Hospital Readmission Prediction

Anh Vo

I. Introduction

Hospital readmissions are a critical issue in the healthcare industry. Several factors contribute to a patient's readmission to a hospital, including the number of procedures received, medications prescribed, and basic demographics. A deeper understanding of these factors can help healthcare professionals provide better treatment and prevent further visits.

In this study, we aim to predict whether a patient will be readmitted based on their first diagnosis and other factors during their first hospital visit. By analyzing these factors, we can identify the key predictors of readmission and develop strategies to reduce the readmission rates. Ultimately, this study can help healthcare professionals provide better care to patients and improve the overall healthcare system's efficiency.

II. Dataset Investigation

1. Introduction

The diabetes dataset comes from Health Facts Database of Cerner Corporation, Kansas City, MO. It is a collection of hospital records from 130 hospitals in the United States in the range of 10 years, from 1999 to 2008, though it does not contain information on specific year. Before being provided to investigators and published platforms such as Kaggle, all data were already deidentified "in compliance with the Health Insurance Portability and Accountability Act of 1996."

This dataset contains 101,766 observations of 50 variables. Not every observation is unique, as patients who are admitted twice have the same patient id for their revisit. Both inpatient, outpatient, and patients who passed away are recorded. Some additional information that isn't in the dataset is that most hospitals in the study are within the Northeast region (58), 28 in the South, 18 in the Midwest, and 16 in the West. 38 hospitals have bed size less than 100, 78 have bed size between 100 and 499, and 14 have bed size larger than 500.

To get more in-depth information about the data structure and collection process, please refer to these links:

Diabetes 130 US hospitals for years 1999-2008 and Research Paper

2. Type transformation and Missing Values

Upon investigating the data, I noticed some serious problems with some of the variables – some columns are number-coded while not actually being numeric. These were transformed into appropriate data types.

Another problems with the data is the missing values, as always. There are many missing values, or values with no meaning. These include values that are marked as "?", or "Unknown/Invalid" anywhere in the dataset.

Let's take a look at the percentage of missing values are within each columns.

	Missing Percentage
weight	96.858479 %
medical_specialty	49.082208 %
payer_code	39.557416 %
ad_type_description	10.215593 %
ad_source_description	6.944363 %
discharge_description	4.598785 %
race	2.233555 %
diag_3	1.398306 %
diag_2	0.351787 %
diag_1	0.020636 %
gender	0.002948 %
encounter_id	0 %
age	0 %
discharge_disposition_id	0 %

It can be seen that the weight variable has almost 97% missing values, medical specialty with almost 50%, and payer code has almost 40%, we can ignore these variables and drop them from our process. If we include these, it might lead to wrong interpretation. All three of these variables aren't candidates for imputation either.

Also, the missing percentage for Race, diag_3, diag_2, diag_1, and gender is very small and insignificant, so we can just drop all NAs.

However it can be dangerous to ignore of drop the NA values in ad_type_description, ad_source_description, and discharge_description, as they contains both Null values, Not Mapped values (for some reasons), and/or Unknown Values. Therefore, we are going to group the NA values together and name the type = "Other" for these three columns

3. Variables Investigation

Upon investigating, I notice that the columns examine and citoglipton only has 1 unique value within that column. Further examination were conducted to see the distribution of values within each variable. 19 out of 24 features for medications were removed due to extreme imbalance. They aren't going to be useful to our model. Additionally, we want only want to look at the first diagnosis, since we are predicting what happens before and during diagnosis 1 that leads to readmission. That being said, diag_2 and diag_3 are also dropped from our data.

Learning where the patients are transferred would help with predicting whether or not they are going to come back. Discharge disposition includes 29 different categories, and they are collapsed to 4 main categories: Home, Transferred (to another facility or department), Dead (patients in hospice care or pronounced dead at the hospital), and Other (NA). Patients who passed away, however, are not going to be readmitted, so they are removed from the study. I also decided to collapse Admission Sources into 4 main categories: Transfer, Emergency, Referral, and Other.

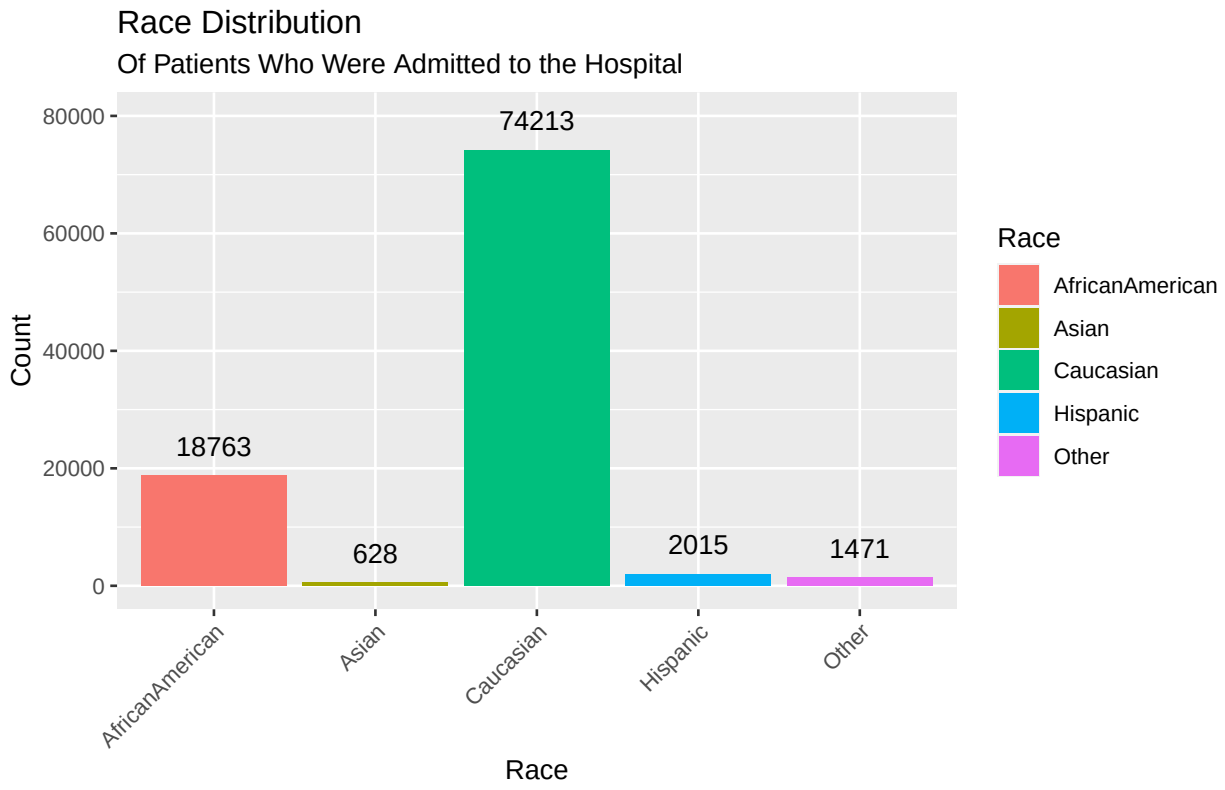
One of the very important variable, diag_1, represents the First Diagnosis of the Patients when they are first admitted to the hospital. This variable is ICD9 coded, meaning that they would have no useful information if we don't map these codes to the actual diagnosis. Here is the table in the research study that shows the complicated mapping of the diagnosis.

A multicollinearity between numerical variables were also checked, and nothing serious were found..

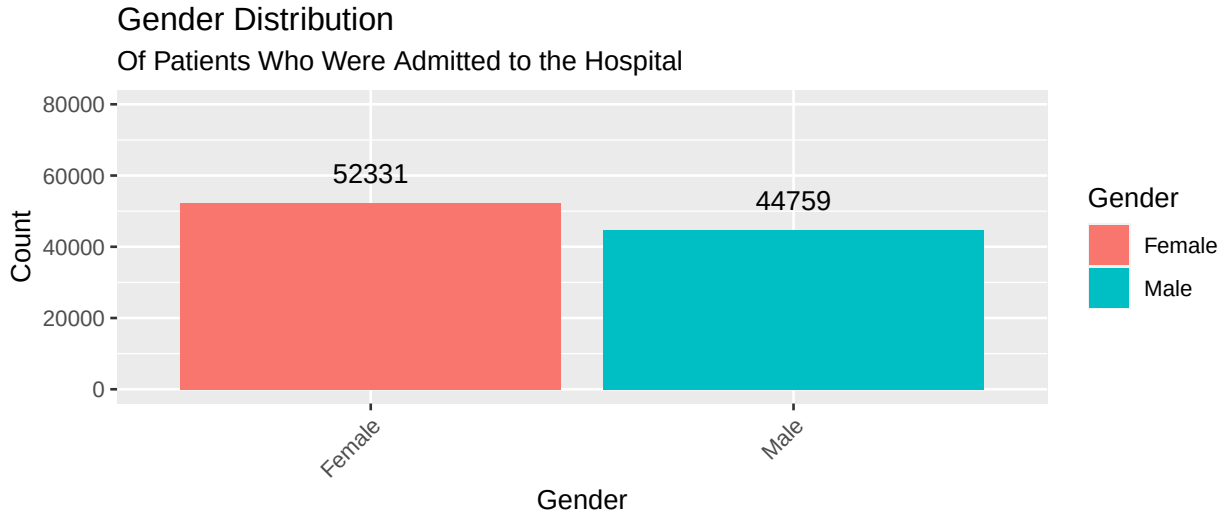
After this stage, the data is finally clean and is ready for our machine learning models!

II. Exploratory Data Analysis

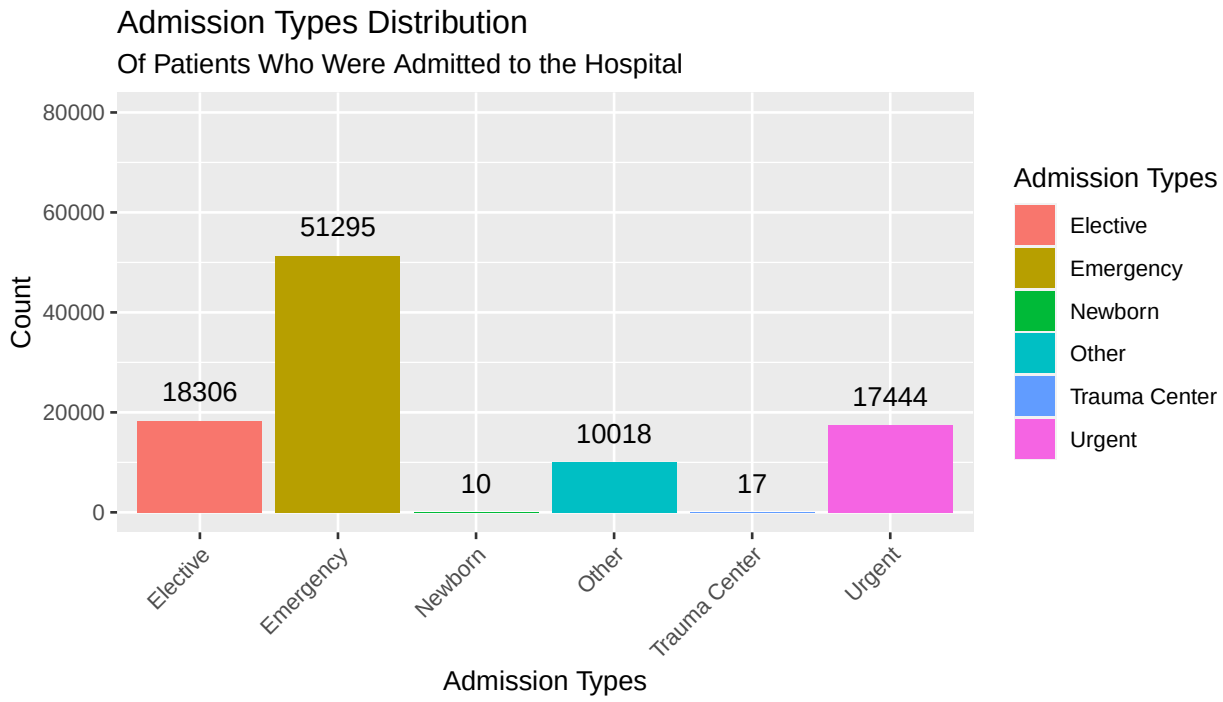
1. Categorical Variables



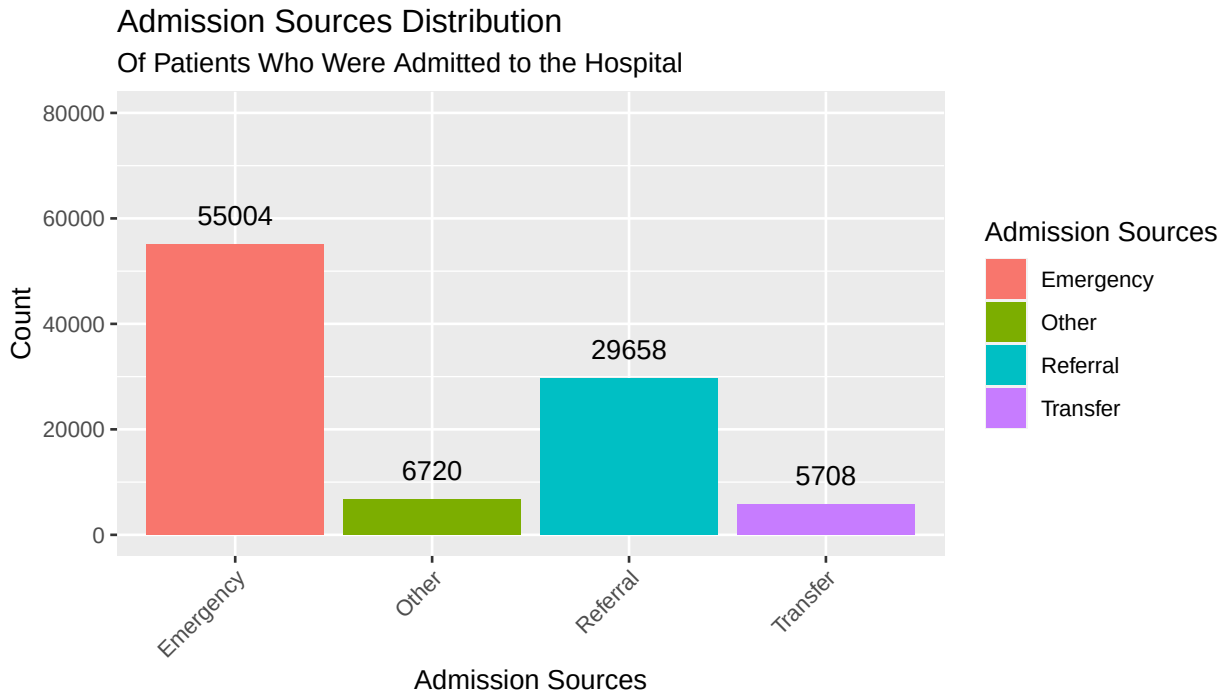
Our population comes mainly from Caucasian and African American patients, so we need to be conscious in interpreting this demographic composition of the population.



There is a moderate balance between our population of females and males in the study.

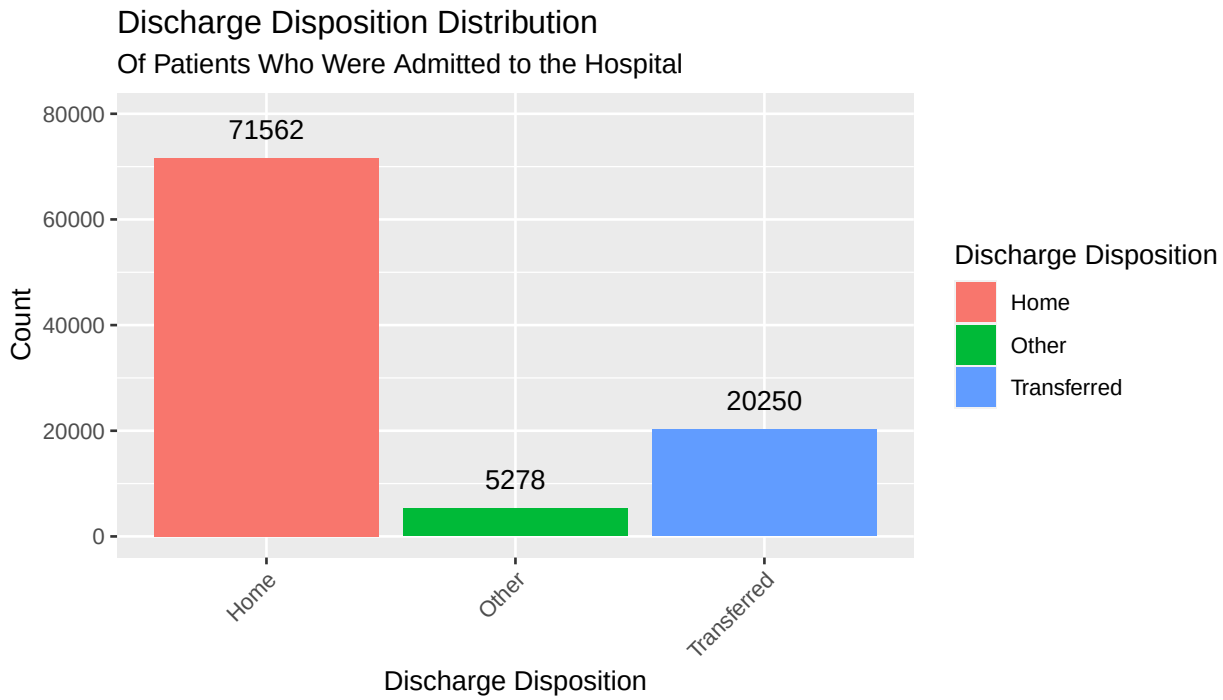


After being collapsed into 6 main categories, there is a very high volume of Emergency admission in our data. We have very low number of New Born and Trauma Center admissions, so we have to be mindful about this in our interpretation.

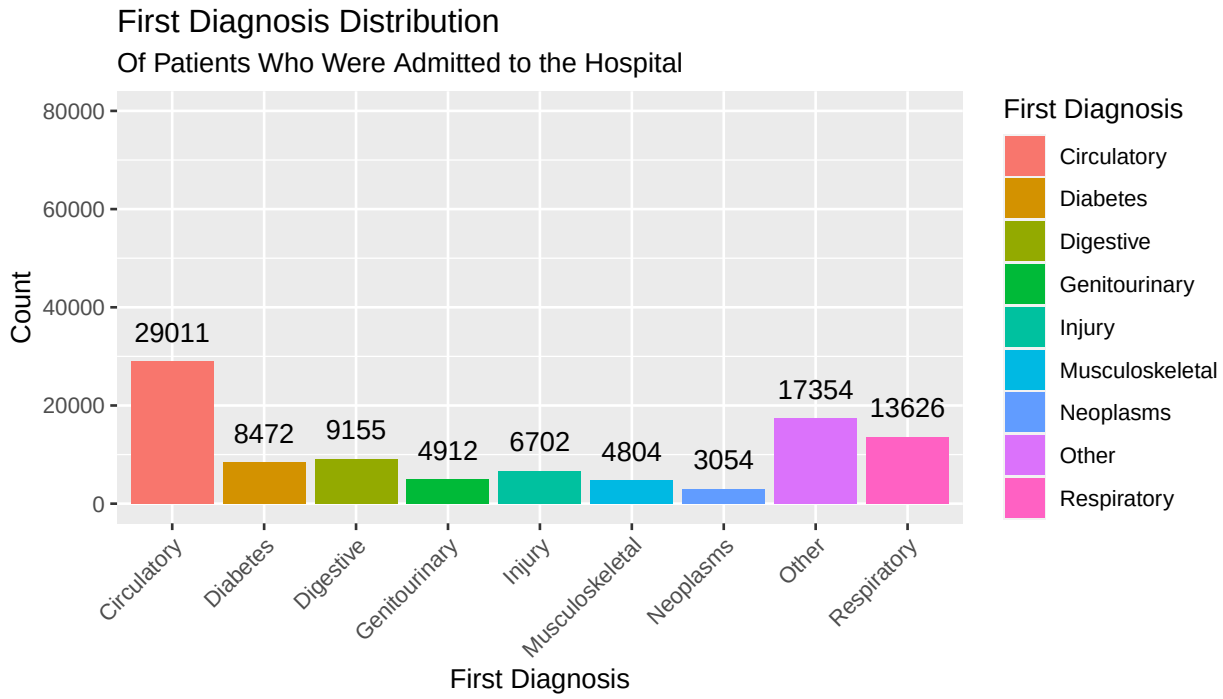


As the admission type is plotted above, the Admission Source is just another general breakdown of why the patients are first admitted, whether it's an emergency (which dominates the population, as also shown in the previous plot), whether they are referred by a physician or clinic, whether they are transferred, and others.

There is also a high volume of patients who are referred to seek medical help.

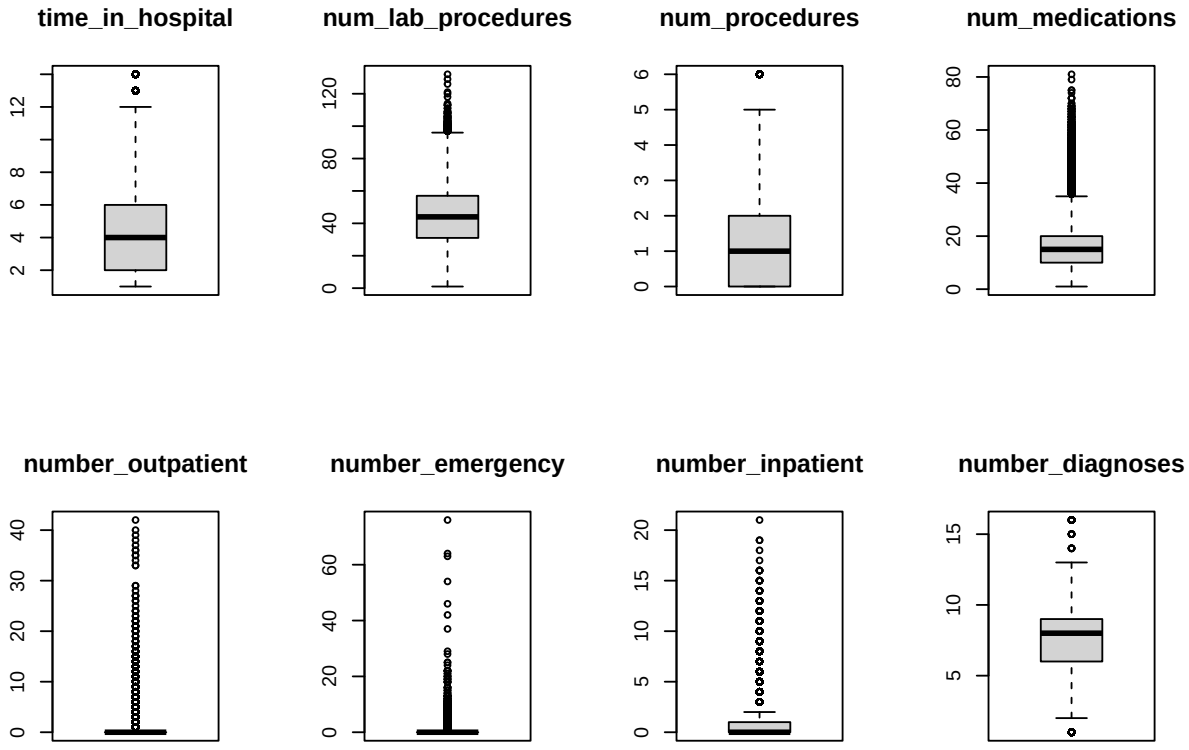


A vast majority of patients are sent home, while some were transferred.

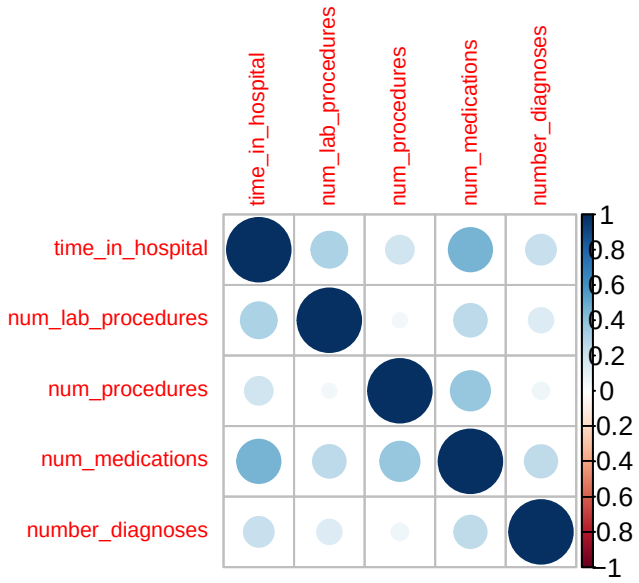


This is the break down of the first diagnosis of patients when they are admitted to the hospital. Diagnoses are pretty balance among all diseases, but we have a large number of patients who are diagnosed with Circulatory, Respiratory, and other diseases that don't belong in the 8 dominant ones.

2. Numerical Variables

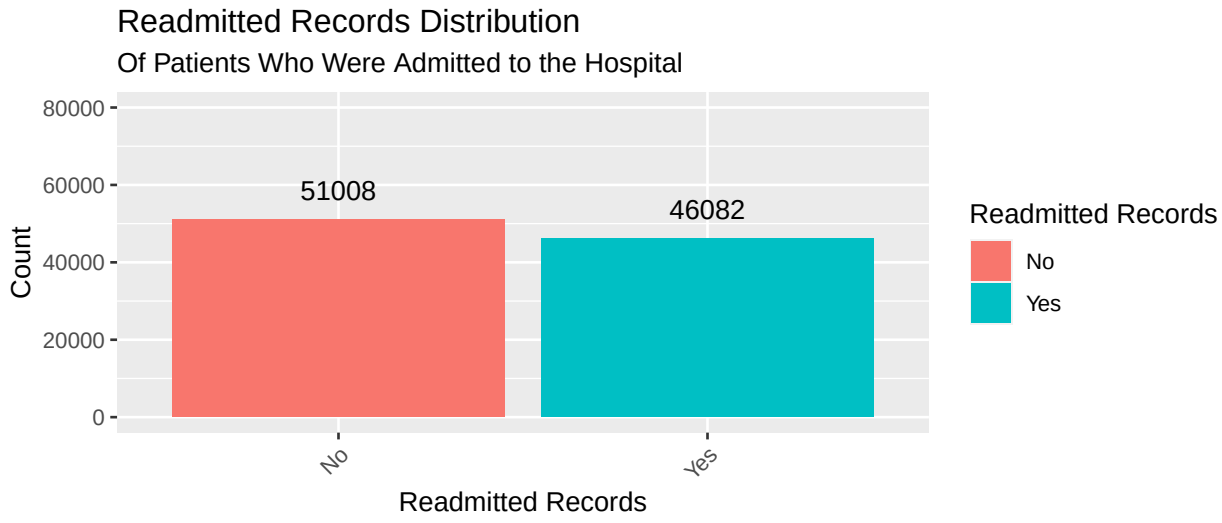


The three box plots regarding number of outpatients, inpatients, and emergency shows the extreme imbalance. It can lead to misunderstanding and misinterpretation if we still include them in the process of building our model. Therefore, these three are eventually dropped.

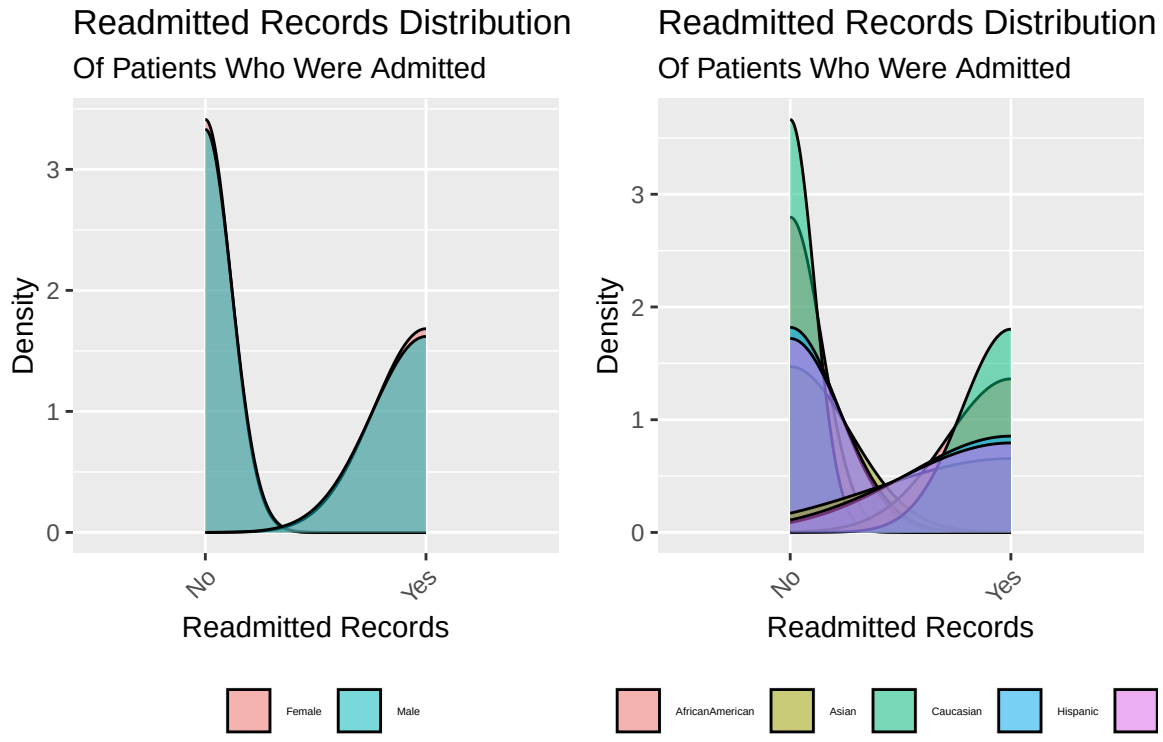


3. Response Variable

One of the most important variable, and is also our class variable, is readmitted. Originally, it represents the days to inpatient readmission, having nominal values with 3 levels: "<30" if the patient was readmitted in less than 30 days, ">30" if the patient was readmitted in more than 30 days, and "No" for no record of readmission. Because we want to look at whether or not the patient is eventually readmitted, this variable was re-factored into two levels: "No" for no record of readmission, and "Yes" for found record of readmission. Here are the general plots of the distribution of values of our response variable.



The readmitted record is pretty balance between patients who are readmitted and patients who are not. Below are the density plots of how readmitted records would look like if we separately by demographics.



III. Models

Although we would like to use all data, it takes a lot of time to train and test these models. Therefore, I reduced the sizes of the data by half for it to be easier in computing.

It should be noted that the patients' ids are not going to be one of our predictor variables, since it is a unique identifier and differs for every individual. There are a total of 97090 observations in our study, after our data cleaning process. After I reduce the data in half size, there are 48545 observations. All models were trained on 38836 observations and test on 9709 observations.

1. Logistic Regression Model

The first and most logical model to start with is the logistic model. Logistic regression is the most simple model in this report, and it is easily interpretable. Because our data has both numerical and categorical variables, logistic regression model can handle this to perform Classification. The data is pre-processed by centering and scaling the variables to have mean 0 and standard deviation 1. 5-fold Cross Validation method was used to estimate the model's performance. 80% of our data is trained using this model and 20% left to testing. Below is the function for building a logistic regression model.

```
logit <- train(factor(readmitted) ~ .,
                    data=train,
                    method="glm",
                    family=binomial(link=logit),
                    preProcess=c("center", "scale"),
                    trControl = trainControl(method="cv", 5))
```

Below is the table for the misclassification rate.

MCR
0.4181687

2. Elastic Net Model

The second model that is used in this report is elastic net – the combination of the strengths of L1 regularization (which encourages sparse solutions by setting some coefficients to zero) and L2 regularization (which shrinks the coefficients towards zero). The data is pre-processed by centering and scaling the variables to have mean 0 and standard deviation 1. 5-fold Cross Validation method was used to estimate the model's performance. 80% of our data is trained using this model and 20% left to testing. Tuning parameters are used to find the best set of values that optimize a performance metric: alpha = 0, 0.2, 0.4, 0.6, 0.8, and 1; lambda: 0, 0.02, 0.04, 0.06, 0.08, 0.10. Below is the function for building an elastic net model.

```
enet <- train(factor(readmitted) ~ .,
                    data=train,
                    method = "glmnet",
                    trControl = trainControl(method="cv", 5),
                    preProcess = c("center", "scale"),
                    tuneGrid = expand.grid(alpha=seq(0, 1, by=0.2),
                                           lambda=seq(0, 0.1, by=0.02)))
```

Below is the table for the best tuning alpha and lambda, as well as the calculated misclassification rate.

alpha	lambda	MCR
0.2	0	0.4186837

3. Random Forest Model

The third model that is used in this report is random forest. Because there are many predictors with complex interactions and non-linearity relationships between the predictors and the response, random forest is appropriate to use. Random forests can also handle outliers very well, as there are some variables with strong outliers. Validation method that was used to estimate the model's performance was "Out-of-Bag" Estimation, since the response is pretty balanced and the dataset is very large. In essence, there isn't a need to separate data into testing and training set here, but I would like to still separate them for calculating misclassification rate. Tuning parameters are used to find the best set of values that optimize a performance metric: mtry: 15 to 18 (4 total). This model was trained and tested on smaller subsets and found that only mtry > 10 produce the best accuracies, and mtry has to be < 19 (19 preds - 1 response). 50 is the number of ntree that is used. Below is the function for building a random forest model.

```
rf <- train(factor(readmitted) ~ .,
                    data=train,
                    method="rf",
                    trControl=trainControl("oob"),
                    tuneGrid=data.frame(mtry=15:18),
                    ntree=50)
```

Below is the table for the best tuning mtry, as well as the calculated misclassification rate.

mtry	MCR
18	0.4270265

4. Neural Networks

The last that is used in this report is neural networks. It is a flexible algorithm that can model highly complex relationships between the predictors and the response. This model is very computationally expensive to train, and we have a large amount of data. But because our data is so large, this can help avoid overfitting. 5-fold Cross Validation method was used to estimate the model's performance. 80% of our data is trained using this model and 20% left to testing. Tuning parameters are used to find the best set of values that optimize a performance metric: size $= 2$ and 6, decay $= 0$ and 0.06. These were chosen based on subsets of the training data that were divided and tested, because this model takes very long to run. It is found that size $= 2$ and 6 tends to perform the best, as well as decay in the with values 0 and 0.06. Below is the function for building a neural networks model.

```
cnn <- train(factor(readmitted) ~ .,
                data=train,
                method="nnet",
                trControl=trainControl("cv",5),
                tuneGrid=expand.grid(size = c(2,6),
                                     decay=c(0, 0.06)),
             trace =F)
```

Below is the table for the best tuning size and decay, as well as the calculated misclassification rate.

size	decay	MCR
6	0.06	0.406015

5. Important Metrics

After our four models were built, each individual model's metrics were calculated to evaluate performance. MCR is the most important one, as it reflects how good the model can fit the data to predict whether or not a patient would be readmitted to the hospital. Accuracy is just 1 - MCR, but it is good to include it also. Three additional metrics were calculated: Precision, Recall, and F1.

Precision is the fraction of true positives (correctly classified positive instances) over the total predicted positives. In other words, it measures how many of the predicted positive instances are actually positive.

Recall, (also known as sensitivity), is the fraction of true positives over the total actual positives. In other words, it measures how many of the actual positive instances are correctly classified.

F1 Score is the harmonic mean of precision and recall, and is a way to balance the importance of precision and recall.

In our situation, we would have to consider these five metrics to evaluate which models performs the best.

In the context of predicting readmission to the hospital, false negatives (patients who are readmitted but not identified by the model) may have serious consequences, such as increased morbidity or mortality. Therefore, in this context, recall may be more important than precision, as it is more important to identify as many readmissions as possible, even if some non-readmissions are identified as readmissions. It is okay to call in someone to "check up", rather than falsely predicting that they don't need readmission, just to regret later in the worst case scenario.

However, false positives (patients who are predicted to be readmitted but do not actually get readmitted) may also have negative consequences, such as unnecessary treatments, additional testing, and increased healthcare costs. If the costs of false positives are high, then precision may be more important, as it is more important to correctly identify only those patients who are actually at high risk of readmission. Healthcare in the U.S. can cost so much, so it's best to balance between precision and recall, which f1 does best.

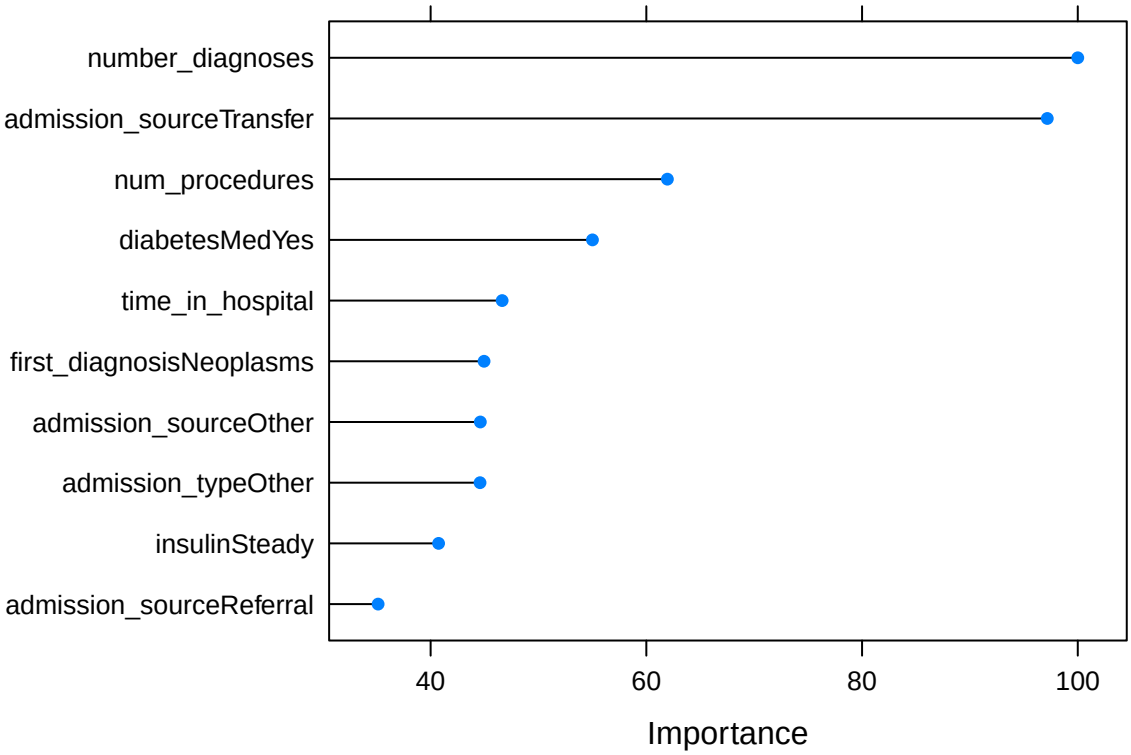
Model	Accuracy	MCR	Precision	Recall	F1
Logistic Regression	0.5818	0.4182	0.6014	0.6374	0.6189
Elastic Net	0.5813	0.4187	0.6006	0.6384	0.6189
Random Forests	0.5730	0.4270	0.5929	0.6326	0.6121
Neural Networks	0.5940	0.4060	0.6196	0.6155	0.6176

As seen from the table above, the logistic regression model has the second highest accuracy –> second lowest MCR. It also has high recall and high f1 score. Therefore, the Logistic Regression Model would be the best model, as it is the perfect balance between precision and recall, has high accuracy and lower misclassification rate. It is also computationally efficient to train, as we observe that it is the fastest in terms of building a model, and it is also very simple and easy to interpret.

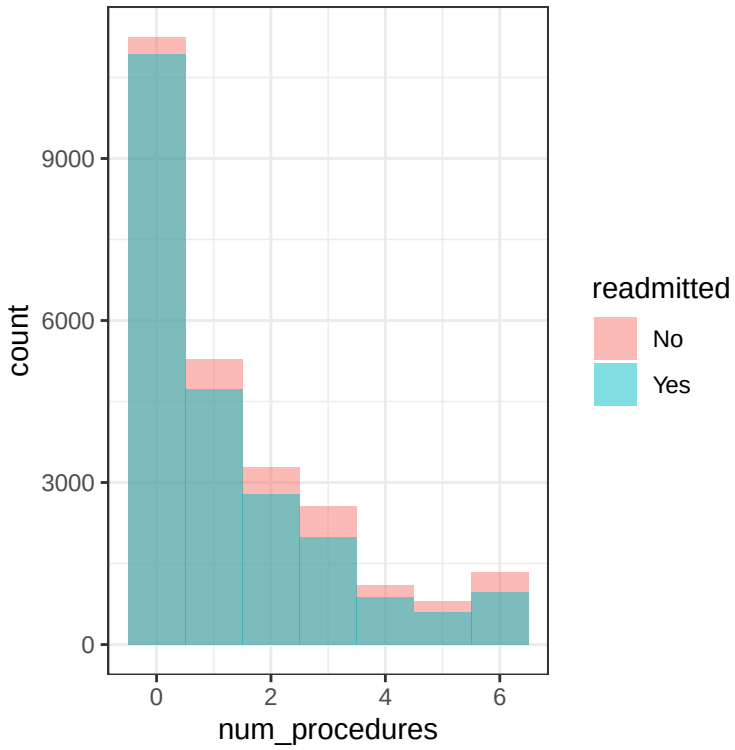
IV. Conclusion

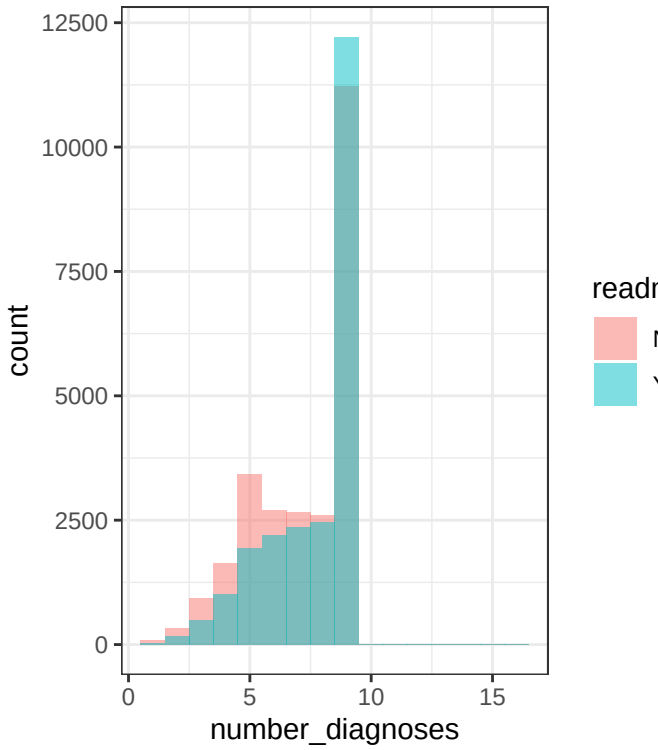
1. Variable Importance

It is essential to test which variables out of the 19 trained and tested ones are the best in predicting whether or not a patient is readmitted. These relevant features can produce insights into the underlying factors driving the model's predictions.



It can be seen that 4 of the most important variables are: number_diagnoses, admission_source, num_procedures, and diabetesMed. Below are the plots of the distribution whether a patient is readmitted, factor out by the two variable: num_procedures and number_diagnoses.





That concludes our study of predicting whether a patient should or should not be readmitted to the hospital based on their patient information. The best model for prediction is Logistic Model, and top six most important variable to look for are: number_diagnoses, admission_source, num_procedures, and diabetesMed.