

REDUCTION OF READMISSIONS TO HOSPITALS BASED ON ACTIONABLE
KNOWLEDGE DISCOVERY AND PERSONALIZATION

by

Mamoun Al-Mardini

A dissertation submitted to the faculty of
The University of North Carolina at Charlotte
in partial fulfillment of the requirements
for the degree of Doctor of Philosophy in
Computing and Information Systems

Charlotte

2018

Approved by:

Dr. Zbigniew W. Raś

Dr. Wlodek Zadrozny

Dr. Jing Yang

Dr. Maciej Noras

ABSTRACT

MAMOUN AL-MARDINI. Reduction of Readmissions to Hospitals Based on Actionable knowledge Discovery and Personalization. (Under the direction of DR. ZBIGNIEW W. RAŚ)

Healthcare spending has been increasing in the last few decades. One of the main reasons for this increase is hospital readmissions, which is defined as a re-hospitalization of a patient after being discharged from a hospital within a short period of time. The excessive amount of money spent every year on hospital readmissions and the urge to enhance healthcare quality make reducing hospital readmissions a necessity. The approach used in this work is entirely novel and was designed specifically to reduce the number of readmissions by applying the concept of personalization and actionable patterns to guide the health domain experts in their decision-making process. Our goal is not to build a system that replaces physicians, but a system that provides them with new insights discovered from the H-CUP medical dataset.

First, we investigate a two-fold problem that predicts the risk of mortality and hospital readmission for newly admitted patients. Several machine learning algorithms are used on our medical dataset to build an accurate classifier. In addition to that, feature selection techniques and boosting were applied to enhance the prediction accuracy and utilize the processing performance.

Second, we build the procedure graph, which shows all possible procedure paths that a new patient may undertake during the course of treatment. Additionally, we cluster patients into subgroups that exhibit similar properties in order to improve the predictability of the next procedures. We further devise a metric system that eval-

uates the level of desirability for procedures along procedure paths, which we would subsequently map to a metric system for the extracted clusters. Finally, we present a novel algorithm that discovers actionable knowledge (medical recommendations) that can be provided to physicians to put patients on a treatment path that would result in optimal reduction of the number of readmissions on average case.

Third, we predict the primary medical procedure for a newly admitted patient according to the similarities with the other patients in our medical dataset. This procedure might differ from the primary medical procedure assigned by a physician. We propose three new approaches to identify the patients, from the dataset, that are similar to the newly admitted patient. Finally, we find the procedures that are highly correlated with the primary medical procedure, and provide them as recommendations to physicians to enhance the final status of patients.

ACKNOWLEDGMENTS

First and foremost, I would like to express my highest appreciation for my advisor Prof. Zbigniew W. Raś for all his efforts, time, guidance, and support throughout my doctoral degree. Prof. Raś was more than an advisor. His caring and kind personality has left a profound influence on my personality and performance. I have learned a lot from him and he will always be a role model.

Moreover, I would like to thank all the respected members in my dissertation committee for their valuable feedback that have enhanced the quality of my dissertation.

Furthermore, I would like to thank SAS Institute for their generous fund to support my research and for their valuable feedback.

My greatest appreciation and thanks go to my parents (Tawfiq and Hanan), brother (Mamdouh), and sister (Diala) for their encouragement and endless support.

Finally, I am grateful for my lovely wife (Amal) for her love, support and patience throughout the doctorate journey. She has been an inspiration to me and I am sure that I wouldn't have reached this step without her. I aspire to read her doctorate dissertation in the near future.

TABLE OF CONTENTS

LIST OF FIGURES	ix
LIST OF TABLES	x
CHAPTER 1: INTRODUCTION	1
CHAPTER 2: Overview of the Related Research	4
2.1. Categories of Hospital Readmission Applications	5
2.1.1. Diagnosis Prediction	6
2.1.2. Risk Prediction	7
2.1.3. After Procedure Prediction	7
2.2. Healthcare Datasets	8
2.2.1. Structured EHR	9
2.2.2. Unstructured EHR	10
2.3. Review of the Existing Applications	11
2.3.1. Diagnosis Prediction Applications	11
2.3.2. Risk Prediction Applications	14
2.3.3. After Procedure Prediction	16
2.4. Data Mining Algorithms Used in Hospital Readmission Reduction	17
2.5. Limitations and Future Directions	19
CHAPTER 3: H-CUP Dataset Description	22
CHAPTER 4: PREDICTING THE RISK OF MORTALITY AND HOSPITAL READMISSIONS	27
4.1. Data Preprocessing	27

	vii
4.2. Predicting the Risk of Mortality and Hospital Readmission	29
4.3. Feature Selection	30
4.4. AdaBoost	32
CHAPTER 5: INTRODUCING PROCEDURE PATHS AND PROCEDURE GRAPH	35
5.1. Unique Procedure Paths	37
5.2. Lengths of Procedure Paths	38
CHAPTER 6: INTRODUCING PATIENTS PERSONALIZATION	43
6.1. Exact Matching Clustering	44
6.2. Rough Matching Clustering	46
6.3. Clusters Filtering	51
6.3.1. Percentage Filtering	52
6.3.2. Co-morbidity Filtering	53
CHAPTER 7: PROCEDURES AND CLUSTERS EVALUATIONS	55
7.1. Procedures' Scoring Function	55
7.2. Clusters' Scoring Function	59
CHAPTER 8: ACTIONABLE KNOWLEDGE DISCOVERY	61
8.1. Entropy-Based Weighted Score for a New Patient	63
8.2. Extracting Recommendations Algorithm	66
8.3. Results from the H-CUP dataset	71

	viii
CHAPTER 9: PRIMARY MEDICAL PROCEDURE PREDICTION AND DISCOVERY OF ITS ASSOCIATED PROCEDURES	77
9.1. Predicting the Primary Procedure	78
9.1.1. Minimum Similarity Match	78
9.1.2. Jaccard Similarity Match	81
9.1.3. Selective Similarity Match	84
9.2. Procedures Associations	88
9.2.1. Matrix Representation of Procedures	89
9.2.2. Personalized Procedures Associations	92
CHAPTER 10: CONCLUSION	96
REFERENCES	98

LIST OF FIGURES

FIGURE 1: Waste in healthcare spending as listed by Price Waterhouse Coopers (PWC) research institute [4]	2
FIGURE 2: The benefits of reducing the number of hospital readmissions	4
FIGURE 3: Categories of the applications used in reducing the number of hospital readmissions	6
FIGURE 4: The clinical data formats and the different types of patients' information found in EHRs	11
FIGURE 5: Depiction showing the timeline of a patient's visits	25
FIGURE 6: The effect of the chi-square filtering on the mortality dataset	32
FIGURE 7: Depiction of a procedure graph	36
FIGURE 8: Number of unique paths per procedure	38
FIGURE 9: Number of patients per procedure code	38
FIGURE 10: Maximum path length per procedure	40
FIGURE 11: Average path length per procedure	40
FIGURE 12: Construction of the patients' clusters	50
FIGURE 13: Depiction of the clusters on the procedure graph	51
FIGURE 14: Procedure graph with procedure scores	57
FIGURE 15: Extracting Recommendations for procedure 5059 (liver transplant)	62

LIST OF TABLES

TABLE 1: Comparison between EHR formats	9
TABLE 2: Overview of the surveyed systems	18
TABLE 3: Description of the used core table features	23
TABLE 4: Procedures' categories as listed in the H-CUP	26
TABLE 5: Predicting mortality accuracies	30
TABLE 6: Predicting readmission accuracies	30
TABLE 7: Descriptions of the feature selection techniques	31
TABLE 8: The most significant features in the mortality and readmission predictions. The prefix in the CCS code columns indicates whether the code refers to a procedure (PR) or a diagnosis (DX)	33
TABLE 9: The effect of AdaBoost on the decision tree classifier	34
TABLE 10: Procedure paths with the highest probabilities for procedure 105 (Kidney transplant)	42
TABLE 11: Some of the most common set of exact diagnoses for procedure 158 (spinal fusion)	44
TABLE 12: Number of clusters and the entropy for different element clusters and different ranges for procedure 158 (spinal fusion)	53
TABLE 13: Categories of the comorbid conditions	54
TABLE 14: A hypothetical example to demonstrate the calculation of the new patient's score	65
TABLE 15: A sample of three clusters with their scores and their included and excluded sets for procedure 105 (kidney transplant)	72
TABLE 16: A summary of the number of clusters that patients belong to on average and the average number of desired clusters, for sample of five procedures	73

TABLE 17: The set of the diagnostic codes and their description for our sample patient	75
TABLE 18: A sample of recommendations based on our multi-cluster entropy-based algorithm. Score of patient prior to following any of the recommendations is 0.367	75
TABLE 19: Dataset S, containing all existing patients	79
TABLE 20: Prediction accuracy of the minimum similarity match using the N most probable primary procedures	80
TABLE 21: Jaccard similarity calculations example	84
TABLE 22: An example of one of the tested patients	87
TABLE 23: Sample of main procedures with their frequencies and accuracies for a testing sample of 1,000 instances, using a training set of size 10,000 instances	88
TABLE 24: The most common procedures associated with procedure 105 (Kidney transplant)	92
TABLE 25: The most common procedures associated with procedure 105 (Kidney transplant)	94

CHAPTER 1: INTRODUCTION

Recently, expenditure on healthcare has risen rapidly in the United States. According to Gorman [10], healthcare spending has been rising at twice the rate of growth of our income for the past 40 years. The projection of the growth rate in healthcare spending is 5.8 % during the period 2014-2024, which means that the spending will rise to 5.4 trillion dollars by 2024. The gross domestic product (GDP) growth rate is only 4.7 % (as of 2014) [13]. This increase in healthcare spending can be attributed to several factors as listed by Price Waterhouse Coopers' (PWC) research institute: over-testing, processing claims, ignoring doctors orders, ineffective use of technology, hospital readmissions, medical errors, unnecessary ER visits, and hospital acquired infections [4]. Figure 1 shows that 25 billion dollars are spent annually on readmissions. Hospital readmissions and surgery outcomes prediction has gained a great interest recently in the scientific research community [26, 14, 27, 23]. Analyzing the reasons behind readmissions and reducing them can save a great amount of money. A hospital readmission is defined as a re-hospitalization of the patient after being discharged from the hospital. The period in average is 30 days [23].

One of the reasons for readmissions is negative side effects that may appear after performing procedures and may not be known in advance, as a result patients may require hospital readmissions [11]. The main goal of this research is to identify the patients with high risk of readmission and devise actionable knowledge (medi-

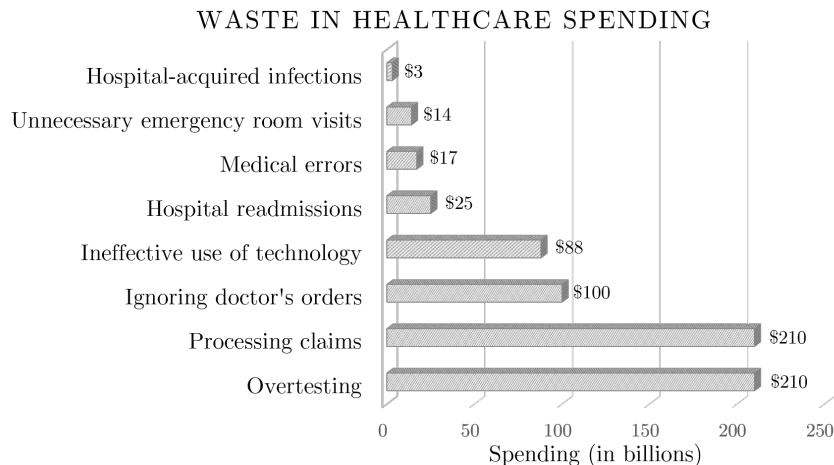


Figure 1: Waste in healthcare spending as listed by Price Waterhouse Coopers (PWC) research institute [4]

cal recommendations) to enhance the final status of patients. We approached this goal through the following main steps: first, predicting the risk of mortality and readmissions through applying several supervised machine learning algorithms. Second, developing novel algorithms to personalize patients according to their diagnoses, evaluate them through defining metric functions, and eventually devise the actionable knowledge (medical recommendations) and provide it to physicians to reduce the average number of hospital readmissions. Third, predicting the primary medical procedure for a newly admitted patient by clustering the patients according to their set of diagnoses. Finally, we identify the procedures that are associated with the predicted primary procedure through personalizing the patients according to their common medical characteristics.

This dissertation is structured as follows: Chapter 2 presents the state-of-the-art literature in this field. Chapter 3 provides information about the H-CUP dataset used in this work. Chapter 4 shows the prediction of the risk of readmission and mor-

tality. Procedure graph and procedure paths are both presented in Chapter 5. The algorithms used to cluster patients based on their diagnoses are presented and explained in Chapter 6. Chapter 7 presents the metric system proposed to evaluate the clusters. Chapter 8 explains the algorithms used to devise the actionable knowledge. The primary medical procedure prediction and procedures associations are presented in Chapter 9. Finally, Chapter 10 concludes the dissertation.

CHAPTER 2: OVERVIEW OF THE RELATED RESEARCH

Big data have played a significant role in different kinds of industries, and recently emerging into the healthcare sector, due to the trend in saving patients records in an electronic format. Making the medical data organized and available to the research community is considered as the tipping point that will speed the wheel of extracting knowledge and solving the pending problems in healthcare. Readmission reduction is one of these problems that can benefit significantly from mining the data. The research of reducing the number of hospital readmissions can result in a multitude of outcomes as shown in Figure 2.



Figure 2: The benefits of reducing the number of hospital readmissions

In this chapter, we survey the major applications in this field to highlight the

significance of this research. We categorize the applications into three main categories based on the prediction achieved and list the advantages and disadvantages of each category. Moreover, we present the different formats of the electronic health records (EHRs) and shed the light on the data mining algorithms used in this field. Finally, we summarize the limitations in the existing applications and highlight the open research directions to be investigated.

2.1 Categories of Hospital Readmission Applications

Several healthcare applications that target the problem of hospital readmission have been investigated in this research. By reviewing the literature, we have found that these applications can be grouped into three main categories according to the prediction that they are targeting. The categories are: diagnosis prediction, risk prediction, and after procedure prediction. These categories are further classified into two other categories: informative and actionable. Informative category includes the applications that provide only information such as the applications found in the diagnosis and risk prediction categories. On the other hand, actionable category refers to the applications that try to make some changes to reduce the problem such as the ones found in the after procedure prediction category, so rather than providing only information, it provides also an action plan for future. Informative category helps the clinicians in taking decisions. However, actionable category takes a role in the decision making process. Figure 3 shows the two approaches followed in categorizing the applications reviewed in this research. Following in this section, we will describe each category by giving the goal, the advantages, and limitations. Later in Section



Figure 3: Categories of the applications used in reducing the number of hospital readmissions

2.3, we will present examples for each category.

2.1.1 Diagnosis Prediction

Applications in this category aim to find the most probable disease based on the patients' data. Not being able to diagnose the patient with a certain disease or giving a wrong diagnosis can result in hospital readmission. Therefore, developing a model that is able to predict the correct disease can help in reducing the number of patients coming back. Following are the advantages of this prediction:

- Helping physicians to better diagnose the patients.
- Reducing errors that may occur due to the wrong diagnosis.
- Early prediction of diseases.
- Giving the patient the necessary care.

This kind of prediction is considered informative, as it only provides the clinicians with information about what could be the disease based on the patient's information. These information are invaluable in reducing the number of hospital readmissions, but wrong diagnosis prediction may consume the hospital's resources to make sure whether the patient has a certain disease or not.

2.1.2 Risk Prediction

Risk prediction aims to identify the patients and diseases with a high risk of hospital readmission. Applications in this category try to find if a certain disease causes a high rate of hospital readmissions. Accordingly, they categorize the patients who suffer from this disease into different groups based on the probability of coming back. Grouping patients into different categories to know the possibility of them being readmitted can help in reducing the number of readmissions. Following are the advantages of this prediction:

- Dedicating extra care to patients with high risk.
- Drawing the attention of the researchers in the medical field to focus on finding new treatments for diseases with high risk of readmission.

This kind of prediction is also considered informative, as it only tells the clinician whether a new patient might be readmitted in the future or not. Although this information is substantial in reducing the number of readmission, as clinicians may dedicate extra care to the patients with high risk of readmission, but this may also add more headache to the hospital and overwhelm its resources.

2.1.3 After Procedure Prediction

Most of the applications fall in the diagnosis and risk prediction categories. However, there are a few applications that are concerned with predicting the status of patients or predicting the outcomes after performing a certain procedure. This kind of prediction is considered actionable, as it looks one step further to predicting what

will happen after applying the course of treatment. Following are the advantages of this prediction:

- Allowing physicians to see all possible paths that a patient may follow after performing different kinds of treatments.
- Providing the physicians with a prediction of the most probable path that a patient may follow based on his/her status.
- Finding another path that is better than the most probable path.

This kind of prediction is still immature. Nevertheless, the preliminary results show a promising era in the field of healthcare prediction. It is worth mentioning here that actionable applications require a higher accuracy rate than the informative applications, as they suggest some possible changes on the patient's medical conditions to achieve better results.

2.2 Healthcare Datasets

Healthcare industry generates a surplus amount of data that keep track of patients' medical information. Most of these data are stored in a hard copy format, which makes it very hard to modify, analyze, manage, and apply research to extract some useful information and find the hidden trends. Recently, there is a great tendency towards digitizing the medical data to make it available in electronic format. This digitization is an essential step in big data research, since it opens the door for researchers to apply the data mining algorithms on the saved data to extract valuable information that will eventually help in improving the healthcare quality. Electronic

Table 1: Comparison between EHR formats

	Structured	Unstructured
Accessibility	X	
Flexibility of adding notes		X
Standardization	X	
Subjective data		X
Ease of modification	X	
Ease of collaboration	X	
Customizable	X	
Accepts diversity		X
Requires more effort		X

Health Records (EHR) are found in two formats: structured, such as databases, and unstructured, such as medical notes or scenario-based texts. The format of data governs several aspects such as the ease of access to the data, flexibility of adding notes, ability to standardize the format, ease of modifying the data, ease of making collaboration between the medical organizations, customizing the data based on the patient's conditions, accepting the diversity that is reflected by the nature of the medical field, and the effort required to process the data. All of these aspects are substantially reflected on the data mining techniques that can be used to process the data and the kind of information that can be extracted [32]. Following is a description of each format and Table 1 shows a comparison between them.

2.2.1 Structured EHR

Structured format is represented by database tables that include patients' records with fixed schema. Each column in the database represents a feature that describes the patient. Structured format is more common than the unstructured format, due to the fact that fixed schema enhance the access performance and allow the stan-

standardization, which makes it easy to introduce collaborations and make the process of installation easy and fast. That being said, storing all kind of patients' data in a structured format is impractical, due to the diversity of patients and diseases. For example, some required fields may not be applicable among all patients or it is hard to fit in a structured format such as the notes that physicians write besides the symptoms or diagnoses [32].

2.2.2 Unstructured EHR

Unstructured format is represented by notes or scenarios written by clinicians to describe the patients' conditions. Some examples of these written notes may include, but not limited to: radiology reports, surgical note, and discharge notes. This format requires applying natural language processing (NLP) methods to extract the needed information. One advantage of the unstructured format over the structured format is the flexibility that allows the clinicians to add customized details and set them free from any kind of restrictions that the structured format may force. Clinical notes could be subjective to the principal clinician who writes them and may lack some important information. In addition, notes may include some grammatical errors or un-understandable abbreviations. Consequently, data pre-processing becomes more sophisticated and requires more efforts [32].

The main component of the EHRs is the medical data, however, there could be other data such as demographic and socio-economic data. These additional data play a fundamental role in enhancing the process of mining the patients' data. Clinical data in general include information about the patient's medical conditions such as

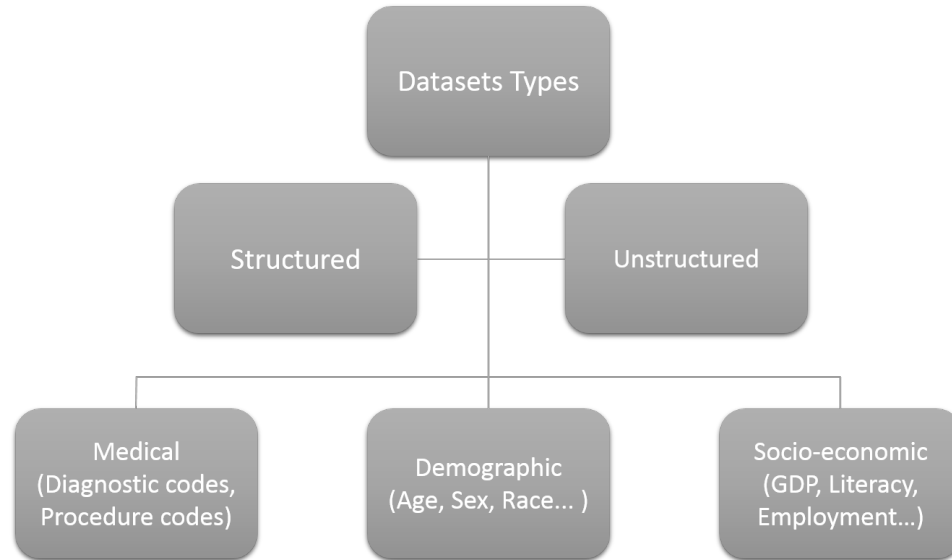


Figure 4: The clinical data formats and the different types of patients' information found in EHRs

diagnoses and procedures. On the other hand, demographic data include information that describe the characteristic of the patient such as age, sex, and race. Social-economic (socio-economic) data refer to the economic status of the patient, such as salary, literacy, and employment. Figure 4 shows the clinical data formats and the different types of information found in EHRs.

2.3 Review of the Existing Applications

In this section, we present the major applications that target the problem of hospital readmissions. These applications are listed in the corresponding categories, explained in Section 2.1, in a reversed chronological order to exhibit the progression in research contribution in each category with time.

2.3.1 Diagnosis Prediction Applications

ISABEL [19] is considered as one of the first computerized tools that processes unstructured medical data. The proposed application processes text-based medical

knowledge to extract some useful information in pediatrics. It mainly processes the standard pediatrics textbooks by applying matching patterns algorithm to produce a set of possible diagnoses for a set of symptoms or conditions. The main goal of this tool is to be a reference to physicians to remind them of any possible diagnosis and save them from any mistake. The information provided by ISABEL are invaluable to clinicians. However, the processed knowledge are not up-to-date, textbooks are not integrated with the daily medical records. Moreover, it is limited in providing the possible diagnoses and does not give any recommendation about what would be the next step in the course of treatment. Additionally, it does not provide a flexible way of interaction with the users.

Chou et al [3] proposed a predictive model that is able to process a structured dataset related to breast cancer, which is considered as a common and serious disease for women. The proposed model works by applying classification methods on the breast cancer dataset. The authors proposed a hybrid predictive model by integrating artificial neural networks and multivariate adaptive regression splines. The purpose of this integration is to take the advantages of each approach in order to enhance the prediction accuracy. Neural networks are known for its enhanced memory characteristic and the outstanding generalization capabilities. However, regression is known to be fast. The system applies a binary classification, this means it tells whether a woman may have breast cancer or not, but it does not cluster patients or rank them based on the severity of the disease.

DeepQA [8] is an application developed by IBM to address some of the challenges that clinicians face in the diagnostic process, in which they need up-to-date medical

knowledge. It builds upon IBM Watson to process unstructured contents found in medical resources, such as books and electronic health records (EHRs), by applying natural language processing (NLP) techniques. This can help the clinicians to work with the most recent medical knowledge and make better decisions to enhance the quality of healthcare. DeepQA depends on the question answering algorithm used previously by IBM Watson. Additionally, it allows the clinicians to interact with the system in a simple way by entering the patient's details and viewing the medical resources that the results were built based on them.

WatsonPaths [14] is the most current application developed by IBM for healthcare. It is similar to IBM Watson in which it accepts questions and provides precise answers with high confidence by applying NLP techniques. Additionally, it provides the ability of interacting with users. The input in WatsonPaths is mainly a scenario-based that describes the patient's status and ends with a punchline question such as what is the diagnosis? or what is the best way for treatment? The algorithm used in WatsonPaths is based on breaking down the input scenario into pieces of information. These information could be demographic, pre-existing conditions, signs, symptoms, and test results. Accordingly, the application starts to interact with the user and ask some subquestions to build on the existing information and conclude new facts. These information are then used to build an asserted graph that includes prioritized nodes. Finally, a probabilistic approach is applied on the asserted graph in order to provide the answers for the punchline questions. The system has been tested against the previous versions of Watson question-answering systems used in the medical domain such as DeepQA. The results show improvements in terms of accuracy and confidence.

2.3.2 Risk Prediction Applications

Risk-O-Meter [33] is a risk calculator application that is designated to people with no medical expertise. It mainly predicts the risk of readmissions for heart failure patients by clustering all possible combinations of patients' attributes and then classify them. Moreover, it discovers the association rules that determine and explain the high risk of readmission. Risk-O-Meter accepts any number of attributes that describe the patient and is able to complete the missing attributes to fit by the trained data. After that, the patient is assigned to the most similar (closest) cluster, and based on this assignment a risk factor is assigned to the patient with an explanation of the result.

Natale et al [16] similarly focus on their proposed predictive model on heart failure patients, but from different perspective. They aim to reduce readmissions through targeting the clinical interventions after the discharge rather than affecting clinicians decisions. In their work, they propose a model to stratify patients with high risk of hospital readmission. However, they did not stop at this point, but further proposed to increase the clinical interventions for those who are classified with high risk of readmission especially after being discharged. They have found, based on the population under study, that a hospital readmission happens not because of insufficient care, but rather of patients non-compliance to the course of treatment. Therefore increasing interventions can play a substantial role in influencing non-compliance patients. Interventions can be as simple as making a follow-up call, or it can go further to visit the patient at home. The developed model mainly starts by applying a binary classifier

that predicts whether a patient has a high possibility of readmission. After that each patient is assigned a cost based on the classification confidence and consequently the level of intervention is determined. Having different levels of interventions utilizes the hospital resources and increases the care to those with high risk of readmission. However, it would be useful to develop a model that works on improving the clinicians' decisions before the hospital discharge on one the hand and takes care of patients after that on the other hand.

Intel and Cloudera proposed in [2] a predictive analytics to help a Large Hospital Group (LHP) in reducing readmission rates. Their approach is similar to Risk-O-Meter in which it depends on classifying patients with high readmission risk. Accordingly, they suggest to provide special care to the patients with higher likelihood of readmission, and hence decrease the risk of readmission. The proposed model does not only rely on the electronic medical record (EMR) as most of the applications do, but it also links them with the socio-economic data, which provide the patients' history and living standards that have significant impact on readmission. Random Forests algorithm has been used to build the proposed model and predict the risk of readmission. This model has been tested on a sample of patients who have one of these three diseases: acute myocardial infarction, pneumonia, and cardiac heart failure. The results show that the predictive model outperforms the standard model used by LHP. This research introduces the importance of using socio-economic data in predicting the risk of readmission, and draws the attention of the researchers to use additional data such as the demographic data. The results exhibit a strong predictive system in the field of healthcare, but dedicating extra care to some patients requires

more resources and may be a burden to hospitals.

Pathway-Finder [15] is the first interactive recommender system that finds all the possible pathways for heart failure patients. The proposed system consists of two layers, offline and online. The core of the offline layer is the Bayesian network module, which is a probabilistic model responsible of finding the dependencies between heart failure attributes. The authors used the state inpatient discharge (SID) dataset of Washington state which consists of 200,000 patients records to build their model. The online layer is mainly the graphical user interface (GUI), which takes the patient's basic socio-demographic information as an input and then starts by displaying the associated comorbidities from the learned Bayesian network in the offline layer. At this stage, the user (physician) is allowed to select all applicable comorbidities. Based on the basic socio-demographic data and the selected comorbidities, the application finds all possible interventions (procedures) that are associated with the given data. The final stage is the display of the risk likelihood based on the physician's selection. Although, this work is an advanced milestone in reducing the number of readmissions, but the authors did not present the accuracy of the system as a whole. Furthermore, the work is limited to a certain kind of patients who suffer from "Heart Failure".

2.3.3 After Procedure Prediction

Panahiazar et al [17] developed a model that is able to predict a medication plan for heart failure patients based on prior healthcare records of similar patients. The model considers different sort of information from the EHR, such as lab results, medications, comorbidities, and demographics, in order to cluster patients and find the similarities

between patients using machine learning techniques. Patients are then assigned a medication plan according to the similarity index assigned to them. Based on this assignment, physicians choose the right or most suitable plan for their patients and can predict the response of patients to a certain medication plan.

Hajja et al [11] similarly proposed a model to cluster patients in order to predict the outcomes after undertaking a certain procedure. The clusters are built based on the side effects that patients had after taking a certain procedure. These side effects could be categorized into positive, negative, or neutral. On the other hand, the similarity is defined based on the diagnoses that patients had before taking the same procedure. Accordingly, clinicians will be able to know in advance what kind of side effects a patient may have after taking the decided procedure. Moreover, the proposed model adds a novel feature in which it allows the transition of a patient who is assigned to a certain cluster to another cluster which may result with less serious side effects. This kind of transition can be achieved by applying action rules [21].

2.4 Data Mining Algorithms Used in Hospital Readmission Reduction

As we have seen in Section 2.3, there are different kinds of data mining algorithms used in the field of healthcare in general and specifically in reducing the number of hospital readmissions. The algorithms have been used to achieve several goals including, prediction, clustering, and finding the relations between different attributes. Data mining algorithms are generally classified into two main categories: supervised and unsupervised. A simple difference between them is whether there is a predefined class or not. Supervised algorithms predict the target label or tag using a pre-labeled

Table 2: Overview of the surveyed systems

Paper	Year	Classification	Clustering	Association Rules	Action Rules
Ramnarayan et al [19]	2003	X			
Chou et al [3]	2004	X			
Ferrucci et al [8]	2013	X			
Lally et al [14]	2014	X	X		
Zolfaghar et al [33]	2013	X		X	
Natale et al [16]	2013	X			
Vedomске et al [28]	2013	X			
Hilbert et al [12]	2014	X			
Demir et al [5]	2014	X			
Fernandes et al [7]	2014	X	X		
Veloso [29]	2014		X		
Panahiazar et al [17]	2014		X		
Hajja et al [11]	2014		X		X
Silva et al [24]	2015	X			
Shams et al [22]	2015	X			
Vukicevic et al [30]	2015	X			
Liu et al [15]	2015	X			
Intel and Cloudera [2]	2015	X	X		

training data such as classification. However, unsupervised algorithms find correlations in the input raw data such as clustering. In this section, we are going to list the most important categories of data mining algorithms, however, we are not going to explain each one as the purpose of this review is to highlight the use of data mining algorithms in the research that targets reducing the number of hospital readmissions, but for more details about the algorithms, you may check [31] and [25]. Table 2 lists the applications described in Section 2.3 in addition to some other recent applications with the data mining algorithms used in a chronological order.

The most common data mining algorithms used in reducing the number of hospital readmissions are classification and clustering algorithms. Classification algorithms are widely used in such kind of research due to the fact that prediction is the main purpose. Clustering is also important in the sense that patients differ in their demographic information, clinical conditions, and reactions to the course of treatment.

Therefore, there is a need to stratify patients in subgroups based on some common characteristics and then classify them. That being said, a new patient can be assigned to the closest group and we would be able to know in advance how he/she will react to the prescribed treatment or if there is a possibility of any kind of risk.

2.5 Limitations and Future Directions

The recent applications developed to reduce the number of hospital readmissions show promising results. However, there is always a door for improvements. According to the reviewed applications, we list in this section some of the observations about the existing models and we highlight on the limitations and open research directions in this area.

- Most of the applications focus on certain diseases, usually dangerous ones such as heart failure, rather than including all diseases or procedures. One of the advantages of mining big data is to extract knowledge and find trends, so by analyzing all kinds of diseases and procedures, we may find some hidden information that can reduce the number of hospital readmissions.
- As we have seen in the literature review, most of the applications developed to reduce the number of hospital readmissions fall in the diagnosis and risk prediction. However, there is a lack of applications that predict what happens after the medical procedure. Drawing the attention beyond the procedure can add a great impact on enhancing the healthcare quality and reducing the number of hospital readmissions.

- There is a lack of applications that can predict all possible paths that a patient may undertake and how many hospital visits the patient is expected to have during the course of treatment.
- Most of the predictive applications use classification and/or clustering algorithms. However, there is a lack of using association and action rules algorithms. Introducing new algorithms may reveal new facts that can be useful in reducing the number of hospital readmissions.
- Medical data are essential in mining patients' records. However, other data such as demographic and socio-economic can add insightful impact to the final outcomes.
- Medical data tend to be complex and heterogeneous in nature, since assembling it requires acquiring the patient's data from different sources, which makes it hard to achieve the completeness of data. Some of the clinical data available currently tend to be complete, but there is still a door for improvement.
- Reducing the number of hospital readmissions can be achieved by either enhancing the quality of medical care that the patient takes during his/her stay at the hospital or by extending the care after the discharge through the follow-ups. A complete system that adopts both approaches can better reduce the number of hospital readmissions.
- Accuracy is very important in this kind of research, due to the sensitive information it provides to both the clinicians and the patients. Although the existing

applications show promising results, but still we need more improvements to make the applications dependable.

CHAPTER 3: H-CUP DATASET DESCRIPTION

In this research, we mined the Florida State Inpatient Databases (SID) that are part of the Healthcare Cost and Utilization Project (H-CUP) [1]. The SID dataset is primarily a state-level discharge data that is collected from non-federal community hospitals, which constitute the majority of hospitals in the USA. The SID includes patients' demographic data, such as age, gender, and race. In addition to the demographic information, SID includes patients' medical data, such as diagnoses, procedures, status of the patient, and the length of stay. The dataset is mainly composed of three tables: *American Hospital Association (AHA) Linkage*, *Charges*, and *Core*. The most important table in the SID is the *Core* table, which is considered as the nucleus of the SID. The *AHA Linkage* table contains hospital specific information and the *Charges* table contains information about the charges associated with the use of hospital resources. The *Core* table contains over 280 features (attributes), however, many of those features are repeated with different values according to the patient's status. There are two types of coding schemes used in the *Core* table for labeling and formatting: the International Classification of Diseases, Ninth Revision, Clinical Modification (ICD-9-CM) and the Clinical Classifications Software (CCS). The ICD-9-CM coding is detailed and uses more codes to label the procedures and diagnoses. On the other hand, CCS is more generalized and it is a collapsed form of the ICD-9-CM. For example, there are 15,072 diagnosis categories and 3,948 procedure

Table 3: Description of the used core table features

Features	Concepts	Value Margin	Value Type
VisitLink	Patient Identifier	8(n)	Integer (8)
DaysToEvent	Temporal visit ordering	nnnn	Integer (4)
LOS	Length of Stay	0 - 365	Integer (3)
DXCCSn	n^{th} Diagnosis, flexible feature	1 - 259	Integer (3)
PRCCSn	n^{th} Procedure, meta-action	1 - 231	Integer (3)
DXPOAn	Present on Admission Indicator	Y/N	Boolean
FEMALE	Indication of sex	1/0	Boolean
RACE	Indication of race	1 - 6	Integer (1)
AGE	Age in years at admission	0 - 124	Integer (3)
DIED	Died during hospitalization	1/0	Boolean

categories in the ICD-9-CM. CCS coding however, collapses these categories into a smaller number of more generalized categories, totaling only 285 diagnoses categories, and 231 procedures categories.

In our experiments, we only used the features listed in Table 3 that are relevant to the examined problem. Visit linkage (*VisitLink*) feature is an encrypted identifier for patients. Each patient has a unique identifier among the hospitals within the same state. Days to event (*DaysToEvent*) feature provides information about the number of days between two consecutive visits for the same patient identified by the *VisitLink* feature. The value of this feature is set randomly for the first visit to preserve the privacy of patients. The de-identification of such information can prevent any attempt to track patients. Therefore, some of the values in the dataset are assigned random values initially. The value of the following visit would be the initial random value assigned for the first visit plus the number of days between the admission dates of the two consecutive visits. For example, the patient can be assigned *DaysToEvent*=12 in the first visit, which is an entirely arbitrary number and does

not provide us with any information about the actual admission date. The value of *DaysToEvent* in the second visit will be the first value (12) plus the number of days between the two visits. If *DaysToEvent*=12 for the first visit and the number of days between the two admission dates is 28 days, then *DaysToEvent* for the second visit will be 12+28=40 days. It is worth mentioning here that *DaysToEvent* represents the number of days between the admission dates, and not between the discharge date and the next admission date. *VisitLink* and *DaysToEvent* features are encrypted identifiers of the patients. They are used together to track patients across multiple visits within the same hospital or multiple hospitals within the same state without revealing the patient's identity. The *Length of Stay (LOS)* feature represents the number of days a patient stays at the hospital, which is the number of days from the admission date to the discharge date. *VisitLink* and *LOS* can be used together to calculate the number of days between the discharge date and the next admission date. Referring back to our example above, depicted in Figure 5, if $DaysToEvent_1=12$ and $LOS=10$ for the first visit, and $DaysToEvent_2=40$ for the second visit, then the number of days between these two visits (between discharge of the first visit to admission of the second visit) is $40-12-10=18$ days. Following is the equation used to calculate the number of days between the discharge date and the next admission date:

$$DischargeToAdmissionDays = DaysToEvent_2 - DaysToEvent_1 - LOS_1 \quad (1)$$

where *DischargeToAdmissionDays* refers to the number of days between the dis-

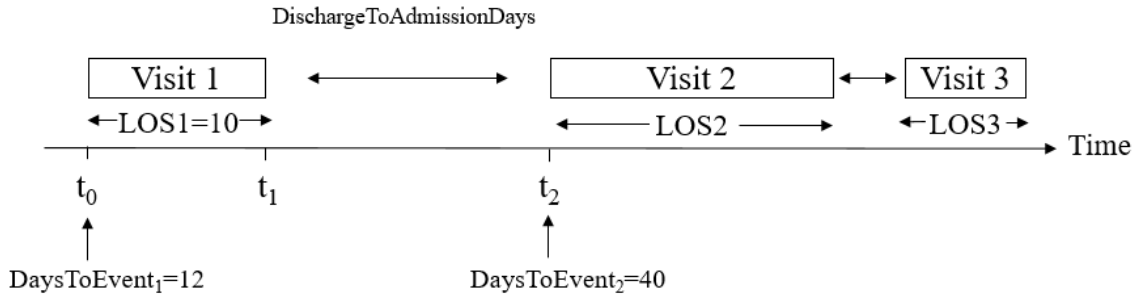


Figure 5: Depiction showing the timeline of a patient's visits

charge date and the next admission date, *DaysToEvent* refers to the number of days between any two consecutive admission dates, and *LOS* refers to the length of stay at the hospital. The subscript in the variable names indicates the visit number, 1 being the first visit and 2 being the second visit.

Calculating the number of days between the discharge date and the next admission date is of substantial importance, especially when the research concerns hospital readmissions. In order to consider that a patient had a readmission, the result of Equation 1 should be less than or equal to 30 days for any two consecutive visits, as shown in Equation 2.

$$ReadmissionIndicator = \begin{cases} Yes, & DischargeToAdmissionDays \leq 30 \\ No, & DischargeToAdmissionDays > 30 \end{cases} \quad (2)$$

The *Core* table reports up to 31 diagnoses (*DXCCSn*) and up to 31 procedures (*PRCCSn*) per discharge as it has 31 diagnosis columns and 31 procedure columns. It is worth mentioning that it is often the case that patients examination returns less than 31 diagnoses, and that the number of procedures they undergo is less than

Table 4: Procedures' categories as listed in the H-CUP

Procedure Category	Description
Minor Diagnostic	Non-operating room procedures(diagnostic)
Minor Therapeutic	Non-operating room procedures (therapeutic)
Major Diagnostic	Operating room procedures performed for diagnostics
Major Therapeutic	Operating room procedures performed for therapeutic

31. Furthermore, even though a patient might have gone through several procedures during a given visit, the primary procedure that occurred at the visit discharge is assumed to be the first procedure (*PRCCS1*). Table 4 shows the different categories of procedures. The Present on Admission (*DXPOAn*) indicator identifies the diagnoses that were present when the patient was admitted. Since the dataset represents discharge data, then this feature is useful for identifying the diagnoses that were present at the time of admission rather than the time of discharge. In addition to the features explained above, there are several demographic data that are reported in the *Core* table as well, such as race, age range, sex, living area, etc.

CHAPTER 4: PREDICTING THE RISK OF MORTALITY AND HOSPITAL READMISSIONS

In this chapter, we investigated a two-fold problem that predicts the risk of mortality and hospital readmission for newly admitted patients. For example, given a medical dataset, we would like to build a model that is capable of predicting with a high accuracy the likelihood that a newly admitted patient could be at risk of death or hospital readmission. We applied a supervised learning model to predict the status of patients when they get discharged. Several machine learning algorithms were used on our medical dataset to build an accurate classifier. The algorithms that are used: Naïve Bayes, Decision Tree, Logistic Regression, Neural Networks, and Support Vector Machine. In addition to that, feature filtering techniques and boosting were applied to enhance the prediction accuracy and utilize the processing performance.

4.1 Data Preprocessing

Data preprocessing is the most important step in machine learning. The better understanding and screening of data helps in avoiding any misleading results in the steps to follow. There are several methods used to prepare the data depending on the nature of data itself. In this work, we used five methods to prepare the data for the prediction step. These methods are: *cleaning*, *formatting*, *sampling*, *under-sampling*, and *aggregation*. Following is an explanation of the steps underwent in each method:

- **Cleaning:** in this step we dealt with the missing instances in the dataset. That being said, we deleted some of them and replaced the remaining with 0's were applicable. The 0 in this case means that the patient does not exhibit a certain diagnosis or did not undergo a certain procedure.
- **Formatting:** most of the columns in the dataset are formatted in a way to include different codes for the diagnoses and procedures, which makes the header of the column insignificant in the prediction process (think about it as a bag-of-word). Therefore, we transposed the data in a way to create columns for all the possible diagnostic and procedure codes. In addition to that, we transformed the data into a binary representation to simplify the classification process and make it faster. At the end of this step, we ended up with 520 features.
- **Sampling:** the Core table in the SID includes a large number of features, which makes it very hard to process. Therefore, we have selected a subset of features that are relevant to the problem. By doing so, we increased the usage of the processing and memory capabilities.
- **Under-Sampling:** one of the problems we faced in the classification process is the skewed distribution of the classes. The number of the positive classes in the mortality and readmission datasets is way less than the negative classes. The classification of an imbalance data, as a result, will give us a non-realistic accuracy. In order to deal with the imbalanced dataset, we reduced the number of the majority classes (negative classes) to be close to the minority classes (positive classes).

- Aggregation: our dataset does not include a feature that gives us an indication whether a patient got readmitted or not. Therefore, we aggregated three features (*VisitLink*, *DaysToEvent*, and *LOS*) into one feature that is used later as a class feature in the readmission prediction model as shown in Eq. 1 in Chapter 3. In order to consider that a patient had a readmission, the result of Eq. 1 should be less than or equal to 30 days for any two consecutive visits, as shown in Eq. 2 in Chapter 3.

4.2 Predicting the Risk of Mortality and Hospital Readmission

Several supervised learning algorithms are used to predict the mortality of patients while they are in hospice care and the likelihood of patients' readmission. The specific algorithms used are: Naïve Bayes, Decision Tree, Logistic Regression, Neural Network, and Support Vector Machine. These algorithms are selected due to their prevalent use in the field of machine learning and high level of support in the form of libraries and frameworks in various programming environments.

Overall, we were able to achieve a reasonably high accuracy in predicting the likelihood of mortality and hospital readmission. Tables 5 and 6 show the accuracies achieved by each algorithm on both the training and testing datasets for the mortality and readmission problems respectively. As can be noticed from both tables, the predictability of mortality is higher than the predictability of readmission. Moreover, the algorithms gave relatively close accuracies, however, some algorithms gave slightly better accuracies than others, such as neural networks and support vector machine, in both the mortality and readmission predictions. In addition to that, decision tree

Table 5: Predicting mortality accuracies

Algorithm	Training Accuracy	Testing Accuracy
Naïve Bayes	87.43%	80.47%
Decision Tree	96.29%	80.34%
Logistic Regression	84.65%	65.21%
Neural Network	84.64%	82.78%
SVM	95.21%	82.68%

Table 6: Predicting readmission accuracies

Algorithm	Training Accuracy	Testing Accuracy
Naïve Bayes	62.76%	61.98%
Decision Tree	91.38%	56.92%
Logistic Regression	64.57%	63.59%
Neural Network	64.38%	63.1%
SVM	76.75%	62.71%

outperforms the other algorithms over the training dataset, which is foreseeable as decision trees tend to over-fit the training data.

4.3 Feature Selection

The high dimensionality of the dataset sometimes has a negative impact on the accuracy of the prediction model. Our dataset has 520 features, as described in Section 4.1, however, not all of them have the same impact on the prediction. Therefore, instead of using the whole set of features, we can select a subset that has significant impact on the prediction. We used three techniques to assign weights for the most significant features with respect to each prediction problem, which are: Entropy-Based, Chi-Squared, and Correlation filtering. Each of these techniques is explained briefly in Table 7.

We applied the aforementioned feature filtering techniques on both the mortality

Table 7: Descriptions of the feature selection techniques

Filtering Method	Description
Entropy-Based Filter	Entropy-Based filtering is used to determine the overall relevance of a feature in predicting the decision class. This technique assigns high weights to the features that return the highest information gain, which is an indication of the homogeneity of the dataset.
Chi-Squared Filter	Chi-Squared filtering uses the chi-square statistical test to measure the independence of two events. Each feature is tested against the decision class feature. If the feature is found to be independent, then it is discarded. Otherwise, the feature is considered significant to the decision class feature.
Correlation Filter	Correlation filtering works similarly to the chi-squared filter, but in an opposite way. A feature is selected if it has a high correlation with the decision class feature. That being said, each filtering technique assigns weights differently.

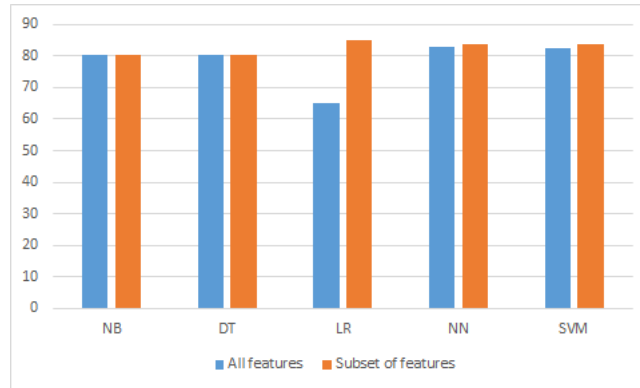


Figure 6: The effect of the chi-square filtering on the mortality dataset

and readmission data. We could reduce the number of features by [50-70]%. We selected the reduction percentage that gave us the best accuracy. Although the results did not show a significant improvement in the testing accuracy as shown in Figure 6. However, there was a noticeable improvement in the processing performance. Another advantage is the ability to list the features that are significant in predicting the risk of mortality and readmissions as shown in Table 8.

4.4 AdaBoost

The ultimate goal that we plan to achieve when dealing with classification algorithms is to build a prediction model, given a training dataset, that is able to generalize on a testing dataset and provide a high accuracy. One way to achieve that is by using boosting, which is simply defined as the process of building a highly accurate classifier model by combining a set of weak classifiers. Adaptive Boost (AdaBoost) algorithm, proposed by Freund and Schapire [9], is the very first implementation of boosting that maximizes the accuracy of a classifier by focusing on the points where the classifier does not perform well. The algorithm works by testing the model repeatedly using different portions of the training dataset. The selection of the data points,

Table 8: The most significant features in the mortality and readmission predictions. The prefix in the CCS code columns indicates whether the code refers to a procedure (PR) or a diagnosis (DX)

Readmission		Mortality	
CCS Code	Description	CCS Code	Description
DX158	Chronic kidney disease	DX131	Respiratory failure, insufficiency, arrest (adult)
DX59	Deficiency and other anemia	PR216	Respiratory intubation and mechanical ventilation
DX99	Hypertension with complications and secondary hypertension	DX249	Shock
PR137	Other procedures to assist delivery	DX2	Septicemia (except in labor)
PR58	Hemodialysis	DX107	Cardiac arrest and ventricular fibrillation
DX55	Fluid and electrolyte disorders	DX157	Acute and unspecified renal failure
PR134	134:Cesarean section	DX55	Fluid and electrolyte disorders
DX181	Other complications of pregnancy	DX122	Pneumonia (except that caused by tuberculosis or sexually transmitted disease)
DX195	Other complications of birth, puerperium affecting management of mother	DX106	Cardiac dysrhythmias

Table 9: The effect of AdaBoost on the decision tree classifier

	Readmission		Mortality	
	Decision Tree	AdaBoost	Decision Tree	AdaBoost
Training accuracy	91.38%	60.57%	95.85%	86.31%
Testing accuracy	56.92%	59.39%	80.46%	86.29%

to be used in building the model, is based on a weight ($\omega_i(x, y)$) given to the points where the model performed poorly. This weight gives these data points advantage over the other data points in the selection process in the next iteration. The process keeps repeating n times in which the weight adapts in each iteration until a stronger classifier is built.

In order to study the effect of the AdaBoost on our dataset, we applied it on the decision tree classifier and noticed the improvement on the accuracy over the training and testing datasets. Table 9 shows a comparison before and after using AdaBoost on the readmission and mortality datasets. As we can notice, AdaBoost could build a stronger classifier that is able to better predict the class label. The accuracy has increased by 4% and 7% for the readmission and mortality predictions respectively. On the other hand, the accuracy of the training dataset was decreased, which means that the generated model does not over-fit the dataset and can generalize on new observations.

CHAPTER 5: INTRODUCING PROCEDURE PATHS AND PROCEDURE GRAPH

Procedure paths are defined as the sequence of procedures that a given patient undertakes to reach a desired treatment. In other words, a procedure path is a detailed description for the course of treatments provided to an admitted patient. The length of any given procedure path is an indicator of the number of readmissions that occurred or will occur throughout the course of treatment. For example, one procedure path for a patient could be the following: $path_x = (p_1, p_3, p_3, p_6)$, where p_i ($i = 1, 3, 6$) indicates a particular procedure. According to the procedure path ($path_x$), the number of readmissions was 3.

In this research, we lay the foundation for predicting procedure paths by devising a system that will anticipate the following procedure (or readmission). We will also introduce a way to extract action rules that describe medical recommendations that will rectify the following procedure for new patients.

The *procedure graph* for some procedure p is defined as the tree of all possible procedure paths extracted from our dataset for patients who underwent procedure p as their first procedure. Figure 7 shows a depiction of the *procedure graph*. $P_{(0,1)}$ is the initial procedure that patients start with, the next procedure could be any procedure from $P_{(1,1)}$ to $P_{(1,n)}$, which is determined by the resulting set of diagnoses after performing the initial procedure $P_{(0,1)}$. The first argument x in the notation

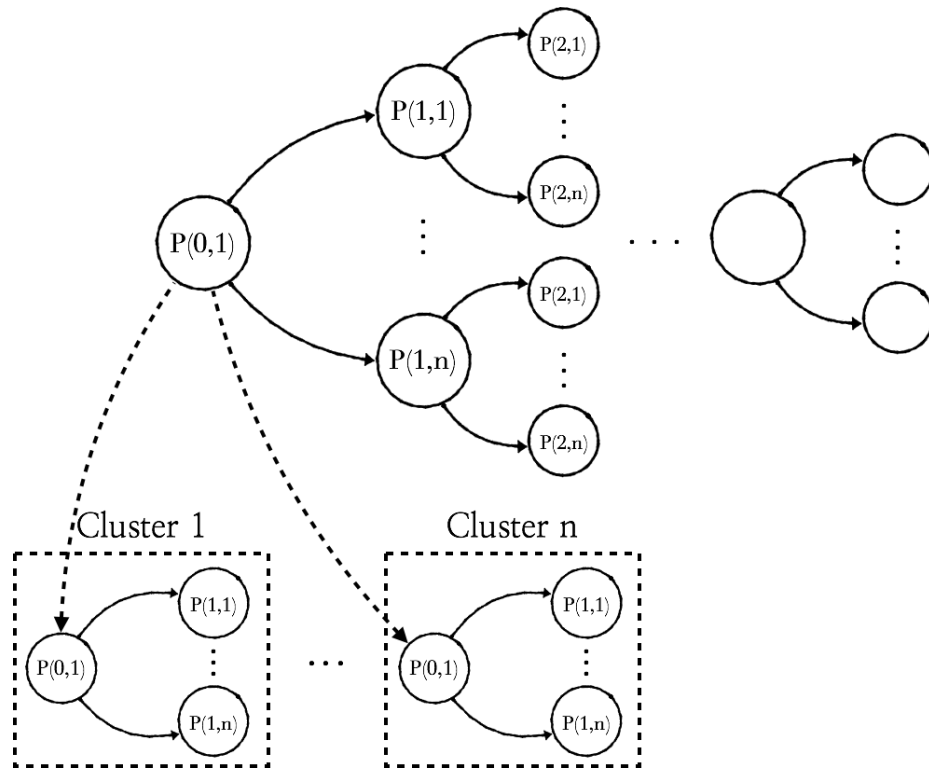


Figure 7: Depiction of a procedure graph

$P_{(x,y)}$ refers to the number (or rather level) of readmission, and the second argument y refers to the procedure identifier at that level. For example, $P_{(1,2)}$ refers to the procedure with identifier 2 that occurred at the first level of readmissions (e.g first readmission following the initial procedure). The portions of the graph that are contained in dashed boxes depict the personalization part that we introduce in the next section. The idea of personalization is to cluster patients that are scheduled to undergo procedure $P_{(0,1)}$ according to their diagnoses. As a result of this clustering, we will be able to anticipate with higher accuracy the following procedure (readmission) that the patient will undergo by identifying which cluster the new patient belongs to. In the following sections, we will provide some information about the number of different possible paths and the length of each path.

5.1 Unique Procedure Paths

The number of all procedure paths is extremely high. This high number of unique procedure paths indicates that it is not true that there exists a single universal course of treatment that patients typically follow to reach the desired state. For example, the number of patients that underwent procedure 222 (Blood Transfusion), as their first procedure, is 72,521 and the number of unique procedure paths that those patients underwent is 1,230 paths. Blood transfusion is considered a minor procedure and this could explain the high number of unique procedure paths. Now, let us consider a major procedure, such as 158 (Spinal fusion), the number of patients who underwent this procedure, as their first procedure, is 72,928 and the number of unique procedure paths that those patients underwent is 443 paths. Although the number of unique paths, in case of the major procedure, is reduced by $1/3$. However, the number is still high and this emphasizes the fact that patients do not follow the same path in their course of treatment. Figure 8 shows each procedure with the number of possible unique paths. The procedure with the highest number of unique paths is 22 (Blood Transfusion) and the average number of unique paths for all procedures is 144.7. Figure 9 shows each procedure with the number of patients who underwent that procedure as their initial procedure. The procedure with the highest number of patients is 137 (Other procedures to assist delivery) with a total of 200,646 patients. It is worth mentioning here that this number of patients does not equal to the number of patients who underwent this procedure in the dataset. However, this number represents the number of patients who underwent this procedure as their first

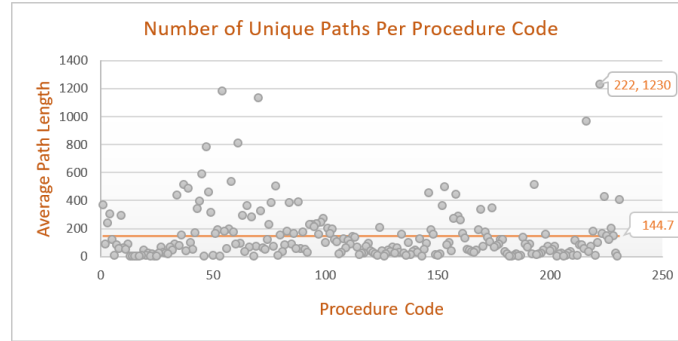


Figure 8: Number of unique paths per procedure

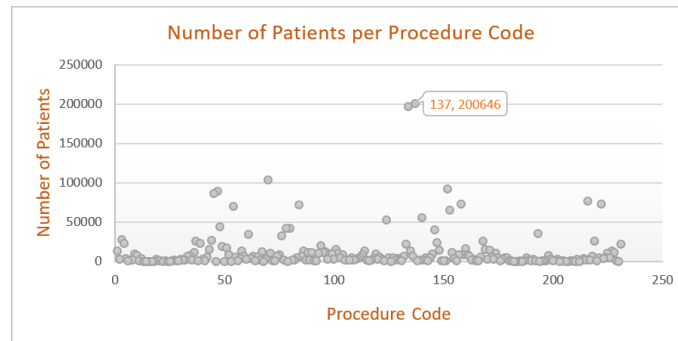


Figure 9: Number of patients per procedure code

procedure.

5.2 Lengths of Procedure Paths

The length of the procedure path is an indicator of the number of readmissions during the course of treatment. A procedure may have different paths' lengths depending on the number of possible unique paths that the patient may follow. Knowing the length of the path is a valuable information for the physicians in the treatment process. The fact that the physicians can now anticipate the next procedure in the course of treatment makes them reconsider their decisions and select a better path that has a shorter length. The relation between the path length (number of readmission) and the number of nodes (procedures) in that path is defined as follows:

$$\text{Path length (Number of readmission)} = \text{Number of nodes (procedures)} - 1$$

Let us now consider a real example from the dataset showing the different paths' lengths for a certain procedure. Suppose that a new patient is admitted to the hospital and the physician prescribed procedure 58 (Hemodialysis) to be the initial procedure based on the patient's diagnoses. Knowing that the patient will undergo procedure 58 as the initial procedure allows us to anticipate what could be the maximum and the average number of possible readmissions, which are 36 and 4.2 respectively. Figure 10 shows the maximum path length for each procedure. Procedure 61 (other OR procedures on vessels other than head and neck) has the maximum path length of 56 readmissions. Figure 11 shows the average path length for each procedure. Procedure 224 (cancer chemotherapy) has the highest average path length of 6.34 readmissions and the average path length for all procedures is 2.88, which is almost 3 readmissions in average. Note that these numbers may include outliers, for example, there could be only one patient who had the maximum path length of 56. In Chapter 7, we will provide a more robust mechanism that provides a score for each procedure, which represents the average number of hospital readmissions

The process of selecting a certain path follows a probabilistic function, where the probability of following any path can be calculated by dividing the number of patients at the end of the path by the number of patients who underwent the initial procedure, as seen in the following equation:

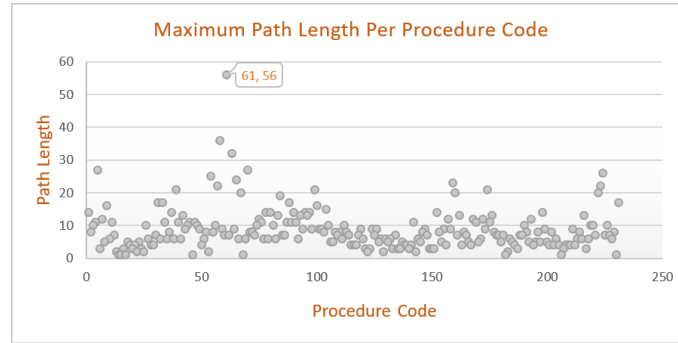


Figure 10: Maximum path length per procedure

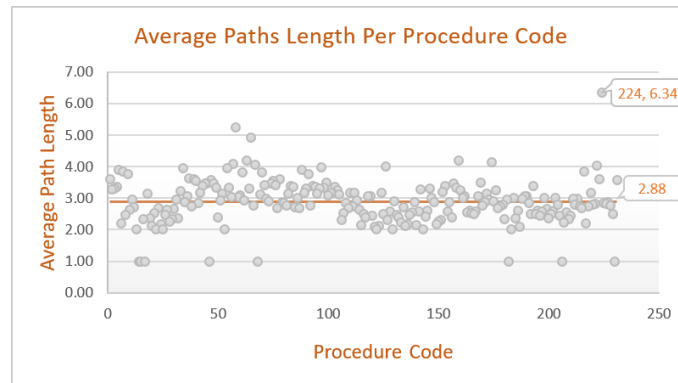


Figure 11: Average path length per procedure

$$prob(path_k) = freq(P_{(l,i)})/freq(P_{(0,1)})$$

where $prob(Path_k)$ refers to the probability of following the k_{th} path, $freq(P_{(l,i)})$ refers to the number of patients in the last node (procedure) and $freq(P_{(0,1)})$ refers to the number of patients in the initial node (procedure). l refers to the number (or rather level) of readmission, which is in this case the last level and the second argument i refers to the procedure identifier at that level.

Table 10 shows the highest probabilities of the paths for procedure 105 (Kidney Transplant). Following is a description of the procedure codes listed in Table 10:

- 54: Other vascular catheterization not heart
- 70: Upper gastrointestinal endoscopy biopsy
- 88: Abdominal paracentesis
- 99: Other OR gastrointestinal therapeutic procedures
- 110: Other diagnostic procedures of urinary tract
- 111: Other nonOR therapeutic procedures of urinary tract
- 193: Diagnostic ultrasound of heart echocardiogram
- 231: Other therapeutic procedures

The number of patients in the nodes (procedures) that form a procedure path decreases as we move to the right until we reach to the last node, where the number

Table 10: Procedure paths with the highest probabilities for procedure 105 (Kidney transplant)

Procedure path	Probability (%)
{105, 231}	0.404
{105, 193}	0.231
{105, 70, 54}	0.173
{105, 110, 111}	0.115
{105, 88, 99}	0.058

of patients becomes very low (approaching 1). This means for a certain path, the probability of predicting the second procedure is higher than the probability of predicting the third procedure and so on until we reach the last procedure. In order to increase the predictability of a certain path or the predictability of the next procedure, we need to apply personalization on the patients by clustering them into groups according to their common characteristics (features). These features could be fixed or flexible based on whether we can change them or not. For example, gender, age, and race are considered fixed features. On the other hand, diagnoses and procedures are considered flexible features. In Chapter 6, we will provide an elaborate description of the clustering approaches we propose.

CHAPTER 6: INTRODUCING PATIENTS PERSONALIZATION

Given a new patient, being able to anticipate the procedure path for that patient is an invaluable asset to medical doctors since it can be used as a mean to inform the patient of his or her course of treatment, and ultimately altering or amending the course of treatment accordingly. Our assumption that we use in this work, which aligns with the definition of our dataset, is that the procedure performed on the patient is determined from the set of diagnoses that the patient is diagnosed with. In addition to that, the set of diagnoses will also determine the state in which the patient ends up in. In this case, the state would be the set of diagnoses after performing the first procedure, which as a result, will determine the second procedure.

Although it is theoretically possible to create a chain of predictions that will provide a complete prediction for the entire procedure path, we only examine predicting the following (next) procedure in this work for higher prediction accuracy. If the following procedure is part of a desired procedure path then no intervention is needed. Otherwise, a set of medical recommendations, if at all feasible, are needed to enhance the final status of the patient.

To predict the procedure path, or rather the following procedure, we start by extracting knowledge from our existing dataset. The approach used to predict the following procedure is an unsupervised clustering technique based on the set of di-

Table 11: Some of the most common set of exact diagnoses for procedure 158 (spinal fusion)

Set of diagnoses	Number of patients	Entropy
{205}	502	3.015
{205, 98}	128	2.752
{205, 663}	86	2.543
{205, 209}	67	2.798
{205, 211}	51	2.510

agnoses that patients exhibit at the time of their first admission. Our assumption is that patients that exhibit similar set of diagnoses will end up with a similar set of diagnoses after the procedure, and again, by definition of our dataset, these set of diagnoses will determine the next procedure. Next, we provide two different clustering approaches: the exact matching clustering approach and the rough matching clustering approach.

6.1 Exact Matching Clustering

In the exact match clustering approach, we define a cluster by a set of diagnoses. For a given patient to belong to any cluster, he or she must have the exact same diagnoses set. Table 11 shows some of the most common sets of exact diagnoses for patients who undertook procedure 158 (spinal fusion). From Table 11, we observe that for procedure 158, the cluster with the exact set of diagnoses {205} contains 502 patients. This means that there are 502 patients that were diagnosed with 205 and nothing else, before undertaking procedure 158. Similarly, the second row of Table 11 means that there are 128 patients that were diagnosed with both diagnostic codes 205 and 98 and nothing else, before having to undergo procedure 158.

Here is a description of each diagnostic code provided in the table:

- 205: Spondylosis, intervertebral disc disorders, other back problems
- 98: Essential hypertension
- 663: Screening and history of mental health and substance abuse codes
- 209: Other acquired deformities
- 211: Other connective tissue disease

We should also mention here that the entropy for the entire system before this personalization attempt is 3.667, which is higher than all the entropy values in the table. The weighted entropy for all exact matching clusters that have size above threshold 50 is 2.892. This implies that by applying this clustering approach, we will be able to have a higher level of predictability of which following (next) procedure is likely to be undertaken. In other words, by knowing the cluster of which a patient belongs to, we would be able to anticipate (with higher accuracy), where that patient is likely to end up after performing the first procedure.

One advantage of using the exact matching clustering is that medical recommendations can be precisely described. For example, if we discovered that patients in cluster $c_1 = \{205\}$ tend to end up in a state that is more desired than patients in cluster $c_2 = \{205, 98\}$, then we can precisely devise the medical recommendations that are needed to be done, in this case, only healing diagnostic code 98.

The fact that patients are usually admitted with other diagnoses that are often irrelevant to the main diagnosis (or diagnoses), makes this approach quite limited,

which is rather evident in the frequencies (number of patients) that exhibit the most common set of exact diagnoses, compared to the number of patients that exhibit the same diagnoses but along with other diagnoses that may be irrelevant. For example, the number of patients that exhibit diagnosis code 205 along with other diagnoses is 13,096, which is substantially larger than the number of patients that only exhibit diagnosis code 205. In the next section, we present a new clustering approach that addresses this limitation.

6.2 Rough Matching Clustering

In this section, we define a novel clustering method, called *Rough Clustering*, to specify the properties that the newly admitted patient needs to exhibit to belong to a given cluster. As mentioned previously in this chapter, the need for clustering is of a great value in this research. The reason is derived from our ultimate goal, which is to increase the predictability of the next procedure and to put the patient on the most optimal procedure path that can eventually reduce the number of hospital readmissions. Unlike the exact matching clustering approach where we define one unique set of diagnoses that needs to exactly match the patient's set of diagnoses.

The rough clustering algorithm works by clustering the patients according to the diagnoses they have and the diagnoses they do not have, which is similar to the *Rough Set Theory* [18] in the sense it has an upper and lower approximations. The rough clustering defines three sets of diagnoses in each cluster to determine whether a new patient belongs to the cluster or not. The first set, which we call the *included set*, describes the set of diagnoses that any given patient needs to exhibit for that patient

to belong to the cluster. The second set, which we call the *excluded* set, is the set of diagnoses that patients cannot exhibit for them to belong to that cluster, and finally the third set, which we call the *optional* set, is the set of diagnoses that patients can, but do not need to, exhibit for them to belong to that cluster.

Since the *optional* set is the complement of the *included* and *excluded* sets combined, we decided not to specify it each time we define a cluster. For example, if the entire set of diagnoses is $D = \{d_1, d_2, \dots, d_{10}\}$, the *included* set of some cluster c is $included(c) = \{d_1, d_2, d_5, d_7\}$, and the *excluded* set of the same cluster c is $excluded(c) = \{d_6, d_8, d_9, d_{10}\}$, then the *optional* set for cluster c is $optional(c) = \{d_3, d_4\}$, which is equal to $D - [included(c) \cup excluded(c)]$.

To further explain the included and excluded set, let us examine the following cluster, which is extracted from our dataset for the patients who underwent procedure 158 (spinal fusion) as their first procedure. The extracted cluster is described by the following diagnoses: $\{98, -49, -138, -211\}$. The included and excluded sets, therefore, can be expressed as follows: *Included*: $\{98\}$ and *Excluded*: $\{-49, -138, -211\}$. In order for a new patient to belong to this cluster, the patient must exhibit all the diagnoses in the *Included* set, which is in this example only diagnostic code 98 (essential hypertension). In addition to that, the patient must not exhibit any of the diagnoses in the *Excluded* set, which is in this example diagnostic code 49 (diabetes mellitus without complication), 138 (esophageal disorders) and 211 (other connective tissue disease).

There are other clustering algorithm available in the literature that we considered

in this research. However, we found out that the rough clustering that we developed is more appropriate taking into account our requirements for personalization and actionability. We are interested in a controlled way of building clusters where stopping conditions are created in real time. The criterion that defines whether a patient belongs to a cluster is not solely dependent on the features that the patient exhibits, but also on the features that the patient does not exhibit. These clusters yield hierarchy of personalized recommender systems which interact with each other. The goal is to find an optimal treatment process for a newly admitted patient right after the primary medical procedure is identified. There is a level of customization and actionability that we need to apply in this research that perfectly fits the clustering strategy we have developed.

The main reason why this rough matching approach is more superior than the exact matching approach is because it is not based on the assumption that each diagnosis must be relevant to the procedure. However, note that although this rough approach was designed to be less strict so that it counteracts or rectifies the main disadvantage of the exact matching approach, this approach is in fact still precise enough to devise the medical recommendations as will be explained later.

The methodology used to extract all combinations of clusters is similar to the association action rules extracting approach presented in [20]. We start by extracting all 1-diagnosis clusters that lie within the range specified (for both the *included* and *excluded* sets). Then we build 2-diagnosis clusters by combining all possible pairs of 1-diagnosis clusters. Next step would be to construct 3-diagnosis clusters by combining all the 2-diagnosis clusters with all the 1-diagnosis clusters, so on and so forth. Using

this procedure however, the number of generated clusters will grow extremely fast. Iterating through all possible clusters for each patient to verify whether that patient belongs to a given cluster or not would be a highly inefficient implementation. Instead, we used a retrieval digital tree implementation that starts the verification process at the root of the tree, which would allow us to discard entire subtrees anytime the patient does not satisfy a node constraint (whether that constraint was an *included* constraint or an *excluded* constraint).

As mentioned earlier in this section, the main goal of clustering patients according to the diagnostic codes is to increase the predictability for the next procedure, which can be measured by calculating the entropy. Constructing more combinations of the diagnostic codes enforces more personalization on the patients and as a result improves the predictability of the next procedure (the entropy decreases). Figure 12 illustrates the process of generating the clusters. The top of the tree, Level 1, shows the 1-element set clusters, which are represented by all possible diagnostic codes for a given procedure, *included sets* : $\{d1\}$, $\{d2\}$, and $\{d3\}$ and their negations, *excluded sets* : $\{-d1\}$, $\{-d2\}$, and $\{-d3\}$. The process continues and in order to generate clusters for any Level ' L_n ', we pair all elements in Level ' $L_{(n-1)}$ ' with elements from ' L_1 '.

It is also worth mentioning here that any diagnostic code can not exist with its negation in the same cluster. For example, we can not have $\{d1\}$ and $\{-d1\}$ in the same cluster, as this means the patients who belong to that cluster have the diagnostic code $d1$ and do not have it at the same time.

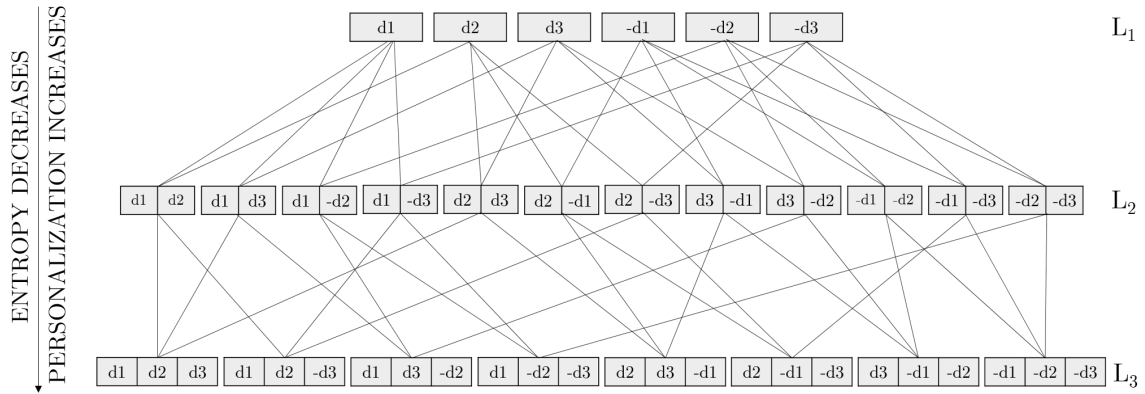


Figure 12: Construction of the patients' clusters

Now let us show the benefits of clustering patients by providing a hypothetical example that mimics a real life scenario. Figure 13 shows the clusters on the procedure graph. The list of procedures shown in each cluster represents the next procedures that the patients in that cluster are more likely to follow. The distribution of the following procedures implies the most probable procedure that a new patient will undergo after the first primary procedure. For example, the patients in the first cluster are more likely to undertake $P_{(1,2)}$ after $P_{(0,1)}$.

The first cluster (Cluster 1) contains 60 patients, from the 60 patients that belong to Cluster 1, 10 out of which ended up undergoing procedure $P_{(1,1)}$, 45 ended up undergoing procedure $P_{(1,2)}$, only 5 patients ended up undergoing procedure $P_{(1,3)}$, and 18 patients did not come back to the hospital. According to our example, this distribution of following procedures implies that if a patient exhibits the set of diagnoses that the first cluster is defined by, then that patient will most likely end up in a state that will require him or her to undergo procedure $P_{(1,2)}$. Similarly, the distribution of Cluster 2 and Cluster 3 will imply that most new patients that will belong to Cluster 2 will end up undergoing procedure $P_{(1,1)}$, and that most new patients that will belong

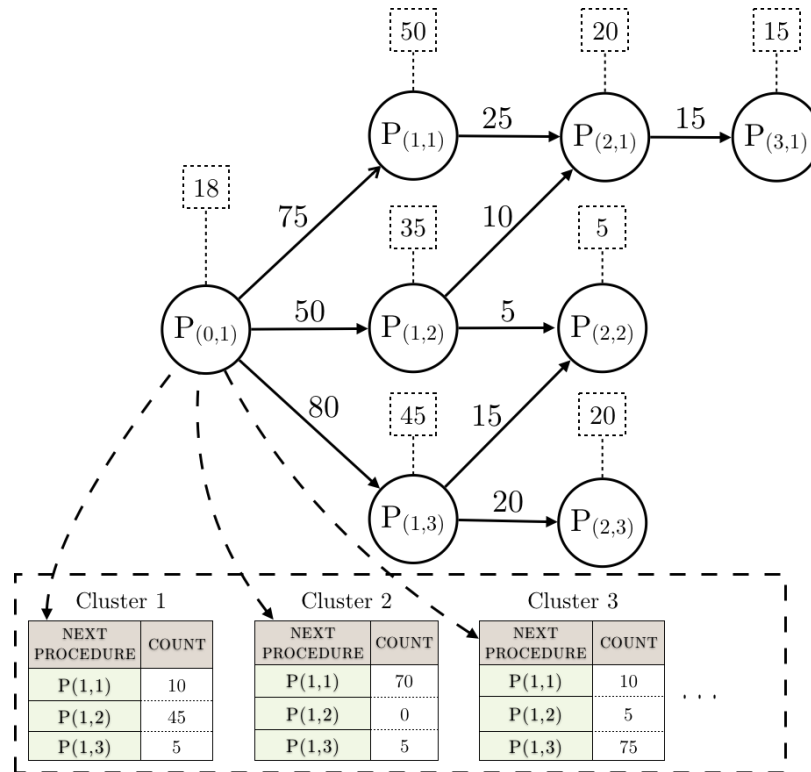


Figure 13: Depiction of the clusters on the procedure graph

to Cluster 3 will end up undergoing $P_{(1,3)}$.

6.3 Clusters Filtering

The main goal of clustering (personalizing) the patients is to group them according to their common characteristics. Patients' personalization can increase the predictability of the next procedure. In order to achieve a high level of personalization, we need to generate more levels of clustering. However, as we go down the tree, the number of generated clusters will increase tremendously. The number of all possible CCS diagnostic codes is 283, which means that there will be 566 clusters in the first level (including the included and excluded one-element sets). In the second level, there will be 159,612 generated clusters and so on. The number of generated clusters in any level can be calculated using the following equation:

$$Num_Clusters = \frac{n!}{l!(n-l)!} - \frac{d!}{(d-l+1)!} \quad (3)$$

where d refers to the total number of all possible diagnoses (283 in case of CCS and 15,072 in case of ICD-9-CM), and n refers to the number of one-element sets on the first level, which is double the number of all possible diagnoses ($n=2*d$).

There is a need to filter the clusters and only select the clusters that can help in predicting the next procedure. In the following sections, we will present two filtering methods.

6.3.1 Percentage Filtering

The filtering criteria we followed is based on the fact that if the majority of patients have a common diagnosis, then this diagnosis will not play a significant role in determining the state for which the patients will end up in. The complement of this statement is also true, that is, if the majority of patients do not have a specific diagnosis, then not having this diagnosis will not play a significant role in determining the state for which the patients will end up in. Therefore, we defined a range between [20-80]% and we only considered the diagnostic codes that lie within this range. This means, we will consider a cluster only if [20-80]% of the total number of patients exhibit the diagnoses in that cluster. Table 12 shows the number of clusters and the entropy for different levels of clustering and different ranges for procedure 158 (spinal fusion). By examining the table, we can notice that as we expand the filtering range, the number of clusters increases. Also, we can observe that the weighted entropy

Table 12: Number of clusters and the entropy for different element clusters and different ranges for procedure 158 (spinal fusion)

	Range					
	20% to 80%		10% to 90%		5% to 95%	
	# of clusters	Entropy	# of clusters	Entropy	# of clusters	Entropy
1-element Clusters	14	5.105	36	5.109	66	5.117
2-element Clusters	37	5.051	379	5.028	1532	5.041
3-element Clusters	50	4.988	2097	4.958	19167	4.971
4-element Clusters	44	4.916	6969	4.905	155028	4.91

decreases as we increase the level of personalization.

6.3.2 Co-morbidity Filtering

The percentage filtering technique could successfully cut down the number of generated clusters. However, we are interested in rectifying our clustering algorithm by keeping the clusters that include the diagnoses that are prevailing in determining where the patient is more likely to end up in. These clusters will be a better representation of the patients' medical conditions and will enhance the predictability of the next procedure. That said, by looking at the different categories of the diagnoses, we found that the *Comorbid* conditions can provide us with this information. A comorbid condition is defined as "a clinical condition that exists before a patient's admission to the hospital, is not related to the principal reason for the hospitalization, and is likely to be a significant factor influencing mortality and resource use in the hospital." [6]. Table 13 shows the different categories of comorbid conditions for the ICD-9-CM coding scheme. The comorbid conditions are summarized into (30) categories that include (1199) ICD-9-CM codes and around 50 CCS codes.

By only selecting the clusters that include these comorbid conditions, we can reduce

Table 13: Categories of the comorbid conditions

Congestive heart failure	Cardiac arrhythmias	Valvular disease	Pulmonary circulation	Peripheral vascular	Hypertension
Paralysis	Neurological disorder	Chronic Pulmonary	Diabetes (uncomplicated)	Diabetes (complicated)	Hypothyroidism
Renal failure	Liver disease	Peptic ulcer disease	AIDS	Lymphoma	Metastatic cancer
Solid tumor	Rheumatoid arthritis	Coagulopathy	Obesity	Weight loss	Fluid/electrolyte disorder
Blood loss anemia	Deficiency anemias	Alcohol abuse	Drug abuse	Psychoses	Depression

the number of clusters and refine our clustering algorithm. In addition to that, the benefit of this filtering technique goes beyond the clustering to basically rectify the extraction of the recommendations (actionable knowledge) that we will describe in Chapter 8.

In the following chapter, we introduce a novel algorithm that assigns a score to each procedure in the procedure graph by taking into consideration the number of patients and the length of the procedure path. This procedures' score is later consequently used to calculate the score for each cluster to determine the risk of readmission, which would guide us to extracting the actionable knowledge (medical recommendations). The main benefit of scoring the generated clusters is that physicians will be able to identify the risk of readmission for newly admitted patients and dedicate more care for them. Additionally, these scores will be used to devise the medical recommendations that will enhance the final status of patients.

CHAPTER 7: PROCEDURES AND CLUSTERS EVALUATIONS

Our ultimate goal in this research is to provide recommendations (actionable knowledge) to physicians to put the patients on the optimal procedure path. In Section 6.2, we presented a novel algorithm to cluster the patients based on the diagnoses they have and they do not have. However, we still do not have details whether the cluster that the patient belongs to will lead the patient through the most optimal procedure path. For this reason, we need a metric (score) system to evaluate the clusters that the patients belong to. In this chapter, we define two functions to first evaluate (score) the procedures in the procedure graph and second evaluate the clusters.

7.1 Procedures' Scoring Function

In Chapter 5, we introduced the procedure graph, which allows physicians to visualize all the possible procedure paths that a new patient may undertake during the course of treatment. Since patients can now be clustered based on their diagnoses, then we can anticipate the most probable procedure path and predict the next procedure. That said, without having a metric system that can evaluate (score) the level of desirability of the next procedure, we would not be able to determine whether there exist(s) following procedures that are more desired than the anticipated following procedure. For this reason, we define a function to evaluate procedures in the procedure graph. We define the procedure score recurrence function as:

$$\text{score}(P_x) = \begin{cases} \sum_{k=1}^n \frac{|P_k|}{|P_x|} * (1 + \text{score}(P_k)) & \text{if } n \geq 1 \\ 0, & \text{otherwise} \end{cases}$$

where n denotes the number of procedures directly following procedure P_x , P_k denotes the k th procedure right after P_x , and $|P_k|$ denotes the number of patients that underwent the k th procedure.

The output of the procedure score function represents the average number of readmissions. The function starts by assigning a zero score to the leaves in the procedure graph, which means that there are no readmissions after these procedures. After that, the function moves backward until it reaches the root (initial procedure) of the tree. Figure 14 shows the scores of the procedures in the procedure graph. The score (1.323) of the first procedure $P_{(0,1)}$ means: if a new patient undergoes this procedure as his or her first procedure, then there will be, on average, 1.323 readmissions.

Let us demonstrate how the function works by walking through an example using the hypothetical procedure graph shown in Figure 14. According to Figure 14, 15 patients have undergone procedure $P_{(3,1)}$, and since there are no procedures following $P_{(3,1)}$, this means that all the 15 patients have an average number of future readmission equals to 0. Hence, the score of node $P_{(3,1)}$ will be equal to zero. Now let us examine node $P_{(2,1)}$, the number of patients that undergone $P_{(2,1)}$ is 35 (25 from $P_{(1,1)}$ and 10 from $P_{(1,2)}$), out of the 35, 20 patients did not come back to the hospital and 15 were readmitted to undergo procedure $P_{(3,1)}$. The score of procedure $P_{(2,1)}$ is the sum of weighted score of each possible procedure directly following the procedure $P_{(2,1)}$ as

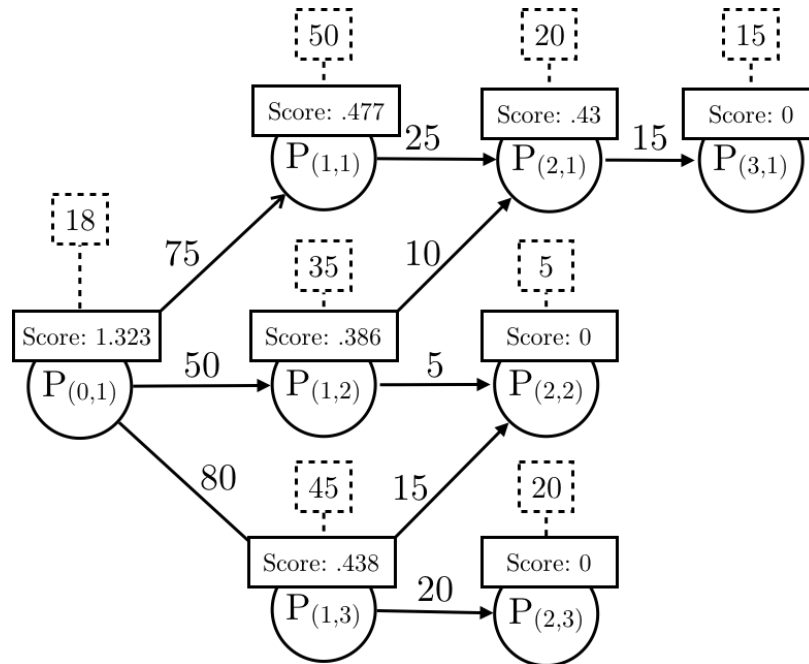


Figure 14: Procedure graph with procedure scores

shown below:

- The weight/probability of the first possibility (no further readmissions) is $20/35$. The score (average number of readmissions) for patients who did not come back to the hospital is zero.
- The weight/probability of the second possibility (undergoing $P_{(3,1)}$) is $15/35$, for which the score (average number of readmissions) will be 1 (which essentially reflects undergoing one more procedure) plus the score of $P_{(3,1)}$, which is zero for this example.

The score of procedure $P_{(2,1)}$ therefore becomes:

$$\text{score}(P_{(2,1)}) = \left(\frac{20}{35} * 0 \right) + \left(\frac{15}{35} * (1 + \text{score}(P_{(3,1)})) \right) = \frac{15}{35} * (1 + 0) = 0.43$$

This essentially means that if a patient were to undergo procedure $P_{(2,1)}$, then the number of following readmission on average is 0.43. Also, in this particular example, since we know that patients can only have one readmission $P_{(3,1)}$, we can also state that since the score is 0.43, then this also means that for any patient who undertakes $P_{(2,1)}$, there will be a 43% chance that he or she will undergo one additional readmission.

Now let us examine one more node: procedure $P_{(1,2)}$. The number of patients that underwent procedure $P_{(1,2)}$ is 50, from which we have three possibilities:

- **Possibility 1:** 35 out of 50 did not come back to the hospital.
- **Possibility 2:** 10 out of 50 were readmitted to undergo procedure $P_{(2,1)}$.
- **Possibility 3:** 5 out of 50 were readmitted to undergo procedure $P_{(2,2)}$.

To calculate the score in this case, we need to calculate the weighted score for each possible following procedure:

- The weight/probability of the first possibility is $35/50$, again however, the score (average number of readmissions) for patients who did not come back to the hospital is zero.
- The weight/probability of the second possibility is $10/50$, for which the score (average number of readmissions) will be 1 (which essentially reflects undergoing $P_{(2,1)}$), plus the score of $P_{(2,1)}$.
- The weight/probability of the third possibility is $5/50$, for which the score (average number of readmissions) will be 1 (which essentially reflects undergoing $P_{(2,2)}$), plus the score of $P_{(2,2)}$. Note here that the score of $P_{(2,2)}$ is zero

since procedure $P_{(2,2)}$ was the last procedure for all patients that went through procedure $P_{(2,2)}$.

The score of procedure $P_{(1,2)}$ hence becomes:

$$\begin{aligned} \text{score}(P_{(1,2)}) &= \left(\frac{35}{50} * 0 \right) + \left(\frac{10}{50} * (1 + \text{score}(P_{(2,1)})) \right) + \left(\frac{5}{50} * (1 + \text{score}(P_{(2,2)})) \right) \\ \Rightarrow \text{score}(P_{(1,2)}) &= 0 + \left(\frac{10}{50} * (1 + .43) \right) + \left(\frac{5}{50} * (1 + 0) \right) = 0.386 \end{aligned}$$

Which again, would mean that for patients that undergo procedure $P_{(1,2)}$, the number of following readmission on average is 0.386, this however does not mean that there is a 39% chance that the patients will undergo additional readmissions, since a single patient may undergo two readmissions.

7.2 Clusters' Scoring Function

In this section, we will use the procedure graph metric system devised in Section 7.1 to introduce a mapping between the scores of nodes in our procedure graph to the scores of clusters. Since clusters contain patients that undergo the same initial procedure but the following procedures may differ for them, the score of a cluster is therefore defined as the sum of the weighted score of the procedures directly following that cluster. So, we define the score of cluster C_x as:

$$\text{score}(C_x) = \sum_{k=1}^m \frac{|P_{(x,k)}|}{m} * \text{score}(P_k)$$

where m denotes the total number of patients in Cluster C_x , and $|P_{(x,k)}|$ denotes the number of patients that underwent the k_{th} next procedure for Cluster C_x . Clearly, $C_x = \bigcup\{P_k : 1 \leq k \leq n\}$.

Now let us go back to Figure 13 shown in Section 6.2 and calculate the scores of the clusters as per the equation above:




- $\text{score}(\text{Cluster 1}) = \frac{10}{60} * .477 + \frac{45}{60} * .386 + \frac{5}{60} * .438 = .406$
- $\text{score}(\text{Cluster 2}) = \frac{70}{75} * .477 + \frac{0}{75} * .386 + \frac{5}{75} * .438 = .474$
- $\text{score}(\text{Cluster 3}) = \frac{10}{90} * .477 + \frac{5}{90} * .386 + \frac{75}{90} * .438 = .439$

The scores calculated above represent the average number of readmissions for the patients in the clusters. The most desired cluster is the one that has the minimum score, which is in this example Cluster 1. If a new patient belongs to Cluster 2, which has the highest score, then this indicates that the patient is in a higher risk of readmission after performing the primary procedure $P_{(0,1)}$ compared to the other patients who are undergoing the same procedure. Knowing that a patient belongs to a cluster with a high score allows the physicians to dedicate more care to the patient and eventually reduce the chances of readmissions in the future. In the following chapter, we will use the scores of the clusters in our recommendation extraction algorithm.

CHAPTER 8: ACTIONABLE KNOWLEDGE DISCOVERY

In this chapter, we present our novel algorithm to extract the actionable knowledge (medical recommendations) that we can provide to physicians to put the patient on the optimal procedure path and reduce the number of anticipated hospital readmissions. In Chapters 6, we presented a clustering algorithm that can group the patients according to the diagnoses (medical problems) that they have and the diagnoses (medical problems) that they do not have. Additionally, the algorithm can group the patients according to their comorbid conditions. We further devised a function to evaluate the clusters, in Chapter 7, to know the desirability of clusters and the risk of readmission. In this chapter however, we build upon the steps presented above to extract the medical recommendations. Let us first present a simple example from our dataset to explain how the recommendations are extracted.

Figure 15 shows an example extracted from the dataset for procedure 5059 (liver transplant). If a new patient belongs to Cluster 1, then there will be no recommendations that we can provide to enhance the patient's medical status, as this cluster has the minimum score (most desirable). However, if the patient belongs to any of the other clusters, then we can provide a set of medical recommendations that can allow the patient to follow the same path as of the patients in Cluster 1. If we do so however, we can reduce the average number of readmission by [48-52]%, which can be calculated by subtracting the score of the cluster the patient belongs to after

Cluster 1	Cluster 2	Cluster 3	Score
0.265	0.515	0.557	
			Clusters
$\{2851, 5715\}$ $\{-7054, -24900, -5722\}$	$\{2851, 5715\}$ $\{-7054, -24900, -5119\}$	$\{2851, 5715\}$ $\{-7054, -2639, -2761\}$	Included/Excluded sets
No Recommendations	2851 \rightarrow 2851 5715 \rightarrow 5715 -7054 \rightarrow -7054 -24900 \rightarrow -24900 \sim \rightarrow -5722	2851 \rightarrow 2851 5715 \rightarrow 5715 -7054 \rightarrow -7054 \sim \rightarrow -24900 \sim \rightarrow -5722	Recommendations
	48.5%	52.4%	Readmission Reduction

2851: Acute posthemorrhagic anemia, 5715: Cirrhosis of liver without mention of alcohol, 7054: Chronic hepatitis C without mention of hepatic coma, 24900: Secondary diabetes mellitus without mention of complication, 5722: Hepatic encephalopathy, 5119: Pleural effusion, 2639: protein-calorie malnutrition, 2761: Hyposmolality

Figure 15: Extracting Recommendations for procedure 5059 (liver transplant) applying the recommendations from the score of cluster the patient belongs to before applying the recommendations and then divide the result by the score of the cluster the patient belongs to before applying the recommendations. Following is a general interpretation for all the possible recommendation forms:

- $(d_1 \rightarrow d_1)$: keep diagnostic code d_1 (the patients in the desired cluster also have this diagnostic code, as it is essential for the specified procedure).
- $(\sim \rightarrow -d_1)$: make sure that the patient does not have diagnostic code d_1 .
- $(d_1 \rightarrow -d_1)$: treat diagnostic code d_1 .
- $(\sim \rightarrow d_1)$: it is essential for the patient to have diagnostic code d_1 to belong to the desired cluster.

In the previous example, we assumed that the patient belongs to only one cluster. However, it is often the case that the patient may belong to multiple clusters. It is

also often the case that when the physician applies the recommendations, the patient will belong to a different set of clusters. Given this new assumption, we need to address the following two-fold question: 1) how to calculate the patient's score? 2) how to calculate the reduction in readmission after applying the recommendations? In the following sections, we will address these questions and explain our updated algorithm for recommendation extraction.

8.1 Entropy-Based Weighted Score for a New Patient

The most straightforward way to calculate the patient's score, given he or she will belong to a set of clusters, would be by calculating the average score of the clusters that the patient belongs to. If we do so, we are mistakenly assuming that all the clusters that the patient belongs to provide us with the same level of information relevance, which is not the case, as there are some clusters that provide more insightful information with respect to the next procedure prediction. Accordingly, we should assign different weights for each cluster depending on the relevance of the cluster's score, and in order to measure the relevance of any cluster, we use the entropy of the clusters with respect to the distribution of the next procedures by using the following equation:

$$e(c_i) = - \sum_{i=1}^m p_i \log(p_i) \quad (4)$$

where p_i is the probability of the i^{th} primary procedure, and m is the number of primary procedures in cluster c_i .

The new weighted score (ws) for cluster c is then calculated by first normalizing

the cluster's entropy, then multiplying the normalized entropy by the cluster's score as shown in the following equation:

$$ws(c) = \frac{e(c)}{\sum_{i=1}^m e(c_i)} * score(c) \quad (5)$$

The final step would be to sum the weighted scores of the clusters that the patient belongs to as shown in the following equation:

$$PatScore = \sum_{i=1}^n ws_i \quad (6)$$

For a new admitted patient, we can calculate the score of the patient (average number of anticipated following readmissions) by adding up all the weighted scores of all the clusters that he or she belongs to. For example, let us assume that the set of diagnoses for a new patient is the following: {3, 6, 9, 11}. Let us also assume that the generated clusters from our dataset are shown in Table 14. We can observe that the new patient will belong to Cluster 2, Cluster 4, and Cluster 6. To calculate the score of the new patient we start by calculating the sum of entropies for Clusters 2, 4, and 6:

$$e(c_2) + e(c_4) + e(c_6) = (2.6) + (1.7) + (2.1) = 6.4$$

Next, we calculate the weighted score for Clusters 2, 4, and 6:

$$ws(c_2) = \frac{2.6}{6.4} * 1.5 = 0.61$$

Table 14: A hypothetical example to demonstrate the calculation of the new patient's score

Name	Included Set	Excluded Set	Score	Entropy
Cluster 1	{11}	{-3, -9}	1.7	1.8
Cluster 2	{3, 9}	{-5}	1.5	2.6
Cluster 3	{}	{-3, -5, -9}	0.7	2.9
Cluster 4	{6, 3}	{-2}	0.9	1.7
Cluster 5	{6, 11}	{-9}	2.0	0.9
Cluster 6	{3, 11}	{-5}	1.0	2.1
Cluster 7	{6}	{-3, -10}	1.17	3.1
Cluster 8	{6, 9, 11}	{-3, -5}	0.5	1.3

$$ws(c_4) = \frac{1.7}{6.4} * 0.9 = 0.24$$

$$ws(c_6) = \frac{2.1}{6.4} * 1.0 = 0.33$$

Finally, we calculate the score of the new admitted patient by adding all the weighted scores for the three clusters:

$$\text{patientScore} = 0.61 + 0.24 + 0.33 = 1.18$$

The patient score value is meant to serve as the most accurate prediction for the average number of following readmissions for the new admitted patient.

8.2 Extracting Recommendations Algorithm

In the previous section, we devised an approach to calculate the score of a patient that belongs to multiple clusters. The next step is to identify clusters with scores that are better than the patient's score (we refer to these clusters by the term *desired clusters*). For all the desired clusters, we identify the required medical recommendations and apply them accordingly. It is often the case however, as we will demonstrate in the following example, that after applying the medical recommendations, the new patient will end up belonging to a new set of clusters rather than only the targeted desired cluster. Although we know that the targeted cluster has a lower (or better) score than the patient's score, the other clusters that the patient will belong to after applying the medical recommendations may, or may not, be desired (exhibit better scores than the patient's original score). The outline of the entire algorithm is shown in Algorithm 1.

Referring back to our example from Table 14, after calculating the patient's score, the next step would be to identify the desired clusters (clusters with score lower than the patient's score). Assuming we are still examining the patient that exhibits the following set of diagnoses: $\{3, 6, 9, 11\}$, whose score was calculated in the previous section to be 1.18, the set of desired clusters that have scores lower than 1.18 are Clusters 3, 7, and 8. Note that we did not select Cluster 4 and Cluster 6, since the patient already belongs to these clusters, although their scores are less than the patient's score. After identifying the clusters that exhibit scores lower than the patient's score, we identify the required medical recommendations that are associated with each one

1. Rough Clustering: extract the clusters according to the initial procedure that the patient will undergo and calculate their scores.
 2. Identify the clusters that the new patient belongs to according to the patient's diagnostic codes.
 3. Filter the clusters using the *percentage filtering* and *comorbidity filtering*.
 4. Calculate the score for the new patient using the entropy-based weighted score method (PatientScoreBefore).
 5. Identify the clusters that have scores better (less) than the patient's score (we refer to these clusters as *desired clusters*) in compliance with the following restrictions.
 - A cluster must have at least one of the patient's comorbidities.
 - A cluster must not have any other comorbidities.
 6. **for** *each desired cluster (dc)* **do**
 - (a) Identify the diagnoses recommendations (modifications).
 - Recommendations must not affect the comorbid conditions.
 - (b) Identify the clusters that the patient will belong to after following the recommendations.
 - (c) Calculate the score for the patient using the entropy-based weighted score method (PatientScoreAfter).
- if** ($PatientScoreAfter \leq PatientScoreBefore$) **then**
- Save recommendations.
 - Calculate score reduction.
- else**
- | Discard the recommendations.
- end**
- end**

Algorithm 1: Recommendation extraction algorithm

of the desired clusters. For each set of recommendations, we further identify the new clusters that the patient will belong to after applying the recommendations, and we calculate the entropy-based weighted score to the new patient. A complete outline for this approach is presented next:

Let us continue our example by examining the medical recommendations that will allow the new patient, described by the set of diagnoses: $\{3, 6, 9, 11\}$, to be similar to the patients in Cluster 3 (the most desired clusters). We need to apply the following medical recommendations:

1. Treat diagnostic code 3
2. Treat diagnostic code 9
3. Make sure that the patient does not have diagnostic code 5

After applying the above medical recommendations, the patient can be described by the following set of diagnoses: $\{6, 11\}$. The next step would be to examine the other clusters and identify the clusters that the patient belongs to after applying the medical recommendations. By examining Table 14, we conclude that after applying the recommendations, the new patient now belongs to Clusters 1, 3, 5, and 7. To calculate the entropy-based weighted score we need to start by calculating the summation of the entropies for Clusters 1, 3, 5, and 7:

$$e(c_1) + e(c_3) + e(c_5) + e(c_7) = (1.8) + (2.9) + (0.9) + (3.1) = 8.7$$

Next, we calculate the weighted score for Clusters 1, 3, 5, and 7:

$$\text{ws}(c_1) = \frac{1.8}{8.7} * 1.7 = 0.35$$

$$\text{ws}(c_3) = \frac{2.9}{8.7} * 0.7 = 0.23$$

$$\text{ws}(c_5) = \frac{0.9}{8.7} * 2.0 = 0.21$$

$$\text{ws}(c_7) = \frac{3.1}{8.7} * 1.17 = 0.42$$

Finally, we calculate the score of the patient after applying the recommendations by adding all the weighted scores for the three clusters:

$$\text{patientScoreAfter} = 0.35 + 0.23 + 0.21 + 0.42 = 1.21$$

Since the score of the patient after applying the medical recommendations (1.21) is greater than the score of the patient before applying the recommendations (1.18), we discard the set of the medical recommendations. Recall that the score is a measure of the average number of following readmission, the lower the value the better the score.

We now examine the second desired cluster (Cluster 8) and identify the required medical recommendations. To allow the new patient to be similar to the patients in

Cluster 8, we need to apply the following set of medical recommendations:

1. Treat diagnostic code 3
2. Make sure that the patient does not have diagnostic code 5

After applying the above medical recommendations, the patient can be described by the following set of diagnoses: {6, 9, 11}. The next step would be to examine the other clusters and identify the clusters that the patient belongs to after applying the medical recommendations. By examining Table 14, we can conclude that after applying the medical recommendations, the new patient now belongs to Clusters 7 and 8. To calculate the entropy-based weighted score, we need to start by calculating the entropy for Clusters 7 and 8:

$$\text{eg}(c_7) + \text{eg}(c_8) = (3.1) + (1.3) = 4.4$$

Next, we calculate the weighted score for Clusters 7 and 8:

$$\text{ws}(c_7) = \frac{3.1}{4.4} * 1.17 = 0.82$$

$$\text{ws}(c_8) = \frac{1.3}{4.4} * 0.5 = 0.15$$

Finally, we calculate the score of the patient after applying the medical recommendations by adding all the weighted scores for the two clusters:

$$\text{patientScoreAfter} = 0.82 + 0.15 = 0.97$$

Notice here that the score of the patient after applying the medical recommendations (0.97) is less than the score of the patient before applying the medical recommendations (1.18). Therefore we keep the set of the medical recommendations as a valid option for reducing the number of readmissions. Our algorithm should continue looping through all desired clusters until we extract all valid medical recommendations. Next, we will show some results through applying our approach on the Florida State Inpatient Databases (SID).

8.3 Results from the H-CUP dataset

In this section, we provide some of the results obtained after extracting clusters of patients from the SID dataset, followed by calculating their scores according to the algorithms presented in the previous section. After calculating the scores for the extracted clusters, any clinically feasible recommendations would essentially be considered a valid set of actions that will help in reducing the average number of following readmissions.

Next, we show a sample of few 3-element clusters extracted from patients who undertook procedure 105 (kidney transplant) as their first procedure, using the filtering range of 5% to 95%, as shown in Table 15.

Here is a description of the diagnostic codes shown in Table 15:

- 3: Bacterial infection
- 48: Thyroid disorders

Table 15: A sample of three clusters with their scores and their included and excluded sets for procedure 105 (kidney transplant)

Cluster Name	Number of Patients	Included Set	Excluded Set	Score
C 1	18	{106}	{-53, -156}	0.1136
C 2	49	{157}	{-3, -48}	0.5397
C 3	18	{53, 106}	{-3}	0.6149

- 53: Disorders of lipid metabolism
- 106: Cardiac dysrhythmias
- 156: Nephritis, nephrosis, renal sclerosis
- 157: Acute and unspecified renal failure

According to all extracted clusters, Cluster 1 (which has a score of 0.1136) is the best cluster for procedure 105 (amongst all 59430 extracted clusters). This means that if a new patient were to belong to Cluster 1, then there is no medical recommendations that could reduce the anticipated number of readmissions for that patient. However, if a new patient belongs to any other cluster, then there is at least one set of medical recommendations that would reduce the anticipated number of following readmissions.

By examining Table 15, we can infer that by applying medical recommendations to the patients in Cluster 3 to make them similar to the patients in Cluster 1 (given that patients only belong to Cluster 3 prior to the recommendation and will only belong to Cluster 1 after applying the recommendations) will reduce the number of following readmissions on average by 81.5%. This means that by treating the diagnostic code 53

Table 16: A summary of the number of clusters that patients belong to on average and the average number of desired clusters, for sample of five procedures

Procedure Code	Clusters Count	Avg Patient Clusters	Avg Desired Clusters
44	5892	2370	776
58	1441	523	178
75	9203	4096	1094
78	4218	2214	460
153	1543	705	160

and making sure that the patients do not have diagnostic code 156 before performing procedure 105, we would decrease the number of following readmissions by 81.5 for the 100 patients in Cluster 3.

As explained earlier in this chapter, it is rather common for a new patient to belong to multiple clusters. Table 16 shows a summary of the number of generated clusters, the number of clusters new patients will belong to on average, and the average number of desired clusters (clusters that have better score), for a sample of five procedures. These numbers are calculated over 100 patients who undertook one of the procedures mentioned in the table. It is worth mentioning here that the number of clusters would be less in case the ICD-9-CM coding was used instead, since it is detailed and uses more codes to label the procedures and diagnoses as explained in Chapter 3.

Here is a description of the procedures codes shown in Table 16:

- 44: Coronary artery bypass graft (CABG)
- 58: Hemodialysis
- 75: Small bowel resection

- 78: Colorectal resection
- 153: Hip replacement total and partial

Now, we will show few examples of what actually happens for a sample patient who is going to undergo procedure code 4399 (total gastrectomy). The set of diagnostic codes for our sample patient is shown in Table 17, and the results are shown in Table 18, followed by an explanation. The weighted score of the sample patient, shown in Table 18, prior to following the recommendations is 0.367. This number represents the average number of readmissions for patients similar to our sample patient. Here is a description of the procedure codes not shown in Table 17:

- 389: Septicemia
- 25000: Diabetes mellitus type 2
- 2724: Hyperlipidemia
- 1519: Malignant neoplasm of stomach
- 41401: Coronary atherosclerosis native vessel
- 496: Chronic airway obstruction

By examining Table 18, we can notice that Cluster C2 has the lowest score (the most desired cluster). The recommendations that our system will provide to the physician, so the patient can end up with the same outcomes as of Cluster C2, would be to treat diagnostic code 5180, make sure that the patient does not have diagnostic

Table 17: The set of the diagnostic codes and their description for our sample patient

Diagnostic Code	Description
1510	Malignant neoplasm of stomach cardia
5180	Pulmonary collapse
V1005	History of colonic malignancy
4019	Hypertension
V1006	History of rectal and anal malignancy
V1582	History of tobacco use

Table 18: A sample of recommendations based on our multi-cluster entropy-based algorithm. Score of patient prior to following any of the recommendations is 0.367

Cluster Name	Score of Targeted Cluster	Entropy of Targeted Cluster	Action Rules	Reduction
C1	0.175	3.32	[] \rightarrow -389 [] \rightarrow -25000 [] \rightarrow -2724 4019 \rightarrow 4019 5180 \rightarrow -5180	10.99%
C2	0.111	3.17	[] \rightarrow -389 [] \rightarrow -25000 [] \rightarrow -1519 4019 \rightarrow 4019 5180 \rightarrow -5180	2.06%
C3	0.21	2.52	[] \rightarrow -389 [] \rightarrow -1519 [] \rightarrow 41401 4019 \rightarrow 4019 [] \rightarrow -496	-9.141E-4%

codes 389, 25000, and 1519, and keep diagnostic code 4019 (as it is a co-morbid condition). Following these recommendations will result, on average, in a (2.06%) readmission reduction. Although the score of Cluster C1 is higher (less desired) than the score of Cluster C2, the set of recommendations associated with Cluster C1 resulted in a better readmission reduction (10.99 % reduction). This can happen due to the fact that the overall weighted scores of the set of clusters that the patient belongs to after following the recommendations that are associated with Cluster C1 is better than the set of recommendations that are associated with Cluster C2. Another observation from the table is that the score of the targeted desired cluster in the last row (Cluster C3) is less (more desired) than the patient's score before following the recommendations. However, the patient ended up with a negative reduction percentage. This is because after following the recommendations, the patient may belong to other clusters that are undesired, and the resulting entropy-based weighted score will end up being higher (less desired) than the initial score of the patient prior following the recommendations. Finally, the repetition of some recommendations among different clusters emphasizes the importance of these recommendations. For example, the recommendation (5180 \rightarrow -5180) is repeated in the first two clusters, which means that it is very important to treat the diagnostic code 5180 (pulmonary collapse) before performing the specified procedure.

CHAPTER 9: PRIMARY MEDICAL PROCEDURE PREDICTION AND DISCOVERY OF ITS ASSOCIATED PROCEDURES

One of the reasons for hospital readmissions is the wrong diagnosis of the patients. It is very important to provide the patients with the proper diagnosis in order to avoid any future readmissions and reduce the healthcare spending. The selection of the primary medical procedure by physicians is the first step in the patient treatment process. In the previous chapters, we introduced a system for physicians that may recommend additional treatment prior to the scheduled procedure which would, as a result, yield to reduction in the number of anticipated hospital readmissions. The input for our system is the set of diagnoses of a newly admitted patient, and the primary medical procedure assigned by the physician for that patient. However, in this chapter, we focus on predicting the primary medical procedure for the new patient by clustering the patients according to their set of diagnoses. This procedure might differ from the primary medical procedure assigned by a physician. We propose three new approaches to identify the patients, from the dataset, that are similar to the newly admitted patient. Additionally, we identify the procedures that are associated with the predicted primary procedure through personalizing the patients according to their common medical characteristics. Our goal is not to build a system which is replacing physicians but a system which provides them with new insights discovered from H-CUP data.

9.1 Predicting the Primary Procedure

In this section, we examine three approaches that address the problem of identifying the right primary procedure for a patient, given his or her set of diagnoses. The goal of our system, which is to accurately predict the primary procedure for a newly admitted patient, is almost wholly determined by its ability to identify other existing patients that are considered similar to our admitted patient. The basis for determining similarities between different patients however, which we will explore next, is an intricate endeavor, given that the input representing our patients is a set of diagnoses that differ greatly in the level of significance.

9.1.1 Minimum Similarity Match

The first approach that we propose to predict the primary procedure, is to have our similarity function be defined in a way that marks a newly admitted patient (p_n) similar to an existing patient (p_e) if and only if the existing patient exhibits every single diagnoses present in the admitted patient:

$$\text{similarity}(p_n, p_e) = \begin{cases} 1, & \text{if } \text{diag}(p_n) \subset \text{diag}(p_e) \\ 0, & \text{otherwise} \end{cases} \quad (7)$$

where 1 indicates that the new patient (p_n) is similar to the existing patient (p_e), and 0 otherwise. Consequently, we can define A_n as the set of all existing patients that the new patient (p_n) is similar to:

Table 19: Dataset S, containing all existing patients

Patient	Diagnoses	Primary Procedure
p_1	{d1, d2, d5, d8}	Procedure 6
p_2	{d1, d2, d3, d5}	Procedure 3
p_3	{d1, d3, d4, d5, d9}	Procedure 6
p_4	{d1, d2, d3, d6}	Procedure 2
p_5	{d1, d3, d4, d5, d8}	Procedure 3
p_6	{d1, d2, d3, d4, d5}	Procedure 2
p_7	{d1, d3, d5, d6, d7}	Procedure 3

$$A_n = \bigcup_{i=1}^m \{p_i : \text{similarity}(p_n, p_i) = 1\} \quad (8)$$

where m is the number of existing patients in our dataset who are similar to p_n , and p_i is the i^{th} existing patient in that set. The final output for our recommender system is a probability distribution for the primary procedures obtained by the set of all similar existing patients (A_n). To demonstrate with an example, say we have a newly admitted patient (p_n) with the following set of diagnoses: $\text{diag}(p_n) = \{\text{d1, d3, d5}\}$, let us also assume that our dataset consists of the set of seven patients shown in Table 19.

Looking at our dataset of patients in Table 19, we can conclude the following:

- $\text{similarity}(p_n, p_i) = 0$, for $i = 1$ and 4
- $\text{similarity}(p_n, p_i) = 1$, for $i = 2, 3, 5, 6$, and 7

According to our previous definitions, A_n will contain the set of elements: p_2, p_3, p_5, p_6 , and p_7 and the primary procedures for these patients are 3, 6, 3, 2, and 3 respectively. Therefore, the output to our recommender system will be 60% Procedure

Table 20: Prediction accuracy of the minimum similarity match using the N most probable primary procedures

N	Frequency	Accuracy
1	152	18.5%
2	192	23.6%
3	216	26.5%
4	233	28.6%
5	243	29.8%
6	250	30.7%
7	260	31.9%
8	265	32.5%
9	268	32.9%
10	270	33.1%

3, 20% Procedure 6, and 20% Procedure 2, which is the probability distribution of the primary procedures of A_n .

Table 20 shows a list of the accuracies for our system when tested on 815 randomly selected instances, each being compared to roughly 4 millions existing patients using our definition of similarity presented earlier. As can be seen in Table 20, the procedure with the highest probability in the existing matches distribution was predicted correctly 18.5% of the time, the correct primary procedure was predicted correctly as one of the two procedures 23.6% of the time, and the correct primary procedure was predicted correctly as one of the three procedures 26.5% of the time. The frequency is the number of instances, out of the 815, for which the primary procedure was predicted correctly.

Although the approach presented in this section is showing reasonably good results, the fact that our definition of similarities requires an existing patient (p_e) to exhibit all diagnoses of the new patient (p_n) makes this system rather limited. Therefore, we

need to apply a more flexible system for identifying similarities that can increase the number of patients that are similar to the newly admitted patient.

9.1.2 Jaccard Similarity Match

According to our dataset, a patient has on average 7.55 diagnoses on admission. In addition, there are 30.99%, 10.67%, and 2.68% of patients having 10 diagnoses or more, 15 diagnoses or more, and 20 diagnoses or more, respectively. Therefore, there is a high probability that a newly admitted patient will exhibit a large number of diagnoses, which would make it hard to identify similar patients in our dataset using the algorithm presented in Section 9.1.1. One way to tackle this limitation is to modify the definition of the similarity between patients in a way that when a new patient gets admitted to the hospital, rather than finding patients who have all the diagnoses that the new patient has, we measure the percentage of diagnoses that are common between the new patient and the existing patients in the dataset. This can be achieved by applying Jaccard similarity index, which measures the similarity between two sets by dividing the cardinality of the intersection by the cardinality of the union. There are several similarity functions in the literature, however, we selected Jaccard similarity function as it meets our requirements of finding the intersections between the patients diagnoses and it is a logical progression of the Minimum Similarity Match explained in Section 9.1.1. Jaccard similarity is described in the following equation:

$$J(A, B) = \frac{|A \cap B|}{|A \cup B|} \quad (9)$$

where A and B represent the two sets of diagnoses, and where $|\cdot|$ represents the

cardinality of the intersection and union of the two sets.

The elements of the sets are the diagnoses of the patients. One of the sets contains the new patient's diagnoses and the other set contains the existing patient's diagnoses. In our dataset, we calculate Jaccard similarity of the new patients with every patient in the dataset. The patient with the highest Jaccard index is considered the most similar to the new patient. Now that we have identified the patient that is most similar to our newly admitted patient, we can consequently use the existing patient's main procedure to predict the main procedure for our new patient. However, doing so is rather unfavorable, since building an entire prediction system based upon one patient, may lead to a bias in our results. Therefore, it would be more robust to instead consider all patients that satisfy a similarity value within a given margin (e.g. x top most similar patients). In our implementation, we have tested using different margins and found that 5 percent would yield the best results. For example, suppose we have a dataset of 40 patients and the similarities of these patients with the newly admitted patient fall in this range [40%-90%]. Given a margin of 5%, we would select the top 2 similarities out of the 40 similarities, which means that we will select the primary procedures of the patients with the two highest similarities.

The best way to explain how this algorithm works is by providing an example. Suppose a new patient pat_0 comes to the hospital with a certain number of diagnoses $\{d_1, d_3, d_5\}$. Also, suppose that our dataset contains 10,000 patients as listed in Table 21. The first step would be to find the intersection and union of pat_0 with all the patients in the dataset, then calculate the Jaccard index. For example, the similarity index of pat_0 and pat_1 is calculated as follows:

$$J(pat_0, pat_1) = \frac{|pat_0 \cap pat_1|}{|pat_0 \cup pat_1|} = \frac{|\{d_1, d_5\}|}{|\{d_1, d_2, d_3, d_5, d_8\}|} = 2/5 = 0.4 \quad (10)$$

Let us make the following assumptions before going through the details:

- The range of similarities is between [40-75] %.
- The patients [$pat_5 - pat_{9998}$] have similarities less than 50% with pat_0 .
- The primary procedures are found in advance for each patient.

As seen in Table 21, the range of the similarities is [40-75]. Given this range, we would select the highest 5% similarities and store the primary procedures associated with them. In our implementation, we have tested using different margins on the training and testing samples and found that 5 percent would yield the best results. Exceeding this margin would result in deviation from the correct prediction. After finding the highest 5% similarities, we find the average similarity for each procedure. Assume now that the patients who fall in the highest 5% similarities are only pat_2 , pat_3 , and pat_4 . As can be noticed from the table, the primary procedures that are associated with these patients are p_6 and p_3 . The next step would be to calculate the average similarity index for these procedures. The average similarity index for p_6 would be the average of 75% and 60%, which is equal to 67.5% and the average similarity index of p_3 would be 75%, given our assumption that pat_4 is the only patient who exhibits p_3 and also lies within our specified similarity margin (highest 5%). Finally, our prediction will be the procedure with the highest average (weight). We have run Jaccard similarity algorithm on a dataset of 10,000 patients who under-

Table 21: Jaccard similarity calculations example

Patient	Diagnoses	Patient's primary procedure	Similarity index
pat_1	$\{d_1, d_2, d_5, d_8\}$	p_6	40%
pat_2	$\{d_1, d_2, d_3, d_5\}$	p_6	75%
pat_3	$\{d_1, d_3, d_4, d_5, d_9\}$	p_6	60%
pat_4	$\{d_1, d_3, d_4, d_5\}$	p_3	75%
...
pat_{9999}	$\{d_1, d_2, d_3, d_6\}$	p_2	40%
pat_{1000}	$\{d_1, d_3, d_4\}$	p_3	50%

went different procedures and measured the accuracy using 10-fold cross validation.

The resulted accuracy was 20.25%, which is slightly better than the accuracy of the Minimum Similarity Match (MSM) method used in Section 9.1.1.

The approaches presented in Section 9.1.1 and 9.1.2 are showing reasonably good results in predicting the primary procedure. In these approaches however, we based our definition of the patients' similarity on the number of diagnoses that the patients exhibit. However, we should shift our focus to the level of importance of each diagnoses with respect to their abilities to predict the primary procedure. There is typically only a small number of subsets that are capable of determining the primary procedure. In the next subsection, we present a new and novel approach on how to identify such sets.

9.1.3 Selective Similarity Match

In this subsection, we introduce an enhanced system for predicting the primary procedure for new patients. Our approach presented here is based on the fact that there is only a selected number of combinations for diagnoses subsets that are capable of predicting primary procedures. This means that for a new patient exhibiting x

number of diagnoses, it would be more likely the case that matching our dataset for patients that exhibit only a subset of the x diagnoses will yield better results, by doing so, our system will not only avoid over-fitting, but it will also result in many more matches in our existing dataset and the same in a higher level of prediction accuracy. The level of predictability for a subset of diagnoses s , can be determined based on the distribution of the primary procedures for existing patients that exhibit s . By calculating the entropy of the main procedures for each possible subset of the diagnoses, we are able to identify subsets that can most accurately predict the primary procedure (subsets that have the least entropy values).

Our system starts by generating all possible combinations of k -diagnosis sets, starting with $k=1$ and ending with $k=3$, then calculating the entropy of the primary procedures for each combination. For each combination of diagnoses s , we identify all existing patients that belong to s , then we calculate the entropy $H(s)$ according to the distribution of the primary procedures for s :

$$H(s) = - \sum_{i=1}^m p_i \log(p_i) \quad (11)$$

where p_i is the probability of the i^{th} primary procedure, and m is the number of primary procedures in s .

The reason for why we stop at the number 3 is because the number of distinct subsets that can be generated from the set of all 285 diagnoses grows exponentially large as k increases. For example, the number of unique 3-diagnoses subsets that can be chosen from 285 diagnoses is roughly 4 millions, the number of unique 4-diagnoses subsets

however, exceeds 250 millions.

For a new admitted patient with x number of diagnoses, we generate all subsets of k -diagnoses for $k = 1, 2$, and 3 , then, using our previously calculated entropies for all possible diagnoses, we identify the subset of the patient diagnoses with the lowest entropy (highest level of predictability), and use its most frequent procedure as the anticipated primary procedure. We have run the Selective Similarity Match (SSM) algorithm on a dataset of 10,000 patients and measured the accuracy using 10-fold cross validation. The resulted accuracy was 25.25%, which is better than both the accuracies of the Minimum Similarity Match (MSM) and Jaccard Similarity Match approaches used in Section 3.1 and Section 3.2 respectively.

Next, we provide a real example from our dataset to demonstrate the algorithm.

Let us first assume that the first step of the algorithm, which is to generate all possible combinations of k -diagnosis sets, starting with $k=1$ and ending with $k=3$ has been performed. Now, say that a new patient (p_n) has been admitted to the hospital with the following set of diagnoses $\{181, 183, 101, 164\}$:

- 181: Other complications of pregnancy.
- 183: Hypertension complicating pregnancy, childbirth and the puerperium.
- 184: Early or threatened labor.
- 189: Previous C-section.

The next step would be to generate all 1-diagnosis, 2-diagnoses, and 3-diagnoses subsets of (p_n), which is shown in the first column of Table 22.

Table 22: An example of one of the tested patients

List of Diagnoses in Cluster	Entropy	Primary Procedure
181	2.414	137
183	2.258	137
184	1.564	137
189	1.293	134
181, 183	2.419	137
181, 184	1.783	137
181, 189	1.224	134
183, 184	1.241	134
183, 189	0.622	134
184, 189	0.884	134
181, 183, 184	-	-
181, 183, 189	0.337	134
181, 184, 189	1.095	134
183, 184, 189	-	-

According to Table 22, the list of diagnoses that has the least entropy is {181, 183, 189}, in which the most probable primary procedure is 134 (Cesarean section), which is indeed the correct primary procedure for our patient (p_n). Following is a description of the procedure codes found in Table 22:

- 134: Cesarean section.
- 137: Other procedures to assist delivery.

Table 7 shows few procedures with their prediction accuracy for a testing sample of 1,000 instances, using a training set of size 10,000 instances. For example, the third row in our table states that we encountered 10 instances (out of our 1,000 testing sample) with main procedure 'Spinal Fusion', and that we were able to predict this procedure with accuracy 60%, meaning that we were able to correctly predict that the main procedure is *Spinal Fusion*, for 6 instances out of 10.

Table 23: Sample of main procedures with their frequencies and accuracies for a testing sample of 1,000 instances, using a training set of size 10,000 instances

Procedure	Frequency	Accuracy
137 (Procedures to Assist Delivery)	23	65%
84 (Cholecystectomy and Common Duct Exploration)	21	62%
158 (Spinal Fusion)	10	60%
152 (Arthroplasty Knee)	14	57%
134 (Cesarean Section)	38	45%
61 (Other OR Procedures on Vessels Other than Head and Neck)	18	44%
45 (Percutaneous Transluminal Coronary Angioplasty PTCA)	45	42%
78 (Colorectal Resection)	32	39%
124 (Hysterectomy Abdominal and Vaginal)	19	32%
70 (Upper Gastrointestinal Endoscopy Biopsy)	148	31%
47(Diagnostic Cardiac Catheterization Coronary Arteriography)	27	30%

9.2 Procedures Associations

In Chapter 5, we introduced the procedure graph, which shows the different sequence of procedure paths that a patient may undertake during the course of treatment. Each procedure path includes a set of primary procedures from different visits. For example, if there exist in our medical dataset a patient or a set of patients who underwent the following set of primary procedures $\{P1, P2, P3\}$ in the first, second, and third hospital visits respectively, then this set of procedures will be represented as a path in the procedure graph. In this section, we will have a deeper look into the procedures that are applied in the same hospital visit. As mentioned in Chapter 3, a patient may have up to 31 procedures in each visit. The goal in this section is to show the associations between these procedures with respect to the primary procedure. That said, given the primary procedure, we would like to predict what are the highly associated procedures to that procedure.

9.2.1 Matrix Representation of Procedures

In order to find the correlation between the primary procedure and the other procedures that might coexist with it in the same hospital visit, we apply a conditional probabilistic function and use matrix representation for all procedures that may coexist with the primary procedure. For example, if we are interested in finding the associated procedures with the primary procedure $p=(105: \text{kidney transplant})$, then we need to construct a dataset $Sub_Dataset(p)$, from the H-CUP data, that only includes the entries where the primary procedure is 105. First, we build Boolean matrix $B = \{B_{i,j} : i \leq m \ \& \ j \leq n\}$ which is used to identify all procedures that coexist with a primary procedure p . Integer n , where $(n \leq 31)$, refers to the total number of different procedures which coexist with p and m refers to the number of patients having p as the primary procedure. The equation below is used to represent and calculate $Sub_Dataset(p)$:

$$Sub_Dataset(p) = \begin{bmatrix} B_{11} & B_{12} & B_{13} & \dots & B_{1n} \\ B_{21} & B_{22} & B_{23} & \dots & B_{2n} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ B_{m1} & B_{n2} & B_{n3} & \dots & B_{mn} \end{bmatrix} * \begin{bmatrix} P_1 \\ P_2 \\ \vdots \\ P_n \end{bmatrix} \quad (12)$$

where B_{ij} is set to 1 when the procedure is applied in the patient's visit. P_i refers to the medical procedure. Each patient's record, denoted by $Patient_i$, is represented as the set of procedures that are applied in his/her visit as shown below:

$$Patient_i = \{P_j : B_{ij} = 1 \ \& \ 1 \leq j \leq n \leq 31\} \quad (13)$$

The reason for using this matrix representation is to make it faster for our computations when we find the associations through applying the probabilistic model. In order to find the correlation between the primary procedure and all the possible procedures that may coexist with it, we use a probabilistic model that uses at its core the conditional probability theory. The most associated procedure with the primary procedure P_r is found using the following equation:

$$Prob_x = max\{X_i : X_i = Prob(P_i|P_r)\} \quad (14)$$

where $Prob_x$ represents the probability of the most associated procedure (let us call it P_x) with the primary procedure P_r . X_i represents the association percentage (or the probability of association) between procedure P_i and the primary procedure P_r . Prob is the conditional probability function.

To find the most probable set of procedures that are associated with the primary procedure, we need to generalize Equation 14 to multiple events. To find two the most associated procedures $[P_x, P_y]$ with the primary procedure P_r , first we find the most associated procedure P_x with the primary procedure P_r and next the most associated procedure P_y with the procedures P_r and P_x . The formula below shows how to find $Prob_y$, which is the probability of the most associated procedure with the set of procedures (P_r, P_x) :

$$Prob_y = \max\{Y_i : Y_i = Prob(P_i|(P_r, P_x))\} \quad (15)$$

The generalization of the last equation to three or more procedures is quite straightforward.

Table 24 shows the most associated procedures of length 1 and 2 with the primary procedure 105 (kidney transplant). As can be noticed from the table, procedure 231 (Other therapeutic procedures) is the most associated procedure with a 98.2% of the cases. In addition to that, the most probable association pattern of length 2 is [231, 102 (Ureteral catheterization)] with a 31.9%. These associations can serve as recommendations to the physicians, especially if these associations come with high correlation percentage, such as procedure 231. On the other hand, knowing that there is no association with the primary procedure is also beneficial to the physicians as they can avoid applying such procedures on the patients. It is worth mentioning here that the association between the primary procedure (105) and another primary procedure is most likely to be zero, as in the case with procedure (106: Genitourinary incontinence procedures). This might look counterintuitive, as these two procedures are strongly related to the kidney disease. However, in our dataset, the first recorded procedure is considered the primary procedure and the other procedures are considered to be normal (or minor). Therefore, there will be only one primary procedure in each visit and it is unlikely to apply two primary procedures during the same visit.

Following are the descriptions of the procedure codes mentioned in Table 24:

Table 24: The most common procedures associated with procedure 105 (Kidney transplant)

One Procedure		Two Procedures	
Procedure	Association	Procedures	Association
231	98.2	231-102	31.9
102	31.9	231-222	24.9
222	24.8	231-58	19.4
58	19.6	231-112	13.9
112	14.1	231-54	12.9
54	13.1	231-99	8.9
99	9.0	231-98	8.7
98	8.5	231-110	6.9
110	7.1	231-61	6.3
106	0	231-106	0

231: Other therapeutic procedures

102: Ureteral catheterization

222: Blood transfusion

58: Hemodialysis

112: Other vascular catheterization not heart

54: Other OR therapeutic procedures of urinary tract

99: Other OR gastrointestinal therapeutic procedures

98: Other nonOR gastrointestinal therapeutic procedures

110: Other diagnostic procedures of urinary tract

61: Other OR procedures on vessels other than head and neck

106: Genitourinary incontinence procedures

9.2.2 Personalized Procedures Associations

In the previous section, we demonstrated how to find the associations between the primary procedure and the other possible procedures. The percentages shown in Table

24 are on the most general level, which means that the associations are found with respect to all the patients in our dataset who have procedure 105 (Kidney transplant) as their primary procedure. Therefore, these associations are not necessary true for all patients.

By analyzing Table 24 again, we can notice that the probability of association decreases as we move from one to two associated procedures with a primary procedure. In addition to that, although there could be a procedure with a small association percentage, but, this percentage could be higher for a small group of patients. If a new patient belongs to such group, then it would make more sense to recommend this procedure as one of the highly associated procedures. In order to overcome this problem and make our recommendations more precise, we need to cluster the patients into groups of patients that exhibit some common medical characteristics. For this purpose, we will use the *Rough Clustering* explained in Section 6.2. Following is the outline of our developed algorithm:

Now, let us demonstrate with a real example from our dataset to show how the algorithm works and present some results of the personalized associations. The following diagnoses are for one of the patients found in our dataset who had procedure 105 (Kidney transplant) as his/her primary procedure:

156: Nephritis, nephrosis, renal sclerosis

158: Chronic kidney disease

210: Systemic lupus erythematosus and connective tissue disorders

1. Identify all the diagnoses that are associated with the primary procedure
 2. Define the *Included* and *Excluded* sets from the diagnoses found in step 1.
 3. Form the clusters according to the *Rough Clustering* algorithm.
 4. Filter the clusters using the *percentage filtering* and *co – morbidity filtering*.
 5. Identify the clusters that the new patient belongs to according to the patient’s diagnostic codes in compliance with the following restriction:
 - A cluster must have at least one of the patient’s co-morbidities.
 6. **for** *each procedure* **do**
 - (a) Calculate the association with the primary procedure.
 - (b) **if** ($Association_Percentage \geq Threshold$) **then**
 - | Save recommendation.
 - else**
 - | Discard recommendation
 - end**
- end**

Algorithm 2: Personalized associated procedure extraction algorithm

Table 25: The most common procedures associated with procedure 105 (Kidney transplant)

One Procedure		Two procedures	
Procedure	Association %	Procedures	Association %
231	98.7	231-102	33.2
102	33.1	231-222	32.2
222	31.9	231-58	23.9
58	24.1	231-54	15.3
54	15.8	231-112	15.2
112	15.5	231-99	9.6
99	9.4	231-98	8.6
98	8.5	231-216	7.3
61	7.7	231-110	7.1

Table 25 shows the associated procedures for the clusters that the patient belongs to. By comparing Table 25 with Table 24, we notice an improvement in the association percentage, which highlights the importance of the patients' personalization as a way to find the procedures' associations. It is worth mentioning here that the numbers in Table 25 may vary depending on the patient's diagnoses and the clusters that he/she belongs to. In addition to that, the frequency of the primary procedure has an effect on the associations. High frequent primary procedures provide more insightful associations and less outliers. On the other hand, less frequent primary procedures may result in unreliable associations.

CHAPTER 10: CONCLUSION

In the recent years, healthcare spending has risen and become a burden on governments especially in US. One of the reasons for this increase is hospital readmissions, which is defined as a re-hospitalization that may happen for a patient after being discharged from a hospital within a short period of time, usually 30 days. Decreasing the number of readmissions can improve the healthcare quality and reduce the healthcare spending. In this dissertation, we presented novel algorithms to reduce the average number of readmissions by applying the concept of personalization and actionable patterns. We surveyed state-of-the-art applications that are concerned in reducing the number of hospital readmissions. These applications are categorized according to the prediction that they are targeting: diagnosis prediction, risk prediction, and after procedure prediction.

This dissertation is divided into three main sections: First, we predicted the risk of mortality and the risk of readmission by applying several supervised machine learning algorithms. Second, which is the core of this research, we built the procedure graph that shows all the possible procedure paths that a new patient may undertake during the course of treatment. Furthermore, we proposed a novel algorithm to cluster (personalize) patients based on their diagnoses. Moreover, we devised scoring functions to evaluate the procedures in the procedure graph and evaluate the clusters. Finally, we proposed an algorithm to devise the medical recommendations (actionable knowl-

edge) that can help the physicians to eventually minimize the anticipated number of hospital readmissions. Third, we proposed a system to predict the primary medical procedure according to the similarities of the new patient with the other patients. We further found the procedures that are highly correlated with the primary medical procedure and provided them as recommendations to the physicians to enhance the final status of patients. The results presented in this dissertation show the ability of our system in reducing the number of hospital readmissions and enhancing the patients status.

REFERENCES

- [1] Overview Of The State Inpatient Databases (SID). <https://www.hcup-us.ahrq.gov/sidoverview.jsp>. Accessed: 2016-11-6.
- [2] Intel and cloudera use predictive analytics to help a large hospital group reduce readmission rates. Technical report, Intel/Cloudera, 2015.
- [3] S.-M. Chou, T.-S. Lee, Y. E. Shao, and I.-F. Chen. Mining the breast cancer pattern using artificial neural networks and multivariate adaptive regression splines. *Expert Systems with Applications*, 27(1):133–142, 2004.
- [4] P. Coopers. The Price of Excess. Identifying Waste in Healthcare Spending, 2006.
- [5] E. Demir. A decision support tool for predicting patients at risk of readmission: A comparison of classification trees, logistic regression, generalized additive models, and multivariate adaptive regression splines. *Decision Sciences*, 45(5):849–880, 2014.
- [6] A. Elixhauser, C. Steiner, D. R. Harris, and R. M. Coffey. Comorbidity measures for use with administrative data. *Medical care*, 36(1):8–27, 1998.
- [7] M. P. Fernandes, C. F. Silva, S. M. Vieira, and J. Sousa. Multimodeling for the prediction of patient readmissions in intensive care units. In *IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)*, pages 1837–1842, 2014.
- [8] D. Ferrucci, A. Levas, S. Bagchi, D. Gondek, and E. T. Mueller. Watson: Beyond jeopardy! *Artificial Intelligence*, 199:93 – 105, 2013.
- [9] Y. Freund and R. E. Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. In *European conference on computational learning theory*, pages 23–37. Springer, 1995.
- [10] L. Gorman. Priceless: Curing the Healthcare Crisis. *Business Economics*, 48(1):81–83, 2013.
- [11] A. Hajja, H. Touati, Z. W. Raś, J. Studnicki, and A. A. Wieczorkowska. Predicting negative side effects of surgeries through clustering. In *New Frontiers in Mining Complex Patterns*, pages 41–55. Springer, 2014.
- [12] J. P. Hilbert, S. Zasadil, D. J. Keyser, and P. B. Peele. Using decision trees to manage hospital readmission risk for acute myocardial infarction, heart failure, and pneumonia. *Applied health economics and health policy*, 12(6):573–585, 2014.
- [13] S. P. Keehan, G. A. Cuckler, A. M. Sisko, A. J. Madison, S. D. Smith, D. A. Stone, J. A. Poisal, C. J. Wolfe, and J. M. Lizonitz. National Health Expenditure Projections, 2014–24: Spending Growth Faster than Recent Trends. *Health Affairs*, 34(8):1407–1417, 2015.

- [14] A. Lally, S. Bachi, M. A. Barborak, D. W. Buchanan, J. Chu-Carroll, D. A. Ferrucci, M. R. Glass, A. Kalyanpur, E. T. Mueller, J. W. Murdock, et al. WatsonPaths: Scenario-based Question Answering and Inference over Unstructured Information. *Yorktown Heights: IBM Research*, 2014.
- [15] R. Liu, R. V. Srinivasan, K. Zolfaghar, S.-C. Chin, S. B. Roy, A. Hasan, and D. Hazel. Pathway-finder: An interactive recommender system for supporting personalized care pathways. In *IEEE International Conference on Data Mining Workshop (ICDMW)*, pages 1219–1222, 2014.
- [16] J. Natale and S. Wang. A decision tree model for predicting heart failure patient readmissions. In *IIE Annual Conference*, page 3518. Institute of Industrial Engineers-Publisher, 2013.
- [17] M. Panahiazar, V. Taslimitehrani, N. L. Pereira, and J. Pathak. Using ehrs for heart failure therapy recommendation using multidimensional patient similarity analytics. *Studies in health technology and informatics*, 210:369–373, 2014.
- [18] Z. Pawlak. Rough sets. *International Journal of Parallel Programming*, 11(5):341–356, 1982.
- [19] P. Ramnarayan, A. Tomlinson, A. Rao, M. Coren, A. Winrow, and J. Britto. Isabel: a web-based differential diagnostic aid for paediatrics: results from an initial performance evaluation. *Archives of disease in childhood*, 88(5):408–413, 2003.
- [20] Z. W. Raś, A. Dardzińska, L.-S. Tsay, and H. Wasyluk. Association Action Rules. In *Data Mining Workshops. ICDMW'08. IEEE International Conference on*, pages 283–290. IEEE, 2008.
- [21] Z. W. Raś and A. Wiczorkowska. Action-rules: How to increase profit of a company. In *Principles of Data Mining and Knowledge Discovery*, pages 587–592. Springer, 2000.
- [22] I. Shams, S. Ajorlou, and K. Yang. A predictive analytics approach to reducing 30-day avoidable readmissions among patients with heart failure, acute myocardial infarction, pneumonia, or copd. *Health care management science*, 18(1):19–34, 2014.
- [23] S. Silow-Carroll, J. N. Edwards, A. Lashbrook, et al. Reducing hospital readmissions: lessons from top-performing hospitals. *CareManagement*, 17(5):14, 2011.
- [24] C. Silva, S. M. Vieira, and J. M. Sousa. Fuzzy decision tree to predict readmissions in intensive care unit. In *CONTROLO2014–Proceedings of the 11th Portuguese Conference on Automatic Control*, pages 365–373. Springer, 2015.
- [25] D. Tomar and S. Agarwal. A survey on data mining approaches for healthcare. *International Journal of Bio-Science and Bio-Technology*, 5(5):241–266, 2013.

- [26] H. Touati, Z. W. Raś, J. Studnicki, and A. A. Wieczorkowska. Mining Surgical Meta-actions Effects with Variable Diagnoses' Number. In *Foundations of Intelligent Systems*, pages 254–263. Springer, 2014.
- [27] M. C. Tremblay, D. J. Berndt, and J. Studnicki. Feature Selection for Predicting Surgical Outcomes. In *System Sciences. HICSS'06. Proceedings of the 39th Annual Hawaii International Conference on*, volume 5, pages 93a–93a. IEEE, 2006.
- [28] M. Vedomske, D. E. Brown, J. H. Harrison, et al. Random forests on ubiquitous data for heart failure 30-day readmissions prediction. In *IEEE 12th International Conference on Machine Learning and Applications (ICMLA)*, volume 2, pages 415–421, 2013.
- [29] R. Veloso, F. Portela, M. F. Santos, Á. Silva, F. Rua, A. Abelha, and J. Machado. A clustering approach for predicting readmissions in intensive medicine. *Procedia Technology*, 16:1307–1316, 2014.
- [30] M. Vukicevic, S. Radovanovic, A. Kovacevic, G. Stiglic, and Z. Obradovic. Improving hospital readmission prediction using domain knowledge based virtual examples. In *Knowledge Management in Organizations*, pages 695–706. Springer, 2015.
- [31] I. H. Witten and E. Frank. *Data Mining: Practical machine learning tools and techniques*. Morgan Kaufmann, 2005.
- [32] P. Yadav, M. Steinbach, V. Kumar, and G. Simon. Mining electronic health records (ehr): A survey. Technical Report TR 15-016, University of Minnesota, MN, 2015.
- [33] K. Zolfaghar, J. Agarwal, D. Sistla, S.-C. Chin, S. Basu Roy, and N. Verbiest. Risk-o-meter: An intelligent clinical risk calculator. In *ACM 19th SIGKDD international conference on Knowledge discovery and data mining*, pages 1518–1521, 2013.