Hospital Re-Admissions: Final Report

Maggie Liu (ml958) & Yubin Kim (ytk3)

Abstract— **Hospital re-admissions is a recurring problem in the healthcare industry. Many patients return to the hospital due to complications from their initial condition or if their initial conditions have worsened.**

The goal of this project is to preform exploratory data analysis on hospital admissions data and develop a predictive model that may help mitigate the hospital re-admissions problem. The model will try to predict the number of visits and accentuate important features that may affect the number of revisits.

The project used feature engineering and cleaned the data that was provided to better represent the data. Then, the data was modeled with a simple linear regression, ridge and lasso regressions, decision tree and random forest models to find out which modeling technique may work best.

Random forest model obtained the lowest error and performed the best. The most influential predictors were the admission type and the reason for discharge. This definitely shows that there is a correlation of return by patient choice. One of the most influential of the reasons for discharge was elective discharge, when patients worked against what was advised by medical professionals.

I. Introduction

Hospital administrations struggle to balance the financial challenges as well as quality patient care. One problem faced by hospitals is when to release patients. The traffic received in intensive care units and inpatient units at hospitals make it impossible to ensure high quality and efficient service for all patients. Therefore discharged patients may end up being readmitted if they were unable to receive the full care necessary on their previous visit. Medicare defines, the percentage of patients that return after discharge within 30 days is as the re-admission rate.

The goal of this project is to develop a predictive model to predict the total number of visits that an individual patient will make to the hospital as well as evaluate any trends that ca be seen in the data. Hopefully this can inform some features that may cause more re-admissions in some patients over others and improve the quality and level of service and care provided to patients.

II. Exploratory Data Analysis

Our dataset is MIMIC III from the MIT Lab for Computational Physiology. The data gives us information on a patient's admission into a hospital. The data reveals what they were diagnosed with the specific times or admission and discharge.

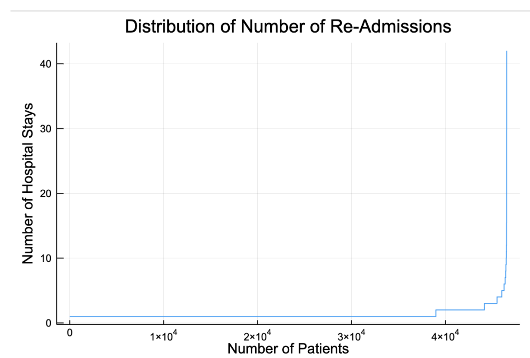

Fig. 1. Distribution of the number of hospital re-admissions.

A. About the Data

Due to the complex nature of healthcare the data was also formatted in a complex way. Unlike most data sets which may be one or two csv's. the data is given as a collection of many tables. The tables of data that we will be using are:

- ADMISSIONS.CSV gives information regarding individual admissions.
- PATIENTS.CSV a tabulated data on individual patient information.
- ICUSTAYS.CSV gives information regarding stays in the ICU.

We will mainly focus on Admissions.csv because it has all the data on admissions at the hospital. The data can be joined with other tables to extrapolate more information.

Our main data, Admissions.csv, was relatively clean with 19 raw features and 58,976 rows of data. The features included: row ID, patient ID, hospital admissions ID, admission time, discharge time, death time, admission time, admission location, discharge location, etc. The data types of these features only consisted of integers and strings. All nominal-valued features were converted to real numbers such that we could more easily fit complex models.

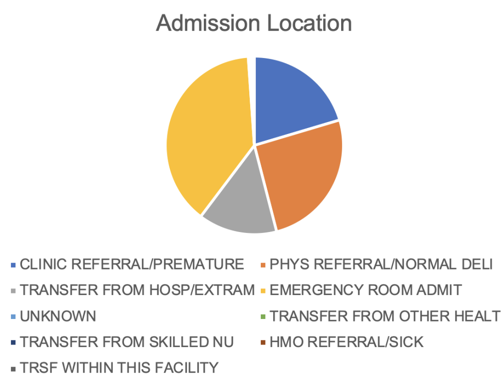

Fig. 2. Distribution of the types of admission locations.

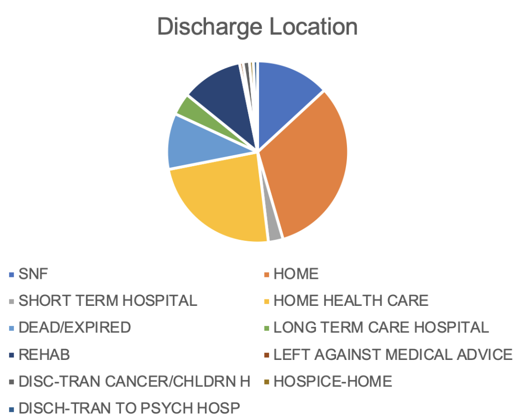

Fig. 3. Distribution of the types of discharge locations.

B. *Limitations in the Data*

Due to patient confidentiality and HIPAA regulations, the data presented is not completely representative. For example, all years in the date are offset into the future by some random offset. Although it would have been interesting to calculate the average number of concurrent patients in the hospital for each patient, this offset makes it impossible to know if certain patients were at the hospital at the same time.

Other limitations that we potentially have is correlation the cannot be detected with just the data. In the real world, there are other factors that cannot be quantified in data,

Additionally, human factors may also limit our ability to fit a model accurately. Two patients who may have the same values in the data may still choose to behave or react to treatment differently and this noise may never be able to be captured into a feature.

C. *Data Cleaning*

Some columns such as admission time and discharge time are fully populated while others such as religion and language contain many missing and corrupted or uninterpretable values. Others such as death time were only populated for patients who died during their stay at the hospital and was missing for all other patients.These missing values and corrupted values were cleaned.

III. Methodology

To develop a reliable predictive model to predict the expected number of visits we will preform feature engineering and transformation on some predictors that will fix over and under fitting. Then we will model the data with linear regression, lasso and ridge regression, decision tree, and a random forest. Our goal is to establish the highest performing model that fits our data and interpret its results.

A. *Using Smaller Data*

We have decided not to use all of the data. The full data has a lot of information that may not be relevant to what we want and may cause more noise and overfitting. If we used all the features present in the data we would have severe overfitting with data that isn't relevant. For example there is data that had vital signs by the hour that may not be relevant in the long run. Having the features would have minimal improvement but increase run time and unnecessarily make the models more complex.

B. Relevance to the Real World

The methodology was carefully thought out and the features that we were going to use. We chose to prioritize some features over others because of the application that it may have. We did not focus a lot of the diagnosis, first, because diagnosis is so variable and there are so many human factors involved, we did not want this model to predict how certain diseases do over others. This would also not be very helpful to explore because of in the real-world, a patient would be more willing to be diagnosed by a doctor and not a machine outputting numbers. There is a psychological and humanisitic aspect that we cannot ignore.

We also tried to focus on features that can potentially be changed. The healthcare industry is a $4 billion dollar industry and still growing. There are so many laws and regulations guarding this industry there are often things that cannot be changed, such as resources. We would like our model to be useful information and not a model that can be used to potentially harm the industry or the care that the customers may be getting.

IV. Feature Engineering and Transformation

Given that our data has much fewer features than the number of data points, we are less concerned with over-fitting our data than with under-fitting. To prevent under-fitting, we converted the existing data features into more useful columns to give us a more meaningful and insightful model. For example, there are the two data features for the time of admission and discharge. These two features are meaningless in context without each other so we convert these data features into one feature calculating the length or duration of stay. Similarly, we have a feature with the specific diagnoses with over 15,000 unique values. In the description of the data, however, it was noted that values in this column should not be used to stratify patients since it is not systematically assigned nor on the same level of granularity throughout i.e. some diagnoses can be more vague or generic than others are.

Other features such as ethnicity and language also had many unique values that could be broadened without any loss in generality. For example with ethnicity, there are 6 general categories defined and recognized (White, Asian, Black/African American, etc.) whereas the data had more specific values such as ”Asian - Vietnamese”. After generalizing, we converted this feature into a one-hot matrix. Our main method of feature engineering was converting categorical/nominal features into real-valued features through one-hot encoding.

All in all, we ended up with around 50 features from feature transformation and engineering. Details of our feature transformations follow as below:

A. Length of Stay

The features, admission time and discharge time were aggregated to form a difference and get the length of stay. This turns two nominal values into a scalable value. This feature was also standardized because of the nature of our other features, are mostly binary or one-hot features and therefore the values of this feature far outweighed all others.

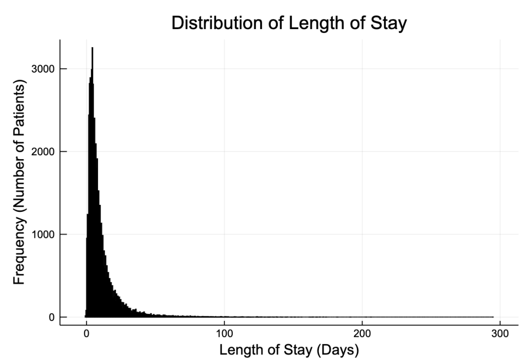

Fig. 4. Distribution of the lengths of stay after feature engineering.

B. Month of Stay

This feature is an addition to determine the month of the patient's stay. This will see if there is any seasonality to the visits, such as during holidays.

C. One-Hot Admission and Insurance Type

There are four existing admission types: Emergency, Elective, Newborn, and Urgent. There are also five types of insurance: Medicare, Private, Medicaid, Government and Self-pay. We transformed these into a one-hot vector as they are categorical and nominal. From initial summary analysis it seems that medicare and private are the most popular insurance types (Fig 5.) and emergency was the majority of the admission types (Fig 6.).

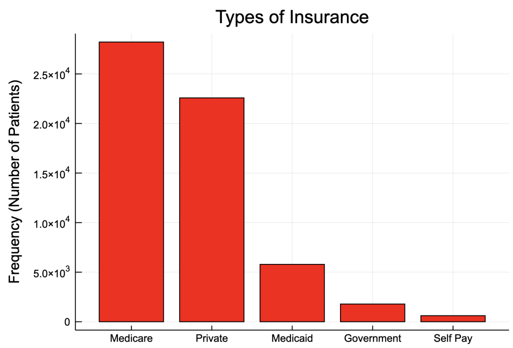

Fig. 5. Distribution of the types of insurance.

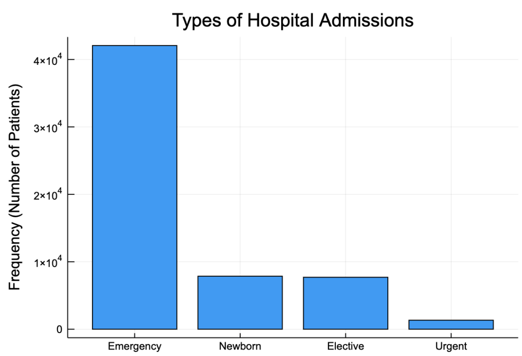

Fig. 6. Distribution of the types of admissions.

D. Convert Language into a Binary

We converted the language feature to have be either English or non-English. With the 76 different languages, making this one-hot didn't seem very necessary because the hospitals are in the US and therefore perform services in one language, English. We have also chosen this method because there seemed to have a lot of corrupted and missing values. Also, converting this into a one-hot vector would create a lot of features which may cause the model to be overfit.

V. Models and Results

We chose to use several different models to evaluate their relative performance and select the most appropriate model. Since we decided to predict the expected number of visits or admissions, which is a real valued number, we chose simple linear regression as our base model to compare all other models to for performance. We chose an ensemble of models that we believed would be well suited to our set of features, the type of prediction that we are pursuing, with varying levels of model interpretability.

We used the mean square error (MSE) and mean absolute error (MAE) as our error metrics since they are well suited to real-valued predictions.

A. Simple Linear Regression

We performed a simple linear regression with an offset to predict the expected number of visits for individual patients. For both MSE and MAE, the train and test error were very similar within 0.01 of each other. We concluded that this model is too simple to predict the expected number of visits for individual patients.

The most interesting finding out of our linear regression model was the feature with the largest coefficient with a value of 2.753: whether or not the patient left against medical advice. This is a logical conclusion: if the patient leaves against the advice of the physician, this is likely a decision that may negatively impact their health or time to recovery and thus increase the chance that a patient will need to be re-admitted to the hospital.

Error	Linear Regression
MSE train	5.716
MSE test	6.053
MAE train	1.141
MAE test	1.146

TABLE I

Error for Linear Regression

B. Ridge and Lasso Regression

Although we do not have too many features, we chose to perform lasso and ridge regression in order to get a better idea of which features were more influential. We tested several regularizers and loss functions and settled on quadratic loss with an l1 regularizer, which seemed to perform the best. Given that there is a not insignificant amount of outliers, we first used an l1 loss function as well, but saw that this did not perform as well in terms of MSE and MAE. We hypothesize that this may be due to the much more significant amount of terms that are not outliers.

Additionally, we believed that the addition of regularization would reduce the sensitivity of our predictions in the presence of corrupted feature values that may have occurred as a result of our feature engineering.

However, given that there were not that many data features to begin with, relative to the number of data points, both lasso and ridge performed more poorly than linear regression, suggesting that features may have been shrunk erroneously or unnecessarily. Ridge and lasso are better suited when there are many more features and one common case is using lasso to determine which features to drop before then performing ridge regression with the remaining features.

Error	Lasso	Ridge
MSE train	5.940	6.263
MSE test	6.289	6.622
MAE train	1.101	1.104
MAE test	1.102	1.102

TABLE II

ERROR FOR LASSO & RIDGE REGRESSION

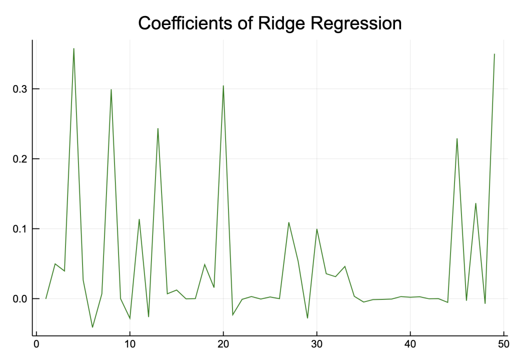

Fig. 7. Coefficients of Ridge Regression

C. Decision Tree

We believe that the pool of patients lends itself well to being structured as a decision tree with different characteristics of different sub-populations. For example, a large portion of patients are actually newborns. The behavior of this sub-population will behave differently than patients coming in for

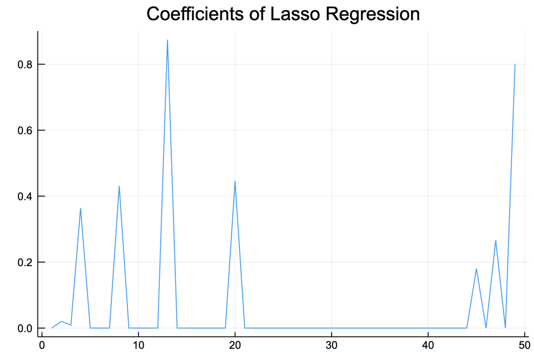

Fig. 8. Coefficients of Lasso Regression

other causes. Therefore, a decision tree is an appropriate structure to try and model. Additionally, decision trees have high interpretability, which makes them a good tool in understanding which features are the most important in our model and can help us to better understand if there exists bias in the data, facilitates use of the model, and other insights.

From the actual tree outputted, we can see which features were split on first, which are the most predictive features. In our model, the first feature split on was the type of admission (e.g. emergency, urgent, etc.), and other notable splits were on admission location (e.g. emergency-room admit).

D. Random Forest

Given that a decision tree is a good model to build, a random forest should perform better given that it builds multiple decision trees and averages the results. Therefore, random forest is a more robust model and as shown in the results, performs the best out of all the models trained. However, random forest models are more difficult to interpret and if one of our goals is the determine which features are the most influential, the lack of interpretability may hinder us.

Error	Decision Tree	Random Forest
MSE train	3.404	3.962
MSE test	7.078	5.668
MAE train	0.774	0.907
MAE test	1.173	1.076

TABLE III

ERRORS FOR DECISION TREE & RANDOM FOREST MODELS

VI. Conclusion and Future Work

In conclusion, we found that a random forest model was able to achieve the lowest train and test error with a decision tree model having a slightly worse performance. Out of our other models, a simple linear regression slightly outperformed lasso and ridge regression, suggesting that the number of features in our model is insufficient to effectively use lasso and ridge.

As noted, some of the most influential features across our models were types of admission (emergency, urgent, etc.) and reason for discharge (e.g. leaving against medical advice).

Additionally, we noted that there were a few outliers in the data where a select number, around 700, of patients had over 10 visits whereas the average number of visits was 1.89. However, to exclude these patients would be unethical since these outliers are not because of an error. Instead, it is likely that these patients have some chronic or rare condition that prompts the large number of visits. Therefore, the MSE of our models was relatively high because of these patients.

In this project, we were very careful about the feature to predict. Among consideration were the probability of a patient being re-admitted, the total number of stays expected for patients, or even a binary "will the patient come back or not within the year". However, with the probability of a patient being re-admitted, we felt that this result may contain the largest risk of being a Weapon of Math Destruction (WMD). This probability could also act as a proxy for the probability that a patient would experience some health event that would cause them to come back to the hospital and this scenario is much more similar to that in the pneumonia example described by Rich Caruana from Microsoft. If our model were to be used in this setting, it could be used as a substitute for letting the patient leave if their probability of re-admission is below some threshold, effectively replacing the judgment of a doctor.

A. Further Improvements and Recommendations

Other possible methodology that we could have used is use a truncated SVD model and other dimension-reducing methodology on all the features of the data to obtain features that would be used in a predictive model. Then use only those features to predict the response. This would also yield all the relevant predictors needed to successfully obtain the expected number of visits.

Some improvements that could be made to our model is further data engineering where we could have classified the diagnosis into triage categories. However, this would mean that we would have to have more knowledge of the medical capabilities which is not in the scope of the project.

B. Weapon of Math Destruction, Fairness, and Other Considerations

In the considerations for deciding whether a model can be a WMD, we must consider three things: how difficult it is to measure outcomes, can its predictions hurt anyone, and whether or not a feedback loop could be created. It is very easy to measure the outcome that we are predicting: the expected number of visits a patient will make and should not create a feedback loop. For example, if we predict that a patient should expect to be re-admitted to the hospital, this should ideally prompt a higher level of service and care to reduce the chance of the patient coming back. In the opposite case, if we instead predict a low number of visits, this should not lower the quality of service provided to the patient as this is unethical and against the Hippocratic oath. Only the second criterion is of some concern and given that it is impossible to have included every influential or all correct features, and that it is very likely we may never know all the features necessary to make such a call, our model should not used as the sole tool for deciding when to release a patient. Therefore, although our predictions may impact a patient's level of care, the predictions made should not influence any individual medical decisions.

Despite this, fairness may not be as important of a criterion since the range of values being predicted is fairly small and is not being used for medical diagnoses or treatment recommendations.

VII. References

MIMIC-III, a freely accessible critical care database. Johnson AEW, Pollard TJ, Shen L, Lehman L, Feng M, Ghassemi M, Moody B, Szolovits P, Celi LA, and Mark RG. Scientific Data (2016). DOI: 10.1038/sdata.2016.35. Available at: http://www.nature.com/articles/sdata201635