Predicting Hospital Readmittance Via Machine Learning Classification Techniques Using Structured and Unstructured Data

Evan Canfield & Christine Personett

DSBA 6156: Applied Machine Learning

November 19, 2019

# Introduction

Hospital readmission rates are an important consideration for hospitals in the United States. Readmission rates are directly connected to Medicare reimbursement, with Medicare being the largest payer of hospital services in the country. Hospitals with high readmission rates incur financial penalties. Readmissions are defined as any return to the hospital within 30 days of being discharged. Currently, the LACE Index is to predict readmissions and provides a score based on four features: the patient’s length of stay, acuity, co-morbidities, and emergency department visits within the last 6 months. This tool is used by physicians currently, and within some studies show poor discrimination in distinguishing between patients that will and will not be readmitted. [1]

We will be exploring the prediction of hospital readmissions through machine learning methods applied to both structured and unstructured data. In analysis of structured data, we will focus on using demographic information and measured readings (i.e., patient temperature, platelet count, etc.). In analyzing unstructured data, we will process digitized doctor reports through natural language processing methods. To both the structured and unstructured data we will be employing modeling methods such as logistic regression, Naive Bayes, and random forest. Through the analysis of both structured and unstructured methods, we plan to improve on the prediction of patients likely to be readmitted.

# Related Work

Currently in some hospitals today hospital readmission is predicted risk assessment tools such as the HOSPITAL score and LACE index. An evaluation of these methods shows the HOSPITAL score showed good discrimination for hospital readmission (C statistic of 0.75) and the LACE index showed poor discrimination (C statistic of 0.58). [1] New models have been researched including using neural networks to predict hospital readmission. The neural network model showed a 20% improvement over LACE. [2]

Additionally, while not research on a predictive model, the following study provided interesting insights and may function as a benchmark case. The research studied the ability of medical practitioners to estimate the risk of a patient being readmitted within 30 days. Depending on the practitioner’s positions, their ability to predict readmission yielded C statistics ranging from 0.50 – 0.58. [3] [4]

For this analysis we will be using the MIMIC-III dataset [5], which will be further discussed in the **Error! Reference source not found.** section. This is a robust data set of medical records used for many different research applications. A difficulty in using this data is that it contains a large amount of missing information. The missing information comes from both human error and cases where data is missing because no data was recorded. For medical reading, only tests that were performed on the patient contain recorded data. This makes it difficult to effectively large amounts of the dataset. The following are instances where people have implemented different strategies for handling the missing data. Methods from all of these instances were considered for handling the missing data:

* MIMIC-Extract: A Data Extraction, Preprocessing, and Representation Pipeline for MIMIC-III [6]
* Benchmark of Deep Learning Models on Large Healthcare MIMC Datasets [7]

# Methods

The analysis in this paper will performed upon two types of data: structured and unstructured. The structured data will be built around patient demographic information as well as laboratory-based measurements and chart data recorded for each patient during each hospital admission. The unstructured data will be built around the electronic reports and notes recorded for each patient during each hospital admission

## Data

The data used for used for this analysis is Medical Information Mart for Intensive Care III (MIMIC-III) database. [5] The MIMIC-III data is a relational database of de-identified and anonymized patient information gathered from Beth Israel Deaconess Medical Center. It contains twenty-six tables with information on patient stays, diagnoses, procedures, prescriptions, lab results, vital signs, and medical notes among other attributes.

The data is freely available to researchers, but requires completion of the CITI “Data or Specimens Only Research” course before allowing access.

## Feature Selection

### Preliminary Exploration

Initial exploration of the data included determining which diagnoses were most common for patients who were readmitted within 30 days and who were not, shown in Figure 1. The diagnosis labels came from the Diagnosis column in the Admissions table.

Figure : Most Common Diagnoses

|  |  |
| --- | --- |
| **Not Readmitted Within 30 Days** | **Readmitted Within 30 Days** |
|  |  |

### Structured Data

The admissions table needed cleaning prior to modeling. The subject id values were given leading zeros if they had five digits, to standardize the ID format. Columns that were not needed were dropped; this included the row id column. Missing values were initially stored as a space; these were replaced with nan values for further processing later. Date values were converted from strings to datetime, this included time of admission, discharge, death, emergency department (ED) registration, and ED discharge. The patients table needed birth time converted from a string to datetime and standardization of subject id before it could be joined to the admissions table.

The admissions table was joined with the patient table in a left join using the subject id as a key. Observations of type newborn were dropped. If there was a value for death time, the observation was dropped. Language, religion, diagnosis and ED date columns were also dropped. There were a significant number of missing values for marital status. The missing values were treated as an additional category instead of dropping or imputing since they may not have been missing at random. Age at admission was a calculated field taken from the difference between admission date and birthdate. Thirty-day readmissions were calculated and stored in the column readmit\_30. Patients were grouped together using subject id and if the difference between a discharge date and admission date was less than or equal to 30 days, then they were counted as having a readmission. Patients with a readmission category of elective were excluded.

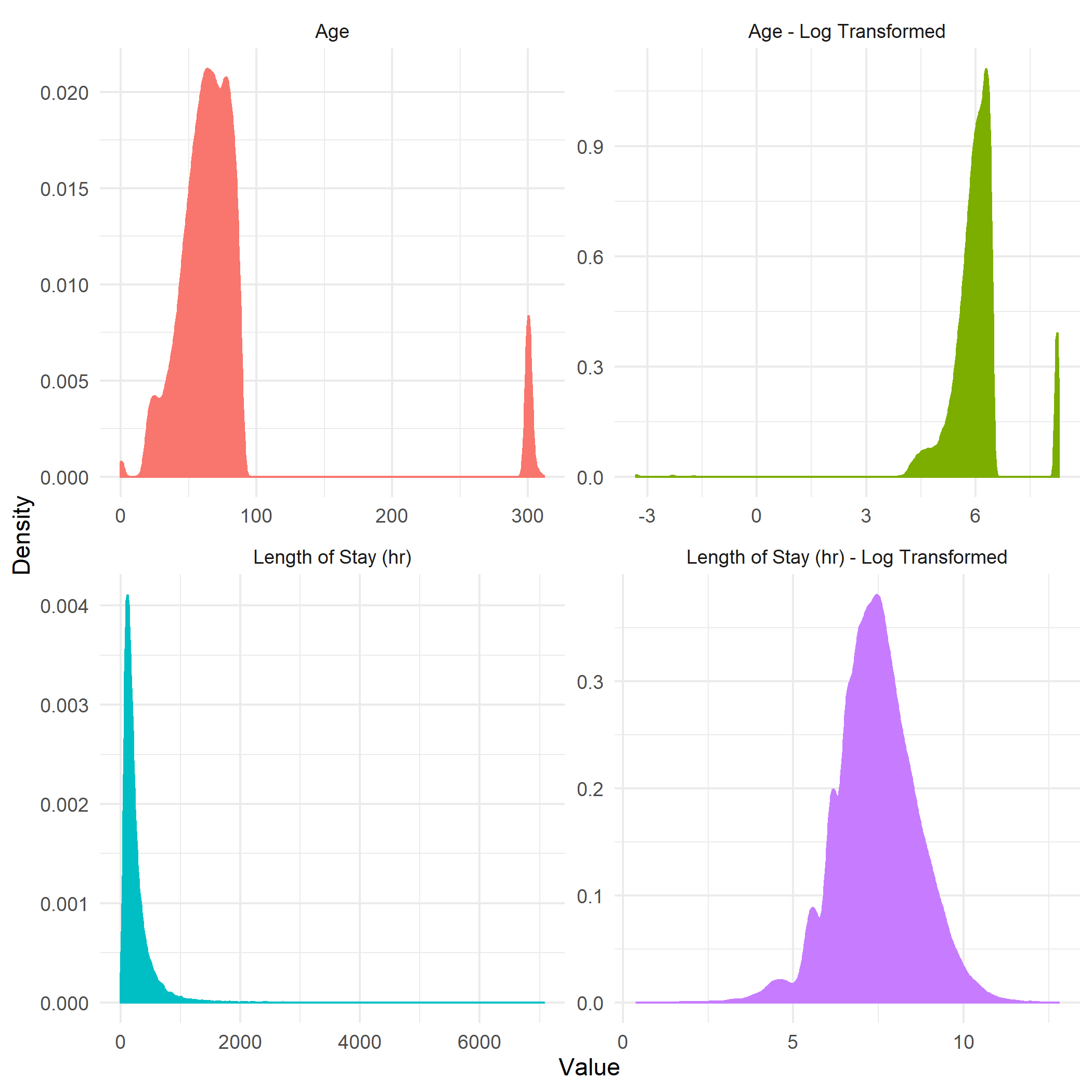
The ICU stay table was joined with the admissions table on hadmid with a right join. This determined whether the patient went to the intensive care unit (ICU) during their hospital stay. If there was nan for the icu stay value, then the hospital visit did not involve the ICU.

The lab events table was also used to gather specific lab values during the hospital visit. The lab events table also included outpatient lab values, therefore if there was not a hadmid associated with the lab, it was excluded. Lab events were filtered by the item id, then the chart time, value and units were extracted for blood urea, platelets, magnesium, calcium and albumin. The hadmid and lab item ids were grouped, and total flags were counted from the flag column in the lab events table. This indicated that there was an abnormal value for the specific lab during the hospital stay. Presence of an abnormal lab value was recorded instead of specific values due to the number of missing lab values. The missing values were not imputed with a normal value because the absence of a lab test, especially in the hospital, does not imply that it would be a normal value. The calculated flag for the lab was categorical with three values including abnormal, no abnormal values and no lab drawn. Each flag value was recorded for the columns of blood urea, platelets, magnesium, calcium and albumin. Then the admissions table was left joined with the modified lab events table.

The microbiology events table was processed similarly to the lab events table. The hadmid column and a calculated column of whether a microbiology test was ordered. The admissions table was left joined to the microbiology events table on hadmid. If there was a lab drawn for the hadmid, then microbiology column was flagged as a one, if there was a null value it was flagged as a zero. The order of a microbiology test indicates that the physician suspected an infection and was sufficient to include regardless of the outcome of the culture.

After the complete data frame was created using the admissions table, patients table, icu table, lab events table, and microbiology events table, additional processing was needed. The distributions of the continuous numeric data inspected. The continuous numeric data was log transformed, and the log transformed distribution also inspected. This can be seen in Figure 2.

Figure : Log Transformations

As the log transformed length of stay having a more normal distribution it was kept, and the standard length of stay was dropped. Log transformation did not help age, and so the age was kept.

The ethnicity variable was condensed to fewer categories. Spaces were also converted to a text character for strings like insurance, marital status, discharge location and admission location. All categorical variables were converted to an n-1 set of dummy variables, these included: admission type, admission location, discharge location, insurance, marital status, ethnicity and gender.

### Unstructured Data

Analysis of the unstructured data uses the Admissions and Noteevents tables from the MIMIC-III database. From the Admissions table, the following two variables are used:

* hadmid, identifying unique hospital admissions
* readmit\_30, derived variable defining whether a patient was readmitted within 30 days of each unique hospital admission (target variable)

From the Noteevents table the following variables are used:

* hadmid, identifying unique hospital admissions
* IsError, identifying a note with an error as identified by a physician
* Category, defines the type of note recorded, with Description
* Description, defines the type of note recorded, with Category
* Text, the note text

The select variables from each table are joined together on hadmid.

All observations that were positive (1) for an error were dropped. There are only 866 notes to which this applies. After dropping notes which contained an error there remained 2,082,294 unique notes.

Two different note datasets were developed. The first dataset uses all of the notes related to each unique hospital admission. The second dataset only uses the discharge summary (and discharge summary addendums) related to each unique hospital admission.

To create these datasets the dataframes were grouped by hadmid and then each note for each hadmid was concatenated together, each note separated by a space. After performing this action, dropping the IsError, Category, and Description columns, and joining the select variables from the Admissions dataset, via the hadmid key, the remaining datasets had the following structure:

* All Notes Dataset: 45,124 observations
* Discharge Summary Dataset: 43,875 observations

Not all hospital admissions had discharge summary reports, resulting in the different sized datasets.

### Text Data Processing

In order to use the text data in the analytical models, natural language processing is required. After cleaning the data, removing all numbers, punctuation, and special characters, the text was converted into tokens. Once tokenized, the text was converted into a numerical matrix two different ways. The first method was to use the frequency count of each token as the matrix value. The second method was to convert the frequency counts using Term Frequency-Inverse Document Frequency (TD-IDF) conversion. Both conversions used a maximum number of features of 3,000.

Further dimensional reduction was also tested. Truncated Single Value Decomposition was performed on the sparse matrices, reducing the 3,000-feature sparse matrix to a 300-feature reduced dense matrix. This dimension reduction was performed to improve speeds during training and validation test.

## Modeling

For this analysis all datasets will be divided into training, validation, and testing sets. The training set will be 60% of the original data, with the validation and test sets each being 20%.

All model tuning is performed on the training and validation sets. Due to data imbalance models are tuned to a weighted F1 score. Final reported metrics are from the test set, which is separated from the training and validation sets until final scoring.

### Models

This is a binary classification exercise. The following classifiers were tested and scored:

* Naïve Bayes
* K Nearest Neighbors
* Logistic Regression
* Random Forest
* Ada Boost
* XGBoost

Support Vector Machines (SVM), Random Forest, and Decision Tree classifiers were also initially tested on the training and validation sets. For the evaluated SVM models, running time far exceed the times for the other classifiers and did not return any additional classification ability compared to the other classifiers. Therefore, SWM classifiers were dropped. Decision Tree and Random Forest classifiers showed extreme overfitting on the training data (AUC = 1.0) and poor performance on the validation data, and were therefore dropped. The other tree-based classifiers (Ada Boost and XGBoost) did not exhibit such extreme overfitting.

### Data Balancing

The data is highly imbalanced. For improved performance, sub-sampling of the negative target values was performed on the training data. This results in a balanced training dataset, with equal numbers of positive and negative target values.

Future plans are to also use the Adaptive Synthetic (ADASYN) sampling approach. This approach is a modification of the Synthetic Minority Over Sampling Technique (SMOTE) approach. The approach performs oversampling on the positive cases, creating synthetic positive cases so that the applied dataset is balanced between the positive and negative target cases.

# Preliminary Results

The following results were generated on the test data sets. For the discharge summary data set, the Count Vectorizer method was used discharge summary notes are approximately the same size for all observations. For the all notes data set, TF-IDF method was used to account for the asymmetric distribution of notes for each hospital admission.

For scoring the models, five-fold cross validation was the preferred method. Due to data imbalance and model performance, some models generated cross validation folds with no positive target predictions. In those cases, scores were generated by a single fold method.

All scoring on the test data set was performed on the non-dimensionally reduced data. Test set scores are the average of five-fold cross validation analysis.

Table 1: Test Set Scores for Discharge Notes Dataset

|  | Naïve Bayes | K Nearest Neighbors | Logistic Regression | Ada Boost  (Decision Tree) | XGBoost |
| --- | --- | --- | --- | --- | --- |
| Accuracy | 0.700 | 0.764 | 0.672 | 0.680 | 0.657 |
| AUC Score | 0.687 | 0.635 | 0.667 | 0.649 | 0.653 |
| Precision | 0.130 | 0.143 | 0.130 | 0.126 | 0.123 |
| Recall | 0.585 | 0.484 | 0.661 | 0.613 | 0.650 |
| F1 Score | 0.212 | 0.221 | 0.218 | 0.209 | 0.207 |
| Prevalence | 0.069 | 0.069 | 0.069 | 0.069 | 0.069 |
| Cross Validation | Yes | No | No | No | No |

Table : Test Set Scores for Complete Notes Dataset

|  | Naïve Bayes | K Nearest Neighbors | Logistic Regression | Ada Boost  (Decision Tree) | XGBoost |
| --- | --- | --- | --- | --- | --- |
| Accuracy | 0.596 | 0.546 | 0.651 | 0.646 | 0.628 |
| AUC Score | 0.616 | 0.618 | 0.657 | 0.645 | 0.637 |
| Precision | 0.104 | 0.100 | 0.123 | 0.118 | 0.113 |
| Recall | 0.638 | 0.701 | 0.664 | 0.643 | 0.648 |
| F1 Score | 0.178 | 0.175 | 0.207 | 0.199 | 0.193 |
| Prevalence | 0.069 | 0.069 | 0.069 | 0.069 | 0.069 |
| Cross Validation | No | No | No | No | No |

# Works Cited

|  |  |
| --- | --- |
| [1] | H. T. Robinson R., "The HOSPITAL score and LACE index as predictors of 30 day readmission in a retrospective study at a university-affiliated community hospital," PeerJ, 2017. |
| [2] | N. A. W. E. S. S. L. E. U. K. Jamei M, "Predicting all-cause risk of 30-day hospital readmission using artificial neural networks," *PLoS One,* vol. 12, 2017. |
| [3] | A. B. J. Steventon, "Preventing hospital readmissions: the importance of considering ‘impactibility,’ not just predicted risk," *BMJ Quality & Safety,* vol. 26, pp. 782-785, 2017. |
| [4] | N. S. J. O. E. e. a. Allaudeen, "Inability of Providers to Predict Unplanned Readmissions," *Journal of General Internal Medicine,* vol. 26, no. 7, pp. 771-776, 2011. |
| [5] | P. T. S. L. L. L. F. M. G. M. M. B. S. P. C. L. a. M. R. Johnson A.E.W., "MIMIC-III, a freely accessible critical care database.," Scientific Data (2016). [Online]. Available: http://www.nature.com/articles/sdata201635. |
| [6] | M. M. C. G. H. M. N. T. G. M. Wang S., "MIMIC-Extract: A Data Extraction, Preprocessing, and Representation Pipeline for MIMIC-III," *ArXiv ,* 2019. |
| [7] | M. C. C. Z. L. Y. Purushotham S., "Benchmark of Deep Learning Models on Large Healthcare MIMIC Datasets," *Journal of Biomedical Informatics,* vol. 83, pp. 112-134, 2018. |
| [8] | "Hospital Readmissions Reduction Program (HRRP)," [Online]. Available: https://www.cms.gov/Medicare/Quality-Initiatives-Patient-Assessment-Instruments/Value-Based-Programs/HRRP/Hospital-Readmission-Reduction-Program. |
| [9] | X. Y. B. &. W. F. Min, "Predictive Modeling of the Hospital Readmission Risk from Patients’ Claims Data Using Machine Learning: A Case Study on COPD," *Scie Rep,* vol. 9, no. 2362, 2019. |