|  |
| --- |
| **Data Analytics** |
| Predictive Analysis of Hospital Readmissions |
| CSCI-4957 |

|  |
| --- |
| Koi Stephanos  4-28-2020 |

Contents

[Introduction 2](#_Toc38985477)

[Preprocessing 3](#_Toc38985478)

[Checking and Reading the Data 3](#_Toc38985479)

[Dealing with Missing Values 4](#_Toc38985480)

[Exploratory Analytics 5](#_Toc38985481)

[Data Visualization 5](#_Toc38985482)

[Patients Readmitted vs Non-Readmitted: 5](#_Toc38985483)

[Rate of Patients Currently on Diabetes Medication: 5](#_Toc38985484)

[Breakdown of Patients by Race: 6](#_Toc38985485)

[Breakdown of Patients by Gender: 6](#_Toc38985486)

[Breakdown of Patients by Age: 7](#_Toc38985487)

[Percentage of Deceased Patients: 7](#_Toc38985488)

[Analyzing Visualizations 8](#_Toc38985489)

[Pattern Mining 8](#_Toc38985490)

[Readmitted Rules 9](#_Toc38985491)

[Non-Readmitted Rules 9](#_Toc38985492)

[Evaluating Rules 10](#_Toc38985493)

[Predictive Analytics 11](#_Toc38985494)

[Splitting the Data 11](#_Toc38985495)

[Training the Models 11](#_Toc38985496)

[Support Vector Machine 11](#_Toc38985497)

[Logistic Regression 12](#_Toc38985498)

[Evaluating Predictions 13](#_Toc38985499)

[Support Vector Machine 14](#_Toc38985500)

[Logistic Regression 15](#_Toc38985501)

[Overall Comparison 16](#_Toc38985502)

[EXTRA: Neural Network 16](#_Toc38985503)

[Conclusion 17](#_Toc38985504)

Introduction

This document catalogs the creation and results of a classification model to predict the re-admittance rate for hospitals patients. The initial dataset provided was a semi-cleaned list of 10k patients that had been admitted to the hospital. Features include basic information about the patient including their medical background and the admitting physician, a series features for current medications, previous diagnoses, test results and the target of the prediction, whether that patient was readmitted. The framework will be built in python using popular data analytics libraries such as *pandas*, *numpy*, *sklearn* and *matplotlib*.

We begin the process by examining the provided data set and apply several preprocessing techniques to clean the data and interpolate missing values when appropriate. Next we begin exploratory analysis by producing plots of several key feature points to understand the relationships constituent in the data. To further explore these relationships, we will perform pattern mining and association rule learning specifically on the factors that contribute to a patient dying or being readmitted. Finally, we will attempt to perform some predictive analysis by building a support vector machine model as well as a logistic regression model both trained on the cleaned data set. We will evaluate both learning algorithms with several training:testing splits, providing several metrics to gauge the accuracy and fit of the models, ultimately providing graphs to summarize our success.

All code and data files used throughout this course of this report may be found at the following repo: [github/predictingHospitalReadmissions](https://github.com/dkStephanos/predictingHospitalReadmissions).

Preprocessing

# Checking and Reading the Data

We begin by reading in the raw data file containing entries for ten thousand patients. Initially we want to test this sample to determine the overall health of the data, determining how much data is present, and reflect believable values. To begin, we had to determine what columns had missing values. The data was not consistent in how it represented this and would vary between leaving cells blank or recording ‘?’ or ‘None’, etc. As a result, the first thing we did was replace all those values with Nan and then determine counts per column. Here are columns with a significant count of missing values:

Figure

# Dealing with Missing Values

Looking at these counts, columns like *weight*, *payer\_code*, *medical\_specialty*, *max\_glu\_serum* and *A1Cresult* should be dropped immediately. Thousands of missing examples just aren’t recoverable. For race, we just set missing values to the Other class. We decided to keep missing values for *diag\_1*, *diag\_2* and *diag\_3*, as an absence in this field could serve predictive value. We chose, however, to drop the diagnosis descriptions, as they would have required natural language processing to contribute and that was beyond the scope of this project.

The remaining columns all have more than 85% of values present, so attempts may be made to overcome those gaps. To do this, we used mode-based imputation on the missing values. Since these fields were categorical, we felt it was best to just assume the most frequently occurring type for each of these columns. Finally, all rows containing numeric features that were outliers outside a standard deviation of 4 were dropped. Ultimately, our resulting data contained **9646** examples with **44** features each. Some missing fields were represented in the remaining dataset, but otherwise, no missing values remain. Below is a flowchart summarizing this process:

Figure

Exploratory Analytics

# Data Visualization

Once our data is clean, the next step is to beginning plotting various features to derive patterns. Since our priority is to predict whether a patient will be readmitted to the hospital, we will prioritize data plots related to that information.

## Patients Readmitted vs Non-Readmitted:

The first thing we wanted to do was see the percentage of patients that were readmitted, since that was what we were trying to predict. Overall, the dataset was evenly split. There were more not readmitted than readmitted, but this likely reflect the problem domain, and the difference does not negatively impact our predictive models. A summary of the breakdown:

Figure

## Rate of Patients Currently on Diabetes Medication:

We next wanted to look at the proportions for a key contributing factor. One of this biggest groups within the dataset were those who were currently taking diabetes medication. Diabetes makes you a higher risk factor for other complications, so this feature holds significant predictive potential:

Figure

## Breakdown of Patients by Race:

Next, we looked at the racial breakdown of patients. Its is unfortunately the case that minority groups in America frequently receive less attentive care, and this could be factor in the results. Overall, we do see a large skew towards Caucasians, followed by African Americans, but overall these results are consistent with demographic trends:

Figure

## Breakdown of Patients by Gender:

We also broke down the patients by gender. There were noticeably quite a few more women the men within the dataset. There is no clear reason for this, and should be considered when weighing the impact of this study:

Figure

## Breakdown of Patients by Age:

Our final demographic breakdown was age. Here again we can see a consistent climb as age rises, which is likely consistent with the problem domain:

Figure

## Percentage of Deceased Patients:

Finally, we did need to look at the percentage of patents that passed away during this study. This is an important detail when considering medical information and can vastly impact the way we review those results. As you can see, only a small fraction of patients died, but that makes it a highly relevant feature:

Figure

## Analyzing Visualizations

Overall by looking at the breakdown of our patient data in multiple dimension, we can get a feel for the skewness, or how much the data reflects reality. Based on the above graphs, we can see for example that most of our patients are female, perhaps a bit more than we might expect for an average sample. The number of patients readmitted looks good, having a healthy amount of each in the data set will allow our predictive model to perform better. We are noticeably skewed towards older Caucasians on some form a diabetic medication. This could be problematic, but more than likely just represents the natural distribution found in the problem domain.

# Pattern Mining

The next step in our analysis is to look for patterns and associations that are prevalent within the data. To do this, we will use the *apriori* and *association\_rules* algorithms from the *mlxtend* library. The *apriori* algorithm looks through our features and produces frequent item sets or features that commonly appear together. Then, *association\_rules* can be called to create rules based on those item sets. These rules tell us that given some group of antecedents, a given consequence is likely to occur. A good measure of how much more likely the consequence is lift, which measure the impact this rule has. Higher lift means a more powerful relationship. We will look at the top rules that affecting readmittance rates.

## Readmitted Rules

Overall, looking at the rules generated for hospital readmittance, a few things stand out. First, the most frequent occurrences in the top row include no nateglinide, no glyburide and discharged/transferred to home with home health service. Without speaking to the medical consequences of medications, the discharged to home with health service. This feature was found in nearly every one of the top 100 rules for readmittance. This very much suggests that one of, if not the best, single metrics for predicting whether an individual patient will need to be readmitted is if they are being sent home with a health service. These patients, in conjunction with the features on specific drugs, could be very helpful in targeting high risk individuals. Evaluating our rules, however, confidence and support are around 51% and 5% respectively. This means that while these associations exist, support is low, and they were seen in only about half of the cases provided. This is not a great indication of these rule’s value, and typically, these would not meet the confidence thresholds ordinarily provided when generating association rules. Below is a list of the top five rules generated, listing all the features associated with the rule as well as it’s lift:

|  |  |
| --- | --- |
| Top Rules Sorted by Lift | |
| ‘nateglinide\_No','chlorpropamide\_No','discharge\_disposition\_id\_Discharged/transferred to home with home health service', 'glyburide\_No' | 1.314706 |
| 'nateglinide\_No', 'discharge\_disposition\_id\_Discharged/transferred to home with home health service', 'glyburide\_No', 'acarbose\_No' | 1.31339 |
| 'nateglinide\_No', 'discharge\_disposition\_id\_Discharged/transferred to home with home health service', 'glyburide\_No', 'glipizide.metformin\_No' | 1.31339 |
| 'nateglinide\_No', 'discharge\_disposition\_id\_Discharged/transferred to home with home health service', 'glyburide\_No' | 1.313334 |
| 'nateglinide\_No', 'discharge\_disposition\_id\_Discharged/transferred to home with home health service', 'acetohexamide\_No', 'glyburide\_No' | 1.314706 |

Table

## Non-Readmitted Rules

Overall, rules for non-readmittance were not as impactful as for admittance. This makes sense, considering most of our samples were not readmitted. What is interesting is while our supports scores are similar, although about a half a percentage better on average, our confidence scores for these rules are much higher. Over our top ten rules, the confidence is 73% on average, over twenty points higher than for the readmitted rules. This means that while this rules may be less impactful, we can have more faith they would apply to future patients. That being said, there are a couple interesting trends within the rules themselves. Most predominately, is diagnosis 3 code 250, which appears in the majority of the first 500 rules. This is clear a strong indicator. The first chunk of rules not to feature it revolve around patients in the 50-60 age group with no changes discharged to home. Discharged to home is also nearly omnipresent in early rules and a strong predictor as well. This makes sense, as it is the least sever discharge case. Below are the top five rules sorted by lift:

|  |  |
| --- | --- |
| Top Rules Sorted by Lift | |
| 'discharge\_disposition\_id\_Discharged to home', 'glipizide\_No', 'pioglitazone\_No', 'diag\_3\_250' | 1.193883 |
| 'discharge\_disposition\_id\_Discharged to home', 'rosiglitazone\_No', 'glipizide\_No', 'diag\_3\_250' | 1.193308 |
| 'discharge\_disposition\_id\_Discharged to home', 'metformin\_No', 'glimepiride\_No', 'diag\_3\_250' | 1.190514 |
| 'discharge\_disposition\_id\_Discharged to home', 'rosiglitazone\_No', 'metformin\_No', 'diag\_3\_250' | 1.189411 |
| 'discharge\_disposition\_id\_Discharged to home', 'glyburide\_No', 'glipizide\_No', 'diag\_3\_250' | 1.193883 |

Table

## Evaluating Rules

Ultimately, many of the desired rules could not be created with very high support or confidence. Efforts were also made to generate rules associated with patients passing away, although despite generating over 8 gigs of associative rules, none related to patient death had high enough support. If more efforts were made to parse diagnostic information, thereby cleaning the feature set, more powerful rules could potentially be generated. The full generated rule sets for admittance can be found in the project repository data files.

Predictive Analytics

# Splitting the Data

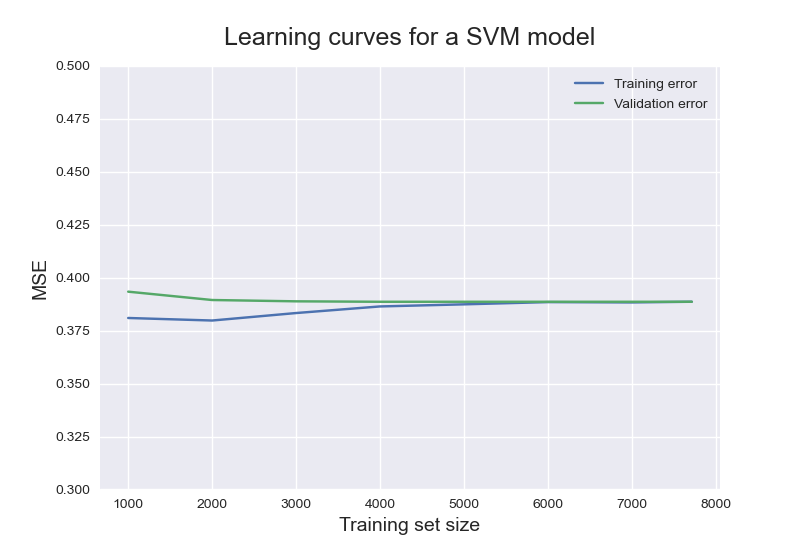
The first step towards training a predictive model is preparing the data. A machine learning algorithm requires both a set of data on which it can train, in order to refine its predictions, as well as a test set, on which the train model can be evaluated. Since we are interested in readmittance rates, we will split the data set trying to preserve this ration in both testing and training sets. Ultimately, a testing split of 80/20 was chosen for both models discussed here, but others were implemented during evaluation.

# Training the Models

Once our data was prepared, we could move on to training the models. We chose two different machine learning algorithms popular for classification, Support Vector Machine (SVM) and Logistic Regression (LogReg). Logistic Regression is a classic machine learning tool for binary classification and can be considered a baseline performance. SVM is a popular variation on logistic regression that uses various kernels to optimize its classification and often outperforms simple Logistic Regression. Both will be discussed separately, and then contrasted.

## Support Vector Machine

To evaluate the quality of our model, we generated a learning curve. This allows us to test the model at a variety of training and validation sizes and see how accuracy is effective. The plot has the training and validation accuracies below. We expect them to converge as the sizes increase, but too close is an indication of overfitting. That is what we see in this graph:

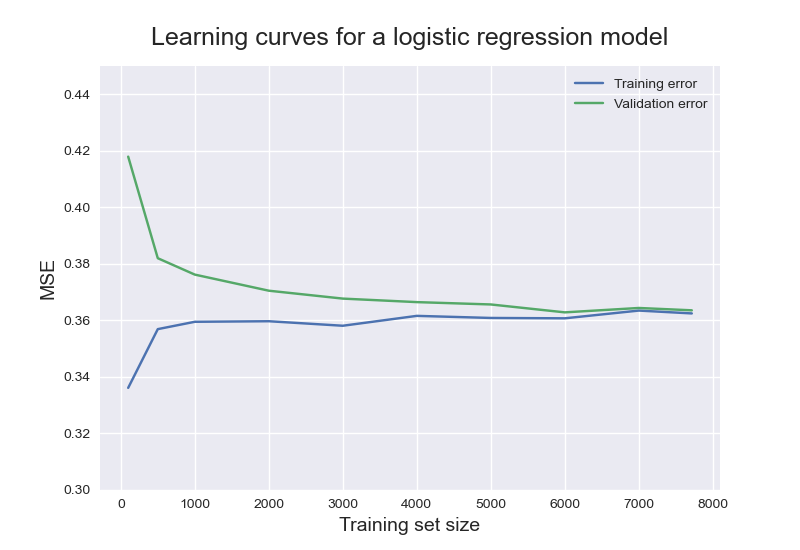


**Figure 9**

Our model converges at around 4000, meaning more training examples don’t affect performance at this point. The tightness of the curve suggests overfitting, which could be overcome with regularization.

## Logistic Regression

We also created a learning curve for the Logistic model. This graph indicates a similar level of performance, with a similar level of drop-off around 4000 samples. It also, however, indicates better sparsity, as the lines do not as closely converge. This model would likely stand up better to unseen data than SVM.



Figure

# Evaluating Predictions

Below are the confusion matrices for both models. As you can see, both perform similarly well. One common weakness is the high counts of false negatives. This makes sense, as we were concerned with upping accuracy, but is a concern considering the problem domain. You don’t want to tell someone they aren’t at risk when they are. This is the strongest indication this model needs to be improved before practical use. We also examine each model individually, here is a summary of their confusion matrices:

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Model |  | Predicted: No | Predicted: Yes | Model |  | Predicted: No | Predicted: Yes |
| SVM | Actual: No | 1110 | 69 | **LogReg** | Actual:  No | 1105 | 66 |
| SVM | Actual:  Yes | 680 | 71 | **LogReg** | Actual:  Yes | 662 | 97 |

Table

## Support Vector Machine

Figure

SVM can be run with a variety of kernels. Kernels essentially provided different metrics by which the data can be determined. Kernel decisions have more of an impact on computation time then on accuracy but can still provide value when fine tuning a model. We tested three popular kernels for binary classification: *polynomial, sigmoid, gaussian.* As the results show, the performance of gaussian and polynomial was quite similar in terms of accuracy measures. The sigmoid kernel massively underperformed and was not considered. Between gaussian and polynomial, gaussian is more popular and ran much quicker in our implementation, so it was chosen as the default for our model moving forward.

Figure

In the algorithm, the gamma rate determines how impactful a single data entry can be on the rest of the predictions. The smaller this value gets, the more impact a data entry can have. This is useful when fine tuning performance and can have a big impact on our false positive/false negative rate. As you can see, our accuracy and recall remain fairly constant, only seeing a minor rise when the rat is .01. Once we get past .1, however, precision sees a massive uptick. This indicates a better, more flexible model. All three metrics reduce when reducing gamma further, so we will continue with .01 as our gamma rate for this model.

## Logistic Regression

When it came to evaluating the Logistic model, we used the same training splits as for SVM. This left C, or the regularization rate, as the biggest variable to optimize. Regularization allows us to smooth out our model, allowing it to be more flexible when introduced to new data. We tested several standard typical regularization rates and recorded the impact on sparsity metrics below:

Figure

Data sparsity is a good thing and means our model will be more receptive to future inputs. Choosing a C value of .01 provides comparable accuracy and the highest sparsity, so it is the clear choice in this case.

## Overall Comparison

Looking at SVM compared to LogReg, both have some advantages. LogReg is similar to implement, whereas SVM offers more opportunities for fine tuning. Currently, we get slightly better accuracy results from the logistic model. SVM, however, typically outperforms LogReg and it is possible with a more comprehensive and predictive feature set, it may overtake the logistic model.

## EXTRA: Neural Network

After performing our predictive analysis, we were curious to see how the results would compare to the neural network. We set up a model with the MLPClassifier from sklearn, with the same test split, a low alpha, lbfgs as the solver and default layers. We were able to secure an accuracy of over 65%, outperforming SVM and LogReg. This potential should be explored further.

Conclusion

Ultimately, the problem of determining which patients will need to be readmitted is novel, complex and compelling. Our efforts to process the raw data, extract patterns, and build predictive models proved productive. We were able to establish trends in the data and make predictions with an accuracy up to 65%. While the practicality of these models in a medical sense has yet to be accomplished, further efforts to add natural language processing to this approach to make better use of diagnosis features could dramatically improve performance. We feel this is a strong step in the direction of prediction readmittance rates, and that an eventual solution using this as a foundation could prove financially marketable.

List of Figures

[Figure 1 3](#_Toc38985407)

[Figure 2 4](#_Toc38985408)

[Figure 3 5](#_Toc38985409)

[Figure 4 6](#_Toc38985410)

[Figure 5 6](#_Toc38985411)

[Figure 6 7](#_Toc38985412)

[Figure 7 7](#_Toc38985413)

[Figure 8 8](#_Toc38985414)

[Figure 9 12](#_Toc38985415)

[Figure 10 13](#_Toc38985416)

[Figure 11 14](#_Toc38985417)

[Figure 12 15](#_Toc38985418)

[Figure 13 16](#_Toc38985419)