASS6, PL_NCHS, ZIPINC_QRTL |
| Outliers removed, transformed to standard units, missing values replaced with mean. | PRDAY1, PRDAY2, PRDAY3, PRDAY4, PRDAY5, PRDAY6, TOTCHG |

For the CABG data set, the EarlyDischarge variable was positive in only 8.3% of the records. So the positive EarlyDischarge records were boosted by a factor of 5.0. After boosting, the records with a positive value for EarlyReadmission accounted for 31.15% of the total records.

*CABG Model Results*

The accuracy measures for how each model performed against both the CABG training and the test data sets are shown in the table below. In this case, the Random Tree model outperformed all of the other models, both in accuracy as well as a significantly better AUC score.

| | Training | | | Test | | |
|---|---|---|---|---|---|---|
| **Model** | **Correct** | **Wrong** | **AUC** | **Correct** | **Wrong** | **AUC** |
| Logistic Regression | 69.83% | 30.17% | 0.671 | 69.47% | 30.53% | 0.658 |
| Random Tree | 77.65% | 22.35% | 0.856 | 72.78% | 27.22% | 0.801 |
| Neural Network | 70.84% | 29.16% | 0.69 | 70.24% | 29.76% | 0.671 |
| Ensemble | 74.31% | 25.69% | 0.778 | 71.29% | 28.71% | 0.729 |

As with the AMI models, the sensitivity score for the CABG Random Tree model was significantly better than the other models. Once again, these results run contrary to the results that Turgeman and May achieved with their models for Congestive Heart Failure. There is no improvement to sensitivity or to accuracy of the CABG Random Tree model when ensembling it with the CABG Neural Network model.

| **Model** | **True Positive** | **False Positive** | **True Negative** | **False Negative** | **Sensitivity** | **Specificity** |
|---|---|---|---|---|---|---|
| Logistic Regression Train | 3707 | 2984 | 46,900 | 18,881 | 0.16 | 0.94 |
| Logistic Regression Test | 918 | 818 | 11,658 | 4,709 | 0.16 | 0.93 |
| Random Tree Train | 17,808 | 11,418 | 38,466 | 4,780 | 0.79 | 0.77 |
| Random Tree Test | 4,117 | 3,418 | 9,059 | 1,510 | 0.73 | 0.73 |
| NN Train | 4812 | 3356 | 46,528 | 17,776 | 0.21 | 0.93 |
| NN Test | 1188 | 947 | 11,528 | 4,439 | 0.21 | 0.92 |
| Ensemble Train | 11344 | 7373 | 42,511 | 11,244 | 0.50 | 0.85 |
| Ensemble Test | 2643 | 2214 | 10,263 | 2,984 | 0.47 | 0.82 |

Let's take a closer look at the transparency provided by the CABG Random Tree model. The following are the top decision rules for identifying those at-risk for an early readmission from the Random Tree model.

1. (DISPUNIFORM_transformed = {3}) and (APRDRG_Severity_transformed = {1,2}) and (HCUP_ED_transformed = {0,1,2})

2. (BODYSYSTEM8_transformed = {1,3,4,6,7,9,11,13,14}) and (LOS_x_transformed > 0.8709167038934117) and (BODYSYSTEM12_transformed = {1,2,3,4,5,6,7,8,9,10,11,12,14}) and (LOS_x_transformed > 0.008034916812512428)
3. (HCUP_ED_transformed = {1,2}) and (CM_RENLFAIL_transformed = {0})
4. (BODYSYSTEM9_transformed = {2,3,4,5,7,8,10,11,14,16}) and (PRDAY2_transformed > 0.0) and (CM_RENLFAIL_transformed = {0})
5. (BODYSYSTEM8_transformed = {0,4,5,7,10,11,13,14,15}) and (LOS_x_transformed > 0.8709167038934117) and (TOTCHG_transformed > 1.3241809618745265E-4) and (DMONTH_transformed <= 1.0066619018883864) and (NCHRONIC_transformed > -0.145853767913513)

In this model, the length of stay variable recurs in several of the top decision rules. However, since it has been transformed to standard units, it would be difficult to use it to provide guidance for care teams. The HCUP_ED variable appears in two of the top rules, so prior admission to the emergency department may be relevant to predicting early readmission. But it is difficult to imagine how this could provide insight to care teams regarding what additional treatment a patient would need. So, while there is transparency to the CABG Random Tree model, it does not appear that this transparency would provide any utility to care teams. The BODYSYSTEMx variables might provide some additional insight, but would require interpretation by someone knowledgeable in the ICD-9 codes, along with a subsequent review with clinical care providers.

## COPD Analysis

### *COPD Dimensionality Reduction*

The COPD models dropped the variables from analysis outlined in the following table.

| **Variable Drop Condition** | **Variables Dropped** |
|---|---|
| 01: Record identifiers | Key_NRD, NRD_DaysToEvent, NRD_VisitLink, HOSP_NRD |
| 02: AMI identifiers | DRG, DRG_NoPOA, DRGVER |
| 03: Single value present in all rows | MDC, MC_NoPOA |
| 04: Single category too large | BODYSYSTEM1, CM_AIDS, CM_ALCOHOL, CM_ARTH, CM_BLDLOSS, CM_COAG, CM_DMCX, CM_DRUG, CM_LIVER, CM_LYMPH, CM_METS, CM_NEURO, CM_PARA, CM_PERIVASC, CM_PSYCH, CM_PULMCIRC, CM_TUMOR, CM_ULCER, CM_VALVE, CM_WGHTLOSS, DIED, ELECTIVE, ORPROC, REHABTRANSFER, RESIDENT, SAMEDAYEVENT, SERVICELINE |
| 05: More than 50% or records with missing values | BODYSYSTEM13, BODYSYSTEM14, BODYSYSTEM15, BODYSYSTEM16, BODYSYSTEM17, BODYSYSTEM18, BODYSYSTEM19, BODYSYSTEM20, BODYSYSTEM21, BODYSYSTEM22, BODYSYSTEM23, BODYSYSTEM24, BODYSYSTEM25, BODYSYSTEM26, BODYSYSTEM27, BODYSYSTEM28, BODYSYSTEM29, BODYSYSTEM30, CHRON13, CHRON14, CHRON15, CHRON16, CHRON17, CHRON18, CHRON19, CHRON20, CHRON21, CHRON22, CHRON23, CHRON24, CHRON25, CHRON26, CHRON27, CHRON28, CHRON29, CHRON30, DX10, DX11, DX13, DX14, DX15, DX16, DX17, DX18, DX19, DX20, DX21, DX22, DX23, DX24, DX25, DX26, DX27, DX28, DX29, DX30, DXCCS13, DXCCS14, DXCCS15, DXCCS16, DXCCS17, DXCCS18, DXCCS19, DXCCS20, DXCCS21, DXCCS22, DXCCS23, DXCCS24, DXCCS25, DXCCS26, DXCCS27, DXCCS28, DXCCS29, DXCCS30, E_CCS1, E_CCS2, E_CCS3, E_CCS4, E_MCCS1, ECODE1, ECODE2, ECODE3, ECODE4, PCLASS1, PCLASS10, PCLASS11, PCLASS12, PCLASS13, PCLASS14, PCLASS15, PCLASS2, PCLASS3, PCLASS4, PCLASS5, PCLASS6, PCLASS7, PCLASS8, PCLASS9, PR1, PR10, PR11, PR12, PR13, PR14, PR15, PR2, PR3, PR4, PR5, PR6, PR7, PR8, PR9, PRCCS1, PRCCS10, PRCCS11, PRCCS12, PRCCS13, PRCCS14, PRCCS15, PRCCS2, PRCCS3, PRCCS4, PRCCS5, PRCCS6, PRCCS7, PRCCS8, PRCCS9, PRDAY1, PRDAY10, PRDAY11, PRDAY12, PRDAY13, PRDAY14, PRDAY15, PRDAY2, PRDAY3, PRDAY4, PRDAY5, PRDAY6, PRDAY7, PRDAY8, PRDAY9, PRMCCS1 |
| 06: Pearson chi-square value less than .95 | AWEEKEND, CHRON4, S_HOSP_U, CM_CHRNLUNG |

| Variable Drop Condition | Variables Dropped |
|---|---|
| 07: Nominal variable with over 100 unique values | DX2, DX3, DX4, DX5, DX6, DX7, DX8, DX9, DX12, DXCCS2, DXCCS3, DXCCS4, DXCCS5, DXCCS6, DXCCS7, DXCCS8, DXCCS9, DXCCS10, DXCCS11, DXCCS12 |

*COPD Transformations*

There were 63 predictor variables that remained after the dimensionality reduction, plus the target variable. The following transformation were performed on these variables, where outliers were defined as any value more than 3 standard deviations.

| Transformations | Variable |
|---|---|
| Outliers removed, transformed to standard units | AGE, DISCWT, DMONTH, LOS_X, N_DISC_U, N_HOSP_U, NCHRONIC, NDX, NECODE, NPR, NRD_STRATUM, S_DISC_U, TOTAL_DISC |
| Missing values replaced with mode. | BODYSYSTEM2, BODYSYSTEM3, BODYSYSTEM4, BODYSYSTEM5, BODYSYSTEM6, BODYSYSTEM7, BODYSYSTEM8, BODYSYSTEM9, BODYSYSTEM10, BODYSYSTEM11, BODYSYSTEM12, CHRON2, CHRON3, CHRON5, CHRON6, CHRON7, CHRON8, CHRON9, CHRON10, CHRON11, CHRON12, PAY1, PL_NCHS, ZIPING_QRTL |
| Outliers removed, transformed to standard units, missing values replaced with mean. | LOS_Y, TOTCHG |

The positive EarlyDischarge records comprised 18.27% of the data set. For this data set, the EarlyDischarge records were boosted by a factor of 2.0, resulting in a final distribution where they comprised 30.9% of the records.

*COPD Model Results*

The COPD accuracy measures for how each model performed against both the training and the test data sets are shown in the table below. The COPD model performance was not

consistent with the performance evidenced against the AMI and CABG data sets. For COPD, the Neural Network model outperformed all other models, both in terms of accuracy and the highest AUC scores.

| | Training | | | Test | | |
|---|---|---|---|---|---|---|
| **Model** | **Correct** | **Wrong** | **AUC** | **Correct** | **Wrong** | **AUC** |
| Logistic Regression | 68.29% | 31.71% | 0.632 | 68.38% | 31.62% | 0.634 |
| Random Tree | 59.89% | 40.11% | 0.654 | 57.9% | 42.1% | 0.622 |
| Neural Network | 67.41% | 32.59% | 0.696 | 67.41% | 32.59% | 0.698 |
| Ensemble | 63.68% | 36.32% | 0.603 | 62.61% | 37.39% | 0.58 |

But interestingly, the AUC score of the Ensemble model at 0.58 was not improved over the AUC score of the Random Tree model at 0.622. And in looking at the sensitivity of the models, the Random Tree again outperformed all the other models by a significant margin. The Neural Network model, despite having the best accuracy and best AUC, had the worst sensitivity score at 0.00. Ensembling the COPD Random Tree model with the Neural Network model only resulted in a lower sensitivity than the Random Tree model alone.

| **Model** | **True Positive** | **False Positive** | **True Negative** | **False Negative** | **Sensitivity** | **Specificity** |
|---|---|---|---|---|---|---|
| Logistic Regression Train | 4,793 | 7,855 | 257,939 | 114,124 | 0.04 | 0.97 |
| Logistic Regression Test | 1,227 | 1,969 | 64,616 | 28,474 | 0.04 | 0.97 |
| Random Tree Train | 76,902 | 112,299 | 153,495 | 42,015 | 0.65 | 0.58 |
| Random Tree Test | 18,324 | 29,159 | 37,428 | 11,377 | 0.62 | 0.56 |
| NN Train | 40 | 6,513 | 259,277 | 118,875 | 0.00 | 0.98 |
| NN Test | 6 | 1,688 | 64,897 | 29,695 | 0.00 | 0.97 |
| Ensemble Train | 38,407 | 59,223 | 206,571 | 80,510 | 0.32 | 0.78 |
| Ensemble Test | 9,127 | 15,426 | 51,161 | 20,574 | 0.31 | 0.77 |

The following are the top decision rules from the COPD Random Tree model.

1. (DISPUNIFORM_transformed = {3}) and (APRDRG_Severity_transformed = {1,2}) and (HCUP_ED_transformed = {0,1,2})
2. (BODYSYSTEM8_transformed = {1,3,4,6,7,9,11,13,14}) and (LOS_x_transformed > 0.8709167038934117) and (BODYSYSTEM12_transformed = {1,2,3,4,5,6,7,8,9,10,11,12,14}) and (LOS_x_transformed > 0.008034916812512428)
3. (HCUP_ED_transformed = {1,2}) and (CM_RENLFAIL_transformed = {0})
4. (BODYSYSTEM9_transformed = {2,3,4,5,7,8,10,11,14,16}) and (PRDAY2_transformed > 0.0) and (CM_RENLFAIL_transformed = {0})
5. (BODYSYSTEM8_transformed = {0,4,5,7,10,11,13,14,15}) and (LOS_x_transformed > 0.8709167038934117) and (TOTCHG_transformed > 1.3241809618745265E-4) and (DMONTH_transformed <= 1.0066619018883864) and (NCHRONIC_transformed > -0.145853767913513)

Like the CABG decision rules, the length of stay variable appears in multiple decision rules. The renal failure variable also reappears. Lastly, BODYSYSTEM8 appears in multiple rules and might be of interest to the clinical care team.

## PNU Analysis

### *PNU Dimensionality Reduction*

The variables in the table below were dropped from the PNU analysis.

| **Variable Drop Condition** | **Variables Dropped** |
|---|---|
| 01: Record identifiers | Key_NRD, NRD_DaysToEvent, NRD_VisitLink, HOSP_NRD |
| 02: AMI identifiers | DRG, DRG_NoPOA, DRGVER |
| 03: Single value present in all rows | CHRON1 |

| **Variable Drop Condition** | **Variables Dropped** |
|---|---|
| 04: Single category too large | APRDRG, BODYSYSTEM1, CM_AIDS, CM_ALCOHOL, CM_ARTH, CM_BLDLOSS, CM_COAG, CM_DMCX, CM_DRUG, CM_LIVER, CM_LYMPH, CM_METS, CM_PARA, CM_PERIVASC, CM_PSYCH, CM_PULMCIRC, CM_TUMOR, CM_ULCER, CM_VALVE, CM_WGHTLOSS, DIED, DXCCS1, ELECTIVE, ORPROC, REHABTRANSFER, RESIDENT, SAMEDAYEVENT, SERVICELINE |
| 05: More than 50% or records with missing values | BODYSYSTEM12, BODYSYSTEM13, BODYSYSTEM14, BODYSYSTEM15, BODYSYSTEM16, BODYSYSTEM17, BODYSYSTEM18, BODYSYSTEM19, BODYSYSTEM20, BODYSYSTEM21, BODYSYSTEM22, BODYSYSTEM23, BODYSYSTEM24, BODYSYSTEM25, BODYSYSTEM26, BODYSYSTEM27, BODYSYSTEM28, BODYSYSTEM29, BODYSYSTEM30, CHRON12, CHRON13, CHRON14, CHRON15, CHRON16, CHRON17, CHRON18, CHRON19, CHRON20, CHRON21, CHRON22, CHRON23, CHRON24, CHRON25, CHRON26, CHRON27, CHRON28, CHRON29, CHRON30, DX10, DX11, DX12, DX13, DX14, DX15, DX16, DX17, DX18, DX19, DX20, DX21, DX22, DX23, DX24, DX25, DX26, DX27, DX28, DX29, DX30, DXCCS12, DXCCS13, DXCCS14, DXCCS15, DXCCS16, DXCCS17, DXCCS18, DXCCS19, DXCCS20, DXCCS21, DXCCS22, DXCCS23, DXCCS24, DXCCS25, DXCCS26, DXCCS27, DXCCS28, DXCCS29, DXCCS30, E_CCS1, E_CCS2, E_CCS3, E_CCS4, E_MCCS1, ECODE1, ECODE2, ECODE3, ECODE4, LOS_y, PCLASS1, PCLASS10, PCLASS11, PCLASS12, PCLASS13, PCLASS14, PCLASS15, PCLASS2, PCLASS3, PCLASS4, PCLASS5, PCLASS6, PCLASS7, PCLASS8, PCLASS9, PR1, PR10, PR11, PR12, PR13, PR14, PR15, PR2, PR3, PR4, PR5, PR6, PR7, PR8, PR9, PRCCS1, PRCCS10, PRCCS11, PRCCS12, PRCCS13, PRCCS14, PRCCS15, PRCCS2, PRCCS3, PRCCS4, PRCCS5, PRCCS6, PRCCS7, PRCCS8, PRCCS9, PRDAY1, PRDAY10, PRDAY11, PRDAY12, PRDAY13, PRDAY14, PRDAY15, PRDAY2, PRDAY3, PRDAY4, PRDAY5, PRDAY6, PRDAY7, PRDAY8, PRDAY9, PRMCCS1 |
| 06: Pearson chi-square value less than .95 | AWEEKEND, N_HOSP_U, S_HOSP_U, |
| 07: Nominal variable with over 100 unique values | DX2, DX3, DX4, DX5, DX6, DX7, DX8, DX9, DXCCS2, DXCCS3, DXCCS4, DXCCS5, DXCCS6, DXCCS7, DXCCS8, DXCCS9, DXCCS10, DXCCS11 |

*PNU Transformations*

There were 59 predictor variables that remained after the dimensionality reduction, plus the target variable. The table below outlines the transformations that were then applied.

| **Transformations** | **Variable** |
|---|---|
| Outliers removed, transformed to standard units | AGE, DISCWT, DMONTH, N_DISC_U, NCHRONIC, NDX, NECODE, NPR, NRD_STRATUM, S_DISC_U, TOTAL_DISC |
| Missing values replaced with mode. | BODYSYSTEM2, BODYSYSTEM3, BODYSYSTEM4, BODYSYSTEM5, BODYSYSTEM6, BODYSYSTEM7, BODYSYSTEM8, BODYSYSTEM9, BODYSYSTEM10, BODYSYSTEM11, CHRON2, CHRON3, CHRON4, CHRON5, CHRON6, CHRON7, CHRON8, CHRON9, CHRON10, CHRON11, DISPUNIFORM, PAY1, PL_NCHS |
| Outliers removed, transformed to standard units, missing values replaced with mean. | LOS_X, TOTCHG |

For the PNU data set, the EarlyDischarge records made up 13.51% of the cases. After boosting by a factor of 3.0, they comprised 31.9% of the records.

*PNU Model Results*

The PNU accuracy measures for how each model performed against both the training and the test data sets are shown in the table below.

| | **Training** | | | **Test** | | |
|---|---|---|---|---|---|---|
| **Model** | **Correct** | **Wrong** | **AUC** | **Correct** | **Wrong** | **AUC** |
| Logistic Regression | 68.15% | 31.85% | 0.66 | 68.03% | 31.96% | 0.659 |
| Random Tree | 59.74% | 40.26% | 0.704 | 58.17% | 41.83% | 0.675 |
| Neural Network | 68.11% | 31.89% | 0.646 | 68% | 32% | 0.646 |
| Ensemble | 63.88% | 36.12% | 0.644 | 63.35% | 36.65% | 0.629 |

For the PNU data sets, the Logistic Regression model was more accurate than all other models for the first time, though with nearly equivalent accuracy scores achieved by the Neural

Network model. Both were significantly more accurate than the Random Tree model, though the Random Tree model had the best AUC score of all of the models.

| Model | True Positive | False Positive | True Negative | False Negative | Sensitivity | Specificity |
|---|---|---|---|---|---|---|
| Logistic Regression Train | 13,477 | 13,360 | 221,419 | 96,414 | 0.12 | 0.94 |
| Logistic Regression Test | 3,299 | 3,284 | 55,414 | 24,306 | 0.12 | 0.94 |
| Random Tree Train | 85,137 | 114,026 | 120,753 | 24,754 | 0.77 | 0.51 |
| Random Tree Test | 20,835 | 29,332 | 29,366 | 6,770 | 0.75 | 0.50 |
| NN Train | 172 | 182 | 234,597 | 109,719 | 0.00 | 1.00 |
| NN Test | 38 | 53 | 58,645 | 27,567 | 0.00 | 1.00 |
| Ensemble Train | 42,623 | 57,234 | 177,545 | 67,268 | 0.39 | 0.76 |
| Ensemble Test | 10,550 | 14,577 | 44,121 | 17,055 | 0.38 | 0.75 |

But once again, the Random Tree model far outperformed all of the other models on the basis of sensitivity. The Neural Network model again provided no sensitivity, and so Ensembling it with the Random Tree model would provide no utility.

The following were to top decision rules for the PNU Random Tree model.

1. (CM_ANEMDEF_transformed = {0}) and (AGE_transformed <= 0.6539579122867222) and (DISPUNIFORM_transformed = {1,2,3,4})
2. (CM_ANEMDEF_transformed = {0}) and (AGE_transformed <= 0.6539579122867222) and (DISPUNIFORM_transformed = {1,2,3,4})
3. (NCHRONIC_transformed > 0.39084095350927206) and (CM_ANEMDEF_transformed = {0}) and (DISPUNIFORM_transformed = {1,2,3,4})
4. (NDX_transformed > 0.240433404741478) and (BODYSYSTEM10_transformed = {0,1,3,4,5,6,7,8,9,10,11,12,13,15,16}) and (CM_CHRNLUNG_transformed = {0}) and

(DISPUNIFORM_transformed = {1,2,3,4}) and (AGE_transformed > -1.3069622635743416)

5. (CM_CHF_transformed = {0}) and (DX1_transformed = {02,03,07,09,10,15,17,18,25,26}) and (AGE_transformed <= 0.6539579122867222) and (DISPUNIFORM_transformed = {1,2,3,4})

These rules had a lot of commonality with the variables of interest. Age appears repeatedly in these rules (though transformed). The DISPUNIFORM variable indicating where the patient was being discharged was present in all of the rules and could merit further investigation into the post-discharge care of pneumonia patients. Anemic deficiencies once again appear in multiple decision rules, as it had with the AMI models.

## Stroke Analysis

*Stroke Dimensionality Reduction*

The variables in the table below were dropped from the Stroke analysis.

| **Variable Drop Condition** | **Variables Dropped** |
|---|---|
| 01: Record identifiers | Key_NRD, NRD_DaysToEvent, NRD_VisitLink, HOSP_NRD |
| 02: AMI identifiers | DRG, DRG_NoPOA, DRGVER |
| 03: Single value present in all rows | BODYSYSTEM1, CHRON1, CM_NEURO, DXCCS1, DXMCCS1, MDC, MC_NoPOA |
| 04: Single category too large | APRDRG, CM_AIDS, CM_ALCOHOL, CM_ARTH, CM_BLDLOSS, CM_COAG, CM_DMCX, CM_DRUG, CM_LIVER, CM_LYMPH, CM_METS, CM_PERIVASC, CM_PSYCH, CM_PULMCIRC, CM_TUMOR, CM_ULCER, CM_WGHTLOSS, DIED, ELECTIVE, ORPROC, RESIDENT, SERVICELINE |

| **Variable Drop Condition** | **Variables Dropped** |
|---|---|
| 05: More than 50% or records with missing values | BODYSYSTEM14, BODYSYSTEM15, BODYSYSTEM16, BODYSYSTEM17, BODYSYSTEM18, BODYSYSTEM19, BODYSYSTEM20, BODYSYSTEM21, BODYSYSTEM22, BODYSYSTEM23, BODYSYSTEM24, BODYSYSTEM25, BODYSYSTEM26, BODYSYSTEM27, BODYSYSTEM28, BODYSYSTEM29, BODYSYSTEM30, CHRON14, CHRON15, CHRON16, CHRON17, CHRON18, CHRON19, CHRON20, CHRON21, CHRON22, CHRON23, CHRON24, CHRON25, CHRON26, CHRON27, CHRON28, CHRON29, CHRON30, DX14, DX15, DX16, DX17, DX18, DX19, DX20, DX21, DX22, DX23, DX24, DX25, DX26, DX27, DX28, DX29, DX30, DXCCS14, DXCCS15, DXCCS16, DXCCS17, DXCCS18, DXCCS19, DXCCS20, DXCCS21, DXCCS22, DXCCS23, DXCCS24, DXCCS25, DXCCS26, DXCCS27, DXCCS28, DXCCS29, DXCCS30, E_CCS1, E_CCS2, E_CCS3, E_CCS4, E_MCCS1, ECODE1, ECODE2, ECODE3, ECODE4, LOS_y, PCLASS10, PCLASS11, PCLASS12, PCLASS13, PCLASS14, PCLASS15, PCLASS2, PCLASS3, PCLASS4, PCLASS5, PCLASS6, PCLASS7, PCLASS8, PCLASS9, PR10, PR11, PR12, PR13, PR14, PR15, PR2, PR3, PR4, PR5, PR6, PR7, PR8, PR9, PRCCS10, PRCCS11, PRCCS12, PRCCS13, PRCCS14, PRCCS15, PRCCS2, PRCCS3, PRCCS4, PRCCS5, PRCCS6, PRCCS7, PRCCS8, PRCCS9, PRDAY10, PRDAY11, PRDAY12, PRDAY13, PRDAY14, PRDAY15, PRDAY2, PRDAY3, PRDAY4, PRDAY5, PRDAY6, PRDAY7, PRDAY8, PRDAY9 |
| 06: Pearson chi-square value less than .95 | AWEEKEND, BODYSYSTEM10, BODYSYSTEM11, BODYSYSTEM12, BODYSYSTEM13, BODYSYSTEM3, BODYSYSTEM6, BODYSYSTEM9, CHRON10, CHRON11, CHRON12, CHRON13, CHRON2, CHRON3, CHRON4, CHRON5, CHRON6, CHRON7, CHRON8, CHRON9, CM_HTN_C, CM_OBESE, DISCWT, DX1, DX10, DX11, DX12, DX13, DX3, DX7, DX8, DX9, DXCCS11, DXCCS12, DXCCS13, DXCCS3, DXCCS9, H_CONTRL, HCUP_ED, HOSP_BEDSIZE, HOSP_NRD, HOSP_UR_TEACH, HOSP_URCAT4, N_DISC_U, N_HOSP_U, NECODE, NRD_STRATUM, PCLASS1, PL_NCHS, REHABTRANSFER, S_DISC_U, S_HOSP_U, SAMEDAYEVENT |
| 07: Nominal variable with over 100 unique values | DX2, DX4, DX5, DX6, DXCCS2, DXCCS4, DXCCS5, DXCCS6, DXCCS7, DXCCS8, DXCCS10, PR1, PRCCS1, PRMCCS1 |

*Stroke Transformations*

There were only 31 predictor variables that remained after the dimensionality reduction of the Stroke data set, plus the target variable. After those variables were dropped, the following transformations were applied to the remaining variables.

| **Transformations** | **Variable** |
|---|---|
| Outliers removed, transformed to standard units | AGE, DMONTH, LOS_X, NCHRONIC, NDX, NPR, PAY1, TOTAL_DISC, |
| Missing values replaced with mode. | BODYSYSTEM2, BODYSYSTEM4, BODYSYSTEM5, BODYSYSTEM7, BODYSYSTEM8, DISPUNIFORM, ZIPINC_QRTL |
| Outliers removed, transformed to standard units, missing values replaced with mean. | PRDAY1, TOTCHG, |

Only 9.56% of the cases in the Stroke data set had a positive value for the EarlyDischarge variable. So these records were boosted by a factor of 3.0, after which they comprised 24.08% of the data set.

*Stroke Model Results*

The Stroke accuracy measures for how each model performed against both the training and the test data sets are shown in the table below.

| | **Training** | | | **Test** | | |
|---|---|---|---|---|---|---|
| **Model** | **Correct** | **Wrong** | **AUC** | **Correct** | **Wrong** | **AUC** |
| Logistic Regression | 76.02% | 23.98% | 0.676 | 76.42% | 23.58% | 0.676 |
| Random Tree | 85.8% | 14.2% | 0.96 | 78.83% | 21.17% | 0.883 |
| Neural Network | 96.12% | 3.88% | 0.972 | 89.75% | 10.25% | 0.893 |
| Ensemble | 90.92% | 9.08% | 0.969 | 84.04% | 15.96% | 0.889 |

For this data set, the Neural Network model was far more accurate than any other model type, and it also had the best AUC scores. And for the first time, the Ensemble of the Random

Tree and Neural Network models had a higher AUC score than the Random Tree model did on its own. But once again, the Random Tree model had the best sensitivity score of all the models, including the Ensemble model.

| Model | True Positive | False Positive | True Negative | False Negative | Sensitivity | Specificity |
|---|---|---|---|---|---|---|
| Logistic Regression Train | 208 | 188 | 9,802 | 2,970 | 0.07 | 0.98 |
| Logistic Regression Test | 47 | 38 | 2,429 | 726 | 0.06 | 0.98 |
| Random Tree Train | 3,001 | 1,693 | 8,297 | 177 | 0.94 | 0.83 |
| Random Tree Test | 653 | 566 | 1,901 | 120 | 0.84 | 0.77 |
| NN Train | 2,685 | 18 | 9,972 | 493 | 0.84 | 1.00 |
| NN Test | 495 | 54 | 2,413 | 278 | 0.64 | 0.98 |
| Ensemble Train | 2,852 | 869 | 9,121 | 326 | 0.90 | 0.91 |
| Ensemble Test | 568 | 312 | 2,155 | 205 | 0.73 | 0.87 |

The following decision rules were the top rules for the Stroke data set.

1. (DISPUNIFORM_transformed = {3,4}) and (BODYSYSTEM7_transformed = {0,1,2,4,5,7,8,9,10,11,12,13,15}) and (LOS_x_transformed > -0.2918108965545197) and (DMONTH_transformed <= 0.9376914555601276) and (NCHRONIC_transformed > -0.17142521620032952)
2. (BODYSYSTEM4_transformed = {0,1,2,4,6,7,12,14,15,16,17}) and (BODYSYSTEM5_transformed = {1,2,3,4,5,9,11}) and (BODYSYSTEM8_transformed = {0,1,6,7,8,10,11,12,14}) and (DQTR <= 3)
3. (PAY1_transformed = {0,1,3,5}) and (DMONTH_transformed <= -0.5028994818991886) and (NCHRONIC_transformed > -0.17142521620032952) and (APRDRG_Severity_transformed = {1,3}) and (DQTR <= 3)

4. (NPR_transformed > -0.6774811067207236) and (CM_RENLFAIL_transformed = {0}) and (APRDRG_Severity_transformed = {0,2,3}) and (DQTR <= 3) and (PAY1_transformed = {0,3,5})
5. (BODYSYSTEM7_transformed = {0,2,7,9,10,11,13,15}) and (LOS_x_transformed > 0.00918840280975509) and (APRDRG_Risk_Mortalit_transformed = {0,1,3}) and (BODYSYSTEM8_transformed = {2,3,5,13,14,15}) and (DMONTH_transformed <= 0.07333689308453782)

The severity of the patient illness as coded in the APRDRG_Severity variable appears in multiple decision rules, as does the number of chronic conditions (NCHRONIC). Length of stay is also present in multiple rules again. While these variables may be relevant to predicting whether a patient is at-risk for an early readmission, prima facie they do not appear to have any clinical utility. There is nothing a care team care team can do, for example, about the severity of the stroke or the number of chronic conditions the patient currently has.

*Additional Stroke Models*

While there were issues using the SVM and C5.0 models for the other disease data sets, these models were able to complete for the Stroke data set. It's not clear if this is because the size of the Stroke data set was smaller than the other data sets, or if there was something else causing issues with the use of those models. But since Turgeman and May specifically used these model types in their analysis, it is worth examining if changing the model types to match their approach would lead to results that better matched their outcomes.

The table below provides the accuracy metrics for a C5.0 decision tree model, and SVM model, and an ensemble of both as applied to the Stroke data set.

| | Training | | | Test | | |
|---|---|---|---|---|---|---|
| **Model** | **Correct** | **Wrong** | **AUC** | **Correct** | **Wrong** | **AUC** |
| C5.0 Tree | 95.88% | 4.12% | 0.99 | 92.07% | 7.93% | 0.959 |
| SVM | 95.53% | 4.47% | 0.983 | 84.66% | 15.34% | 0.888 |
| Ensemble | 96.96% | 3.04% | 0.95 | 89.85% | 10.15% | 0.953 |

From an accuracy standpoint, all three models performed extremely well. The Ensemble model was slightly more accurate than the individual models on the training data set, while the C5.0 decision tree model was more accurate against the test data. The C.50 model also had the best AUC score on both the training and test data sets. So there wasn't a clear benefit to the ensembling of the C5.0 decision tree and SVM models from an accuracy standpoint.

| **Model** | **True Positive** | **False Positive** | **True Negative** | **False Negative** | **Sensitivity** | **Specificity** |
|---|---|---|---|---|---|---|
| SVM Train | 2,798 | 209 | 9,781 | 380 | 0.88 | 0.98 |
| SVM Test | 553 | 277 | 2,190 | 220 | 0.72 | 0.89 |
| C5.0 Train | 2,667 | 32 | 9,958 | 511 | 0.84 | 1.00 |
| C5.0 Test | 597 | 81 | 2,386 | 176 | 0.77 | 0.97 |
| Ensemble Train | 2,844 | 66 | 9,924 | 334 | 0.89 | 0.99 |
| Ensemble Test | 576 | 132 | 2,335 | 197 | 0.75 | 0.95 |

When looking at the model sensitivity though, the Ensemble model had the highest sensitivity score against the training data set. However, the C5.0 model had a slightly higher sensitivity score against the test data set. Given the mixed results, there was not a clear benefit to the ensembling approach with regard to model sensitivity either. In fact, all of these models had lower sensitivity scores than the Random Tree model examined earlier. Thus the model results did not provide support for Turgeman and May's methodology even when the approach followed them more closely by using SVM and C5.0 decision tree models.

## Conclusions

Turgeman and May's mixed-ensemble method for predicting hospital readmissions did not outperform other modeling approaches when applied to multiple disease types using 2014 data from the Nationwide Readmission Database—specifically data sets for AMI, CABG, COPD, PNU and Stroke. This study substituted Neural Network "black box" classification models for SVM models and Random Tree models for C5.0 decision trees. However, even when Turgeman and May's approach was followed and SVM and C5.0 models were used for ensembling, the SVM and C5.0 ensemble model was still outperformed by a Random Tree model. In fact, the Random Tree models had better sensitivity scores than other modeling approaches when used against all five disease data sets.

The Random Tree models could therefore be used on their own without ensembling with other model types to achieve the best sensitivity scores. Additionally, they can provide transparency for clinical teams through the decision rules that they output. However, the utility of this transparency to care teams is questionable. In order to gain any meaningful sensitivity from the models, many of the variables in the data sets were either dropped or transformed. The values of the transformed variables that appear in the decision rules would not be easily interpretable by clinical staff.

Additionally, variables that may be more clearly interpretable, such as length of stay or severity of disease, don't necessarily provide insights that could lead to actionable clinical interventions. In such cases, there is no advantage to the transparency achieved over a black box model that might outperform the Random Tree model (though no black box modeling approach outperformed the Random Tree models in this study).

Lastly, even if we are able to interpret the variables and potentially garner apparent clinical insight from a decision rule, there is a potential danger that these decision rules would be interpreted as causal claims. Clinical staff might then focus interventions on addressing the variables surfaced in the rules. But no causal inference can be made from the application of the Random Tree model. Researchers might find these rules valuable in directing future studies that causal inference could be inferred from, but the models on their own do not provide this. Thus the value that model transparency could provide to clinical care teams is in itself suspect.

*Limitations of this Study and Opportunities for Further Inquiry*

This study limited analysis to records from the 2014 Nationwide Readmission Database. Extending the study out to data that spans admissions from multiple years might yield additional insights into whether the time of year has an impact on readmissions.

Additionally, it would be valuable to have SVM and C5.0 models to compare against all of the disease data sets, rather than only the Stroke data set. While the results of those models applied to the Stroke data set were in-line with the results achieved by the substitute models used against the other data sets, it would provide a better evaluation of Turgeman and May's ensembling methodology. The ultimate cause for the problems encountered with those models in this study is not known, but improved computer processing resources might have resolved the issues.

Another line of inquiry that would be valuable would be the use of actual patient data from electronic medical record systems. That data with actual clinical measures and patient test information might yield decision rules that would be more meaningful to clinical care teams than those derived from ICD-9 coding. But even this approach would require addressing the larger

question about whether decision rule transparency can be used in any meaningful way by clinical care teams given they are not causal statements.

## References

Turgeman, L., & May, J. H. (2016). A mixed-ensemble model for hospital readmission. *Artificial Intelligence in Medicine,* vol 72, pp. 72-82.

Pourhomayoun, M., Alshurafa, N., Mortazavi, B., Ghasemzadeh, H., Sideris, K., Sadeghi, B., … Sarrafzadeh, M. (2014). Multiple Model Analytics for Adverse Event Prediction in Remote Health Monitoring Systems. *2014 Health Innovations and Point-of-Care Technologies Conference*. Seattle, WA USA, October 8-10, 2014.

Maddipatla, R., Hadzikadic, M., Misra, D., & Yao, Dr. L. (2015). 30 Day Hospital Readmission Analysis. *2015 IEEE International Conference on Big Data.* Santa Clara, CA USA, Oct 29-Nov 1, 2015.

Kansagara, D., Englander, H., Salanitro, A., Kagen, D., Theobald, C., Freeman, M., & Kripalani, S. (2011). Risk Prediction Models for Hospital Readmission: A Systematic Review. *JAMA*, vol 306:15, pp. 1688 – 1698.

Beata Strack, Jonathan P. DeShazo, Chris Gennings, et al., "Impact of HbA1c Measurement on Hospital Readmission Rates: Analysis of 70,000 Clinical Database Patient Records," BioMed Research International, vol. 2014, Article ID 781670, 11 pp. 2014. doi:10.1155/2014/781670

Diabetes 130-US hospitals for years 1999-2008 Data Set. UC Irvine Machine Learning Repository. Retrieved from https://archive.ics.uci.edu/ml/datasets/Diabetes+130-US+hospitals+for+years+1999-2008.

## Appendix A – Data Dictionary

The NRD is separated into four distinct flat files. The data dictionary for each file is provided below.

### Core

| Variable | Description | Value | Value Description |
|---|---|---|---|
| AGE | Age in years at admission | 0-124 | Age in years |
| | | . | Missing |
| | | .A | Invalid |
| | | .B | Unavailable from source (coded in 1988-1997 data only) |
| | | .C | Inconsistent: beginning with 1998 data, EAGE02, EAGE03, EAGE04, EAGE05; in 1988-1997 data, ED021, ED3nn, ED4nnn, ED5nn |
| AWEEKEND | Admission day is on a weekend | 0 | Admitted Monday-Friday |
| | | 1 | Admitted Saturday-Sunday |
| | | . | Missing |
| | | .A | Invalid |
| DIED | Died during hospitalization | 0 | Did not die |
| | | 1 | Died |
| | | . | Missing |
| | | .A | Invalid |
| | | .B | Unavailable from source (coded in 1988-1997 data only) |
| DISCWT | Weight to discharges in the universe | nn.nnnn | Weight to discharges in the universe. |
| DISPUNIFORM | Disposition of patient, uniform coding | 1 | Routine |
| | | 2 | Transfer to short-term hospital |
| | | 5 | Transfer other: includes Skilled Nursing Facility (SNF), Intermediate Care Facility (ICF), and another type of facility |
| | | 6 | Home Health Care (HHC) |
| | | 7 | Against medical advice (AMA) |
| | | 20 | Died in hospital |
| | | 21 | Discharged/transferred to court/law enforcement |
| | | 99 | Discharged alive, destination unknown, beginning in 2001 |
| | | . | Missing |
| | | .A | Invalid |
| DMONTH | Discharge month | 12-Jan | Discharge month |
| | | . | Missing |

| | | | |
|---|---|---|---|
| | | .A | Invalid |
| DQTR | Discharge quarter | 1 | First quarter (Jan - Mar) |
| | | 2 | Second quarter (Apr - Jun) |
| | | 3 | Third quarter (Jul - Sep) |
| | | 4 | Fourth quarter (Oct - Dec) |
| | | 0 | Missing or invalid |
| DRG | DRG in use on discharge date | nnn | DRG value |
| DRG_NoPOA | DRG in use on discharge date, calculated without POA | nnn | DRG value |
| DRGVER | DRG or MS-DRG grouper version used on discharge date | 4 | 4th revision, eff. Oct 1, 1987 |
| | | 5 | 5th revision, eff. Oct 1, 1988 |
| | | 6 | 6th revision, eff. Oct 1, 1989 |
| | | 7 | 7th revision, eff. Oct 1, 1990 |
| | | 9 | Version 9, eff. Oct 1, 1991 |
| | | 10 | Version 10, eff. Oct 1, 1992 |
| | | 11 | Version 11, eff. Oct 1, 1993 |
| | | 12 | Version 12, eff. Oct 1, 1994 |
| | | 13 | Version 13, eff. Oct 1, 1995 |
| | | 14 | Version 14, eff. Oct 1, 1996 |
| | | 15 | Version 15, eff. Oct 1, 1997 |
| | | 16 | Version 16, eff. Oct 1, 1998 |
| | | 17 | Version 17, eff. Oct 1, 1999 |
| | | 18 | Version 18, eff. Oct 1, 2000 |
| | | 19 | Version 19, eff. Oct 1, 2001 |
| | | 20 | Version 20, eff. Oct 1, 2002 |
| | | 21 | Version 21, eff. Oct 1, 2003 |
| | | 22 | Version 22, eff. Oct 1, 2004 |
| | | 23 | Version 23, eff. Oct 1, 2005 |
| | | 24 | Version 24, eff. Oct 1, 2006 |
| | | 25 | Version 25, eff. Oct 1, 2007 |
| | | 26 | Version 26, eff. Oct 1, 2008 |
| | | 27 | Version 27, eff. Oct 1, 2009 |
| | | 28 | Version 28, eff. Oct 1, 2010 |
| | | 29 | Version 29, eff. Oct 1, 2011 |
| | | 30 | Version 30, eff. Oct 1, 2012 |

| | | | |
|---|---|---|---|
| | | 31 | Version 31, eff. Oct 1, 2013 |
| | | 32 | Version 32, eff. Oct 1, 2014 |
| | | 33 | Version 33, eff. Oct 1, 2015 |
| DXn | ICD-9-CM | annnn | Diagnosis code |
| | | Blank | Missing |
| | | invl | Invalid: beginning with 1998 data, EDX02 |
| | | incn | Inconsistent: beginning with 1998 data, EAGE04, EAGE05, EDX03 |
| DXCCSn | Clinical Classifications Software (CCS): ICD-9-CM diagnosis classification | 1-259 | CCS Diagnosis Codes |
| | | 260 | CCS E-code Class (1988-1997 data) |
| | | 2601-2621 | CCS E-code Class (beginning with 1998 data) |
| | | . | No diagnosis code |
| | | .A | Invalid diagnosis code: beginning with 1998 data, EDX02 |
| | | .C | Inconsistent: beginning with 1998 data, EAGE04, EAGE05, EDX03 |
| E_CCSn | Clinical Classifications Software (CCS) for ICD-9-CM External Cause of Injury Code | 2601-2621 | CCS E-code Class (beginning with 1998 data) |
| | | . | No diagnosis code |
| | | .A | Invalid diagnosis code: beginning with 1998 data, EDX02 |
| | | .C | Inconsistent: beginning with 1998 data EDX03 |
| ECODEn | ICD-9-CM External Cause of Injury Code | E code | annnn |
| | | Blank | Missing |
| | | Invl | Invalid E code |
| ELECTIVE | Elective versus non-elective admission | 0 | Non-elective admission |
| | | 1 | Elective admission |
| | | . | Missing |
| | | .A | Invalid |
| FEMALE | Indicator of sex | 0 | Male |
| | | 1 | Female |
| | | . | Missing |

| | | | |
|---|---|---|---|
| | | .A | Invalid |
| | | .C | Inconsistent, EDX03, EPR03 |
| HCUP_ED | HCUP indicator of emergency department record | 0 | Record does not meet any HCUP Emergency Department criteria |
| | | 1 | Emergency Department revenue code on record |
| | | 2 | Positive Emergency Department charge (when revenue center codes are not available) |
| | | 3 | Emergency Department CPT procedure code on record |
| | | 4 | Condition code P7 indication of ED admission, point of origin of ED, or admission source of ED |
| HOSP_NRD | HCUP NRD hospital identification number | 5(n) | HCUP NRD hospital identifier |
| KEY_NRD | HCUP NRD record identifier | 14(n) | HCUP NRD record identifier |
| LOS | Length of stay, cleaned | 0 - 365 (for HCUP inpatient data), 0-3 (for HCUP outpatient data) | Days (In the 1988-1997 inpatient data, LOS can be greater than 365 days) |
| | | . | Missing |
| | | .A | Invalid |
| | | .B | Unavailable from source (coded in 1988-1997 data only) |
| | | .C | Inconsistent: beginning with 1998 data, ELOS03, ELOS04; in 1988-1997 data, ED011, ED601, ED911n, ED921 |
| MDC | Major Diagnostic Category in effect on discharge date | nn | MDC value |
| MDC_NoPOA | MDC in use on discharge date, calculated without POA | nn | MDC value |
| NCHRONIC | ICD-9-CM Number of chronic conditions | 0 - nn | ICD-9-CM Number of chronic conditions |
| | | .A | Invalid |
| NDX | Number of ICD-9-CM diagnoses on this discharge | 0 - nn | Number of diagnoses |
| NECODE | Number of ICD-9-CM External of Cause of Injury Codes on this Record | nn | Number of E codes |
| NPR | Number of ICD-9-CM procedures on this discharge | 0 - nn | Number of procedures |

| | | | |
|---|---|---|---|
| NRD_DaysToEvent | Days from "start date" to admission | nnnn | Timing between events |
| | | . | Missing |
| NRD_STRATUM | Stratum used to post-stratify hospital | 3(n) | Stratum number (masked) |
| NRD_VisitLink | Patient linkage variable in the NRD | 7(a) | Verified patient linkage number for linking hospital visits for the same patient across hospitals |
| | | . | Missing |
| ORPROC | Major operating room ICD-9-CM procedure indicator | 0 | No major operating room procedure reported on discharge record |
| | | 1 | Major operating room procedure reported on discharge record |
| | | .A | Invalid |
| PAY1 | Expected primary payer, uniform | 1 | Medicare |
| | | 2 | Medicaid |
| | | 3 | Private insurance |
| | | 4 | Self-pay |
| | | 5 | No charge |
| | | 6 | Other |
| | | . | Missing |
| | | .A | Invalid |
| | | .B | Unavailable from source (coded in 1988-1997 data only) |
| PL_NCHS | Patient Location: NCHS Urban-Rural Code | 1 | "Central" counties of metro areas of >=1 million population |
| | | 2 | "Fringe" counties of metro areas of >=1 million population |
| | | 3 | Counties in metro areas of 250,000-999,999 population |
| | | 4 | Counties in metro areas of 50,000-249,999 population |
| | | 5 | Micropolitan counties |
| | | 6 | Not metropolitan or micropolitan counties |
| | | . | Missing |
| PRn | ICD-9-CM Procedure | nnnn | Procedure code |
| | | Blank | Missing |
| | | invl | Invalid: beginning with 1998 data, EPR02 |
| | | incn | Inconsistent: beginning with EAGE05, EPR03 |
| PRCCSn | Clinical Classifications Software (CCS) | 1 - 231 | CCS procedure class |

| | | | |
|---|---|---|---|
| | for ICD-9-CM Procedures | | |
| | | . | No procedure code |
| | | .A | Invalid procedure code: beginning with 1998 data, EPR02 |
| | | .C | Inconsistent: beginning with 1998 data, EAGE05, EPR03 |
| PRDAYn | Number of days from admission to procedure n | -3 | Days prior to admission |
| | | 0 | Day of admission |
| | | 1 - LOS+3 | Days after admission (In rare instances in the 2008-2015 data, the maximum of LOS+3 was not enforced) |
| | | . | Missing |
| | | .A | Invalid |
| | | .B | Unavailable from source (coded in 1988-1997 data only) |
| | | .C | Inconsistent: beginning with 1998 data, EPRDAY01; in 1998-1997 data, ED7nn, ED8nn |
| REHABTRANSFER | A combined record involving transfer to rehabilitation, evaluation, or other aftercare | 0 | Not a combined record or a combined record not involving rehabilitation, evaluation, or other aftercare |
| | | 1 | Combined record involving transfer to rehabilitation, evaluation, or other aftercare |
| RESIDENT | Identifies patient as a resident of the State in which he or she received hospital care | 0 | Nonresident |
| | | 1 | Resident |
| SAMEDAYEVENT | Identifies transfer and same-day stay collapsed records | 0 | Not a transfer or other same-day stay |
| | | 1 | Transfer involving two discharges from different hospitals |
| | | 2 | Same-day stay involving two discharges from different hospitals |
| | | 3 | Same-day stay involving two discharges at the same hospital |
| | | 4 | Same-day stay involving three or more discharges at the same or different hospitals |
| SERVICELINE | Service line based on ICD-9 codes | 1 | 1: Maternal and neonatal |
| | | 2 | 2: Mental health/substance use |
| | | 3 | 3: Injury |
| | | 4 | 4: Surgical |
| | | 5 | 5: Medical |
| | | .A | Invalid |

| | | | |
|---|---|---|---|
| TOTCHG | Total charges, cleaned | Dollars | Total Charge rounded |
| | | . | Missing |
| | | .A | Invalid |
| | | .B | Unavailable from source (coded in 1988-1997 data only) |
| | | .C | Inconsistent: beginning with 1998 data, ETCHG01, ETCHG02; in 1998-1997 data, ED911, ED912, ED921, ED922 |
| YEAR | Calendar year | yy | 2-digit calendar year in 1988-1997 data |
| | | yyyy | 4-digit calendar year beginning with 1998 data |
| ZIPINC_QRTL | Median household income for patient's ZIP Code (based on current year) | 1 | 0-25th percentile |
| | | 2 | 26th to 50th percentile (median) |
| | | 3 | 51st to 75th percentile |
| | | 4 | 76th to 100th percentile |
| | | . | Other (includes ZIP equal blank A, C, M, F and B) |

## Severity

| Variable | Description | Value | Value Description |
|---|---|---|---|
| APRDRG | All Patient Refined DRG | nnn | APRDRG |
| | | . | Missing |
| APRDRG_Risk_Mortality | All Patient Refined DRG: Risk of Mortality Subclass | 0 | No class specified |
| | | 1 | Minor likelihood of dying |
| | | 2 | Moderate likelihood of dying |
| | | 3 | Major likelihood of dying |
| | | 4 | Extreme likelihood of dying |
| | | . | Missing |

| | | | |
|---|---|---|---|
| APRDRG_Severity | All Patient Refined DRG: Severity of Illness Subclass | 0 | No class specified |
| | | 1 | Minor loss of function (includes cases with no comorbidity or complications) |
| | | 2 | Moderate loss of function |
| | | 3 | Major loss of function |
| | | 4 | Extreme loss of function |
| | | . | Missing |

| | | | |
|---|---|---|---|
| CM_AIDS | | 0 | Comorbidity is not present |

| | | | |
|---|---|---|---|
| | AHRQ comorbidity measure for ICD-9-CM codes: acquired immune deficiency syndrome | 1 | Comorbidity is present |
| | | .A | Invalid |
| CM_ALCOHOL | AHRQ comorbidity measure for ICD-9-CM codes: alcohol abuse | 0 | Comorbidity is not present |
| | | 1 | Comorbidity is present |
| | | .A | Invalid |
| CM_ANEMDEF | AHRQ comorbidity measure for ICD-9-CM codes: deficiency anemias | 0 | Comorbidity is not present |
| | | 1 | Comorbidity is present |
| | | .A | Invalid |
| CM_ARTH | AHRQ comorbidity measure for ICD-9-CM codes: rheumatoid arthritis/collagen vascular diseases | 0 | Comorbidity is not present |
| | | 1 | Comorbidity is present |
| | | .A | Invalid |
| CM_BLDLOSS | AHRQ comorbidity measure for ICD-9-CM codes: chronic blood loss anemia | 0 | Comorbidity is not present |
| | | 1 | Comorbidity is present |
| | | .A | Invalid |
| CM_CHF | AHRQ comorbidity measure for ICD-9-CM codes: congestive heart failure | 0 | Comorbidity is not present |
| | | 1 | Comorbidity is present |
| | | .A | Invalid |
| CM_CHRNLUNG | AHRQ comorbidity measure for ICD-9-CM codes: chronic pulmonary disease | 0 | Comorbidity is not present |
| | | 1 | Comorbidity is present |
| | | .A | Invalid |
| CM_COAG | AHRQ comorbidity measure for ICD-9-CM codes: coagulopathy | 0 | Comorbidity is not present |
| | | 1 | Comorbidity is present |
| | | .A | Invalid |
| CM_DEPRESS | AHRQ comorbidity measure for ICD-9-CM codes: depression | 0 | Comorbidity is not present |
| | | 1 | Comorbidity is present |
| | | .A | Invalid |
| CM_DM | AHRQ comorbidity measure for ICD-9-CM codes: diabetes, uncomplicated | 0 | Comorbidity is not present |
| | | 1 | Comorbidity is present |
| | | .A | Invalid |
| CM_DMCX | | 0 | Comorbidity is not present |

| | | | |
|---|---|---|---|
| | AHRQ comorbidity measure for ICD-9-CM codes: diabetes with chronic complications | 1 | Comorbidity is present |
| | | .A | Invalid |
| CM_DRUG | AHRQ comorbidity measure for ICD-9-CM codes: drug abuse | 0 | Comorbidity is not present |
| | | 1 | Comorbidity is present |
| | | .A | Invalid |
| CM_HTN_C | AHRQ comorbidity measure for ICD-9-CM codes: hypertension (combine uncomplicated and complicated) | 0 | Comorbidity is not present |
| | | 1 | Comorbidity is present |
| | | .A | Invalid |
| CM_HYPOTHY | AHRQ comorbidity measure for ICD-9-CM codes: hypothyroidism | 0 | Comorbidity is not present |
| | | 1 | Comorbidity is present |
| | | .A | Invalid |
| CM_LIVER | AHRQ comorbidity measure for ICD-9-CM codes: liver disease | 0 | Comorbidity is not present |
| | | 1 | Comorbidity is present |
| | | .A | Invalid |
| CM_LYMPH | AHRQ comorbidity measure for ICD-9-CM codes: lymphoma | 0 | Comorbidity is not present |
| | | 1 | Comorbidity is present |
| | | .A | Invalid |
| CM_LYTES | AHRQ comorbidity measure for ICD-9-CM codes: fluid and electrolyte disorders | 0 | Comorbidity is not present |
| | | 1 | Comorbidity is present |
| | | .A | Invalid |
| CM_METS | AHRQ comorbidity measure for ICD-9-CM codes: metastatic cancer | 0 | Comorbidity is not present |
| | | 1 | Comorbidity is present |
| | | .A | Invalid |
| CM_NEURO | AHRQ comorbidity measure for ICD-9-CM codes: other neurological disorders | 0 | Comorbidity is not present |
| | | 1 | Comorbidity is present |
| | | .A | Invalid |
| CM_OBESE | AHRQ comorbidity measure for ICD-9-CM codes: obesity | 0 | Comorbidity is not present |
| | | 1 | Comorbidity is present |
| | | .A | Invalid |

| | | | |
|---|---|---|---|
| CM_PARA | AHRQ comorbidity measure for ICD-9-CM codes: paralysis | 0 | Comorbidity is not present |
| | | 1 | Comorbidity is present |
| | | .A | Invalid |
| CM_PERIVASC | AHRQ comorbidity measure for ICD-9-CM codes: peripheral vascular disorders | 0 | Comorbidity is not present |
| | | 1 | Comorbidity is present |
| | | .A | Invalid |
| CM_PSYCH | AHRQ comorbidity measure for ICD-9-CM codes: psychoses | 0 | Comorbidity is not present |
| | | 1 | Comorbidity is present |
| | | .A | Invalid |
| CM_PULMCIRC | AHRQ comorbidity measure for ICD-9-CM codes: pulmonary circulation disorders | 0 | Comorbidity is not present |
| | | 1 | Comorbidity is present |
| | | .A | Invalid |
| CM_RENLFAIL | AHRQ comorbidity measure for ICD-9-CM codes: renal failure | 0 | Comorbidity is not present |
| | | 1 | Comorbidity is present |
| | | .A | Invalid |
| CM_TUMOR | AHRQ comorbidity measure for ICD-9-CM codes: solid tumor without metastasis | 0 | Comorbidity is not present |
| | | 1 | Comorbidity is present |
| | | .A | Invalid |
| CM_ULCER | AHRQ comorbidity measure for ICD-9-CM codes: peptic ulcer disease excluding bleeding | 0 | Comorbidity is not present |
| | | 1 | Comorbidity is present |
| | | .A | Invalid |
| CM_VALVE | AHRQ comorbidity measure for ICD-9-CM codes: valvular disease | 0 | Comorbidity is not present |
| | | 1 | Comorbidity is present |
| | | .A | Invalid |
| CM_WGHTLOSS | AHRQ comorbidity measure for ICD-9-CM codes: weight loss | 0 | Comorbidity is not present |
| | | 1 | Comorbidity is present |
| | | .A | Invalid |
| HOSP_NRD | HCUP NRD hospital identification number | 5(n) | HCUP NRD hospital identifier |
| KEY_NRD | HCUP NRD record identifier | 14(n) | HCUP NRD record identifier |

## DX_PR

| Variable | Description | Value | Value Description |
|---|---|---|---|
| BODYSYSTEMn | ICD-9-CM Body system n | 1 | 1 = Infectious and parasitic disease |
| | | 2 | 2 = Neoplasms |
| | | 3 | 3 = Endocrine, nutritional, and metabolic diseases and immunity disorders |
| | | 4 | 4 = Diseases of blood and blood-forming organs |
| | | 5 | 5 = Mental disorders |
| | | 6 | 6 = Diseases of the nervous system and sense organs |
| | | 7 | 7 = Diseases of the circulatory system |
| | | 8 | 8 = Diseases of the respiratory system |
| | | 9 | 9 = Diseases of the digestive system |
| | | 10 | 10 = Diseases of the genitourinary System |
| | | 11 | 11 = Complications of pregnancy, childbirth, and the puerperium |
| | | 12 | 12 = Diseases of the skin and subcutaneous tissue |
| | | 13 | 13 = Diseases of the musculoskeletal system |
| | | 14 | 14 = Congenital anomalies |
| | | 15 | 15 = Certain conditions originating in the perinatal period |
| | | 16 | 16 = Symptoms, signs, and ill-defined conditions |
| | | 17 | 17 = Injury and poisoning |
| | | 18 | 18 = Factors influencing health status and contact with health services |
| | | . | Missing |
| | | .A | Invalid |
| | | .C | Inconsistent |

| | | | |
|---|---|---|---|
| CHRONn | Chronic Condition Indicators - body system | 1 | 1 = Infectious and parasitic disease |
| | | 2 | 2 = Neoplasms |
| | | 3 | 3 = Endocrine, nutritional, and metabolic diseases and immunity disorders |
| | | 4 | 4 = Diseases of blood and blood-forming organs |
| | | 5 | 5 = Mental disorders |
| | | 6 | 6 = Diseases of the nervous system and sense organs |
| | | 7 | 7 = Diseases of the circulatory system |

| | | | |
|---|---|---|---|
| | | 8 | 8 = Diseases of the respiratory system |
| | | 9 | 9 = Diseases of the digestive system |
| | | 10 | 10 = Diseases of the genitourinary System |
| | | 11 | 11 = Complications of pregnancy, childbirth, and the puerperium |
| | | 12 | 12 = Diseases of the skin and subcutaneous tissue |
| | | 13 | 13 = Diseases of the musculoskeletal system |
| | | 14 | 14 = Congenital anomalies |
| | | 15 | 15 = Certain conditions originating in the perinatal period |
| | | 16 | 16 = Symptoms, signs, and ill-defined conditions |
| | | 17 | 17 = Injury and poisoning |
| | | 18 | 18 = Factors influencing health status and contact with health services |
| | | . | Missing |
| | | .A | Invalid |
| | | .C | Inconsistent |

| | | | |
|---|---|---|---|
| DXMCCSn | Multi-Level CCS for ICD-9-CM Diagnoses | nn.nn.nn.nn | Multi_Level CCS value |
| | | blank | Missing |
| | | .A, A | Invalid |
| | | .C, C | Inconsistent |

| | | | |
|---|---|---|---|
| E_MCCSn | Multi-Level CCS for ICD-9-CM External Cause of Injury Code | nn.nn.nn.nn | Multi-Level CCS value |
| | | blank | Missing |
| | | .A, A | Invalid |
| | | .C, C | Inconsistent |

| | | | |
|---|---|---|---|
| HOSP_NRD | HCUP NRD hospital identification number | 5(n) | HCUP NRD hospital identifier |
| KEY_NRD | HCUP NRD record identifier | 14(n) | HCUP NRD record identifier |

| | | | |
|---|---|---|---|
| PCLASSn | ICD-9-CM Procedure class | 1 | Minor Diagnostic |
| | | 2 | Minor Therapeutic |
| | | 3 | Major Diagnostic |
| | | 4 | Major Therapeutic |
| | | . | Missing |
| | | .C | Inconsistent |

| | | | |
|---|---|---|---|
| | | .A | Invalid |

| | | | |
|---|---|---|---|
| PRMCCSn | Multi-Level CCS for ICD-9-CM Procedures | nn.nn.nn | Multi-Level CCS value |
| | | blank | Missing |
| | | .A, A | Invalid |
| | | .C, C | Inconsistent |

## Hospital

| Variable | Description | Value | Value Description |
|---|---|---|---|
| H_CONTRL | Control/ownership of hospital | 1 | Government, nonfederal |
| | | 2 | Private, not-profit |
| | | 3 | Private, invest-own |
| | | . | Missing |

| | | | |
|---|---|---|---|
| HOSP_BEDSIZE | Bedsize of hospital | 1 | Small |
| | | 2 | Medium |
| | | 3 | Large |
| | | . | Missing |

| | | | |
|---|---|---|---|
| HOSP_NRD | HCUP NRD hospital identification number | 5(n) | HCUP NRD hospital identifier |

| | | | |
|---|---|---|---|
| HOSP_URCAT4 | Hospital urban-rural designation | 1 | Large metropolitan areas with at least 1 million residents |
| | | 2 | Small metropolitan areas with less than 1 million residents |
| | | 3 | Micropolitan areas |
| | | 4 | Not metropolitan or micropolitan (non-urban residual) |
| | | 6 | Collapsed category for any urban-rural location (only applicable to the NEDS, beginning in 2014) |
| | | 7 | Collapsed category of small metropolitan and micropolitan, (only applicable to the NEDS, beginning in 2011) |
| | | 8 | Metropolitan, collapsed category of large and small metropolitan |
| | | 9 | Non-metropolitan, collapsed category of micropolitan and non-urban |

| | | | |
|---|---|---|---|
| HOSP_UR_TEACH | Teaching status of hospital | 0 | Metropolitan non-teaching |

| | | | |
|---|---|---|---|
| | | 1 | Metropolitan teaching |
| | | 2 | Non-metropolitan hospital |

| | | | |
|---|---|---|---|
| N_DISC_U | Number of discharges in the universe for the stratum | 7(n) | Number of discharges in the universe for the stratum |
| N_HOSP_U | Number of hospitals in the universe for the stratum | 3(n) | Number of hospitals in the universe for the stratum |
| S_DISC_U | Number of discharges in the sample for the stratum | 6(n) | Number of discharges in the sample for the stratum |
| S_HOSP_U | Number of hospitals in the sample for the stratum | nn | Number of hospitals in the sample for the stratum |
| TOTAL_DISC | Total hospital discharges | 5(n) | Total hospital discharges |

| | | | |
|---|---|---|---|
| YEAR | Calendar year | yy | 2-digit calendar year in 1988-1997 data |
| | | yyyy | 4-digit calendar year beginning with 1998 data |

## Appendix B – Python Code

Initial data preparation was performed using Python on a Microsoft Azure Data Science virtual machine to provide the processing power to work on the full data set. The code was separated into six work files:

- **load_nrd.py**: join NRD data files, select records with readmissions, drop invalid records.
- **id_readmits.py**: calculate days since discharge and identify readmissions in less than 30 days.
- **ami_records.py**: filter full data set for AMI records and attach early admission flag.
- **cabg_records.py**: filter full data set for CABG records and attach early admission flag.
- **copd_records.py**: filter full data set for COPD records and attach early admission flag.
- **pnu_records.py**: filter full data set for pneumonia records and attach early admission flag.
- **stroke_records.py**: filter full data set for stroke records and attach early admission flag.

**load_nrd.py**

```
import pandas as pd

file1 = 'NRD_2014_Core_mod.csv'
file2 = 'NRD_2014_Severity_mod.csv'
file3 = 'NRD_2014_DX_PR_GRPS_mod.csv'
file4 = 'NRD_2014_Hospital_mod.csv'
nrd_core = pd.read_csv(file1, dtype={'AGE':int, 'AWEEKEND':int,
'DIED':int, 'DISCWT':float, 'DISPUNIFORM':int, 'DMONTH':int,
'DQTR':int, 'DRG':int, 'DRG_NoPOA':int, 'DRGVER':int, 'DX1':str,
'DX2':str, 'DX3':str, 'DX4':str,'DX5':str, 'DX6':str, 'DX7':str,
'DX8':str, 'DX9':str, 'DX10':str, 'DX11':str, 'DX12':str, 'DX13':str,
'DX14':str, 'DX15':str, 'DX16':str, 'DX17':str, 'DX18':str,
'DX19':str, 'DX20':str, 'DX21':str,
'DX22':str, 'DX23':str, 'DX24':str, 'DX25':str, 'DX26':str,
'DX27':str, 'DX28':str, 'DX29':str,'DX30':str, 'DXCCS1':int,
'DXCCS2':int, 'DXCCS3':int, 'DXCCS4':int, 'DXCCS5':int, 'DXCCS6':int,
'DXCCS7':int, 'DXCCS8':int, 'DXCCS9':int, 'DXCCS10':int,
'DXCCS11':int, 'DXCCS12':int, 'DXCCS13':int,
'DXCCS14':int, 'DXCCS15':int, 'DXCCS16':int, 'DXCCS17':int,
'DXCCS18':int, 'DXCCS19':int, 'DXCCS20':int,'DXCCS21':int,
'DXCCS22':int, 'DXCCS23':int, 'DXCCS24':int, 'DXCCS25':int,
```

```
'DXCCS26':int, 'DXCCS27':int,'DXCCS28':int, 'DXCCS29':int,
'DXCCS30':int, 'ECODE1':str, 'ECODE2':str, 'ECODE3':str,
'ECODE4':str,'ELECTIVE':int, 'E_CCS1':int, 'E_CCS2':int, 'E_CCS3':int,
'E_CCS4':int, 'FEMALE':int, 'HCUP_ED':int, 'HOSP_NRD':int,
'KEY_NRD':int, 'LOS':int, 'MDC':int, 'MDC_NoPOA':int, 'NCHRONIC':int,
'NDX':int,'NECODE':int, 'NPR':int, 'NRD_DaysToEvent':int,
'NRD_STRATUM':int, 'NRD_VisitLink':str, 'ORPROC':int,'PAY1':int,
'PL_NCHS':int, 'PR1':str, 'PR2':str, 'PR3':str, 'PR4':str, 'PR5':str,
'PR6':str, 'PR7':str, 'PR8':str, 'PR9':str, 'PR10':str, 'PR11':str,
'PR12':str, 'PR13':str, 'PR14':str, 'PR15':str, 'PRCCS1':int,
'PRCCS2':int, 'PRCCS3':int, 'PRCCS4':int, 'PRCCS5':int, 'PRCCS6':int,
'PRCCS7':int,'PRCCS8':int, 'PRCCS9':int, 'PRCCS10':int, 'PRCCS11':int,
'PRCCS12':int, 'PRCCS13':int, 'PRCCS14':int,'PRCCS15':int,
'PRDAY1':int, 'PRDAY2':int, 'PRDAY3':int, 'PRDAY4':int, 'PRDAY5':int,
'PRDAY5':int, 'PRDAY6':int, 'PRDAY7':int, 'PRDAY8':int, 'PRDAY9':int,
'PRDAY10':int, 'PRDAY11':int, 'PRDAY12':int,
'PRDAY13':int, 'PRDAY14':int, 'PRDAY15':int,'REHABTRANSFER':int,
'SAMEDAYEVENT':int, 'SERVICELINE':int,'TOTCHG':int, 'YEAR':int,
'ZIPINC_QRTL':int})
nrd_sev = pd.read_csv(file2, dtype={'APRDRG':int,
'APRDRG_Risk_Mortality':int, 'APRDRG_Severity':int, 'CM_AIDS':int,
'CM_ALCOHOL':int, 'CM_ANEMDEF':int, 'CM_ARTH':int, 'CM_BLDLOSS':int,
'M_CHF':int, 'CM_CHRNLUNG':int, 'CM_COAG':int, 'CM_DEPRESS':int,
'CM_DM':int, 'CM_DMCX':int,
'CM_DRUG':int, 'CM_HTN_C':int, 'CM_HYPOTHY':int, 'CM_LIVER':int,
'CM_LYMPH':int, 'CM_LYTES':int, 'CM_METS':int, 'CM_NEURO':int,
'CM_OBESE':int, 'CM_PARA':int, 'CM_PERIVASC':int, 'CM_PSYCH':int,
'CM_PULMCIRC':int, 'CM_RENLFAIL':int, 'CM_TUMOR':int, 'CM_ULCER':int,
'CM_VALVE':int, 'CM_WGHTLOSS':int, 'HOSP_NRD':int, 'KEY_NRD':int})
nrd_dx = pd.read_csv(file3, dtype={'BODYSYSTEM1':int,
'BODYSYSTEM2':int, 'BODYSYSTEM3':int, 'BODYSYSTEM4':int,
'BODYSYSTEM5':int, 'BODYSYSTEM6':int, 'BODYSYSTEM7':int,
'BODYSYSTEM8':int, 'BODYSYSTEM9':int, 'BODYSYSTEM10':int,
'BODYSYSTEM11':int, 'BODYSYSTEM12':int, 'BODYSYSTEM13':int,
'BODYSYSTEM14':int, 'BODYSYSTEM15':int, 'BODYSYSTEM16':int,
'BODYSYSTEM17':int, 'BODYSYSTEM18':int, 'BODYSYSTEM19':int,
'BODYSYSTEM20':int, 'BODYSYSTEM21':int, 'BODYSYSTEM22':int,
'BODYSYSTEM23':int, 'BODYSYSTEM24':int, 'BODYSYSTEM25':int,
'BODYSYSTEM26':int, 'BODYSYSTEM27':int, 'BODYSYSTEM28':int,
'BODYSYSTEM29':int, 'BODYSYSTEM30':int, 'CHRON1':int, 'CHRON2':int,
'CHRON3':int, 'CHRON4':int, 'CHRON5':int, 'CHRON6':int, 'CHRON7':int,
'CHRON8':int, 'CHRON9':int, 'CHRON10':int, 'CHRON11':int,
'CHRON12':int, 'CHRON13':int, 'CHRON14':int, 'CHRON15':int,
'CHRON16':int, 'CHRON17':int, 'CHRON18':int, 'CHRON19':int,
'CHRON20':int, 'CHRON21':int, 'CHRON22':int, 'CHRON23':int,
'CHRON24':int, 'CHRON25':int, 'CHRON26':int, 'CHRON27':int,
'CHRON28':int, 'CHRON29':int, 'CHRON30':int, 'DXMCCS1':str,
'E_MCCS1':str, 'HOSP_NRD':int, 'KEY_NRD':int, 'PCLASS1':int,
'PCLASS2':int, 'PCLASS3':int, 'PCLASS4':int, 'PCLASS5':int,
'PCLASS6':int, 'PCLASS7':int, 'PCLASS8':int, 'PCLASS9':int,
```

```
'PCLASS10':int, 'PCLASS11':int, 'PCLASS12':int, 'PCLASS13':int,
'PCLASS14':int, 'PCLASS15':int, 'PRMCCS1':str})
nrd_hos = pd.read_csv(file4)

# identify records with duplicate visits (readmissions)
nrd_core_visits = nrd_core['NRD_VisitLink']
nrd_core_dups =
nrd_core_visits[nrd_core_visits.duplicated(keep=False)]
print(nrd_core_dups.shape)
nrd_core_subset = nrd_core.iloc[nrd_core_dups.index,:]
del nrd_core
nrd_sev_subset = nrd_sev.iloc[nrd_core_dups.index,:]
del nrd_sev
nrd_dx_subset = nrd_dx.iloc[nrd_core_dups.index,:]
del nrd_dx

# join DataFrames
nrd_core_sev = pd.merge(nrd_core, nrd_sev, how='inner', on=['KEY_NRD',
'HOSP_NRD'])
print(nrd_core_sev.shape)
del nrd_core_subset
del nrd_sev_subset
nrd_core_sev_dx = pd.merge(nrd_core_sev, nrd_dx, how='inner',
on=['KEY_NRD', 'HOSP_NRD'])
print(nrd_core_sev_dx.shape)
del nrd_core_sev
del nrd_dx_subset
nrd_full = pd.merge(nrd_core_sev_dx, nrd_hos, how='inner',
on=['HOSP_NRD', 'NRD_STRATUM'])
print(nrd_full.shape)
del nrd_core_sev_dx
del nrd_hos
filename = 'nrd_full.csv'
nrd_full.to_csv(filename, index=False, encoding='utf-8')


# drop missings or invalids
nrd_full = nrd_full[nrd_full.AGE != 0]
print(nrd_full.shape)
nrd_full = nrd_full[nrd_full.AWEEKEND != -9]
print(nrd_full.shape)
nrd_full = nrd_full[(nrd_full.DIED != -8) & (nrd_full.DIED != -9)]
print(nrd_full.shape)
nrd_full = nrd_full[(nrd_full.DISPUNIFORM != -8) &
(nrd_full.DISPUNIFORM != -9)]
print(nrd_full.shape)
nrd_full = nrd_full[(nrd_full.ELECTIVE != -8) & (nrd_full.ELECTIVE !=
-9)]
print(nrd_full.shape)
nrd_full = nrd_full[(nrd_full.PAY1 != -8) & (nrd_full.PAY1 != -9)]
print(nrd_full.shape)
```

```
nrd_full = nrd_full[nrd_full.PL_NCHS != -9]
print(nrd_full.shape)
nrd_full = nrd_full[(nrd_full.TOTCHG != -999999999) & (nrd_full.TOTCHG
!= -888888888) & (nrd_full.TOTCHG != -666666666)]
print(nrd_full.shape)
filename = 'nrd_full_filtered.csv'
nrd_full.to_csv(filename, index=False, encoding='utf-8')
```

**id_readmits**

```
import pandas as pd
# import numpy as np

# note that you used the 'filtered' data set
file1 = 'nrd_full_filtered.csv' #
nrd = pd.read_csv(file1)
nrd_subset = nrd[['LOS','NRD_VisitLink','NRD_DaysToEvent']]
nrd_subset =
nrd_subset.sort_values(['NRD_VisitLink','NRD_DaysToEvent'],
ascending=[False, False])

# calculate the number of days since discharge
nrd_subset.set_value((len(nrd)-1), 'DaysFromDischarge', 0)
i = 0
while i < len(nrd_subset):
    visit1 = nrd_subset.iloc[i]['NRD_VisitLink']
    visit2 = nrd_subset.iloc[i+1]['NRD_VisitLink']
    if visit1 == visit2:
        DaysSinceDischarge = nrd_subset.iloc[i]['NRD_DaysToEvent'] -
(nrd_subset.iloc[i+1]['LOS']+nrd_subset.iloc[i+1]['NRD_DaysToEvent'])
        nrd_subset.iloc[[i],[3]]= DaysSinceDischarge
    else:
        nrd_subset.iloc[[i], [3]] = 0
    i+=1
# write file with days since readmit value
filename = 'nrd_discharge.csv'
nrd_subset.to_csv(filename, index=False, encoding='utf-8')

# flag records where patient readmitted < 30 days
nrd_subset.set_value((len(nrd_subset)-1), 'EarlyReadmit', 0)
i = 1
while i < len(nrd_subset):
    previous_visit = i - 1
    if (nrd_subset.iloc[i]['DaysFromDischarge'] < 30) &
(nrd_subset.iloc[i]['DaysFromDischarge'] > 0) :
        nrd_subset.iloc[[previous_visit], [4]] = 1
        print(i)
    else:
        nrd_subset.iloc[[previous_visit], [4]] = 0
    i+=1
# write file with early readmits flag
```

```
filename = 'nrd_dis_w_flag.csv'
nrd_subset.to_csv(filename, index=False, encoding='utf-8')

# restarted Python, so reloading datasets
filename = 'nrd_dis_w_flag.csv'
nrd_subset = pd.read_csv(filename)
nrd_subset = nrd_subset.drop('LOS',1) #drop LOS
file1 = 'nrd_full.csv' #this is the full merged dataset including
records without readmissions
nrd_full = pd.read_csv(file1)

nrd_full_w_flags = pd.merge(nrd_full, nrd_subset, how='left',
on=['NRD_DaysToEvent', 'NRD_VisitLink'])

# check dataframe
nrd_full_w_flags.shape
nrd_full_w_flags.head()

# set the nulls to 0 after the merge
nrd_full_w_flags['DaysFromDischarge'].fillna(0, inplace=True)
nrd_full_w_flags['EarlyReadmit'].fillna(0, inplace=True)

# set full columns for DaysFromDischarge & EarlyReadmit
# nrd['DaysFromDischarge']=0
# nrd['EarlyReadmit']=0
# transfer DaysFromDischarge & EarlyReadmit values from subset to full
# example -- df[df[['A']]<0] = 0
# df['points'] = np.where( ( (df['gender'] == 'male') & (df['pet1'] ==
df['pet2'] ) ) |
# ( (df['gender'] == 'female') & (df['pet1'].isin(['cat','dog'] ) ) ),
5, 0)

#nrd['DaysFromDischarge'] = np.where((nrd['NRD_DaysToEvent'] ==
nrd_subset['NRD_DaysToEvent']) & (nrd['NRD_VisitLink'] ==
nrd_subset['NRD_VisitLink']), nrd_subset.['DaysFromDischarge'])
# nrd['EarlyReadmit'] = np.where((nrd['NRD_DaysToEvent'] ==
nrd_subset['NRD_DaysToEvent']) & (nrd['NRD_VisitLink'] ==
nrd_subset['NRD_VisitLink']), nrd_subset.['EarlyReadmit'])


filename = 'nrd_full_w_flags.csv'
nrd_full_w_flags.to_csv(filename, index=False, encoding='utf-8')

file1 = 'nrd_full_w_flags.csv' #
nrd_full_w_flags = pd.read_csv(file1)

# create field to flag the discharge prior to the early readmit
filename = 'nrd_dis_w_flag.csv'
nrd_flags = pd.read_csv(filename)
nrd_flags["EarlyDischarge"] = nrd_flags["EarlyReadmit"].shift(-1)
filename = 'nrd_dis_w_flag2.csv'
```

```
nrd_flags.to_csv(filename, index=False, encoding='utf-8')
```

**ami_records**

```
import pandas as pd

file1 = 'NRD_2014_Core_mod.csv'
file2 = 'NRD_2014_Severity_mod.csv'
file3 = 'NRD_2014_DX_PR_GRPS_mod.csv'
file4 = 'NRD_2014_Hospital_mod.csv'
nrd_core = pd.read_csv(file1, dtype={'AGE':int, 'AWEEKEND':int,
'DIED':int, 'DISCWT':float, 'DISPUNIFORM':int, 'DMONTH':int,
'DQTR':int, 'DRG':int, 'DRG_NoPOA':int, 'DRGVER':int, 'DX1':str,
'DX2':str, 'DX3':str, 'DX4':str, 'DX5':str, 'DX6':str, 'DX7':str,
'DX8':str, 'DX9':str, 'DX10':str, 'DX11':str, 'DX12':str, 'DX13':str,
'DX14':str, 'DX15':str, 'DX16':str, 'DX17':str, 'DX18':str,'DX19':str,
'DX20':str, 'DX21':str, 'DX22':str, 'DX23':str, 'DX24':str,
'DX25':str, 'DX26':str, 'DX27':str, 'DX28':str, 'DX29':str,'DX30':str,
'DXCCS1':int, 'DXCCS2':int, 'DXCCS3':int, 'DXCCS4':int, 'DXCCS5':int,
'DXCCS6':int, 'DXCCS7':int, 'DXCCS8':int, 'DXCCS9':int, 'DXCCS10':int,
'DXCCS11':int, 'DXCCS12':int, 'DXCCS13':int, 'DXCCS14':int,
'DXCCS15':int, 'DXCCS16':int, 'DXCCS17':int, 'DXCCS18':int,
'DXCCS19':int, 'DXCCS20':int, 'DXCCS21':int, 'DXCCS22':int,
'DXCCS23':int, 'DXCCS24':int, 'DXCCS25':int, 'DXCCS26':int,
'DXCCS27':int, 'DXCCS28':int, 'DXCCS29':int, 'DXCCS30':int,
'ECODE1':str, 'ECODE2':str, 'ECODE3':str, 'ECODE4':str,
'ELECTIVE':int, 'E_CCS1':int, 'E_CCS2':int, 'E_CCS3':int,
'E_CCS4':int, 'FEMALE':int, 'HCUP_ED':int,'HOSP_NRD':int,
'KEY_NRD':int, 'LOS':int, 'MDC':int, 'MDC_NoPOA':int, 'NCHRONIC':int,
'NDX':int, 'NECODE':int, 'NPR':int, 'NRD_DaysToEvent':int,
'NRD_STRATUM':int, 'NRD_VisitLink':str, 'ORPROC':int,'PAY1':int,
'PL_NCHS':int, 'PR1':str, 'PR2':str, 'PR3':str, 'PR4':str, 'PR5':str,
'PR6':str, 'PR7':str, 'PR8':str, 'PR9':str, 'PR10':str, 'PR11':str,
'PR12':str, 'PR13':str, 'PR14':str, 'PR15':str,'PRCCS1':int,
'PRCCS2':int, 'PRCCS3':int, 'PRCCS4':int, 'PRCCS5':int, 'PRCCS6':int,
'PRCCS7':int, 'PRCCS8':int, 'PRCCS9':int, 'PRCCS10':int,
'PRCCS11':int, 'PRCCS12':int, 'PRCCS13':int, 'PRCCS14':int,
'PRCCS15':int, 'PRDAY1':int, 'PRDAY2':int, 'PRDAY3':int, 'PRDAY4':int,
'PRDAY5':int, 'PRDAY5':int,'PRDAY6':int, 'PRDAY7':int, 'PRDAY8':int,
'PRDAY9':int, 'PRDAY10':int, 'PRDAY11':int, 'PRDAY12':int,
'PRDAY13':int, 'PRDAY14':int, 'PRDAY15':int, 'REHABTRANSFER':int,
'SAMEDAYEVENT':int, 'SERVICELINE':int,
                        'TOTCHG':int, 'YEAR':int, 'ZIPINC_QRTL':int})

fields=['DRG']
nrd_drg = pd.read_csv(file1, usecols=fields)

# ami = nrd_drg.loc[(nrd_drg['DRG']>=410) & (nrd_drg['DRG']<411)]
ami = nrd_drg.loc[(nrd_drg['DRG']>=280) & (nrd_drg['DRG']<286)]

nrd_core_ami = nrd_core.iloc[ami.index,:]
```

```
del nrd_core

nrd_sev = pd.read_csv(file2, dtype={'APRDRG':int,
'APRDRG_Risk_Mortality':int, 'APRDRG_Severity':int, 'CM_AIDS':int,
'CM_ALCOHOL':int, 'CM_ANEMDEF':int, 'CM_ARTH':int, 'CM_BLDLOSS':int,
'M_CHF':int, 'CM_CHRNLUNG':int, 'CM_COAG':int, 'CM_DEPRESS':int,
'CM_DM':int, 'CM_DMCX':int, 'CM_DRUG':int, 'CM_HTN_C':int,
'CM_HYPOTHY':int, 'CM_LIVER':int, 'CM_LYMPH':int,'CM_LYTES':int,
'CM_METS':int, 'CM_NEURO':int, 'CM_OBESE':int, 'CM_PARA':int,
'CM_PERIVASC':int, 'CM_PSYCH':int, 'CM_PULMCIRC':int,
'CM_RENLFAIL':int, 'CM_TUMOR':int, 'CM_ULCER':int, 'CM_VALVE':int,
'CM_WGHTLOSS':int, 'HOSP_NRD':int,'KEY_NRD':int})
nrd_sev_ami = nrd_sev.iloc[ami.index,:]
del nrd_sev

nrd_dx = pd.read_csv(file3, dtype={'BODYSYSTEM1':int,
'BODYSYSTEM2':int, 'BODYSYSTEM3':int, 'BODYSYSTEM4':int,
'BODYSYSTEM5':int, 'BODYSYSTEM6':int, 'BODYSYSTEM7':int,
'BODYSYSTEM8':int, 'BODYSYSTEM9':int, 'BODYSYSTEM10':int,
'BODYSYSTEM11':int, 'BODYSYSTEM12':int, 'BODYSYSTEM13':int,
'BODYSYSTEM14':int, 'BODYSYSTEM15':int, 'BODYSYSTEM16':int,
'BODYSYSTEM17':int, 'BODYSYSTEM18':int, 'BODYSYSTEM19':int,
'BODYSYSTEM20':int, 'BODYSYSTEM21':int, 'BODYSYSTEM22':int,
'BODYSYSTEM23':int, 'BODYSYSTEM24':int, 'BODYSYSTEM25':int,
'BODYSYSTEM26':int, 'BODYSYSTEM27':int, 'BODYSYSTEM28':int,
'BODYSYSTEM29':int, 'BODYSYSTEM30':int,'CHRON1':int, 'CHRON2':int,
'CHRON3':int, 'CHRON4':int, 'CHRON5':int, 'CHRON6':int,
'CHRON7':int, 'CHRON8':int, 'CHRON9':int, 'CHRON10':int,
'CHRON11':int, 'CHRON12':int, 'CHRON13':int, 'CHRON14':int,
'CHRON15':int, 'CHRON16':int, 'CHRON17':int, 'CHRON18':int,
'CHRON19':int, 'CHRON20':int, 'CHRON21':int, 'CHRON22':int,
'CHRON23':int, 'CHRON24':int, 'CHRON25':int, 'CHRON26':int,
'CHRON27':int, 'CHRON28':int, 'CHRON29':int, 'CHRON30':int,
'DXMCCS1':str, 'E_MCCS1':str, 'HOSP_NRD':int, 'KEY_NRD':int,
'PCLASS1':int, 'PCLASS2':int, 'PCLASS3':int, 'PCLASS4':int,
'PCLASS5':int, 'PCLASS6':int, 'PCLASS7':int, 'PCLASS8':int,
'PCLASS9':int, 'PCLASS10':int, 'PCLASS11':int, 'PCLASS12':int,
'PCLASS13':int, 'PCLASS14':int, 'PCLASS15':int, 'PRMCCS1':str})
nrd_dx_ami = nrd_dx.iloc[ami.index,:]
del nrd_dx

nrd_hos = pd.read_csv(file4)

# join DataFrames
nrd_core_sev = pd.merge(nrd_core_ami, nrd_sev_ami, how='inner',
on=['KEY_NRD', 'HOSP_NRD'])
nrd_core_sev_dx = pd.merge(nrd_core_sev, nrd_dx_ami, how='inner',
on=['KEY_NRD', 'HOSP_NRD'])
nrd_full_ami = pd.merge(nrd_core_sev_dx, nrd_hos, how='inner',
on=['HOSP_NRD', 'NRD_STRATUM'])
```

```
filename = 'nrd_full_ami.csv'
nrd_full_ami.to_csv(filename, index=False, encoding='utf-8')

filename = 'nrd_dis_w_flag2.csv'
nrd_flags = pd.read_csv(filename)
nrd_ami_w_flags = pd.merge(nrd_full_ami, nrd_flags, how='left',
on=['NRD_DaysToEvent', 'NRD_VisitLink'])

nrd_ami_w_flags['DaysFromDischarge'].fillna(0, inplace=True)
nrd_ami_w_flags['EarlyReadmit'].fillna(0, inplace=True)

filename = 'nrd_full_ami_w_flags.csv'
nrd_ami_w_flags.to_csv(filename, index=False, encoding='utf-8')
```

**cabg_records.py**

```
import pandas as pd

file1 = 'NRD_2014_Core_mod.csv'
file2 = 'NRD_2014_Severity_mod.csv'
file3 = 'NRD_2014_DX_PR_GRPS_mod.csv'
file4 = 'NRD_2014_Hospital_mod.csv'
nrd_core = pd.read_csv(file1, dtype={'AGE':int, 'AWEEKEND':int,
'DIED':int, 'DISCWT':float, 'DISPUNIFORM':int, 'DMONTH':int,
'DQTR':int, 'DRG':int, 'DRG_NoPOA':int, 'DRGVER':int, 'DX1':str,
'DX2':str, 'DX3':str, 'DX4':str,'DX5':str, 'DX6':str, 'DX7':str,
'DX8':str, 'DX9':str, 'DX10':str, 'DX11':str, 'DX12':str, 'DX13':str,
'DX14':str, 'DX15':str, 'DX16':str, 'DX17':str, 'DX18':str,
'DX19':str, 'DX20':str, 'DX21':str,'DX22':str, 'DX23':str, 'DX24':str,
'DX25':str, 'DX26':str, 'DX27':str, 'DX28':str, 'DX29':str,'DX30':str,
'DXCCS1':int, 'DXCCS2':int, 'DXCCS3':int, 'DXCCS4':int, 'DXCCS5':int,
'DXCCS6':int, 'DXCCS7':int, 'DXCCS8':int, 'DXCCS9':int, 'DXCCS10':int,
'DXCCS11':int, 'DXCCS12':int, 'DXCCS13':int,'DXCCS14':int,
'DXCCS15':int, 'DXCCS16':int, 'DXCCS17':int, 'DXCCS18':int,
'DXCCS19':int, 'DXCCS20':int,'DXCCS21':int, 'DXCCS22':int,
'DXCCS23':int, 'DXCCS24':int, 'DXCCS25':int, 'DXCCS26':int,
'DXCCS27':int, 'DXCCS28':int, 'DXCCS29':int, 'DXCCS30':int,
'ECODE1':str, 'ECODE2':str, 'ECODE3':str, 'ECODE4':str,'ELECTIVE':int,
'E_CCS1':int, 'E_CCS2':int, 'E_CCS3':int, 'E_CCS4':int, 'FEMALE':int,
'HCUP_ED':int, 'HOSP_NRD':int, 'KEY_NRD':int, 'LOS':int, 'MDC':int,
'MDC_NoPOA':int, 'NCHRONIC':int, 'NDX':int,'NECODE':int, 'NPR':int,
'NRD_DaysToEvent':int, 'NRD_STRATUM':int, 'NRD_VisitLink':str,
'ORPROC':int, 'PAY1':int, 'PL_NCHS':int, 'PR1':str, 'PR2':str,
'PR3':str, 'PR4':str, 'PR5':str, 'PR6':str, 'PR7':str,'PR8':str,
'PR9':str, 'PR10':str, 'PR11':str, 'PR12':str, 'PR13':str, 'PR14':str,
'PR15':str, 'PRCCS1':int, 'PRCCS2':int, 'PRCCS3':int, 'PRCCS4':int,
'PRCCS5':int, 'PRCCS6':int, 'PRCCS7':int,'PRCCS8':int, 'PRCCS9':int,
'PRCCS10':int, 'PRCCS11':int, 'PRCCS12':int, 'PRCCS13':int,
'PRCCS14':int,'PRCCS15':int, 'PRDAY1':int, 'PRDAY2':int, 'PRDAY3':int,
'PRDAY4':int, 'PRDAY5':int, 'PRDAY5':int,'PRDAY6':int, 'PRDAY7':int,
'PRDAY8':int, 'PRDAY9':int, 'PRDAY10':int, 'PRDAY11':int,
```

```
'PRDAY12':int, 'PRDAY13':int, 'PRDAY14':int, 'PRDAY15':int,
'REHABTRANSFER':int, 'SAMEDAYEVENT':int, 'SERVICELINE':int,
'TOTCHG':int, 'YEAR':int, 'ZIPINC_QRTL':int})

fields=['DRG']
nrd_drg = pd.read_csv(file1, usecols=fields)

# cabg = nrd_drg.loc[(nrd_drg['DRG']>=414) & (nrd_drg['DRG']<415)]
cabg = nrd_drg.loc[(nrd_drg['DRG']>=233) & (nrd_drg['DRG']<237)]

nrd_core_cabg = nrd_core.iloc[cabg.index,:]
del nrd_core

nrd_sev = pd.read_csv(file2, dtype={'APRDRG':int,
'APRDRG_Risk_Mortality':int, 'APRDRG_Severity':int, 'CM_AIDS':int,
'CM_ALCOHOL':int, 'CM_ANEMDEF':int, 'CM_ARTH':int, 'CM_BLDLOSS':int,
'M_CHF':int, 'CM_CHRNLUNG':int, 'CM_COAG':int, 'CM_DEPRESS':int,
'CM_DM':int, 'CM_DMCX':int, 'CM_DRUG':int, 'CM_HTN_C':int,
'CM_HYPOTHY':int, 'CM_LIVER':int, 'CM_LYMPH':int,'CM_LYTES':int,
'CM_METS':int, 'CM_NEURO':int, 'CM_OBESE':int, 'CM_PARA':int,
'CM_PERIVASC':int, 'CM_PSYCH':int, 'CM_PULMCIRC':int,
'CM_RENLFAIL':int, 'CM_TUMOR':int, 'CM_ULCER':int, 'CM_VALVE':int,
'CM_WGHTLOSS':int, 'HOSP_NRD':int, 'KEY_NRD':int})
nrd_sev_cabg = nrd_sev.iloc[cabg.index,:]
del nrd_sev

nrd_dx = pd.read_csv(file3, dtype={'BODYSYSTEM1':int,
'BODYSYSTEM2':int, 'BODYSYSTEM3':int, 'BODYSYSTEM4':int,
'BODYSYSTEM5':int, 'BODYSYSTEM6':int, 'BODYSYSTEM7':int,
'BODYSYSTEM8':int, 'BODYSYSTEM9':int, 'BODYSYSTEM10':int,
'BODYSYSTEM11':int, 'BODYSYSTEM12':int, 'BODYSYSTEM13':int,
'BODYSYSTEM14':int, 'BODYSYSTEM15':int, 'BODYSYSTEM16':int,
'BODYSYSTEM17':int, 'BODYSYSTEM18':int,'BODYSYSTEM19':int,
'BODYSYSTEM20':int, 'BODYSYSTEM21':int, 'BODYSYSTEM22':int,
'BODYSYSTEM23':int, 'BODYSYSTEM24':int, 'BODYSYSTEM25':int,
'BODYSYSTEM26':int, 'BODYSYSTEM27':int, 'BODYSYSTEM28':int,
'BODYSYSTEM29':int, 'BODYSYSTEM30':int, 'CHRON1':int, 'CHRON2':int,
'CHRON3':int, 'CHRON4':int, 'CHRON5':int, 'CHRON6':int,
'CHRON7':int, 'CHRON8':int, 'CHRON9':int, 'CHRON10':int,
'CHRON11':int, 'CHRON12':int, 'CHRON13':int, 'CHRON14':int,
'CHRON15':int, 'CHRON16':int, 'CHRON17':int, 'CHRON18':int,
'CHRON19':int, 'CHRON20':int, 'CHRON21':int, 'CHRON22':int,
'CHRON23':int, 'CHRON24':int, 'CHRON25':int, 'CHRON26':int,
'CHRON27':int, 'CHRON28':int, 'CHRON29':int, 'CHRON30':int,
'DXMCCS1':str, 'E_MCCS1':str, 'HOSP_NRD':int, 'KEY_NRD':int,
'PCLASS1':int, 'PCLASS2':int, 'PCLASS3':int, 'PCLASS4':int,
'PCLASS5':int, 'PCLASS6':int, 'PCLASS7':int, 'PCLASS8':int,
'PCLASS9':int, 'PCLASS10':int, 'PCLASS11':int, 'PCLASS12':int,
'PCLASS13':int, 'PCLASS14':int, 'PCLASS15':int, 'PRMCCS1':str})
nrd_dx_cabg = nrd_dx.iloc[cabg.index,:]
del nrd_dx
```

```
nrd_hos = pd.read_csv(file4)

# join DataFrames
nrd_core_sev = pd.merge(nrd_core_cabg, nrd_sev_cabg, how='inner',
on=['KEY_NRD', 'HOSP_NRD'])
nrd_core_sev_dx = pd.merge(nrd_core_sev, nrd_dx_cabg, how='inner',
on=['KEY_NRD', 'HOSP_NRD'])
nrd_full_cabg = pd.merge(nrd_core_sev_dx, nrd_hos, how='inner',
on=['HOSP_NRD', 'NRD_STRATUM'])

filename = 'nrd_full_cabg.csv'
nrd_full_cabg.to_csv(filename, index=False, encoding='utf-8')

filename = 'nrd_dis_w_flag2.csv'
nrd_flags = pd.read_csv(filename)
nrd_cabg_w_flags = pd.merge(nrd_full_cabg, nrd_flags, how='left',
on=['NRD_DaysToEvent', 'NRD_VisitLink'])

nrd_cabg_w_flags['DaysFromDischarge'].fillna(0, inplace=True)
nrd_cabg_w_flags['EarlyReadmit'].fillna(0, inplace=True)

filename = 'nrd_full_cabg_w_flags.csv'
nrd_cabg_w_flags.to_csv(filename, index=False, encoding='utf-8')
```

**copd_records.py**

```
import pandas as pd

file1 = 'NRD_2014_Core_mod.csv'
file2 = 'NRD_2014_Severity_mod.csv'
file3 = 'NRD_2014_DX_PR_GRPS_mod.csv'
file4 = 'NRD_2014_Hospital_mod.csv'
nrd_core = pd.read_csv(file1, dtype={'AGE':int, 'AWEEKEND':int,
'DIED':int, 'DISCWT':float, 'DISPUNIFORM':int, 'DMONTH':int,
'DQTR':int, 'DRG':int, 'DRG_NoPOA':int, 'DRGVER':int, 'DX1':str,
'DX2':str, 'DX3':str, 'DX4':str,'DX5':str, 'DX6':str, 'DX7':str,
'DX8':str, 'DX9':str, 'DX10':str, 'DX11':str, 'DX12':str, 'DX13':str,
'DX14':str, 'DX15':str, 'DX16':str, 'DX17':str, 'DX18':str,
'DX19':str, 'DX20':str, 'DX21':str,'DX22':str, 'DX23':str, 'DX24':str,
'DX25':str, 'DX26':str, 'DX27':str, 'DX28':str, 'DX29':str,
'DX30':str, 'DXCCS1':int, 'DXCCS2':int, 'DXCCS3':int, 'DXCCS4':int,
'DXCCS5':int, 'DXCCS6':int,'DXCCS7':int, 'DXCCS8':int, 'DXCCS9':int,
'DXCCS10':int, 'DXCCS11':int, 'DXCCS12':int, 'DXCCS13':int,
'DXCCS14':int, 'DXCCS15':int, 'DXCCS16':int, 'DXCCS17':int,
'DXCCS18':int, 'DXCCS19':int, 'DXCCS20':int,'DXCCS21':int,
'DXCCS22':int, 'DXCCS23':int, 'DXCCS24':int, 'DXCCS25':int,
'DXCCS26':int, 'DXCCS27':int,'DXCCS28':int, 'DXCCS29':int,
'DXCCS30':int, 'ECODE1':str, 'ECODE2':str, 'ECODE3':str, 'ECODE4':str,
'ELECTIVE':int, 'E_CCS1':int, 'E_CCS2':int, 'E_CCS3':int,'E_CCS4':int,
'FEMALE':int, 'HCUP_ED':int,'HOSP_NRD':int, 'KEY_NRD':int, 'LOS':int,
```

```
'MDC':int, 'MDC_NoPOA':int, 'NCHRONIC':int, 'NDX':int, 'NECODE':int,
'NPR':int, 'NRD_DaysToEvent':int, 'NRD_STRATUM':int,
'NRD_VisitLink':str, 'ORPROC':int,'PAY1':int, 'PL_NCHS':int,
'PR1':str, 'PR2':str, 'PR3':str, 'PR4':str, 'PR5':str, 'PR6':str,
'PR7':str,'PR8':str, 'PR9':str, 'PR10':str, 'PR11':str, 'PR12':str,
'PR13':str, 'PR14':str, 'PR15':str,'PRCCS1':int, 'PRCCS2':int,
'PRCCS3':int, 'PRCCS4':int, 'PRCCS5':int, 'PRCCS6':int, 'PRCCS7':int,
'PRCCS8':int, 'PRCCS9':int, 'PRCCS10':int, 'PRCCS11':int,
'PRCCS12':int, 'PRCCS13':int, 'PRCCS14':int,'PRCCS15':int,
'PRDAY1':int, 'PRDAY2':int, 'PRDAY3':int, 'PRDAY4':int, 'PRDAY5':int,
'PRDAY5':int, 'PRDAY6':int, 'PRDAY7':int, 'PRDAY8':int, 'PRDAY9':int,
'PRDAY10':int, 'PRDAY11':int, 'PRDAY12':int,'PRDAY13':int,
'PRDAY14':int, 'PRDAY15':int, 'REHABTRANSFER':int, 'SAMEDAYEVENT':int,
'SERVICELINE':int, 'TOTCHG':int, 'YEAR':int, 'ZIPINC_QRTL':int})

fields=['DRG']
nrd_drg = pd.read_csv(file1, usecols=fields)

# copd = nrd_drg.loc[(nrd_drg['DRG']>=490) & (nrd_drg['DRG']<497)]
copd = nrd_drg.loc[(nrd_drg['DRG']>=190) & (nrd_drg['DRG']<194)]

nrd_core_copd = nrd_core.iloc[copd.index,:]
del nrd_core

nrd_sev = pd.read_csv(file2, dtype={'APRDRG':int,
'APRDRG_Risk_Mortality':int, 'APRDRG_Severity':int, 'CM_AIDS':int,
'CM_ALCOHOL':int, 'CM_ANEMDEF':int, 'CM_ARTH':int, 'CM_BLDLOSS':int,
'M_CHF':int, 'CM_CHRNLUNG':int, 'CM_COAG':int, 'CM_DEPRESS':int,
'CM_DM':int, 'CM_DMCX':int, 'CM_DRUG':int, 'CM_HTN_C':int,
'CM_HYPOTHY':int, 'CM_LIVER':int, 'CM_LYMPH':int,'CM_LYTES':int,
'CM_METS':int, 'CM_NEURO':int, 'CM_OBESE':int, 'CM_PARA':int,
'CM_PERIVASC':int, 'CM_PSYCH':int, 'CM_PULMCIRC':int,
'CM_RENLFAIL':int, 'CM_TUMOR':int, 'CM_ULCER':int, 'CM_VALVE':int,
'CM_WGHTLOSS':int, 'HOSP_NRD':int, 'KEY_NRD':int})
nrd_sev_copd = nrd_sev.iloc[copd.index,:]
del nrd_sev

nrd_dx = pd.read_csv(file3, dtype={'BODYSYSTEM1':int,
'BODYSYSTEM2':int, 'BODYSYSTEM3':int, 'BODYSYSTEM4':int,
'BODYSYSTEM5':int, 'BODYSYSTEM6':int, 'BODYSYSTEM7':int,
'BODYSYSTEM8':int, 'BODYSYSTEM9':int, 'BODYSYSTEM10':int,
'BODYSYSTEM11':int, 'BODYSYSTEM12':int, 'BODYSYSTEM13':int,
'BODYSYSTEM14':int, 'BODYSYSTEM15':int, 'BODYSYSTEM16':int,
'BODYSYSTEM17':int, 'BODYSYSTEM18':int, 'BODYSYSTEM19':int,
'BODYSYSTEM20':int, 'BODYSYSTEM21':int, 'BODYSYSTEM22':int,
'BODYSYSTEM23':int, 'BODYSYSTEM24':int, 'BODYSYSTEM25':int,
'BODYSYSTEM26':int, 'BODYSYSTEM27':int, 'BODYSYSTEM28':int,
'BODYSYSTEM29':int, 'BODYSYSTEM30':int, 'CHRON1':int, 'CHRON2':int,
'CHRON3':int, 'CHRON4':int, 'CHRON5':int, 'CHRON6':int,'CHRON7':int,
'CHRON8':int, 'CHRON9':int, 'CHRON10':int, 'CHRON11':int,
'CHRON12':int, 'CHRON13':int, 'CHRON14':int, 'CHRON15':int,
```

```
'CHRON16':int, 'CHRON17':int, 'CHRON18':int,'CHRON19':int,
'CHRON20':int, 'CHRON21':int, 'CHRON22':int, 'CHRON23':int,
'CHRON24':int, 'CHRON25':int, 'CHRON26':int, 'CHRON27':int,
'CHRON28':int, 'CHRON29':int, 'CHRON30':int,'DXMCCS1':str,
'E_MCCS1':str, 'HOSP_NRD':int, 'KEY_NRD':int, 'PCLASS1':int,
'PCLASS2':int, 'PCLASS3':int, 'PCLASS4':int, 'PCLASS5':int,
'PCLASS6':int, 'PCLASS7':int, 'PCLASS8':int,'PCLASS9':int,
'PCLASS10':int, 'PCLASS11':int, 'PCLASS12':int, 'PCLASS13':int,
'PCLASS14':int, 'PCLASS15':int, 'PRMCCS1':str})
nrd_dx_copd = nrd_dx.iloc[copd.index,:]
del nrd_dx

nrd_hos = pd.read_csv(file4)

# join DataFrames
nrd_core_sev = pd.merge(nrd_core_copd, nrd_sev_copd, how='inner',
on=['KEY_NRD', 'HOSP_NRD'])
nrd_core_sev_dx = pd.merge(nrd_core_sev, nrd_dx_copd, how='inner',
on=['KEY_NRD', 'HOSP_NRD'])
nrd_full_copd = pd.merge(nrd_core_sev_dx, nrd_hos, how='inner',
on=['HOSP_NRD', 'NRD_STRATUM'])

filename = 'nrd_full_copd.csv'
nrd_full_copd.to_csv(filename, index=False, encoding='utf-8')

filename = 'nrd_dis_w_flag2.csv'
nrd_flags = pd.read_csv(filename)
nrd_copd_w_flags = pd.merge(nrd_full_copd, nrd_flags, how='left',
on=['NRD_DaysToEvent', 'NRD_VisitLink'])

nrd_copd_w_flags['DaysFromDischarge'].fillna(0, inplace=True)
nrd_copd_w_flags['EarlyReadmit'].fillna(0, inplace=True)

filename = 'nrd_full_copd_w_flags.csv'
nrd_copd_w_flags.to_csv(filename, index=False, encoding='utf-8')
```

**pnu_records.py**

```
import pandas as pd

file1 = 'NRD_2014_Core_mod.csv'
file2 = 'NRD_2014_Severity_mod.csv'
file3 = 'NRD_2014_DX_PR_GRPS_mod.csv'
file4 = 'NRD_2014_Hospital_mod.csv'
nrd_core = pd.read_csv(file1, dtype={'AGE':int, 'AWEEKEND':int,
'DIED':int, 'DISCWT':float, 'DISPUNIFORM':int, 'DMONTH':int,
'DQTR':int, 'DRG':int, 'DRG_NoPOA':int, 'DRGVER':int, 'DX1':str,
'DX2':str, 'DX3':str, 'DX4':str,'DX5':str, 'DX6':str, 'DX7':str,
'DX8':str, 'DX9':str, 'DX10':str, 'DX11':str, 'DX12':str, 'DX13':str,
'DX14':str, 'DX15':str, 'DX16':str, 'DX17':str, 'DX18':str,
'DX19':str, 'DX20':str, 'DX21':str, 'DX22':str, 'DX23':str,
```

```
'DX24':str, 'DX25':str, 'DX26':str, 'DX27':str, 'DX28':str,
'DX29':str, 'DX30':str, 'DXCCS1':int, 'DXCCS2':int, 'DXCCS3':int,
'DXCCS4':int, 'DXCCS5':int, 'DXCCS6':int,'DXCCS7':int, 'DXCCS8':int,
'DXCCS9':int, 'DXCCS10':int, 'DXCCS11':int, 'DXCCS12':int,
'DXCCS13':int, 'DXCCS14':int, 'DXCCS15':int, 'DXCCS16':int,
'DXCCS17':int, 'DXCCS18':int, 'DXCCS19':int, 'DXCCS20':int,
'DXCCS21':int, 'DXCCS22':int, 'DXCCS23':int, 'DXCCS24':int,
'DXCCS25':int, 'DXCCS26':int, 'DXCCS27':int,'DXCCS28':int,
'DXCCS29':int, 'DXCCS30':int, 'ECODE1':str, 'ECODE2':str,
'ECODE3':str, 'ECODE4':str,'ELECTIVE':int, 'E_CCS1':int, 'E_CCS2':int,
'E_CCS3':int, 'E_CCS4':int, 'FEMALE':int, 'HCUP_ED':int,
'HOSP_NRD':int, 'KEY_NRD':int, 'LOS':int, 'MDC':int, 'MDC_NoPOA':int,
'NCHRONIC':int, 'NDX':int,'NECODE':int, 'NPR':int,
'NRD_DaysToEvent':int, 'NRD_STRATUM':int, 'NRD_VisitLink':str,
'ORPROC':int, 'PAY1':int, 'PL_NCHS':int, 'PR1':str, 'PR2':str,
'PR3':str, 'PR4':str, 'PR5':str, 'PR6':str, 'PR7':str,
'PR8':str, 'PR9':str, 'PR10':str, 'PR11':str, 'PR12':str, 'PR13':str,
'PR14':str, 'PR15':str, 'PRCCS1':int, 'PRCCS2':int, 'PRCCS3':int,
'PRCCS4':int, 'PRCCS5':int, 'PRCCS6':int, 'PRCCS7':int,
'PRCCS8':int, 'PRCCS9':int, 'PRCCS10':int, 'PRCCS11':int,
'PRCCS12':int, 'PRCCS13':int, 'PRCCS14':int,'PRCCS15':int,
'PRDAY1':int, 'PRDAY2':int, 'PRDAY3':int, 'PRDAY4':int, 'PRDAY5':int,
'PRDAY5':int, 'PRDAY6':int, 'PRDAY7':int, 'PRDAY8':int, 'PRDAY9':int,
'PRDAY10':int, 'PRDAY11':int, 'PRDAY12':int,'PRDAY13':int,
'PRDAY14':int, 'PRDAY15':int, 'REHABTRANSFER':int, 'SAMEDAYEVENT':int,
'SERVICELINE':int, 'TOTCHG':int, 'YEAR':int, 'ZIPINC_QRTL':int})

fields=['DRG']
nrd_drg = pd.read_csv(file1, usecols=fields)

# pnu = nrd_drg.loc[(nrd_drg['DRG']>=480) & (nrd_drg['DRG']<487)]
pnu = nrd_drg.loc[(nrd_drg['DRG']>=193) & (nrd_drg['DRG']<196)]

nrd_core_pnu = nrd_core.iloc[pnu.index,:]
del nrd_core

nrd_sev = pd.read_csv(file2, dtype={'APRDRG':int,
'APRDRG_Risk_Mortality':int, 'APRDRG_Severity':int, 'CM_AIDS':int,
'CM_ALCOHOL':int, 'CM_ANEMDEF':int, 'CM_ARTH':int, 'CM_BLDLOSS':int,
'M_CHF':int, 'CM_CHRNLUNG':int, 'CM_COAG':int, 'CM_DEPRESS':int,
'CM_DM':int, 'CM_DMCX':int, 'CM_DRUG':int, 'CM_HTN_C':int,
'CM_HYPOTHY':int, 'CM_LIVER':int, 'CM_LYMPH':int,'CM_LYTES':int,
'CM_METS':int, 'CM_NEURO':int, 'CM_OBESE':int, 'CM_PARA':int,
'CM_PERIVASC':int, 'CM_PSYCH':int, 'CM_PULMCIRC':int,
'CM_RENLFAIL':int, 'CM_TUMOR':int, 'CM_ULCER':int, 'CM_VALVE':int,
'CM_WGHTLOSS':int, 'HOSP_NRD':int, 'KEY_NRD':int})
nrd_sev_pnu = nrd_sev.iloc[pnu.index,:]
del nrd_sev

nrd_dx = pd.read_csv(file3, dtype={'BODYSYSTEM1':int,
'BODYSYSTEM2':int, 'BODYSYSTEM3':int, 'BODYSYSTEM4':int,
```

```
'BODYSYSTEM5':int, 'BODYSYSTEM6':int, 'BODYSYSTEM7':int,
'BODYSYSTEM8':int, 'BODYSYSTEM9':int, 'BODYSYSTEM10':int,
'BODYSYSTEM11':int, 'BODYSYSTEM12':int, 'BODYSYSTEM13':int,
'BODYSYSTEM14':int, 'BODYSYSTEM15':int, 'BODYSYSTEM16':int,
'BODYSYSTEM17':int, 'BODYSYSTEM18':int, 'BODYSYSTEM19':int,
'BODYSYSTEM20':int, 'BODYSYSTEM21':int, 'BODYSYSTEM22':int,
'BODYSYSTEM23':int, 'BODYSYSTEM24':int, 'BODYSYSTEM25':int,
'BODYSYSTEM26':int, 'BODYSYSTEM27':int, 'BODYSYSTEM28':int,
'BODYSYSTEM29':int, 'BODYSYSTEM30':int,
'CHRON1':int, 'CHRON2':int, 'CHRON3':int, 'CHRON4':int, 'CHRON5':int,
'CHRON6':int, 'CHRON7':int, 'CHRON8':int, 'CHRON9':int, 'CHRON10':int,
'CHRON11':int, 'CHRON12':int, 'CHRON13':int, 'CHRON14':int,
'CHRON15':int, 'CHRON16':int, 'CHRON17':int, 'CHRON18':int,
'CHRON19':int, 'CHRON20':int, 'CHRON21':int, 'CHRON22':int,
'CHRON23':int, 'CHRON24':int, 'CHRON25':int, 'CHRON26':int,
'CHRON27':int, 'CHRON28':int, 'CHRON29':int, 'CHRON30':int,
'DXMCCS1':str, 'E_MCCS1':str, 'HOSP_NRD':int, 'KEY_NRD':int,
'PCLASS1':int, 'PCLASS2':int, 'PCLASS3':int, 'PCLASS4':int,
'PCLASS5':int, 'PCLASS6':int, 'PCLASS7':int, 'PCLASS8':int,
'PCLASS9':int, 'PCLASS10':int, 'PCLASS11':int, 'PCLASS12':int,
'PCLASS13':int, 'PCLASS14':int,'PCLASS15':int, 'PRMCCS1':str})
nrd_dx_pnu = nrd_dx.iloc[pnu.index,:]
del nrd_dx

nrd_hos = pd.read_csv(file4)

# join DataFrames
nrd_core_sev = pd.merge(nrd_core_pnu, nrd_sev_pnu, how='inner',
on=['KEY_NRD', 'HOSP_NRD'])
nrd_core_sev_dx = pd.merge(nrd_core_sev, nrd_dx_pnu, how='inner',
on=['KEY_NRD', 'HOSP_NRD'])
nrd_full_pnu = pd.merge(nrd_core_sev_dx, nrd_hos, how='inner',
on=['HOSP_NRD', 'NRD_STRATUM'])

filename = 'nrd_full_pnu.csv'
nrd_full_pnu.to_csv(filename, index=False, encoding='utf-8')

filename = 'nrd_dis_w_flag2.csv'
nrd_flags = pd.read_csv(filename)
nrd_pnu_w_flags = pd.merge(nrd_full_pnu, nrd_flags, how='left',
on=['NRD_DaysToEvent', 'NRD_VisitLink'])

nrd_pnu_w_flags['DaysFromDischarge'].fillna(0, inplace=True)
nrd_pnu_w_flags['EarlyReadmit'].fillna(0, inplace=True)

filename = 'nrd_full_pnu_w_flags.csv'
nrd_pnu_w_flags.to_csv(filename, index=False, encoding='utf-8')
```

**stroke_records.py**

```
import pandas as pd
```

```
file1 = 'NRD_2014_Core_mod.csv'
file2 = 'NRD_2014_Severity_mod.csv'
file3 = 'NRD_2014_DX_PR_GRPS_mod.csv'
file4 = 'NRD_2014_Hospital_mod.csv'
nrd_core = pd.read_csv(file1, dtype={'AGE':int, 'AWEEKEND':int,
'DIED':int, 'DISCWT':float, 'DISPUNIFORM':int, 'DMONTH':int,
'DQTR':int, 'DRG':int, 'DRG_NoPOA':int, 'DRGVER':int, 'DX1':str,
'DX2':str, 'DX3':str, 'DX4':str,'DX5':str, 'DX6':str, 'DX7':str,
'DX8':str, 'DX9':str, 'DX10':str, 'DX11':str, 'DX12':str, 'DX13':str,
'DX14':str, 'DX15':str, 'DX16':str, 'DX17':str, 'DX18':str,
'DX19':str, 'DX20':str, 'DX21':str,'DX22':str, 'DX23':str, 'DX24':str,
'DX25':str, 'DX26':str, 'DX27':str, 'DX28':str, 'DX29':str,'DX30':str,
'DXCCS1':int, 'DXCCS2':int, 'DXCCS3':int, 'DXCCS4':int, 'DXCCS5':int,
'DXCCS6':int, 'DXCCS7':int, 'DXCCS8':int, 'DXCCS9':int, 'DXCCS10':int,
'DXCCS11':int, 'DXCCS12':int, 'DXCCS13':int,'DXCCS14':int,
'DXCCS15':int, 'DXCCS16':int, 'DXCCS17':int, 'DXCCS18':int,
'DXCCS19':int, 'DXCCS20':int, 'DXCCS21':int, 'DXCCS22':int,
'DXCCS23':int, 'DXCCS24':int, 'DXCCS25':int, 'DXCCS26':int,
'DXCCS27':int, 'DXCCS28':int, 'DXCCS29':int, 'DXCCS30':int,
'ECODE1':str, 'ECODE2':str, 'ECODE3':str, 'ECODE4':str,'ELECTIVE':int,
'E_CCS1':int, 'E_CCS2':int, 'E_CCS3':int, 'E_CCS4':int, 'FEMALE':int,
'HCUP_ED':int, 'HOSP_NRD':int, 'KEY_NRD':int, 'LOS':int, 'MDC':int,
'MDC_NoPOA':int, 'NCHRONIC':int, 'NDX':int,'NECODE':int, 'NPR':int,
'NRD_DaysToEvent':int, 'NRD_STRATUM':int, 'NRD_VisitLink':str,
'ORPROC':int, 'PAY1':int, 'PL_NCHS':int, 'PR1':str, 'PR2':str,
'PR3':str, 'PR4':str, 'PR5':str, 'PR6':str, 'PR7':str,'PR8':str,
'PR9':str, 'PR10':str, 'PR11':str, 'PR12':str, 'PR13':str, 'PR14':str,
'PR15':str, 'PRCCS1':int, 'PRCCS2':int, 'PRCCS3':int, 'PRCCS4':int,
'PRCCS5':int, 'PRCCS6':int, 'PRCCS7':int,'PRCCS8':int, 'PRCCS9':int,
'PRCCS10':int, 'PRCCS11':int, 'PRCCS12':int, 'PRCCS13':int,
'PRCCS14':int, 'PRCCS15':int, 'PRDAY1':int, 'PRDAY2':int,
'PRDAY3':int, 'PRDAY4':int, 'PRDAY5':int, 'PRDAY5':int,'PRDAY6':int,
'PRDAY7':int, 'PRDAY8':int, 'PRDAY9':int, 'PRDAY10':int,
'PRDAY11':int, 'PRDAY12':int,'PRDAY13':int, 'PRDAY14':int,
'PRDAY15':int, 'REHABTRANSFER':int, 'SAMEDAYEVENT':int,
'SERVICELINE':int, 'TOTCHG':int, 'YEAR':int, 'ZIPINC_QRTL':int})

fields=['DRG']
nrd_drg = pd.read_csv(file1, usecols=fields)
# drg_codes = [430, 431, 433.01, 433.11, 433.21, 433.31, 433.8,
433.91, 434.00, 434.01, 434.10, 434.11, 434.90, 434.91, 436]
drg_codes = [61, 62, 63]
stroke = nrd_drg.loc[nrd_drg['DRG'].isin(drg_codes)]

# stroke = nrd_drg.loc[(nrd_drg['DRG']>=434) & (nrd_drg['DRG']<435)]

nrd_core_stroke = nrd_core.iloc[stroke.index,:]
del nrd_core

nrd_sev = pd.read_csv(file2, dtype={'APRDRG':int,
'APRDRG_Risk_Mortality':int, 'APRDRG_Severity':int, 'CM_AIDS':int,
```

```
'CM_ALCOHOL':int, 'CM_ANEMDEF':int, 'CM_ARTH':int, 'CM_BLDLOSS':int,
'M_CHF':int, 'CM_CHRNLUNG':int, 'CM_COAG':int, 'CM_DEPRESS':int,
'CM_DM':int, 'CM_DMCX':int, 'CM_DRUG':int, 'CM_HTN_C':int,
'CM_HYPOTHY':int, 'CM_LIVER':int, 'CM_LYMPH':int,'CM_LYTES':int,
'CM_METS':int, 'CM_NEURO':int, 'CM_OBESE':int, 'CM_PARA':int,
'CM_PERIVASC':int, 'CM_PSYCH':int, 'CM_PULMCIRC':int,
'CM_RENLFAIL':int, 'CM_TUMOR':int, 'CM_ULCER':int, 'CM_VALVE':int,
'CM_WGHTLOSS':int, 'HOSP_NRD':int, 'KEY_NRD':int})
nrd_sev_stroke = nrd_sev.iloc[stroke.index,:]
del nrd_sev

nrd_dx = pd.read_csv(file3, dtype={'BODYSYSTEM1':int,
'BODYSYSTEM2':int, 'BODYSYSTEM3':int, 'BODYSYSTEM4':int,
'BODYSYSTEM5':int, 'BODYSYSTEM6':int, 'BODYSYSTEM7':int,
'BODYSYSTEM8':int, 'BODYSYSTEM9':int, 'BODYSYSTEM10':int,
'BODYSYSTEM11':int, 'BODYSYSTEM12':int, 'BODYSYSTEM13':int,
'BODYSYSTEM14':int, 'BODYSYSTEM15':int, 'BODYSYSTEM16':int,
'BODYSYSTEM17':int, 'BODYSYSTEM18':int,'BODYSYSTEM19':int,
'BODYSYSTEM20':int, 'BODYSYSTEM21':int, 'BODYSYSTEM22':int,
'BODYSYSTEM23':int, 'BODYSYSTEM24':int, 'BODYSYSTEM25':int,
'BODYSYSTEM26':int, 'BODYSYSTEM27':int, 'BODYSYSTEM28':int,
'BODYSYSTEM29':int, 'BODYSYSTEM30':int,'CHRON1':int, 'CHRON2':int,
'CHRON3':int, 'CHRON4':int, 'CHRON5':int, 'CHRON6':int,'CHRON7':int,
'CHRON8':int, 'CHRON9':int, 'CHRON10':int, 'CHRON11':int,
'CHRON12':int, 'CHRON13':int, 'CHRON14':int, 'CHRON15':int,
'CHRON16':int, 'CHRON17':int, 'CHRON18':int,'CHRON19':int,
'CHRON20':int, 'CHRON21':int, 'CHRON22':int, 'CHRON23':int,
'CHRON24':int, 'CHRON25':int, 'CHRON26':int, 'CHRON27':int,
'CHRON28':int, 'CHRON29':int, 'CHRON30':int,
'DXMCCS1':str, 'E_MCCS1':str, 'HOSP_NRD':int, 'KEY_NRD':int,
'PCLASS1':int, 'PCLASS2':int, 'PCLASS3':int, 'PCLASS4':int,
'PCLASS5':int, 'PCLASS6':int, 'PCLASS7':int, 'PCLASS8':int,
'PCLASS9':int, 'PCLASS10':int, 'PCLASS11':int, 'PCLASS12':int,
'PCLASS13':int, 'PCLASS14':int, 'PCLASS15':int, 'PRMCCS1':str})
nrd_dx_stroke = nrd_dx.iloc[stroke.index,:]
del nrd_dx

nrd_hos = pd.read_csv(file4)

# join DataFrames
nrd_core_sev = pd.merge(nrd_core_stroke, nrd_sev_stroke, how='inner',
on=['KEY_NRD', 'HOSP_NRD'])
nrd_core_sev_dx = pd.merge(nrd_core_sev, nrd_dx_stroke, how='inner',
on=['KEY_NRD', 'HOSP_NRD'])
nrd_full_stroke = pd.merge(nrd_core_sev_dx, nrd_hos, how='inner',
on=['HOSP_NRD', 'NRD_STRATUM'])

filename = 'nrd_full_stroke.csv'
nrd_full_stroke.to_csv(filename, index=False, encoding='utf-8')

filename = 'nrd_dis_w_flag2.csv'
```

```
nrd_flags = pd.read_csv(filename)
nrd_stroke_w_flags = pd.merge(nrd_full_stroke, nrd_flags, how='left',
on=['NRD_DaysToEvent', 'NRD_VisitLink'])

nrd_stroke_w_flags['DaysFromDischarge'].fillna(0, inplace=True)
nrd_stroke_w_flags['EarlyReadmit'].fillna(0, inplace=True)

filename = 'nrd_full_stroke_w_flags.csv'
nrd_stroke_w_flags.to_csv(filename, index=False, encoding='utf-8')
```

## Appendix C – SPSS Modeler Stream for AMI Analysis

The Modeler Canvas for the AMI analysis is shown in the image below. Only the AMI image is included here since all of the other streams are identical, with the exception of the Stroke analysis which included extra nodes for the SVM, C5.0 and additional Ensemble models.