Predicting Hospital Readmissions: Final Report

Cierra Beck (cnb44), James Calixto (jc2386), Dana Kim (dgk58)

***Abstract*—Hospital readmission is a large problem currently faced by the healthcare industry. Often, when a patient is released from a hospital, they return within a short period of time because of the same issue or from complications resulting from their original condition. One estimate suggests that for patients discharged from a hospital after treatment for heart failure, 23% return again within 30 days. This situation is not ideal as it is often more expensive for the patient, the hospital, and insurance companies to have the patient return as opposed to having the patient stay in a short-stay unit at the hospital. Reprocessing a patient adds a lot of overhead in recollecting vital signs and collating the patient's continued medical record.**

Our goal with this project is to develop a model that predicts which patients are likely to return to the hospital. Based on this, we would be able to recommend certain patients for longer initial visits or alternative care methods to prevent the risk of return. These results will be beneficial to both the hospital and the patient, as the hospital will ideally be able to adapt their discharge process to prevent patients from having to return. Hospitals will ideally be able to reduce occurrences of patient readmission, improve hospital rating, avoid Medicare penalties, save money on medical costs, and improve their resource allocation.

I. Introduction

For this project, we are trying to develop a predictive model that can assess a patient's likelihood of returning to the hospital within a given time. Medicare defines a hospital readmission as an "all-cause" return to the hospital system within 30 days - if a patient is admitted to any hospital within 30 days of a previous hospital visit for any reason, even if the second hospital visit is completely unrelated to the first, then that counts as a readmission under Medicare regulations.

High hospital readmission rates adversely affect everyone involved in the healthcare system, and reducing these rates will help cut costs for both patients and health care providers. Hospital admission is a relatively long process and it is much more efficient for patients in need of short-term subsequent treatment to stay in a short-stay unit within the hospital. With this model, we hope to be able to make decisions on whether a patient should be discharged (low probability of readmission) or be retained (high probability of readmission), as well as predict what treatment they will need in the near future.

II. Challenges in constructing a predictive model

There are several main challenges in building a predictive model of hospital readmissions, especially one that could potentially be used in the real-world setting.

A. *Large, messy feature space*

Due to the vast range of medical conditions and procedures that exist a lot of the combined data in MIMIC-III are not in a straightforward format. While some features such as age and marital status are just simple values (some discrete/continuous values or some nominal values), there are also features such as doctor notes (transcribed from handwritten notes) or chart events, where a single hospital admission event can have timeseries data of several types associated with it. This precludes easy model generation; to fully utilize the dataset requires subprocessing steps with NLP, timeseries analysis, and so on for each data point in the whole dataset. For example, a pregnant mother might have fields of information on the progression of her pregnancy that would not be present for a male patient.

B. *Potential correlations*

A lot of the data is likely to be strongly correlated. Many of the metrics collected in the hospital setting (blood pressure, heart rate, temperature, etc.) are measurements of the human body, and most illnesses and conditions cause an extensive response throughout the body. For example, the simple flu can cause congestion, nausea, headaches, etc. that are all symptoms of a single underlying condition. Additionally, these conditions can be caused by an innumerable array of illnesses, such as how anything from heat exhaustion to bacterial infection can cause fevers. Therefore any model should be wary of overfitting to data, even if there are many features in the dataset. Regularization and dimensional reduction should be used to reduce the amount of redundant data and ensure model accuracy.

C. Real-world considerations

The healthcare industry is incredibly complicated; it is a $3 trillion industry in the United States alone. Any readmission model will need to conform to countless layers of government oversight and regulation, strict requirements for privacy and ethics, and decades of established procedures and practices. This might place constraints on our modeling approaches beyond the purely technical considerations. For example, while the MIMIC-III dataset that we used is publicly available, real patient data is extremely sensitive and subject to strict HIPAA regulations that might prevent us from storing it on a server to process it or from keeping it in memory to update an algorithm. Thus modeling approaches that can deal with streaming data to update the model without actually keeping data in memory might be more favored approaches to real-time healthcare applications.

III. Our approach

For this project, we will be using the MIMIC-III (Medical Information Mart for Intensive Care III) dataset, which collates data from over 40,000 patients at critical care units at the Beth Israel Deaconess Medical Center between 2001 and 2012. This dataset is very thorough; it includes information such as patient demographics, laboratory tests and results, prescribed medications, vital sign measurements, and so forth. Due to its thoroughness as well as the complicated nature of hospital records systems, the data is split up between 26 different .csv files which, together, total 43.3 GB. Additionally, text fields such as caregiver notes are not standardized and often contain misspellings as well as non-standard characters such as newlines. Thus, the MIMIC-III dataset constitutes a big and messy dataset and requires cleaning and preprocessing.

We chose to look at a subset of the entire MIMIC-III dataset. The CSV files that we used were:

- ADMISSIONS.csv, giving information regarding patients' individual admission to the hospital
- D_ICD_DIAGNOSES.csv, a definition table for ICD diagnoses
- DIAGNOSES_ICD.csv, a lookup table for the ICD-9 codes associated with each diagnosis
- PATIENTS.csv, a lookup table for information on each patient's chart data
- SERVICE.csv, a table that lists services that a patient was admitted and transferred under

There were several reasons why we chose to look at a subset of the data, and why we specifically excluded other csv files that were provided in the MIMIC-III dataset:

A. Computational intensiveness of big data

While there are many techniques to working with large datasets, from dimensionality and size reduction techniques such as PCA to using Python libraries designing for out-of-core data manipulation (such as Dask) to simply obtaining access to better hardware with more processors and RAM, our largest data file, CHARTEVENTS.csv, was 34.5 GB. Even simple search and sorting operations would take a while given the 330 million rows of this file. Other files were smaller (such as NOTEEVENTS.csv, INPUTEVENTS_CV.csv, LABEVENTS.csv) were an order of magnitude smaller but even then it would take a while to perform useful operations on the data. Of course, working with big and messy datasets can be a necessity, but we forewent the largest of our datasets after also considering the next point:

B. Usefulness of smaller datasets

More importantly, many of the larger datasets were event logs, where events such as hourly vital signs, lab results, and so forth are logged. Hourly chart events were surprisingly irrelevant to our overall goal of predicting hospital readmission rates, as even in aggregate this would not yield much information about a patient's hospital stay as a whole. The information contained in these events such as heart rate, blood pressure, etc. are strongly correlated with other factors such as age and gender (both features that we did include) and there would be minimal improvement from adding gigabytes worth of data.

C. Relevancy to real-world implementation

A major factor that we carefully considered was the real-world utilization of our models. Our big data analytics system will have to coexist with other functions of hospitals and health service companies, and we analyzed our methodologies and procedures in that context. For example, a particularly computationally intensive method might give better accuracy, but the capital involved in the procurement and upkeep of the servers required might be more efficiently spent on the short-stay units themselves. Additionally, we factored in the usage of our models on both the healthcare service provider side as well as the patient side. Integration with health apps would favor the usage of simpler metrics to predict hospital readmission, allowing the patient to be more proactive in deciding whether the should stay at the hospital or not. As such, we favored streamable algorithms and/or algorithms with largely precomputable steps: algorithms that can be applied to realtime data with low overhead since constantly processing multi-gigabyte history files in realtime would be prohibitively resource-consuming.

Again, the MIMIC-III dataset is a real-world dataset with many features. There were many possible directions we could have taken with this project so we decided to focus on specific predictions in order to reduce the scope of our project to a manageable size.

While looking through the data we noticed that some fields were missing. Due to the nature of health care, patients come in with many different conditions and not all fields will be applicable to all patients. For example, if the patient is admitted directly into the emergency department, they will have an associated in/out time specific to the emergency department that other patients will not. Additionally, some patients are simply missing data; ex. marital status, religion, etc. We will have to ensure that when performing aggregate analyses over the whole dataset, we look at features common to all patients; using data specific to certain classes of patients is more helpful when analyzing that particular subgroup.

Our data also does not contain a feature that we think would be extremely helpful in analysis: length of stay. To add this to our dataset, we will be adding this field where the value is equal to the difference in time between admission time and discharge time. Our data does not explicitly contain age, so we will be adding a field to represent the patients age at the time of their admission.

Since our dataset is a medical dataset, it has to comply with HIPAA data de-identification standards. Because of this, patient name, phone number, and address was removed from the dataset. This is not a problem for our analysis, since this information was not important to us, we just needed to unique identifier for each patient. However, HIPAA also required that the dates were changed, which posed a problem for us. All dates were shifted into the future by random offsets, that resulted in the stay dates occurring between 2100 and 2200. In addition, all patients with ages over 89 had their D.O.B. shifted, so all patients with ages over 89 appear to be over 300 years old. For these patients, we will consider all to be aged 89, so this will be the max age in our dataset.

IV. Summary Statistics and Visualizations

After preparing our dataset, we have the statistics for 58,976 total admissions, which includes 46,520 unique patients. These total admissions includes 3,390 readmissions within 30 days or less. Some summary statistics are shown below in figures 1 through 4.

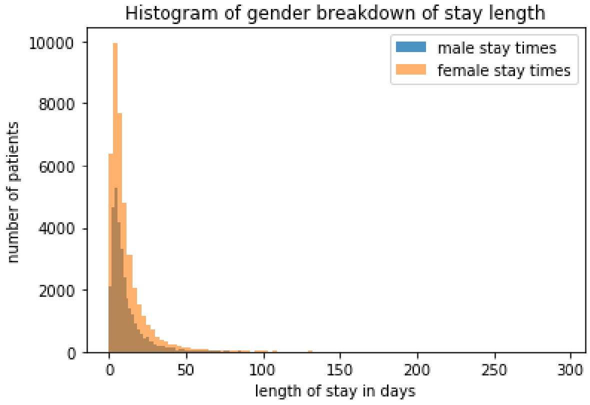

Fig. 1. Histogram of male vs. female stay times

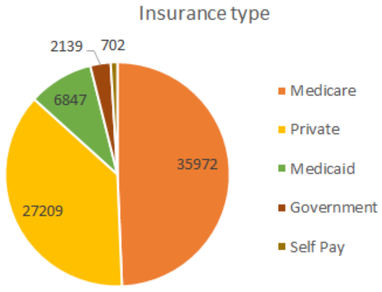

Fig. 2. Type of insurance used by patient

V. Model Development and Testing

In the course of our data analysis, we constructed several models and tested and validated them on our sample data.

A. Categorical features to one-hot encoding

Many of the features in MIMIC-III are categorical, such as marital status, insurance status, ethnicity, and so on. These values are not ordinal, so in order to use these features with classification and regression algorithms we used one-hot encoding to transform these features into boolean columns.

B. Truncated SVD dimensionality reduction

We used a truncated SVD method from the sklearn package to perform dimensionality reduction on the sparse matrix resulting from one-hot encoding. We chose truncated SVD versus a standard PCA implementation because one-hot encoding produces very sparse boolean matrices that do not require centering and thus the SVD method works quickly and efficiently. We can visualize the percentage of explained variance vs. the number of eigenvalues (aka the number of dimensions to reduce to) as in figure 5.

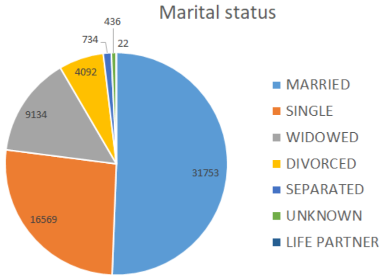

Fig. 3. Marital status of patient

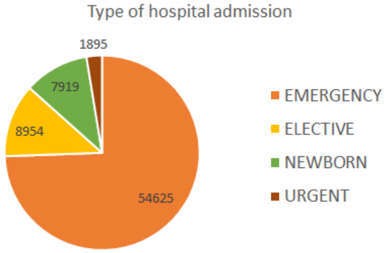

Fig. 4. Type of hospital admission for patient, as logged

Performing dimensionality reduction in this way has two benefits:

- We can reduce the size of the matrix significantly; without a sparse matrix implementation, our one-hot encoded columns for the categorical variables require over 2 GB of memory to store. We can reduce the number of columns from 15766 to 100 while still preserving 83% of the variance.
- Since dimensionality reduction combines columns in linear weights, we can examine these combinations to determine conjunctions of factors that SVD reduction identifies as being strongly correlated. This allows us to discover important clusters of correlated factors that could potentially be strong predictors of stay length.

C. Model creation

We chose to transform the readmission variable into a boolean $[0, 1]$ feature, where 1 indicates that the patient later returned to the hospital within 30 days for any reason and 0 otherwise. There were several reasons why we chose this as our prediction variable:

- As previously noted, this is the Medicare definition for readmission. This makes our analyses consistent with other published literature on readmission rates, and allows us to compare our results with other publicly released data. Additionally, compliance with Medicare target readmission rates is a large driving factor behind implementations of readmission predictors; the government penalizes hospitals that have higher readmission rates. These fines total in the millions of dollars (there was $528 million in total estimated Medicare fines in the United States in 2017), which is a huge financial incentive for hospitals to use this standard for compliance.
- Regression on the actual time to patient return would be inconvenient, as some patients never return to the hospital in this dataset. We would either have to use a mixed variable with the option of taking a numerical value or infinity (which would be cumbersome to work with and lead to strange coefficients) or set some arbitrary value to represent a patient not returning to the hospital (which would also affect the coefficients).

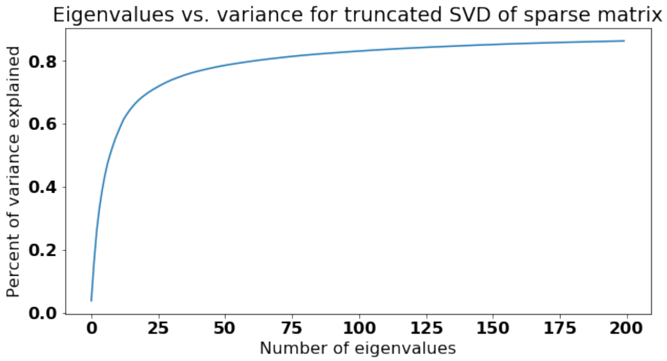

Fig. 5. Plot of variance vs number of eigenvalues

To classify our data, we used several classification methods that we learned in class to construct a model for our data:

- Three versions of the perceptron algorithm. One had no regularization, one used L1 regularization, and one used L2 regularization. The perceptron algorithm is fast, takes up very little memory, and can ”stream” data by continuously updating its coefficient matrix on incoming data. The L1 and L2 regularization prevents overfitting, considering how many features we have in our feature vectors, and we suspected that L2 regularization would work better on a dataset that had already undergone dimensional reduction.
- The k-nearest neighbors classifier. We suspected that similar patients might cluster together in our feature space, allowing us to classify them more accurately, but we also knew that if these clusters were sparse or nonexistent then the classifier would likely do badly due to the high dimensionality. (We set $k = 5$.)
- Random forest and extra trees classifers. The random forest classifier is an ensemble tree method

that creates multiple decision trees with randomly chosen subsets of the features in the feature vector. This prevents any strong predictors from overly affecting the growth of each decision tree and prevents overfitting. Extra trees takes the principle behind random forest one step further - each feature split of the tree is given by a random split instead of the optimal split as measured by the Gini impurity (as is the default in the sklearn package).

VI. Results

A. Predictive features

Using the resulting coefficient matrix given by performing truncated SVD dimensional reduction on the sparse one-hot encoded dataset, we can obtain that the ten features with the largest coefficients (and therefore the most effect on classification) are given in table I. (We filtered by categories with 1,000 or more patients to eliminate thinly populated categories that had undue influence on the classification.)

TABLE I
Strongest predictive features

feature/categorical variable	coefficient	# patients
ADMISSION_TYPE_URGENT	0.025438	1895
INSURANCE_Medicaid	0.020718	5705
INSURANCE_Private	0.01924	21249
ADMISSION_TYPE_ELECTIVE	0.018049	8951
DIAGNOSIS_PNEUMONIA	0.017817	1886
DIAGNOSIS_CONG. HEART FAIL.	0.015021	1273
DIAGNOSIS_SEPSIS	0.013411	1405
CURR_SERVICE_ORTHO	0.012064	1101
MARITAL_STATUS_DIVORCED	0.011594	4092
CURR_SERVICE_TSURG	0.011359	1338

Many of these categories make intuitive sense. People who choose to enter the hospital system of their own accord (ADMISSION_TYPE_ELECTIVE), for instance, tend to be people who are proactive towards their health, so they are likely to return if they have susbequent issues or complications. Note that these are the one-hot encoded headings, so the coefficients are naturally small since many categorical features had a lot of associated categories and possible values.

B. Classifiers and predictive models

We ran a variety of different classifiers on the dataset consisting of the age and stay length variables as well as the 100-feature matrix generated by applying SVD dimensionality reduction to the boolean one-hot categorical feature representations. We cross-validated the accuracy of these classifiers using k-fold cross-validation where $k = 10$.

TABLE II
Classifier results on sparsified matrix

method	accuracy	std. dev
perceptron (no regularization)	0.7149	0.0021
perceptron (L1 regularization)	0.7149	0.0021
perceptron (L2 regularization)	0.7216	0.0022
k-nearest neighbors classifier	0.6942	0.0045
random forest classifier	0.7499	0.0037
extra trees classifier	0.7377	0.0046

As we can see, all methods performed relatively well on the dataset, achieving classification accuracies (as measured by zero-one loss) of about 70%. Furthermore, the standard deviation was very low across the k-fold validation (less than half a percentage point), indicating that these methods are consistently classifying at about 70% accuracy.

The perceptron methods ran extremely quickly and had roughly the same accuracy. Using L2 regularization gave slightly better accuracy as compared to L1 regularization or no regularization. This makes sense since we have already done dimensional reduction from 15766 features to 102 and L1 favors sparse coefficient matrices with large coefficients; we have already reduced the dataset to features with maximum variance so using L2 (which tends to give less sparse coefficient matrices than L1) uses more of the data. Furthermore, the k-nearest neighbors classifier was slightly worse than the other methods, likely due the high dimensionality of the feature vectors even after SVD.

We can ask more specific questions to further assess the characteristics of these classifiers. First we can consider precision and recall instead of a general accuracy metric. Precision is defined as the ratio of true positives to all positive responses, while recall is defined as the ratio of true positives to all positive events. These are often better metrics when the postive/negative event ratio is lopsided; with an imbalanced enough ratio, a ”classifier” that always outputs positive (or negative) can appear to do well with an accuracy metric.

We ran k-fold cross-validation on each model with $k = 10$ separately for precision and recall to obtain the results in table III. (The plus/minus term is the standard deviation).

Note that the precision and recall percentages are much lower than the overall accuracy percentages. The best percentage in this table is the random forest classifier's precision value of about 0.5, indicating that only half of its positive responses predict the patient's behavior accurately. While this is pessimistic in terms of our model's accuracy, it is encouraging when considering that only about 25% of the patients are readmitted within

TABLE III
PRECISION AND RECALL MEASURES

method	precision	recall
perceptron (no regularization)	0.188 ± 0.014	0.039 ± 0.003
perceptron (L1 regularization)	0.188 ± 0.014	0.039 ± 0.003
perceptron (L2 regularization)	0.210 ± 0.019	0.038 ± 0.003
k-nearest neighbors classifier	0.308 ± 0.014	0.172 ± 0.008
random forest classifier	0.503 ± 0.009	0.351 ± 0.008
extra trees classifier	0.474 ± 0.010	0.353 ± 0.007

a month, meaning that there is a lot of noise when it comes to the positive events that the model is trying to predict. In that regard, our random forest is a decent classifier.

The precision and recall metrics are also important in the real-world context of deciding which algorithm or model to use. The accuracy metric weighs false negatives and false positives equally, which is not a realistic assumption to make. In practice, the cost for false positives (predicting a patient will return to the hospital when they will not) is likely much lower than the cost for false negatives (predicting a patient will not return to the hospital when they will not). The only cost for the former is the cost involved in having the capacity to process the (nonexistent) returning patient, whereas in the latter case, the hospital and healthcare service providers face Medicaid penalty fees, the costs of reprocessing and retreating the patient, and so forth. A proper analysis of any readmission model should therefore include some sort of cost function from empirically determined data, and the precision and recall metrics will help the model optimize its parameters based on the cost function.

Another question we can ask to assess the usefulness of these models is *when* they are most accurate. Although they all appear to be about 70% accurate across all patients, perhaps they are more or less effective on subsets of the patient population. To investigate this, we wrote a function to loop through all categories of our categorical variables and apply the perceptron algorithm with L2 regularization and the random forest classifier. (We chose these two because the perceptron algorithm is fast and streamable while the random forest classifier had the best accuracy over the whole test set.) We cross-validated the accuracy of these classifiers using k-fold cross-validation where $k = 5$.

Note that entries marked with an asterisk (*) had too few data points in the category to give a useful accuracy rating when cross-validating. Also remember that the entries in the tables are not the readmission rates themselves, but the mean model accuracy for both models for all patients in that category. The plus/minus variables are the standard deviations.

TABLE IV
ADMISSION_TYPE ACCURACY

category	perceptron, L2	random forest
EMERGENCY	0.267 ± 0.000	0.736 ± 0.005
ELECTIVE	0.825 ± 0.003	0.831 ± 0.009
URGENT	0.391 ± 0.125	0.700 ± 0.028
NEWBORN	0.967 ± 0.039	0.967 ± 0.062

TABLE V
INSURANCE ACCURACY

category	perceptron, L2	random forest
PRIVATE	0.641 ± 0.195	0.753 ± 0.006
MEDICARE	0.437 ± 0.212	0.745 ± 0.003
MEDICAID	0.742 ± 0.004	0.714 ± 0.012
SELF PAY	0.724 ± 0.296	0.844 ± 0.015
GOVERNMENT	0.651 ± 0.207	0.735 ± 0.021

This is much more useful to the creation of a predictive model than our overall classifier accuracies from table II. For example, note that when considering the ADMISSION_TYPE feature as in table IV, readmissions for patients admitted under the EMERGENCY category are very badly predicted by the perceptron model but have well-predicted by the random forest model. Conversely, patients with MEDICAID insurance (as seen in table V) are better predicted with the perceptron model than the random forest model.

We can also look at what categories of patients are better predicted than others. Babies given the NEWBORN status are extremely predictable in terms of readmission, but patients admitted under services ENT or OMED (in table VII) are much less predictable than the average patient. We can use this data to speculate on sources of uncertainty in the the readmission process and find further leads for investigation. For example, patients admitted to ENT (ear, nose, throat) services might be unpredictable since a lot of ear, nose, and throat issues are infections, which may or may not clear up within a month (necessitating a subsequent visit if the condition does not improve).

Using this data, we can refine our prediction approach.

VII. FURTHER IMPROVEMENTS

There are several natural directions for further improvements to create a more accurate readmission model.

A. Improvement of precision and recall metrics

Analysis of empirical data of costs incurred as a result of patient readmissions could yield a cost function that gives the average cost of readmitting a patient versus the overhead cost in increasing hospital capacity. Thus we could use this cost function to optimize the

TABLE VI
MARITAL_STATUS ACCURACY

category	perceptron, L2	random forest
MARRIED	0.627 ± 0.178	0.744 ± 0.003
SINGLE	0.542 ± 0.247	0.744 ± 0.007
NaN	*	*
DIVORCED	0.746 ± 0.007	0.742 ± 0.008
WIDOWED	0.636 ± 0.166	0.756 ± 0.010
SEPARATED	0.686 ± 0.012	0.645 ± 0.031
UNKNOWN (DEFAULT)	0.681 ± 0.053	0.677 ± 0.040
LIFE PARTNER	0.466 ± 0.339	0.600 ± 0.326

TABLE VII
SERVICES ACCURACY

category	description	perceptron, L2	random forest
MED	internal med.	0.621 ± 0.189	0.750 ± 0.004
CSURG	cardiac surgery	0.747 ± 0.335	0.908 ± 0.004
NSURG	neurologic surgical	0.550 ± 0.286	0.776 ± 0.005
CMED	cardiac medical	0.517 ± 0.041	0.710 ± 0.019
NMED	neurologic medical	0.734 ± 0.020	0.747 ± 0.024
TSURG	thoracic surgical	0.729 ± 0.008	0.694 ± 0.024
PSURG	plastic surgery	0.676 ± 0.246	0.731 ± 0.075
GU	genitourinary	0.579 ± 0.113	0.568 ± 0.097
TRAUM	trauma	0.759 ± 0.008	0.769 ± 0.029
SURG	surgical	0.791 ± 0.006	0.777 ± 0.009
ORTHO	orthopaedic surg.	0.524 ± 0.077	0.613 ± 0.037
OMED	orthopaedic med.	0.495 ± 0.032	0.511 ± 0.032
VSURG	vascular surgery	0.356 ± 0.180	0.696 ± 0.028
NBB	newborn baby	0.982 ± 0.014	0.988 ± 0.012
ENT	ear, nose, throat	0.421 ± 0.122	0.547 ± 0.044
GYN	gynecological	0.522 ± 0.034	0.503 ± 0.034
NaN	unmarked	*	*
OBS	obstetrics	0.548 ± 0.107	0.684± 0.109
DENT	dental	*	*
PSYCH	psychiatric	*	*

model for minimum cost instead of simply using an accuracy percentage rate to rank our models. This could also use patient classes to improve the fineness of the the readmission model; for example if patients with certain diseases prove to be exceedingly expensive to readmit, the hospital could invest in short-stay units to house patients who test positive for those diseases. The introduction of empirical data could ground this model in a realistic hospital setting and make it more applicable to the everyday operations of a hospital.

B. Regression and ordinal models

We have previously listed the reasons why we chose to use a simpler classification model instead of a more complex regression model, but a regression model could be more useful. One in-between approach could use an ordinal ranking of readmission times; ex. creating an ordinal prediction variable with value 0 for no readmission within a year, 1 for readmission within a year, 2 for readmission within a month, and 3 for readmisison within a week. This could avoid problems with having a mixed variable due to the necessity of having a value for ”patient never returns” while still providing more information than a binary classifier.

C. Further developing specific case models

We could see that the accuracy metric varied widely among different patient populations and the algorithm with the best accuracy varied depending on the population. To develop a better model, we could create a pseudo-decision tree starting with our categorical variables and branching to the models created with our random forest and perceptron techniques. This would essentially make specialized models for each categorical subpopulation in the patient population, allowing us to leverage different features and techniques to achieve the best accuracy for each subpopulation. For example, we might develop a specialized technique for determining readmission rates for pregnant women based on features and metrics not present in other patients. This will also shed insight into the reasons for readmission for individual groups, and in a real-world application this could enable techniques such as market segmentation and targeted initiatives to specifically reach out to a certain group in our care network.

D. Optimizing speed and portability with different features

With the rise of personal health trackers such as Fitbits and the Apple Watch, patients are able to monitor several health-related metrics such as heart rate, blood pressure, blood oxygenation, etc. It may be worth creating an algorithm optimized for these features and extensively preprocessed to allow for this algorithm to be run on portable hardware such as the aforementioned health trackers. This would allow patients to be proactive with their health monitoring approaches, and could mitigate some of the uncertainty associated with determining whether a patient will be readmitted to the hospital by making the patient more aware of their own health and letting them make informed decisions in non-emergency health-related situations. For example, if their heart rate and blood pressure have been unusually high for an extended period of time, the algorithm could recognize this as warning signs of a certain condition and prompt them to schedule a doctor’s appointment, allowing the hospital to process the patient without them entering urgent care or the emergency room.

VIII. Conclusion

In this paper, we performed feature selection and dimensional reduction on the MIMIC-III dataset to obtain a sparse dataset using one-hot encoding on the categorical features. We then used classifiers to predict the readmission rates of patients, and performed subanalyses to determine the efficacy of these models on metrics such as precision and recall and also to compare these models on different subpopulations of hospital patients. We determined that due to the properties of our dataset and the real-world considerations involved in running these algorithms, our perceptron with L2 regularization and our random forest classifier were the two most effective classifiers for this task, with overall accuracies of about 70% (but much lower precision and recall accuracies). We also suggested directions for further improvement especially given the context of the healthcare industry. These models could be put into use at a real hospital to augment the conclusions and opinions of doctors and nurses, and with the suggested improvements these models could potentially become accurate enough to become the main predictors of hospital readmission used by the hospital.

IX. Bibliography

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