Adapting and Interpreting Machine Learning Techniques in the Biomedical and Clinical Domains

by

Collin J. Engstrom

A dissertation submitted in partial fulfillment of
the requirements for the degree of

Doctor of Philosophy

(Computer Sciences)

at the

UNIVERSITY OF WISCONSIN–MADISON

2019

Date of final oral examination: 12/17/2019

The dissertation is approved by the following members of the Final Oral Committee:

David Page, Professor, Biostatistics and Medical Informatics
Brian Patterson, Assistant Professor, Emergency Medicine
Jignesh Patel, Professor, Computer Sciences
Bilge Mutlu, Associate Professor, Computer Sciences

To my loved ones.

ACKNOWLEDGMENTS

I would like to take this opportunity to thank everyone who has helped me along the path that has led to where I am today.

I would first like to acknowledge my advisors. Thanks to David Page for serving as an advisor, mentor, and guide. Your encouraging and patient disposition, along with your academic gifts, have made you a strong role model for me as a researcher, teacher, and as a person. Similarly, thank you to Brian Patterson for inviting me on to your team and providing a positive, inquisitive atmosphere. Your focus on using machine learning methods for the betterment of patients at the UW Hospital has inspired and challenged me. Thank you to Frank Liao at the UW Health Enterprise Analytics Department as well. Your experience in healthcare has greatly aided us all in implementing our models within the UW Health system. To Brian and Frank: I look forward to continuing our work. To David: I look forward to keeping in touch as you and Lauren move on to the next chapter of your lives!

My work with David, Brian, and Frank has also given me the opportunity to work with three amazing research teams. Thank you to the Biostatistics and Medical Informatics Department and to the students and researchers in David Page's research group (Xiayuan, Aubrey, Wei, Ross, Arezoo, and everyone else). You all have been an invaluable resource. My thanks also go to Brian Patterson's team (Gwen, Apoorva, Alex, Marvin, Chandra, etc.) and the other faculty (Manish Shah and everyone else) and staff (Sharon, Jessie, Treena, Becky, and everyone else) at the Emergency Department and HIP (Maureen Smith and everyone else) who have created a department that is a great place to work and learn. I also owe a debt of gratitude to Frank Liao's team (Sabrina, Joel, and Corey) in the Enterprise Analytics Department at UW Health, along with everyone else at UW Health. I have greatly benefitted from your patience and technical

skills this summer and fall, and I look forward to continuing our collaboration. I would also like to thank Scott Hebbring and everyone else at the Marshfield Clinic for helping Xiayuan and myself with our research.

Similarly, the work outlined in Chapter 3 of this dissertation would not have been possible without our collaborators: James Thomson, Michael Schwartz, Vitor Santos Costa, Ron Stewart, and everyone else involved in the work in predicting neural toxins.

I would further like to acknowledge my committee members, Jignesh Patel and Bilge Mutlu. Your advice and direction in this process is of great help and importance to me as a graduate student and researcher.

I would like to thank the rest of the faculty and staff (Beth, Cathy, Desmond, Shelley, Chris, Pradeep, Kamacho, Hector, and everyone else) at the Computer Sciences Department and the Biostatistics and Medical Informatics Department. Your hard work and dedication have made graduate school a positive experience. I would like to mention how much I appreciate the Computer Sciences Department for funding me before I became involved in research.

To my new friends at CIBM, including Mark Craven: I value your help in the post-doctoral application process and look forward to working with you all!

I would also like to thank Angela Thorp for her time and efforts helping me navigate the complexities of the academic system and also for being a great friend and an invaluable source of advice.

Thank you to Kyle as well for being a source of lightheartedness, encouragement, and introspection that has helped me evolve as a person.

To all my other friends, both those in the UW CS Dept (Erin, Jaclyn, Erdem, Tyler, Austen, Monica, Jia, Debbie, Finn, Erik, Akshay, and the rest), and those in the non-academic community (Abby, Alexis, Brad, Shanda, Alper, Amanda S, Jason, Tiffany, the "Bryce and Josh group," Amanda A, Scott, Peter, Katie A, Paula, Colin, Hiro, Joe, Laura, Katie H, Chelsea,

Robert, Heather, Nate, Tom, Susan, Carol, "Bear," and everyone else), you have all helped me through the most challenging times and celebrated with me the many wonderful moments I have experienced living here in Madison and, earlier, in Minnesota. You have all given me much happiness, support, and a sense of belonging. This has truly made Madison a home away from home for me! To JJ: thank you for being such a great friend–and reading this document not once, but twice.

I would also like to thank the faculty and staff at the Dassel-Cokato (DC) schools in Minnesota. You played an integral role in laying a foundation upon which I have been able to build in life. While every teacher from my past has shown great passion for students and the teaching profession, I would particularly like to mention Jon Ring and Ron Hungerford. It was the two of you who helped me understand that graduate school was a path I was capable of pursuing.

In a similar vein, I would like to acknowledge the faculty and staff at my alma mater, Southwest Minnesota State University. You helped me build on the foundation laid by teachers at DC and demonstrated that a liberal arts education can be used to make us not just better academians, but also more critical thinking, empathetic, and productive members of society. I would like to draw particular attention to my advisor, Dan Kaiser, who demonstrated great passion for the field of computer science and supported my undergraduate journey.

Finally, I wish to recognize all of my family members for being my first teachers. To my mother: you first showed me the potential in myself for teaching and graduate school. You are also simultaneously the kindest and strongest soul I have known. To my father: you taught Derek and me the value of hard work and determination; no one has shown me the concept of "sisu" better than you. To Derek, Alyssa, Graham, and Eva: you have been a source of pride and happiness for me, and you have always given me an honest look at myself when I needed it. To Randy and Lori–our

"adopted" parents: you have built a welcoming and loving home for us all, and you are both truly amazing people. To my grandparents: thank you for teaching Derek and me valuable life lessons gained from your years of experience. Grandma: I have very much enjoyed our phone conversations these past few years. They have kept me grounded, given me perspective, and provided important life tips (like how to quickly thaw a frozen turkey in the bathtub). To my aunts, uncles, siblings, cousins, and the rest of my family: you have all played a vital part in my life before and during graduate school, and I cannot thank you enough!

I would like to conclude by saying that I have done my best to mention as many people by name as possible. Rest assured that even if I did not mention you specifically, though, you are still very much loved and valued.

CONTENTS

LIST OF TABLES

LIST OF FIGURES

ABSTRACT

In recent decades, escalating healthcare costs have drawn the attention of providers and policymakers. These increased expenditures are often due to inefficiencies in patient care, a dilemma that has catalyzed new approaches to healthcare. Key among these are new avenues for leveraging electronic health record (EHR) data. In particular, applying machine learning methods to biomedical and clinical needs has shown remarkable promise. These techniques often present challenges that must be addressed, however. This dissertation discusses certain guiding principles we have gleaned from our own work in applying predictive machine learning models.

First, we demonstrate how neural toxins can be detected by means of a linear support vector machine (SVM). Such an SVM can be trained based on gene expression levels in neural constructs, interrogated by means of RNA-Seq technology. In light of the fact that 60 compounds were analyzed, we use a 60-fold leave-one-compound-out cross-validation strategy, wherein all compounds except one are trained on per iteration. The one remaining compound serves as the test point for that iteration. This leave-one-compound-out process reduces overall variance in predictive measures. In order to do this correctly, though, we observe how traditional leave-one-out cross-validation must be tweaked to meet the needs of the application. Since each compound has two replicates, these must both be held out when the compound is used as a test point. Doing this produces a model effective at discerning toxic from non-toxic compounds. In addition to the modified leave-one-out technique, we also employ a blinded test set, upon which the same trained model attains 90% accuracy. In both of these cases, existing techniques were *adapted* to meet the needs of a current scenario and increase model effectiveness, while at the same time ensuring a fair evaluation.

Next, we demonstrate how the at-home falls risk among elderly patients seen at UW Hospital's emergency department (ED) can be mitigated by machine learning-assisted risk stratification. In this case, six models were trained on actual retrospective ED data, and they evinced strong performance in terms of machine learning metrics. From the models' output, number needed to treat (NNT) (a measure of interest to clinicians) can be projected based on how many patients per week must be referred. In this way, we emphasize how model performance must be correctly *interpreted* to best suit the needs at hand.

The models used in this study are currently being transitioned into production at UW Hospital's ED by our group in that department, along with the predictive analysts at UW Health. During this transition, simplifying steps have had to be made, leading us to conclude that *translational considerations* must play a role in how a model is implemented. As a corollary to this, what has also become clear is that for such models to be of practical use, the *model choice* must conform to the properties of both the data and the end user expectations (i.e., physicians may prefer a model whose choice of features is more transparent, like tree-based methods). Finally, even a highly practical, well-chosen model will likely not see widespread adoption if it fails to consider the day-to-day *workflow* into which it is placed.

While these models trained on traditional metrics (e.g., hinge loss in the case of SVMs) have performed well in their own right in use cases like falls in the ED, the question has been raised as to whether a model that has been directly trained on NNT (or a measure similar enough to it) would perform better. In the final part of this dissertation, we demonstrate how precision can be used as a proxy for NNT. Once this substitution is made, it is possible to perform an optimization. In making this *adaptation*, we uncover how performance in a specific segment of patients (namely the top k at highest risk) surpasses the same measure in a more traditional

machine learning model.

In aggregate, these principles of machine learning used in the biomedical and healthcare domains can be taken as guiding principles for other researchers seeking to design and implement similar models. Moving forward, considering these observations and those gained from other applications will be an important tool in not only advancing strictly academic work, but also in tackling the cost and efficiency concerns that currently beset healthcare in the US.

1 INTRODUCTION

Recent years have seen elevated interest in the use of machine learning methods in the biomedical and clinical fields. Machine learning methodologies have, in many cases, seen application in predicting phenotypic characteristics like disease risk (Ban et al., 2010; Imielinski et al., 2009) or drug dosing (IWPC, 2009). Healthcare organizations have also experienced an increased use of machine learning algorithms for the improvement of patient care. While the burgeoning use of machine learning techniques for these two domains has received much attention, what has been given less treatment is the intersection between machine learning and the biomedical and clinical domains.

As has been pointed out in the literature, with the advent of non-machine learning researchers using machine learning models, there has been a tendency for some researchers to incorrectly apply techniques or draw dubious conclusions (Luo et al., 2016). To mitigate such pitfalls, the issue of machine learning accessibility is crucial.

From the machine learning standpoint, researchers have long emphasized the symbiotic relationship between pure academic research and pure application. Pure research, in this case, can be thought of as the conglomeration of *theory*, *algorithms*, and *methodology*. On the other side, application tends to place high value on the *end user* and *use cases*. These two paradigms need not be thought of as completely separate or even opposing extremes set along a continuum. Rather, they should be considered as two entities mutually benefitting from one another. Viewed as a dynamic process like this, applicatory fields benefit from new methodologies born out of pure research, while the machine learning research is simultaneously driven by real-world data and end user needs (Provost and Kohavi, 1998).

Our work has centered on three applications: one in the biomedical

setting and two in the healthcare setting. The second of these settings, in particular, has demonstrated an increasing need for the inclusion of machine learning research. Thought of in terms of the work mentioned in Provost and Kohavi (1998), healthcare becomes the application-focused entity, or the "use case," while the physicians and other clinical providers are considered the "end user." In recent years, work revolving around these ideas has emerged in the form of what has been termed *learning health systems*. The Institute of Medicine defines a learning health system as:

> A system that "learns," or more concretely, as "one in which knowledge generation is so embedded into the core practice of medicine that it is a natural outgrowth and product of the healthcare delivery process and leads to continual improvement in care" (Olsen et al., 2007).

UW Health and its academic partners in the School of Medicine and Public Health (SMPH) represent one example of a transition toward learning health systems. Recent work in our group at the Department of Emergency Medicine at the UW Hospital has focused on at-home falls in elderly patients visiting the Emergency Department. This work has culminated in the first in-house models derived for the purpose of risk stratification at UW Health; these models have been further transitioned into production. As such, the partnership also constitutes a novel collaboration between the academic side (SMPH) and the production side (UW Health) within the context of learning health systems. Crucial to helping this process along has been input from the Industrial Engineering Department as well, using the SEIPS model (Carayon et al., 2006; Holden et al., 2013) to aid in integrating models into workflow.

From this collaboration have emerged specific observations that will be explored in depth in subsequent chapters. These take-away points can

be distilled down to the following:

- Clinical context will influence *model choice*;
- *Translational considerations* must be taken into account, and the model may need to be adjusted accordingly;
- Implementing models in a real clinical setting requires careful study of existing workflow(s) in place and ensuring that *model placement in the workflow* is done properly;
- Model results must be *interpreted* correctly; and
- Often, common models and traditional procedures for using them must be *adapted* to suit the needs of clinicians.

The following five sections expand on these points further, and the resulting thesis statement is given thereafter.

1.1 Model Choice

When attempting to apply machine learning techniques, the question of how to select a model arises. The answer largely hinges on what one hopes to achieve from the modeling process. A researcher totally unfamiliar with all of the intricacies of machine learning, for instance, may choose a tree-based methodology. As we discuss in Chapter 2, tree-based models (e.g., random forests) have the benefit of being one of the more intuitive models in terms of how features are iteratively split on until some notion of "purity" in the leaves is satisfied. In our aforementioned falls project (Patterson et al., 2019), we were interested in models that would perform well "in the wild," or upon being deployed into production. For this reason, we evaluated six different models and discovered that tree-based methods tended to outperform the others. As discussed in the background material

of Chapter 2, this can likely be attributed to decision trees representing collinear relationships in the data in ways that the other models cannot.

These results, while encouraging, were put to the test by the transition from our academic environment to production, and we needed to take some translational concerns into consideration.

1.2 Translational Considerations

As our work in Patterson et al. (2019) uncovered, models on the academic (SMPH) end of the spectrum led to a strong final performance, particularly with the tree-based methodologies. One challenge that arose when attempting to translate the models into production (UW Health) was that while the original dataset of patient visits was common to both SMPH and UW Health, SMPH had employed an intermediate cleaning process before making the dataset available for their researchers to use. Additionally, new features derived from those in the parent dataset were included in the released dataset. Neither the data cleaning nor the feature creating processes were easily available. For this reason, features needed to be reconstructed de novo on the production side using the following process:

1. Identify a feature in the original (i.e., SMPH side) model;
2. Locate Clarity (Epic's back-end database) table(s) containing this feature; and
3. Using SQL queries, bring the feature into the production side.

Only after this was it possible to build and test a final model based on these features. In short, features could not be mapped en masse, but instead they required manual translation. The original study included 725 features in all six of the models. Subsequent work, however, showed

that similar results could be achieved with a "parsimonious," or slimmed-down, feature set of 15 features relating to demographic data, as well as questionnaire data from the time of visit (Clegg et al.). For this reason, the parsimonious model was selected for production. When this model was translated to UW Health's side, one difference was that the tree-based algorithms now underperformed relative to the penalized regression ones. In this case, adding more features to expand the parsimonious model would require further hand-mapping of those features by the process outlined above.

After translational considerations such as these are taken into account and the models finalized, researchers still must take care that models used in production adhere to existing protocols and workflow within the organization.

1.3 Model Placement in the Workflow

The learning health system and its instantiation(s) at healthcare organizations determine in what stage of the workflow a predictive model should be deployed. This placement should be done in accordance with existing protocols and paradigms used. UW Health has formalized the workflow into a six-module toolkit. In addition to UW Health, our work has also involved a collaboration with the UW Industrial Engineering Department, as mentioned above. These issues will be discussed later in Chapter 2.

1.4 Interpreting Machine Learning Models

In certain cases, existing machine learning techniques may be sufficient to handle a scenario without intrinsically changing the model training or testing phases. In essence, one need only use a straightforward application of one or more models. While some tasks lend themselves to such

application (Patterson et al., 2019), there still may be a need for discerning how the model's output can be used to answer the question at hand. In the cited work, for instance, it was shown how modeling techniques have been integrated into healthcare decisions, helping providers risk stratify patients where resources are constrained. In this case, the fundamental nature of the models was not changed, nor was the primary output metric (i.e., area under the receiver operating characteristic curve, or AUROC). Still, proper thresholding aided in constructing further metrics used to assess patient risk.

1.5 Adapting Machine Learning Models

Increasingly, researchers in multidisciplinary fields would like to use machine learning techniques. In some cases, however, these researchers may find that AUROC is a less meaningful measure for treating patients than others typically used. Still others may encounter research applications that simply do not lend themselves to traditional model evaluation metrics. In short, they are interested in a way to *adapt* modeling techniques to fit their needs.

In cases like these, machine learning models can sometimes be altered in straightforward ways. For instance, recent work has demonstrated that traditional regression-based techniques overemphasize feature importance in determining the penalty term, while downplaying or not considering domain knowledge of these features. To address this, one approach has been proposed to modify the penalty term in a lasso logistic regression model to instead consider what a domain expert (in this case a clinician) deems to be more or less relevant in predicting some condition (Wang et al., 2018).

In other cases, however, researchers may be interested in adjusting models to use metrics more traditionally aligned with their own field of

study, in which case the *objective functions* in the learning process must be adapted (Engstrom et al.).

1.6 Thesis Statement

The purpose of this thesis will be the exploration of the following statement:

> Machine learning research and application are two interdependent components in a greater dynamic process. Fully leveraging the interplay between these two components depends on considering clinical context, translational considerations, organizational workflow, as well as model interpretability and adaptability.

1.7 Dissertation Organization

This dissertation is organized as follows.

Chapter 1 (the current chapter) introduces machine learning and discusses its role in the biomedical and clinical fields.

Chapter 2 discusses the machine learning and medical background upon which the remainder of the dissertation builds.

Chapter 3 introduces an application of machine learning to discerning neurotoxins.

Chapter 4 describes how a machine learning approach can be used to risk stratify patients who may encounter a fall in the near future.

Chapter 5 outlines an approach to adapting traditional machine learning models to suit specific needs in the medical domain.

Chapter 6 lays out ideas to be pursued in future work.

Chapter 7 gives concluding remarks for the dissertation.

2 BACKGROUND

In recent decades, machine learning techniques have gained traction in a myriad of domains outside of computer science. In particular, the field of medicine has seen a sharp increase in such techniques. With the advent of electronic health record (EHR) data has come the recognition that the data stemming from these records can aid in improving healthcare. In 2014, the National Institute for Healthcare Management (NIHCM) noted that 5% of patients in the US account for roughly half of all healthcare expenditure. Such observations have led for calls to better allocate resources and more efficiently administer healthcare (Schoenman and Chockley, 2012).

One potential solution known as precision medicine has garnered attention among clinicians. Precision medicine can be broadly defined as the process of tailoring medical care to fit individual patients' needs, as opposed to a one-size-fits-all approach. Central to realizing this targeted care is the implementation of machine learning techniques.

While machine learning methods have seen successful application in other scenarios like credit card fraud monitoring (Benson Edwin Raj and Annie Portia, 2011), self-driving cars (Stilgoe, 2018), and film preferences (Bennett et al., 2007), using them in the domain of medicine presents unique challenges. Specifically, as discussed in Chapter 1, models and techniques must be properly understood for there to be any hope of properly designing and executing the machine learning portion of a study. Moreover, researchers must take care not to draw false or misleading conclusions from the model's output (Luo et al., 2016).

The challenges of models used, then, can be distilled into five central issues. The first three of these issues relate to *model choice*, along with *translational considerations* and constraints with a model's placement in an organization's *workflow*.The fourth is the issue of *interpretability*, or making certain that researchers who are not experts in machine learning do not

misconstrue what models are saying. The final related challenge is that of *adaptability*, or exploring new ways of making machine learning applicable to the field of medicine in cases where desiderata specific to this domain may not align with traditional approaches to machine learning.

The background information requisite for exploring these issues in subsequent chapters will be given in the following two sections. Section 2.1 discusses the machine learning models and techniques employed in this dissertation. Section 2.2 expands on this, delving into machine learning in the medical domain and how challenges can lead to novel techniques that intrinsically account for the needs of the medical community.

2.1 Machine Learning Models and Techniques

This section first outlines the machine learning models used in the subsequent chapters. Related to this, it elaborates on the process of training, tuning, and testing the models. Finally, it reviews methods and metrics for interpreting the final output of said models.

2.1.1 Models Used

Support Vector Machines

Support vector machines (SVMs) are a type of machine learning methodology whose aim is to maximize a margin between two classes of training instances (Cortes and Vapnik, 1995a). To see this, consider Figure 2.1. Note that there are three labeled regions of interest. The first is the region of positive training instances (denoted by the green '+' symbols); these represent one of the two classes. The third region, conversely, is the region of negative training instances (denoted by the red '−' symbols); these represent the second of the two classes. Dividing these two classes is the second region; this is known as the margin. In the center of the margin

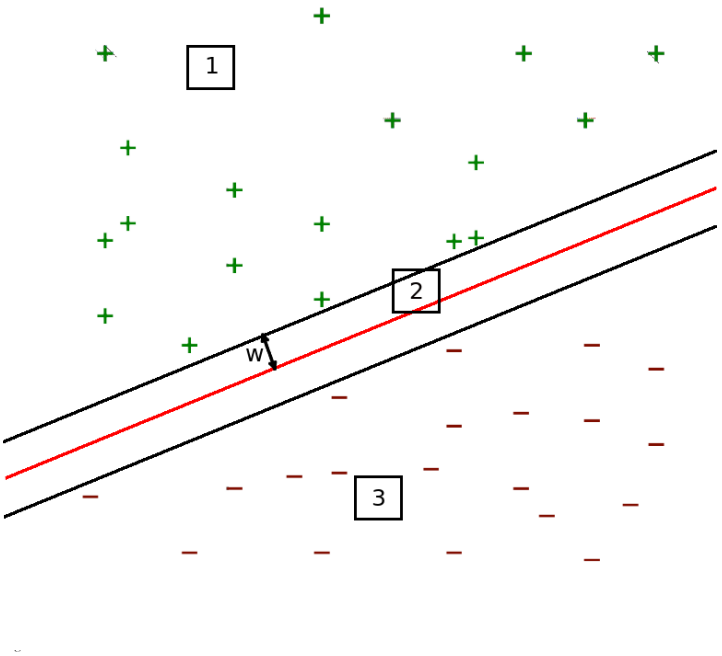


Figure 2.1: A simple linear support vector machine.

resides the hyperplane (represented by a red line in the figure). In the linear case, the hyperplane is nothing more than a line. More generally, any number of dimensions can be represented by SVMs. In cases where a higher dimension is needed to differentiate between classes, the hyperplane is of dimension $n-1$, where n is the number of dimensions in the space to which the features are mapped.

In terms of the figure, “maximizing the margin," translates into expanding region 2 (i.e., maximizing $\frac{1}{||w||}$) to the greatest extent possible, without absorbing examples of either class. In the most basic formulation, instances also should not fall on the wrong side of the hyperplane. In practice, though, this is often not a realistic assumption. To loosen this restriction, a further parameter, called the C-parameter, is introduced to allow for some training instances to reside on the other side of the hyper-

$$\begin{gathered} \mathrm{argmin}_{w,b} \frac{1}{2}||w||^2 + C \sum_{n=1}^{N} \xi_n \\ \mathrm{s.t.} y_n(w^T x_n + b) \geqslant 1 - \xi_n (n = 1, ..., N) \\ \xi_n \geqslant 0 \end{gathered}$$

Figure 2.2: SVM Primal Formulation

plane. This parameter can be tuned according to how much or how little of a penalty there should be for misclassified instances. This is all summed up in Figure 2.2, which shows the primal (or most basic) formulation of the SVM objective function.

Note in the primal formulation that ξ, or the "slack variable," accounts for the magnitude of those instances falling on the wrong side of the hyperplane, while C is the overall penalty on allowing those instances to be placed among those of the opposite class. In certain cases, a higher-dimension kernel may be necessary to more cleanly divide the two classes.

The complexity of fine-tuning the C-parameter as well as other kernel choices is a known drawback to using SVMs. Additionally, discrete parameters must be normalized to have accurate training of SVMs. Even in cases where both of these requirements are met, SVMs may underperform other models in datasets where the number of training instances is dwarfed by the number of features per instance (Burges, 1998).

In spite of these caveats, SVMs have shown predictive utility in medicine, particularly in studies where sufficient data points exist. For instance, prediction of diabetic and pre-diabetic patients has been achieved with SVMs (Yu et al., 2010). They have also been used to successfully uncover cases of heart disease (Maglogiannis et al., 2009). SVMs have also seen application in the field of bioinformatics. In Chapter 3, we introduce an application focused on predicting which compounds are neurotoxins using an SVM model.

Regression-Based

Regression-based models have been applied in scenarios ranging from pediatric asthma (Patel et al., 2018), to sepsis (Taylor et al., 2016), to heart failure (Mortazavi et al., 2016). Such applications have shown promise, but certain preconditions must be satisfied in order for the models to perform as expected. For instance, there should not be a preponderance of multicollinearity or strongly influential outliers. Should these criteria not be met, regression-based models may underperform other methodologies (Stoltzfus, 2011). Below, the various types of regression learning are discussed.

Linear Regression: Linear regression is also known as ordinary least squares linear regression. Used within the context of machine learning, linear regression is identical to that in traditional statistics: an algorithm is generated to fit a line between one or more explanatory variables and a dependent variable. Each explanatory variable value is multiplied by a regression coefficient, and coefficients are varied to find the best fitting line. The fit of this line is evaluated by calculating the sum of the squares of the distances between predicted values and observed values from the data (Marill, 2004a,b). This "sum of least squares" is referred to as the error metric. During learning, the algorithm adjusts coefficients to minimize the error metric, producing the best possible fit line.

Logistic Regression: In the context of machine learning, logistic regression is very similar in principle to linear regression; however, instead of fitting a straight line between continuous data, the algorithm generates a linear predictor function from the inputs and coefficients which is transformed to create a logistic model bounded at 0 and 1 to suit the output requirements for categorical data. Its formula is the sigmoid function:

$$\frac{1}{1+e^{-y}}$$

where $y = b_0 + b_1X_1 + ... + b_nX_n$. Here, each of the X-values represents one of the features (inputs), and the associated b-value is the coefficient. Unlike linear regression, which has a closed-form solution, gradient descent is required to find the optimal logistic regression parameters.

Ridge-Penalized Logistic Regression: Based on the structure of the error metric in linear and logistic regression, collinearity between predictor variables may result in poor regression performance, with either under or overfitting. Ridge regression introduces another term (the "L2 penalty") into the error metric. This penalty takes the form of a constant entered by the user multiplied by the square of the coefficient. In machine learning, such constants which are added by users are termed hyperparameters. In order to find a best fit line during learning, the goal is not only minimization of the original error term, but minimization of the combination of error and penalty term. The structure of the penalty term in ridge regression is such that it has the effect of reducing the magnitude of individual coefficients among multicollinear variables, improving performance of the system (Hoerl and Kennard, 1970; Friedman et al., 2010). Again, this can be solved by gradient descent.

Lasso-Penalized Logistic Regression: Lasso is an acronym for "least absolute shrinkage selector operator." Conceptually, it is very similar to ridge regression, with the introduction of an L1 penalty (as opposed to L2 penalty). The L1 penalty consists of a hyperparameter multiplied by the absolute value of the regression coefficient. Including this term in the error metric results in reducing some coefficients in the model to zero.

This allows the learning algorithm to perform feature selection, leading to a parsimonious model including only those features most important to prediction (Tibshirani, 1996, 2011). This cannot be solved by standard gradient descent since the penalty is not differentiable at 0, but it can be solved by coordinate descent or various modified versions of gradient descent to handle the undifferentiable places.

Tree-Based

In large datasets with anticipated but unknown variable interaction, decision tree methodologies offer the ability to aggregate diverse types of data to make accurate predictions, and they often compare favorably with advanced regression techniques (Kingsford and Salzberg, 2008). To build decision trees, data features are selected based on identifying the most discriminative variable to form the nodes in a classification tree (Breiman et al., 1984), with each terminal node being assigned to a class. Inductive learning of decision trees is accomplished by sequentially adding variables until the terminal nodes achieve sufficient predictive capability, and then pruning the resulting trees to avoid overfitting (Colombet et al., 2000; Lewis, 2000). Tree-based methods offer the ability to deal with complex variable interactions and nonlinear effects in large datasets (Cairney et al., 2014). The primary disadvantage of decision trees is a potential for overfitting models to training data despite pruning techniques.

Random Forest: Random forest algorithms address potential overfitting by iteratively sampling within a dataset to build trees from multiple subpopulations, creating a "forest" of potential trees (Breiman, 2001) and predicting for new cases by an unweighted vote of the predictions of all the trees (Genuer et al., 2010).

Random forests have seen wide application in medicine for predicting conditions like diabetes (Devi and Shyla, 2016), breast cancer (Hsu et al.,

2015), and even the effect pharmaceutical molecular structures have on biological activity (Svetnik et al., 2004). In tasks like these, the performance of random forests tends to be competitive with other machine learning methods. One caveat that has been pointed out is that careful selection of features may be necessary for realizing optimal performance (Alam et al., 2019; Kaur et al., 2019).

AdaBoost: AdaBoost (or "Adaptive Boosting") is a machine learning meta-algorithm, or a wrapper, that can be used with many types of machine learning models, including decision trees. AdaBoost works by using a boosting procedure to subsample the training data and create many "weak" learners (those with predictive accuracy slightly above guessing) of the model type being used. These various weak classifiers ultimately take a weighted vote on the final classification for an instance. AdaBoost is said to be adaptive, because as more and more weak models are created, it shifts focus to the instances that were misclassified by the previously created weak models (Freund and Schapire, 1997).

Researchers have successfully applied AdaBoost in breast cancer prediction and survival (Abuhasel et al., 2015). While successful in using this type of adaptive approach, researchers in these studies have cautioned that outliers in a noisy dataset may exert an outsized impact on the learning process (Adegoke et al., 2017; Dietterich, 2000), particularly in the case of class noise (McDonald et al., 2003).

XGBoost: XGBoost (short for "eXtreme Gradient Boosting"), as its name implies, is a framework for accomplishing gradient boosting for machine learning models (Chen and Guestrin, 2016). Generally speaking, gradient boosting algorithms are built on top of "weak learners," as was the case with AdaBoost (Friedman et al., 2000). Unlike AdaBoost, however, gradient boosting techniques do not iteratively re-weight instances based on predictive performance. Instead, they typically function by *adding more*

weak learners in a greedy fashion in an attempt to reduce an overall loss function (Chen and Guestrin, 2016).

XGBoost is unique in its approach to gradient boosting. Unlike other similar methods, it strives for scalability in all scenarios. In pursuit of this, XGBoost leverages sparsity-awareness, cache-awareness, as well as an approximate split-finding algorithm in cases where a full dataset does not fit into memory (Chen and Guestrin, 2016). It also improves on existing shortcomings in traditional boosting methods. As detailed on AdaBoost, noise in datasets may have an inordinately high impact on a boosted model's predictive power. XGBoost defies this tendency, though, demonstrating a relatively high level of robustness to noise when compared to AdaBoost and other traditional gradient boosting (Gómez-Ríos et al., 2017).

Due to XGBoost's high performance and predictive capacity, it has seen wide application, often meeting or exceeding accuracy levels of other methodologies in competitions such the Kaggle competition (Mangal and Kumar, 2016). In the medical domain, XGBoost and XGBoost-like approaches have been employed in drug prediction for precision medicine in fighting cancer (Janizek et al., 2018), predicting atrial fibrillation (Chen et al., 2018), and also in pathway analysis (Dimitrakopoulos et al., 2018).

2.1.2 Training and Testing the Models

Central to using machine learning models are the training and testing procedures used to develop these models. In subsequent chapters, methods pertaining to supervised learning are used. Supervised learning can be defined as the following:

> Given a training set of M example input-output pairs
> $(x_1,y_1),(x_2,y_2),\ldots,(x_M, y_M)$,
> where each y_j is generated by an unknown function $y = f(x)$,

> find a function $h(x)$ that approximates the true function (Russell and Norvig, 2016).

Implicit in this definition of supervised learning is that the function $h(x)$ (often called a hypothesis) is being drawn from some greater space H (or hypothesis space) of other functions that would also potentially suit the task of approximating $f(x)$. From this arises the need to assess the "goodness" of the chosen function $f(x)$, and a held-aside test set is used to accomplish this.

At the most basic level, if one considers a set of instances:

$$(x_1,y_1),(x_2,y_2),\ldots,(x_M, y_M),$$

one would use only a portion of these instances for the training set:

$$(x_1,y_1),(x_2,y_2),\ldots,(x_m, y_m),$$

while holding aside the remainder for the test set:

$$(x_{m+1},y_{m+1}),(x_{m+2},y_{m+2}),\ldots,(x_M, y_M),$$

where $m < M$. This train-test split is shown in Figure 2.3. Note that the test set can be further divided to incorporate a tuning set, which can be used to set the model's hyperparameters.

In most cases, the instances are randomly shuffled prior to breaking into the training and test sets. One drawback to this division is that it assumes a static underlying data distribution over time. In certain cases, this assumption may not hold. Various medical studies, in particular, have shown that a calibrated model, while showing good performance initially, may deteriorate over time due to population traits shifting (Kukar, 2003; Davis et al., 2017).

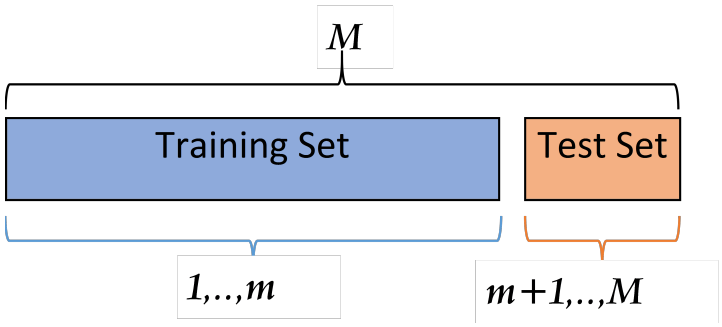


Figure 2.3: A depiction of training-test split.

Chronological Splitting: To account for this drifting effect on models, one strategy is to chronologically arrange a dataset taken from a large time span. With this approach, then, older data points become training instances, while newer ones serve as test points. As opposed to randomizing the instances between the two datasets, this acts as a more conservative way of testing the model, accounting for any drift the model may have encountered for the duration of time measured in the dataset.

Iteration					
1	Train	Train	Train	Train	Test
2	Train	Train	Train	Test	Train
3	Train	Train	Test	Train	Train
4	Train	Test	Train	Train	Train
5	Test	Train	Train	Train	Train

Figure 2.4: A depiction of 5-fold cross-validation.

Cross-Validation Another common approach of training and testing is to use n-fold cross-validation, where n is typically 5 or 10. As shown

in Figure 2.4, this approach uses n iterations with n partitions over the dataset. In each of these iterations, one of the successive folds "takes its turn" acting as the test fold. The remainder of the folds serve as training data during this iteration. Using this approach as opposed to a single train-test split reduces variability and also bolsters model generalization by reducing overfitting (Kohavi et al., 1995).

Leave-One-Out Cross-Validation: A variant of the n-fold cross-validation, as its name implies, sets the number of folds (or n) to the number of instances in the dataset. In each of the n iterations, one instance is held out as a test instance. In cases where examples are not independent, such as predicting protein structure or properties of web pages, one might group examples and perform leave-one-group-out (King et al., 2000; Craven et al., 1998). An example of this is seen in Chapter 3.

2.1.3 Interpreting Results

Another crucial factor in deciding which machine learning framework to ultimately use is how that model performs. In this dissertation, the following measures of performance are used.

Accuracy: Accuracy is simply the fraction of instances that the model predicted correctly. That is:

$$\frac{\mathrm{NumberofCorrectPredictions}}{\mathrm{TotalNumberofPredictions}}$$

True Positive Rate (TPR): Also called sensitivity, the true positive rate can be defined as the fraction of all positive instances the model actually flags as being from the positive class:

$$\frac{\mathit{TruePositives}}{\mathit{TruePositives} + \mathit{FalseNegatives}}$$

False Positive Rate (FPR): The false positive rate can be defined as the fraction of all negative instances the model flags as being from the positive class:

$$\frac{\mathit{FalsePositives}}{\mathit{FalsePositives} + \mathit{TrueNegatives}}$$

Receiver Operating Characteristic (ROC) Curve: Taken together, TPR and FPR can be plotted into what is called a receiver operating characteristic (ROC) curve. The metric of interest that can be derived from this curve is known as the area under the ROC curve, or AUROC for short. (Note that this is often further shortened to AUC in contexts when it is clear that the ROC curve is being used. Throughout this dissertation, this is often the case.) For any given curve, an AUROC near 1.0 is optimal, while an AUROC around 0.5 indicates a performance no better than guessing.

An example ROC curve is shown in Figure 2.5. Note that the dashed diagonal line represents an ROC curve based purely on guessing. (I.e., it would have an AUROC of 0.5.) Conversely, an ROC curve with an inflection point near the top left corner would have an AUROC near 1.0; thus, curves that lean upward and to the left are preferable to those that lean toward the dashed line.

Precision: The final metric of interest is precision, which can be thought of as the fraction of positive instances a model correctly flags as positive

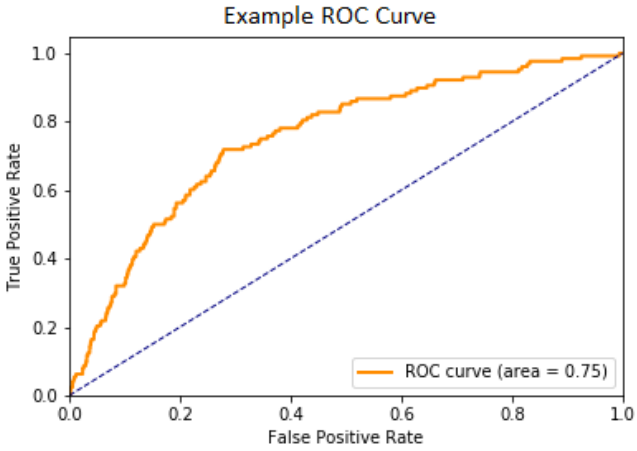


Figure 2.5: A depiction of an ROC curve.

out of all those it flagged as positive:

$$\frac{TruePositives}{TruePositives + FalsePositives}$$

2.2 Clinical Background

As alluded to in Chapter 1, healthcare systems suffer from issues relating to safety, quality, and inefficiency (Carayon et al., 2006; Baker, 2001; Ball and Balogh, 2016). Machine learning, in the context of the learning health system, has been proposed as one strategy for improving health system efficiency by focusing resources where they are most beneficial.

2.2.1 Learning Health Systems

Healthcare researchers have argued that, for too long, one factor contributing to concerns like efficiency and quality of patient care in academic

hospitals is that there has been a single-minded focus on discovery, often to the detriment of scalable concrete solutions that put new knowledge into practice. As one researcher encapsulated the problem, the tendency is to favor "all breakthrough and no follow-through" (Grumbach et al., 2014). How, then, should the challenges of inefficient and costly care be addressed?

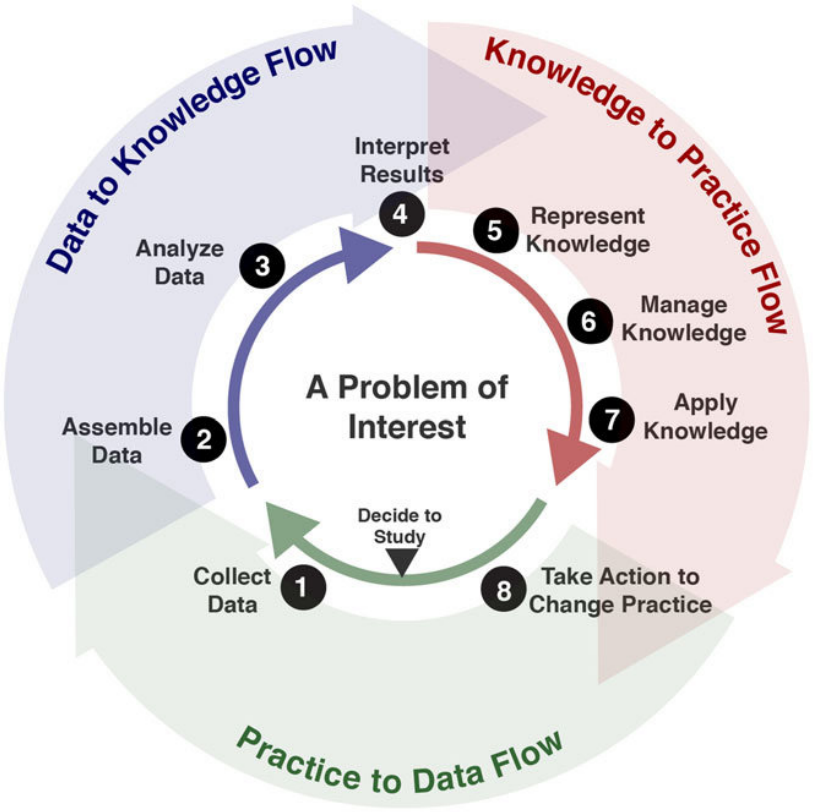


Figure 2.6: A depiction of the learning health cycle. Image courtesy of Flynn et al. (2018).

The learning health system paradigm hopes to ameliorate these issues and others by decreasing the time it takes to implement new techniques and knowledge (i.e., to bring them directly to the patient). Recall that the basic idea behind a learning health system is that it is simply a healthcare

entity attempting to learn about itself. This can be seen in Figure 2.6. Note that the self-learning process shown in the figure unfolds in three broad steps, wherein:

1. Real-world practice generates data;
2. Data can be used to provide knowledge about some question; and
3. The knowledge gleaned influences the practice.

In this way, the health system is both learning about itself and putting the knowledge into practice to improve its performance (Flynn et al., 2018). To expedite the learning process, predictive models are often employed.

While machine learning can be used, one might reasonably wonder about where in the learning health systems cycle such models should live. As noted in Chapter 1, correct placement and functioning of a model in an existing workflow is crucial to preventing the workflow from being disrupted.

A central concern revolving around an organization's workflow is how it addresses quality assurance issues. One iterative process called PDCAs (alternatively PDSAs), or Deming cycles, attempts to encapsulate quality assurance in a process similar to the scientific method (Deming et al., 1986). Like the scientific method, PDCA cycles explain how to solve a problem (e.g., quality assurance issues that arise). This is done in four steps:

1. **P**lan: establish procedures and what must be accomplished from them;
2. **D**o: execute the plan laid out in step 1;
3. **C**heck: analyze results and data collected; and
4. **A**ct: improve the initial process.

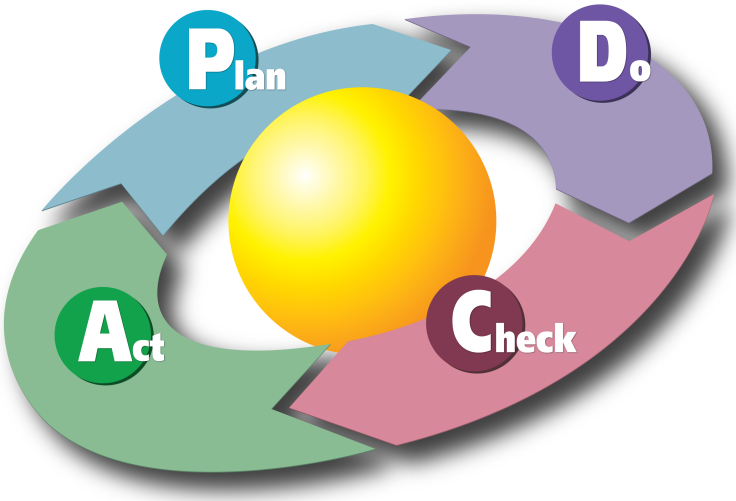


Figure 2.7: The PDCA cycle. Diagram by Karn G. Bulsuk (`http://www.bulsuk.com/`).

This procedure is depicted in Figure 2.7. Note the arrow between "Act" and "Plan." This indicates that the cycle may need to be repeated more than once to achieve the set objective(s). If such is the case, the cycle continues back at the "Plan" step, taking into account anything learned from the prior iteration.

The PDCA cycle has been viewed as an essential part of the learning health system in that it can be viewed as a means of addressing the efficiency and quality concerns mentioned above (Taylor et al., 2014). In particular, this process has seen implementation at UW Health.

UW Health is a health system partnered with the UW School of Medicine and Public Health (SMPH) that serves more than 600,000 patients every year (UW Health, 2019). Treatment of such a vast number of patients is accomplished through a close working relationship between these two affiliates. Our work, in particular, has benefitted from this partnership, embodying the PDCA process outlined above by translating a set of pre-

dictive models from the academic (SMPH) side to the clinical (UW Health) side. Crucial to helping achieve this translation is work that has been done by researchers at UW Health focused on moving towards the goals of a learning health system.

UW Health predictive analysts have developed a framework that closely adapts the PDCA process outlined above. A toolkit that they have created formalizes this, and it is available to help researchers at UW and elsewhere implement their models in real-world contexts (Adelaine et al., 2019). There are six modules to this toolkit, as shown in Figure 2.8. This toolkit adapts the FOCUS-PDCA methodology, an extension of PDCAs that has seen application in healthcare for improving patient care (Bader et al., 2002).

The first module (*"find"*) aims to ensure that what the predictive model is measuring aligns with organizational goals. In essence, it seeks to *find* any facets of the model that must be made to conform to what the organization hopes to accomplish. Our work focuses on machine learning models designed for the purpose of more effectively using resources (e.g., referral slots for the UW Mobility and Falls Clinic), as is discussed in our work in Chapter 4. This is of practical interest to UW Health from the standpoint of the quality of care patients receive, so our model satisfies the goals of this module.

The second module (*"organize"*) involves identifying and involving the right people in the project at hand. The work we have engaged in has, in fact, included researchers from many areas, both academic and clinical. Challenges encountered have necessitated help from clinicians, health communications researchers, and predictive analysts, among others. Often, including such an array of researchers promotes a clearer understanding of how to integrate the model into the existing workflow, which is embodied by the third module (*"clarify"*). Indeed, deploying our model has meant understanding where in the UW Hospital Emergency

Figure 2.8: The learning health system embodied by a UW Health toolkit.

Department's workflow the predictive model should be included.

A related concern revolves around ambiguity or variation in protocols that must be addressed for a successful rollout of the model. This is handled in the fourth module (*"understand"*).

After these considerations for workflow and usability have been taken into consideration, there still may be a disconnect between academians and clinicians in deciding what makes a model "good." This challenge is handled by the fifth module (*"select"*) . The impetus for this module closely relates to two of our take-away points in Chapter 1: interpretability and adaptability. As we discuss later in Chapters 4 and 5, an open area of research in clinical informatics is how machine learning models must be

properly interpreted or even changed to match the needs of a real-world problem.

Finally, after all of these modules are executed, a researcher moves into the sixth module (*"Plan, Do, Check, Act"*). Here, as its name implies, planning, doing, checking, and acting are key to finishing the six-fold implementation process. Often, the health system will need to learn more about itself based on how the current project unfolds. Another common issue that this stage draws attention to is data drift that may occur in the underlying characteristics over time. Note that this last step is a direct application of PDCA cycles.

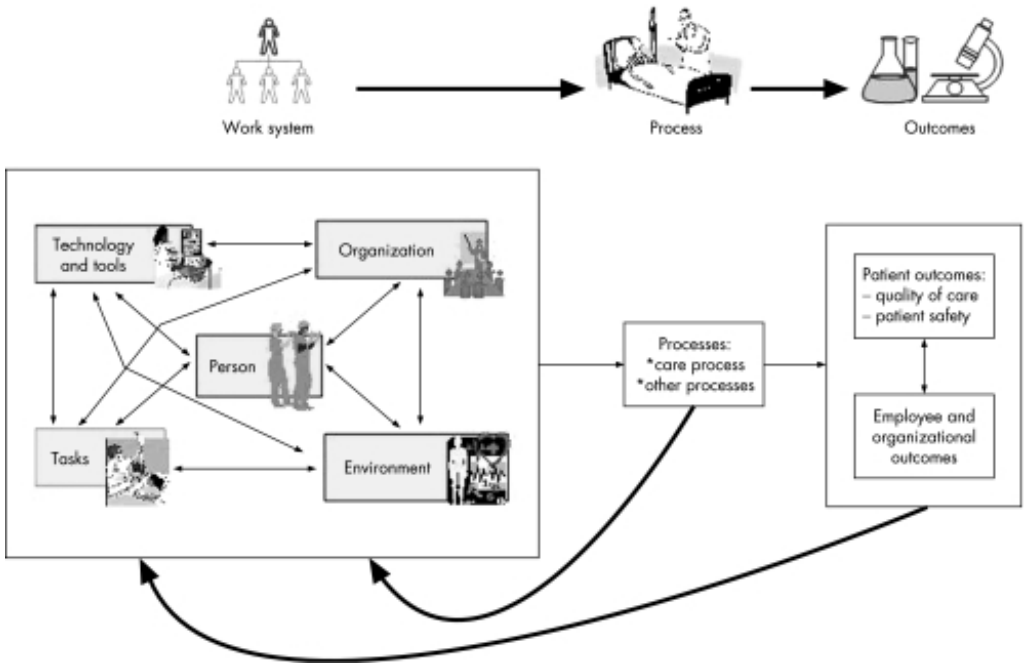


Figure 2.9: The SEIPS model as a means of improving outcomes. Figure courtesy of Carayon et al. (2006).

As alluded to in Chapter 1, our collaboration with the UW Industrial Engineering Department has also guided model development. A cornerstone to the work emerging from the partnership has centered on the

SEIPS model (Holden et al., 2013; Carayon et al., 2006) and its concomitant focus on human factors. These human factors include not only the impact a healthcare system can have on patient safety, but other employee and organizational outcomes as well (Holden et al., 2013).

The SEIPS model is shown in Figure 2.9. Notice that, when considering the development of an intervention, all interactions between people and their environment must be considered. These various interactions are depicted under the first section of the SEIPS model in Figure 2.9 titled "Work system." The organizational structure of the work system, in turn, affects the patient's care and safety, which falls under the second section titled "Process." Finally, the processes which are taking place affect the patient, employee, and organizational outcomes that emanate from the process. This is shown in the "Outcomes" section. Note that both processes and outcomes can feed back into the work system as well.

In our projects, we have taken these workflow paradigms into consideration when designing the tools which communicate model results with providers This is explored further in Chapters 4 and 5.

2.2.2 Machine Learning in the Healthcare Domain

Machine learning models have been used in medical tasks ranging from predicting diabetes risk (Lai et al., 2019; Xie et al., 2019; Anand et al., 2018) to forecasting the chances of mortality (Kim et al., 2019; Hill et al., 2019). Crucial to the modeling process in studies such as these is understanding how the models performed. As pointed out in Section 2.1, AUROC is one of the primary metrics used to ascertain how well a model predicted true positives relative to the number of false positives. While the AUROC metric lends itself to many machine learning tasks, in the healthcare milieu, AUROC may or may not be the most appropriate metric to use. Still, the analysis of a screening test performance as it is understood in epidemiology can be thought of in similar terms as machine learning methodology

analysis. Like machine learning, one still must consider *true positives* and *false positives*, along with *true negatives* and *false negatives*. From these, the measures of *sensitivity* and *specificity* can be derived.

Sensitivity: Sensitivity can be defined as the number of true positives relative to all positives (i.e., all patients with some condition). Mathematically, this can be expressed as:

$$\frac{\mathsf{TruePositives}}{\mathsf{TruePositives} + \mathsf{FalseNegatives}}$$

Notice that this is the same measure as *true positive rate*, as defined above in Section 2.1. "Goodness" in terms of sensitivity, then, is a test that captures as many true positives as possible without taking on false negatives (Mausner and Kramer, 1985).

Specificity: Ideally, one would also like for a test to not capture false positives. This is expressed as *specificity*. The mathematical definition of specificity is:

$$\frac{\mathsf{TrueNegatives}}{\mathsf{TrueNegatives} + \mathsf{FalsePositives}}$$

That is, of all the patients without the condition in question, how many were actually labeled as such (Mausner and Kramer, 1985).

These two measures can be better understood from Figure 2.10. In this figure, the left half represents all instances that are actually negative,

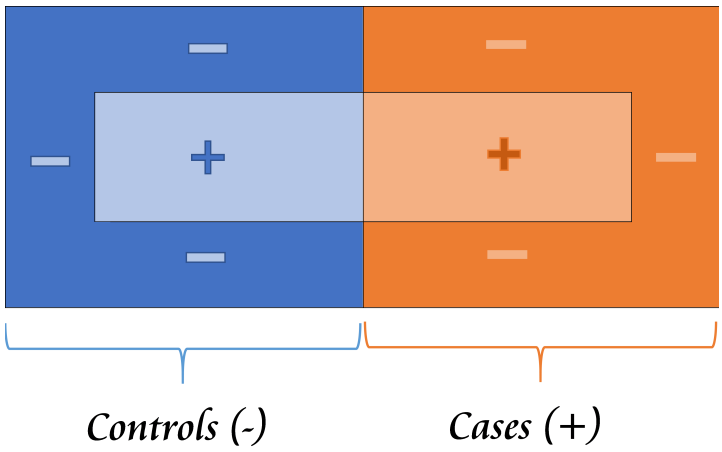


Figure 2.10: True/false positives and true/false negatives used to define sensitivity and specificity.

while the right represents those that are positive. The innermost rectangles both represent instances that were labeled as being from the positive class. Graphically, then, sensitivity would be the light salmon region on the right divided by the whole right side. Specificity, on the other hand, would be the outer dark blue region on the left divided by the whole left side.

Positive Predictive Value (PPV): Another measure of interest to clinicians is positive predictive value (or PPV). It is the ratio of the number of true positives to all patients flagged as positive. Alternatively, one can express it as:

$$\frac{\mathrm{TruePositives}}{\mathrm{TruePositives} + \mathrm{FalsePositives}}$$

This is mathematically equivalent to *precision*, as defined in Section 2.1. It is important to realize that this measure not only accounts for the quality

of the test being used, it is also predicated on how prevalent the condition is. The ideal value for this test is 1 or 100% (Fletcher et al., 2012).

Negative Predictive Value (NPV): Negative predictive value (or NPV) gives the ratio of the number of true negatives to the number of patients flagged as being negative, or:

$$\frac{TrueNegatives}{TrueNegatives + FalseNegatives}$$

As with PPV, NPV's ideal value is 1 or 100%. Its measure is also dependent on how prevalent the condition being measured is (Fletcher et al., 2012).

C-statistic: The C-statistic is another measure of a test's efficacy. It is found by determining the area under the ROC curve; thus, it is synonymous with AUROC in machine learning terminology.

NNT: The NNT (or number needed to treat) metric plays a crucial role in ascertaining treatment effectiveness. It can also be of use in risk stratification tasks. NNT can be thought of as the number of patients to which a treatment must be applied before one patient sees the benefit. Thus, a smaller NNT is desirable. Mathematically, NNT is expressed as:

$$\frac{1}{RelativeRiskReduction * AbsoluteRisk}$$

Here, relative risk reduction is the fraction by which a treatment reduces some condition, while the absolute risk represents the initial portion of some population afflicted by the condition of interest. Later, we show that the absolute risk, as it is presented in the literature, is equivalent to precision.

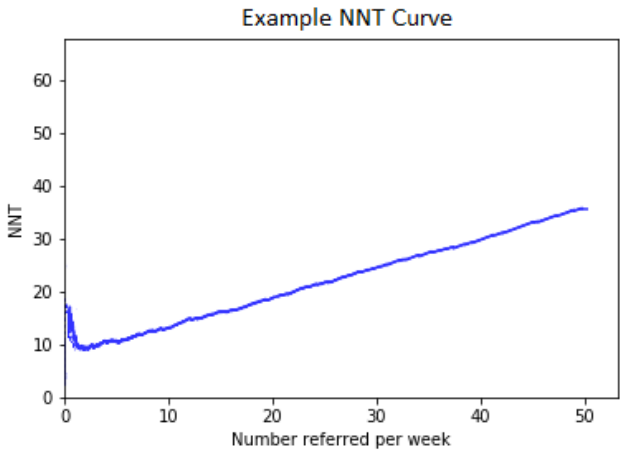


Figure 2.11: A depiction of an NNT curve.

The NNT metric is particularly useful, because it allows for exploring different thresholds in assessing treatment effectiveness for a certain condition. That is, if the relative risk reduction remains fixed, the absolute risk (i.e., the number of patients exhibiting the trait) can be varied to see the how various risk levels translate into effectiveness. To understand this, refer to Figure 2.11. Note that in this example, the treatment being applied is referring patients for follow-up care meant to reduce the likelihood of a condition. Patients are assigned a risk score by an underlying model, and the patients are ranked according to their respective risk. The cutoff is then set to some point along the risk scale, and this determines which patients are referred. As one moves further and further to the right on the NNT graph, more and more patients are referred. Referring more patients in this way results in a higher NNT, since more and more patients are

included that are of lower and lower risk.

We later endeavor to show in Chapters 4 and 5 how NNT can be derived from existing performance measures known to machine learning experts. We further demonstrate how NNT can be shifted into the training phase of the machine learning process, thereby making this alternate performance metric an intrinsic part of a machine learning model being employed.

3 PREDICTING NEURAL TOXICITY

One of the more rudimentary uses of machine learning in the biomedical context is that of genomic analysis. Often, modeling techniques can be combined with other technologies to perform some type of analysis. In the case of genomic analysis, a procedure called RNA-Seq allows for DNA to be used in gene expression analysis, which can include machine learning methods. This chapter explores the marriage of RNA-Seq data with machine learning with an end goal of predicting compounds that are neurotoxins.

The material in this chapter first appeared in Schwartz et al. (2015). Authors Collin Engstrom, David Page, and Vitor Santos Costa provided the machine learning analysis, which is the basis of this chapter. The work pertaining to tissue generation and RNA-Seq analysis was done by the remaining authors. A summary of this RNA-Seq work is interspersed throughout the following sections to provide context for the machine learning component of the project.

3.1 Introduction

In recent years, there has been a pressing need for improved methods to assess the safety of drugs and other compounds (Fabre et al., 2014; Judson et al., 2014; Crofton et al., 2011; Grandjean and Landrigan, 2014; Bal-Price et al., 2015). Success rates for drug approval are declining despite higher R&D spending (Hay et al., 2014), and clinical trials often fail due to toxicities that were not identified through animal testing (Olson et al., 2000). In addition, most of the chemicals in commerce have not been rigorously assessed for safety, despite growing concerns over the potential impact of industrial and environmental exposures on human health (Judson et al., 2014; Crofton et al., 2011; Grandjean and Landrigan, 2014; Bal-Price et al.,

2015). Animal models are costly, time-consuming, and do not recapitulate many aspects of human physiology, which motivated the National Institutes of Health (NIH) and the U.S. Environmental Protection Agency (EPA) to initiate programs that emphasize human cellular approaches for assessing the safety of drugs (Fabre et al., 2014) and environmental chemicals (Judson et al., 2014; Crofton et al., 2011). In vitro cellular models that accurately reflect human physiology have the potential to improve the prediction of drug toxicity early in the development pipeline (Fabre et al., 2014) and would provide a cost effective approach for testing other sources of chemical exposure, including food additives, cosmetics, pesticides, and industrial chemicals (Judson et al., 2014; Crofton et al., 2011; Grandjean and Landrigan, 2014; Bal-Price et al., 2015).

In this study, reproducible neural constructs with vascular and microglial components were fabricated for developmental neural toxicity screening. Machine learning was used to build a predictive model from RNA-Seq data for neural constructs exposed to a training set of 60 toxic and non-toxic chemicals, which then correctly classified 9/10 blinded compounds.

3.2 Background

In this chapter, RNA-Seq is the method used to interrogate gene expression levels. This technique has been in existence since the mid-2000s and has been used in many studies ranging from gene regulation (Trapnell et al., 2013) to viral detection (Radford et al., 2012; Capobianchi et al., 2013; Khoury et al., 2013).

Broadly speaking, RNA-Seq is concerned with the movement of genetic information from the DNA stage to final protein products. For this reason, it has become one of the dominant techniques used to ascertain gene expression levels. This is achieved in a series of steps (Griffith et al., 2015),

which are as follows:

1. Identifying biological sample(s) that should be used for a study;
2. Isolating the RNA in the sample(s);
3. Enriching the RNA;
4. Converting the RNA to cDNA fragments;
5. Aggregating the cDNA fragments into a single library;
6. Sequencing the fragments;
7. Generating a single read from the fragments;
8. Assembling all the reads; and finally
9. Performing a final analysis using the reads.

This process is summarized in Figure 3.1. In this chapter, we focus on the final step, as it is these final reads that yield the gene expression levels that can be used as input features to an SVM model.

3.3 Methods

In this study, RNA-Seq and linear support vector machines were used to build a predictive model for neurotoxicity based on changes in global gene expression by neural constructs exposed to known toxins and non-toxic controls. Neurotoxicity was evaluated using a set of 31 control compounds and 39 toxins with previous literature support for neurotoxicity. Control chemicals included pharmaceuticals with no known neurotoxicity or common food additives. Pluripotent stem cells were used to construct a model for developing brain tissue including seven cell types and vascularization (Schwartz et al., 2015). This process started on day 0. For each compound

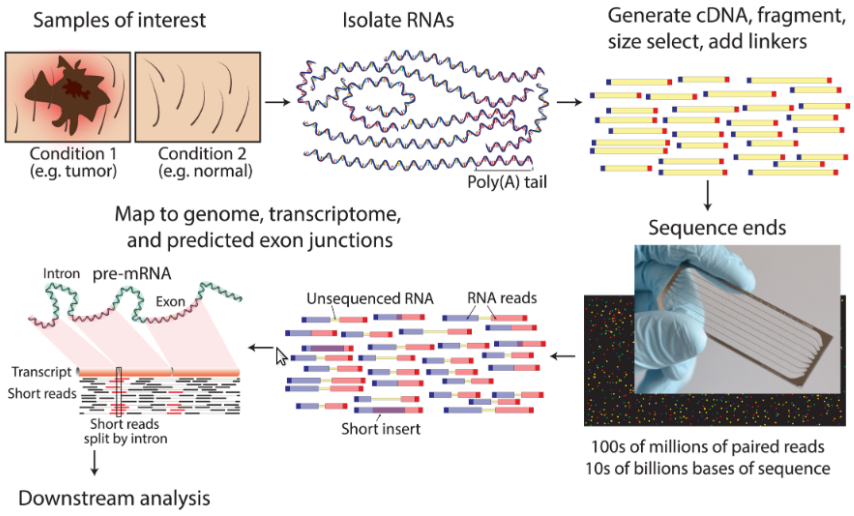


Figure 3.1: A visual depiction of the RNA-Seq process. Figure courtesy of Griffith et al. (2015).

(control or neurotoxin), two replicates were done. For each replicate, day 14 neural constructs were exposed continuously to test compounds through day 16 (two day exposure) or day 21 (seven day exposure), and then harvested for RNA-Seq and machine learning analysis.

The overall machine learning process is summed up by the algorithm below.

Algorithm 3.1 Neural Toxicity Classification

Given: RNA-Seq gene expression measurements for roughly 19K genes on one day or on several different days following exposure to various drugs, together with a neural toxicity label on each drug.

Do: Construct a model based on gene expression data from training set drugs to predict if future drugs are neural toxic.

To better understand this process, one might consider each drug as being a row in a table, with each column representing one of the 19,000 genes (feature inputs to the machine learning model). The last column,

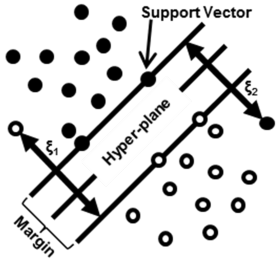
then, would be the predicted status of the test compound; in this case, that would be either "toxic" or "control." This rendering of the problem is shown in Figure 3.2.

Compound	Gene 1	...	Gene 19K	Class
MeHg	3.42	...	5.39	Toxic
Caffeine	2.04	...	4.98	Control

Figure 3.2: Compounds labeled by SVM procedure.

SVMs: As mentioned above, linear SVMs were used for the predictive model. Recall from Chapter 2 that SVMs use a hyperplane to subdivide the two classes of interest. The margin that parallels this hyperplane is maximized on either side so that the buffer between the two classes is as great as possible. A two-dimensional linear support vector machine (SVM) is illustrated in Figure 3.3, where the hyperplane reduces to a line that separates examples (circles) of the two classes (filled or open) and maximizes the margin, or distance between the closest points of different classes (the support vectors are the examples that fix the position and orientation of the hyperplane). The x_is are the examples (circles; gene expression for the current study), the y_is are their labels (filled or open; toxic or non-toxic for the current study), and w is the weight vector, or vector of coefficients on the features (the dimensions). We use soft margin SVMs (Cortes and Vapnik, 1995b) that allow for errors in the training set. The effect of a misclassified example x_j is measured through its distance to the hyperplane, ξ_j. The red portions in the equation are the additions required to support the soft margin SVM (Cortes and Vapnik, 1995b). The SVM is trained to minimize the sum of the margin and errors (weighted by parameter C). The parameter C is tuned with an inner loop of cross-validation, repeated on every fold of the outer, evaluative cross-validation procedure, using only the training data for each fold. As noted in other studies (Struyf

et al., 2008; Hardin et al., 2004), tuning parameters in this way usually yields better results than arbitrarily choosing a single default parameter value, and it gives a much fairer estimate of future performance than does another approach sometimes seen, of performing cross-validation with multiple values of a parameter and "cherry picking" the value that yields the best result. The linear SVM's output is the weight vector w and the other coefficient, b. To make a prediction, the SVM outputs the number $w'x_i - b$, and outputs the label 0 (non-toxic, for our application) if this number is less than 0, and 1 otherwise. While the numerical output does not have a probabilistic interpretation as does the output of logistic regression, it is common to build a logistic regression model with one input variable (the SVM's output) from the same training set to output a probability (probability of toxic), which we do here.



Soft-margin support vector machine (SVM)
Given $\mathbf{x_i}$ for all $1 \leq i \leq \ell$, find $\mathbf{w}$ and b to:

$$\begin{aligned} \min \quad & \tfrac{1}{2} \parallel \mathbf{w} \parallel^2 + \mathbf{C} \textstyle\sum_{\mathbf{i}=1}^{\ell} \xi_{\mathbf{i}} \\ \text{s.t.} \quad & y_i(\mathbf{w}'\mathbf{x_i} - \mathbf{b}) \geq \mathbf{1} - \xi_{\mathbf{i}} \; \forall \mathbf{i} = \mathbf{1} \ldots \ell \\ & \xi_i \geq 0 \end{aligned}$$

Figure 3.3: Linear SVM depiction.

We employed two standard holdout testing methods for evaluation to avoid overly-optimistic prediction of accuracy (Struyf et al., 2008; Hardin et al., 2004; Golub et al., 1999):

1. A nearly unbiased (slightly pessimistic) leave-one-out cross-validation and

2. An unbiased blinded trial with a single hold-out set.

Leave-one-out cross-validation: For leave-one-out cross-validation, there are N data points (compounds) in a training set, and the method proceeds in N steps. In each step, a different data point is held out of the training set, the SVM is trained on the remaining data points, and then it makes its prediction on the held-aside data point. Hence, every data point is a test case exactly once–for a model trained without that data point. Results are aggregated over all the folds, or test cases, to estimate how well the SVM model trained on all the data will perform on a new data point (compound). Because we had two replicates for each compound, on each fold we in fact held out both replicates of a given compound for testing. To do otherwise, leaving one replicate in the training set and the other in the test set, again could give overly-optimistic estimates of future predictive accuracy. So, in effect, we performed "leave one compound out" cross-validation, taking the average of the two output probabilities, from the two replicates, as the probability for the compound being toxic. Finally, we also averaged predictions for each compound from both days 16 and 21 to produce a further improved prediction of toxicity. Again, evaluation of this approach proceeded in the "leave one compound out" fashion.

Using the cross-validation methodology, we can compute the numbers of true positive (toxic) predictions (TP), as well as false positive (FP), true negative (non-toxic, TN), and false negative predictions (FN). From these we can compute accuracy (fraction of predictions that are correct) as well as the following: Sensitivity (true positive rate, or recall; TP/(TP+FN)), specificity (TN/(TN+FP)), and precision (or positive predictive value; TP/(TP+FP)), as well as other metrics such as F-measure and negative predictive value. Nevertheless, all of these metrics depend on not only the model that produces probabilistic predictions for toxicity but also the probability threshold at which we make positive predictions, such as 0.5. Hence, it is common in machine learning and statistical classification to report "thresholdless" curves and or metrics, the most popular being the

receiver operating characteristic (ROC) curve and the area under this curve (AUC). The ROC curve plots true positive rate on the y-axis against the false positive rate ($1 - specificity$) on the x-axis as the threshold is varied. Random uniform guessing produces a diagonal from lower left to upper right corner and AUC of 0.5, while perfect prediction produces a graph that goes up to the upper left corner and then across and AUC of 1.0. The ROC curve is produced by ranking the examples by their predicted probability of being toxic and then varying the threshold. After aggregating over all the folds, the performance estimates were summarized in the form of such an ROC curve.

Blinded trial: In addition to constructing an SVM model, we also aimed to estimate how well the model predicts the developmental neural toxicities of other compounds. Merely reporting its accuracy on the training set would be overly-optimistic. An unbiased method was employed by collecting RNA-Seq data for a set of 10 compounds that were not in the training set but whose neural toxicities were known, and then testing the predictive model on the unknown samples after the model had been constructed and optimized using the training set. This is the blinded trial, so called because the researchers running the SVM do not know the identity of the chemicals, their ground truth labels, or the number of toxic compounds within the blinded set. Predictions were made using a probability threshold of 0.5; while 0.5 is the most common threshold to use, alternatively a threshold could have been chosen based on the training data, using the threshold that would maximize accuracy in the ROC curve above. The toxicity assignment was revealed for the blinded chemicals only after the SVM's predictions with the 0.5 threshold were made.

This unbiased blinded trial uses the predictive model generated from the training set to make predictions on a separate hold-out set, including estimates of accuracy and area under the ROC curve (AUC). The leave-one-

out cross-validation method has lower variance than a single train/test split because it tests on all the compounds of the training set, but is a slightly pessimistic estimate of future performance because each training set is slightly smaller (one less) than the actual training set.

3.4 Results

Leave-one-out cross-validation was used to evaluate neural constructs exposed to a training set of 34 toxins and 26 non-toxic controls. The area under the ROC curves for the training compounds were 0.86 on day 16, 0.88 on day 21, and 0.91 for data averaged from both days. Thus, the SVM produced an estimate for future data of $\geqslant 0.86$ for each day individually and 0.91 using data from two developmental time points. This is graphically depicted in Figure 3.4.

After the testing phase, an unbiased hold-out testing was then used to predict toxicity for a set of ten blinded compounds that were not in the initial training set (5 toxins, 5 non-toxic controls) and were unknown to researchers generating the support vector machine model until after the predictions were made. The averaged days 16 and 21 data was chosen to make predictions using the training set, which produced probabilities for ranking the blinded compounds from most likely to least likely toxic. In addition, we used a threshold of 0.5 to make definitive predictions, labeling every chemical with probability $\leqslant 0.5$ as non-toxic and all others toxic. The area under the ROC curve generated for the ranking of the blinded set was 0.92, which is in agreement with the training experiment. Importantly, all compounds except oleic acid (a false positive) were properly assigned as toxic or non-toxic. Therefore, the accuracy of the prediction on the blinded compound set was 0.9 (9/10 compounds correctly classified).

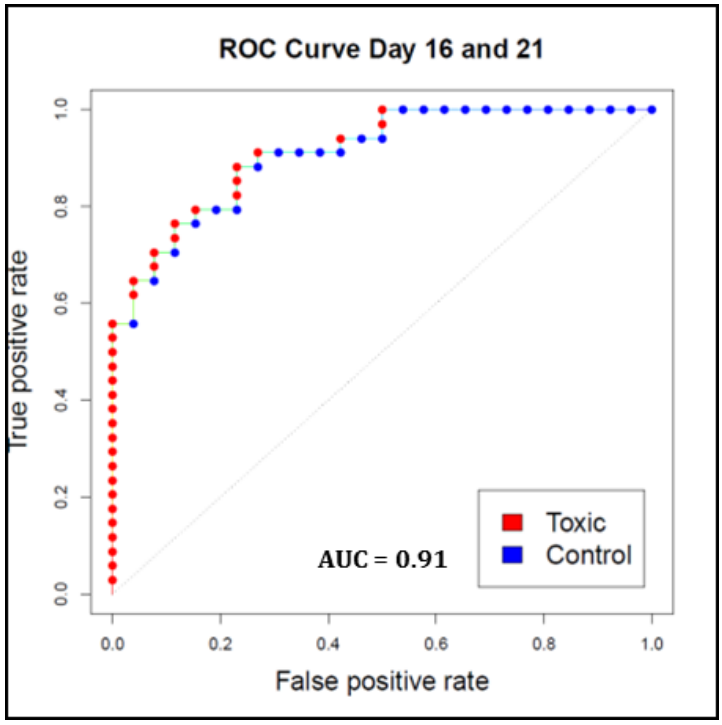


Figure 3.4: ROC for average performance of days 16 and 21.

3.5 Discussion

The high dynamic range for RNA-Seq provides sensitive detection of gene expression changes for cells within the neural constructs, even for relatively rare cell subpopulations, while linear support vector machines have previously been shown to perform well using gene expression data (Struyf et al., 2008; Hardin et al., 2004). The machine learning model correctly identified 9/10 blinded chemicals as toxic or non-toxic (with one false positive), which compares favorably to the expected accuracy when using animal testing to predict human neurotoxicity (Olson et al., 2000).

While it has been demonstrated that accurate machine learning algorithms can be constructed with significantly fewer examples (data points, or training compounds here) than features (variables, or genes here) (Furey

et al., 2000; Struyf et al., 2008; Hardin et al., 2004), 60 compounds is still an exceptionally small dataset given roughly 19,000 genes. Therefore, it is a reasonable expectation that predictive accuracy can be improved further by adding more toxins and controls to the training set. For example, our day 16 model correctly predicts the training compound cadmium to be toxic, but our day 21 model does not. Nevertheless, there are alternative linear separators for day 21 data (with nearly as large margins) that would correctly classify cadmium. Such an alternative linear separator is constructed for the full training set that includes cadmium, such as that used to make predictions for the blinded compounds; this observation supports the assumption that the model would be improved with additional training data. By expanding the training set to include additional compounds with characteristics similar to cadmium, the learning algorithm would construct such an alternative linear separator even if cadmium were held out. Similar improvements might be expected by including other compounds to account for distinct toxic effects that are either underrepresented or not represented at all with the current training set, and incorporation of such information to improve the predictive model is a particular advantage of the machine learning approach.

Machine learning algorithms are also dependent on training compounds that can be definitively assigned. Therefore, initial misclassification of a compound would result in an incorrect prediction even if the machine learning algorithm makes an accurate assessment, such as if a control compound was dosed at a toxic concentration. For example, oleic acid was chosen as a non-toxic control for the blinded set and was dosed at a lower concentration than values reported for human serum (Teubert et al., 2013), but was predicted to be toxic by the machine learning algorithm. It was previously reported that free oleic acid content transiently increases in the brains of postnatal day 1 rats, which was correlated to a neurotrophic role during axonogenesis (Polo-Hernández et al., 2010).

Thus, a potential toxic outcome for oleic acid might be envisioned if the neurotrophic effect is tightly regulated, since elevated expression could disrupt normal developmental timing.

3.6 Conclusion

This chapter has presented an application for machine learning methods in the biomedical setting. As observed in Section 3.1, neural toxic compounds can have a devastating impact on human health, particularly on neural tissue. In this study, RNA-Seq data was harvested from neural constructs exposed to several compounds, some of which were known to be neural toxins and others that are thought to be safe. From this RNA-Seq data, gene expression levels were derived in response to each compound, and an SVM was built on top of these gene expression levels for 19,000 genes.

Our models were able to achieve a final AUROC of 0.91 when expression levels from two days were combined. In this particular study, the end results were in part due to modifications to traditional machine learning approaches. Specifically, we modified the typical leave-one-out cross-validation as presented in Chapter 1 to accommodate for duplicates of each compound, effectively creating a leave-one-compound-out cross-validation strategy. This served to make the model more robust in terms of predicting future compounds. Additionally, an unbiased blind testing phase was included to validate final model performance. Both of these variations exemplify how machine learning techniques must sometimes be *adapted* to suit the needs of a specific application, often in a multi-disciplinary setting.

4 FALLS IN THE ED

As we saw in Chapter 3, machine learning techniques often find a place in predicting characteristics relating to gene expression. In that case, the objective was classifying compounds as toxic or non-toxic to neural constructs. Applications such as these often lead to discoveries that are beneficial to public health.

Another area of exploration with respect to solutions to public health concerns is within the clinical setting. In this chapter, we discuss how machine learning algorithms can aid in predicting at-home falls among the elderly. These models are not only of academic interest. They are currently being transitioned into production, aiming to proactively address at-home falls among elderly patients seen at the UW Hospital Emergency Department (ED).

The work in this chapter is based on research in Patterson et al. (2019). Machine learning analysis was done by authors Collin Engstrom, Varun Sah, David Page, and Brian Patterson. The remaining authors were involved in other aspects of the project relating to dataset curation, background material, etc.

4.1 Introduction

Falls among older adults are a major public health concern, with significant morbidity and mortality (Genuer et al., 2010; Sterling et al., 2001). Despite guidelines (Panel on Prevention of Falls in Older Persons et al., 2011) and quality measures (Centers for Medicare & Medicaid Services and others, 2016), screening for fall risk remains inconsistent in the primary care setting (Phelan et al., 2015; Landis and Galvin, 2014). ED patients are generally at higher risk of outpatient falls than the general population (Carpenter et al., 2009, 2014; Tiedemann et al., 2013), making the ED an im-

portant additional setting to identify high risk patients. While guidelines recommend screening for fall risk in the ED (Weigand and Gerson, 2001; Panel on Prevention of Falls in Older Persons et al., 2011; American Geriatrics Society et al., 2014), this practice has not been widely implemented for many reasons, including the burden of screening in the high-intensity, high-volume ED setting and limited availability of referrals for intervention (Carpenter et al., 2011). Despite previous efforts, no existing screening tools have satisfied the need for a scalable, adaptable, and measurable instrument suitable for widespread implementation (Carpenter and Lo, 2015).

One potential solution to increase screening rates without requiring significant additional resources in the ED is through the development and implementation of an algorithm to screen patients using information present in the electronic health record (EHR) at the time of an ED visit. Recently, healthcare has seen a sharp rise in the implementation of machine learning-derived algorithms for predicting risk across a broad range of clinical scenarios (Goldstein et al., 2017; Churpek et al., 2016; Ting et al., 2017; Li et al., 2016). Often, performance of these algorithms is evaluated by comparing the area under a receiver operating characteristic (ROC) curve, using the terms area under the curve (AUC) or C-statistic, with the concept that algorithms offering superior classification based on AUC are suitable for implementation (Wu et al., 2010; Weng et al., 2017). AUC as a single number may do a poor job of conveying an algorithm's performance for a predictive task in a clinical context which may require a particular balance of sensitivity and specificity (Lobo et al., 2008; Kruppa et al., 2012). Clinicians are generally interested in applying an algorithm to aid in risk stratification for a particular scenario, such as ruling out a rare disease, confirming a particular diagnosis, or reducing population risk via an intervention–in this case, referral for a fall risk reduction intervention.

Such an intervention already exists at our institution in the form of

a multidisciplinary falls clinic. Based on prior literature, we estimate a relative risk reduction of 38% for future falls for patients enrolled in such a program (Close et al., 1999). Currently, very few referrals are made to the falls clinic from the ED. Prior to initiating an automated referral program, decision makers must understand both the anticipated number of referrals generated and the effectiveness of such referrals in preventing future falls. To do so, decision makers may be better served by extrapolations of a model's performance in a given population than by test characteristics such as AUC. This information would allow a clinical site to select the most appropriate risk stratification algorithm, and most appropriate threshold point, to maximize patient benefit within the constraints of available resources and acceptable effectiveness. In this study, we developed several machine learning models to predict six month fall risk after an ED visit. We evaluated these models both using AUC analysis and by interpreting model performance to describe potential clinical trade-offs more concretely in terms of referrals per day and numbers needed to treat (NNT) for prevention of a fall.

4.2 Methods

4.2.1 Study Design and Setting

We performed a retrospective observational study using patient EHR data at a single academic medical center ED with level 1 trauma center accreditation and approximately 60,000 patient visits per year. The goal of developing the models was to create an alert at the time of an ED visit suggesting referral of patients who are at heightened risk of a fall for an existing multidisciplinary falls intervention. In our case, based on discussions with our falls clinic, an estimated 10 referrals per week was seen as operationally feasible. Using the available EHR data, we created risk stratification models for fall revisits to the ED. Our outcome of interest

was a fall visit to the same ED in the 6 months after an index visit (i.e., the first time the patient came to the ED). While this chapter focuses on predicting fall revisits, the methodology we describe is robust and lends itself to any clinical risk stratified prediction task.

4.2.2 Data Selection and Retrieval

EHR data for patients aged 65 years and older who visited the study ED were acquired for a duration of 43 months starting January 2013, with an additional six months of followup data collected for outcome determination. Available EHR features were evaluated for inclusion under the conceptual framework of the Andersen Behavioral Model of Health Services Use, a well-established model which provides a context for characterizing the many factors which lead to healthcare utilization (Aday and Andersen, 1974; Andersen, 1995; Ricketts and Goldsmith, 2005; Andersen et al., 2011). This model has been used to frame numerous prior studies involving ED use and falls among older adults (Stephens et al., 2011; Chatterjee et al., 2012). For each visit, discrete data available within the EHR at the time of the ED visit were collected to create data features including patient demographics, historical visits and visit patterns and diagnoses, as well as visit-specific information including timing, lab tests performed and results thereof, vital signs, chief complaint, and discharge diagnoses. Features were selected based on their availability, clinical relevance, and potential to provide predictive value for fall-revisit risk estimation. Another important criteria for feature selection was to exclude attributes that contained information obtained after an index visit.

The data were organized and analyzed at the level of an ED visit (as opposed to patient level) since our objective was to stratify risk for a fall-revisit based on index visit data alone. Visits by patients who were transferred from other healthcare facilities were rejected as part of our primary exclusion criteria. We excluded visits that resulted in hospital

admissions, as our algorithm would only be implemented for patients who were discharged from the ED. Finally, we excluded patients who did not have a primary care provider (PCP) in our network, as our intervention was specifically aimed towards referring in-network patients. At the end of the exclusion procedures, we were left with 10,030 records.

4.2.3 Feature Preparation

The encoding process for features depended on whether they were numerical or categorical in nature. Numerical features such as age, vital signs during the index ED visit, duration of the index visit, and number of primary care or hospital visits in the six months prior to the index visit were treated as continuous values. For each of the vital signs (e.g., blood pressure, heart rate, respiration rate, and temperature), three features were created: one for the first measurement of the vital sign taken during the visit, one for the last measurement taken, and one final feature for an average of all measurements taken for the sign during the visit. Attributes related to Elixhauser comorbidity index, Hendrich II score, patients' demographics, medications, and lab results were treated as categorical variables. The Elixhauser and Hendrich scores were based on the necessary diagnoses being made either once during an inpatient visit or twice in a six-month period for outpatient visits. In the case of numerical features, we dropped records that had missing values due to the relatively small number of records that were incomplete in this regard, which left us with 9,687 records. However, for categorical variables, missing values were considered as a separate category; in general, the absence of most categorical features could be potentially informative for decision making by the predictive models. At the end of the feature engineering process, we obtained our final dataset which was comprised of 725 features. The feature preparation phase was completely independent of outcome status.

4.2.4 Model Development

Once our features were selected and prepared, we created predictive models from the data. We tested several regression-based methodologies, including thresholded linear regression and logistic regression, both unregularized and including lasso (Tibshirani, 1996) and ridge (Hoerl and Kennard, 1970) penalties. We also included two tree-based methodologies: random forests (Breiman, 2001) and AdaBoost (Schapire and Freund, 1995). Models were generated using the Scikit-learn package in Python (Pedregosa et al., 2011). The dataset created at the end of feature preparation was split into training and test sets in a 3:1 ratio. We split data chronologically, with the final 25% of visits kept as a holdout test set, and the earliest 75% of data retained as a training set. The training set was further split, again chronologically in a 2:1 ratio, to create a tuning set for interim validation.

Models were initially trained on the smaller training set, where tunable parameters were varied using a grid search pattern to achieve best results within the tuning set. Finally, we picked the six models that performed best on the tuning set, and trained one model of each type on the entire training set. These models were then evaluated on the test data that had been held out during the previous phase. Since our dataset was skewed, with more patients who did not fall than those who did, we up-sampled the positive class records while training models to provide a weighting effect to incentivize correct classification of fall cases. This was achieved by randomly duplicating positive cases in the training set until their frequency equaled that of negative cases. Up-sampling was carried out only after the training set had been split into a tuning set, to ensure that no duplicate records created as a result of up-sampling on the entire training set were members of both the training and tuning set. Further, the tuning validation set was not subjected to any up-sampling, to maintain the true population distribution in the evaluation set to simulate performance assessment on

future data.

4.2.5 Model Evaluation

Our initial evaluation of the trained models involved comparing the AUC. 95% confidence intervals were generated in STATA (College Station, TX) using a nonparametric bootstrapping with the Rocreg command and 1,000 iterations (Janes et al., 2009). We then generated classification statistics for each model at each potential threshold value, consisting of performance within the evaluation set in terms of true positives (TP), false positives (FP), true negatives (TN), and false negatives (FN). We were able to use these data to extrapolate both referrals per week and NNT (Cook and Sackett, 1995). Estimated referrals per week were calculated by taking the total percentage of TP and FP results (all patients flagged "positive") at a given threshold from each model and multiplying by the weekly visit volume. NNT was estimated by assuming that the falls reduction clinic would provide a relative risk reduction of 38% (95%CI 21%-52%) based on the results of the PROFET randomized clinical trial which studied a similar intervention in practice and found the percentage of fallers decreased from 52% to 32% in a high risk cohort of patients discharged from the ED (Close et al., 1999). Relative risk reduction and confidence intervals were generated from the reported PROFET data using STATA. The absolute fall risk for a population of patients above a given risk threshold in our models was calculated as the ratio of true positives (patients we predicted would fall who did go on to fall) as compared to all model-identified positives for all patients at or above the risk threshold in the test dataset (TP/TP+FP). This absolute risk was multiplied by the relative risk reduction of 0.38 to estimate an absolute risk reduction, and the inverse of the absolute risk reduction was taken to generate the number needed to treat (Cook and Sackett, 1995). For instance, if the absolute fall risk in the flagged positive group was 60%, the estimated NNT was 1/(0.38* 0.6) = 4.4 referrals per

fall prevented. These projected performance measures were used to create plots that visually described the trade-off between risk reduction gained per referral and number of referrals expected per day.

4.3 Results

We had 32,531 visits to the ED during the study period by adults aged 65 and older, of which 9,687 were both discharged and had a PCP in our network and full numerical data, making up our study population, as shown in Figure 4.1. Within this population, 857 patients returned within 6 months for a fall-related visit; the overall return rate for fall within 6 months was 8.8%. Demographics of patients by outcome are presented in Table 4.1. As compared to patients who did not return for falls, those with falls were similar with regards to gender and insurance status, but were older, more likely to have fallen on their index visit, and more likely to have been brought to the ED by an ambulance.

	All Analyzed Visits	Visits without 180-Day Return for Fall	Visits with 180-Day Return for Fall
N (%)	9687	8830	857
Mean Age (sd)	76.0 (8.4)	75.7 (8.3)	79.3 (8.9)
Female (%)	5863 (60.5%)	5286(59.9%)	577 (67.3%)
White Race (%)	8980 (92.7%)	8187 (92.7%)	793 (92.5%)
Insurance Status			
Medicare	8444 (87.2%)	7705 (87.3%)	739 (86.2%)
Commercial/ Worker's Comp	1210 (12.5%)	1095 (12.4%)	115 (13.4%)
Other/Self Pay	26 (0.3%)	23 (0.3%)	3(0.4%)
Mode of Arrival			
Family or Self	6641 (68.6%)	6263 (70.9%)	378 (44.1%)
EMS or Police	30 (31.4%)	2567 (29.1%)	479 (55.9%)
Fall at Index Visit	1543 (15.9%)	1267 (14.4%)	272 (31.7%)

Table 4.1: Characteristics of analyzed visits.

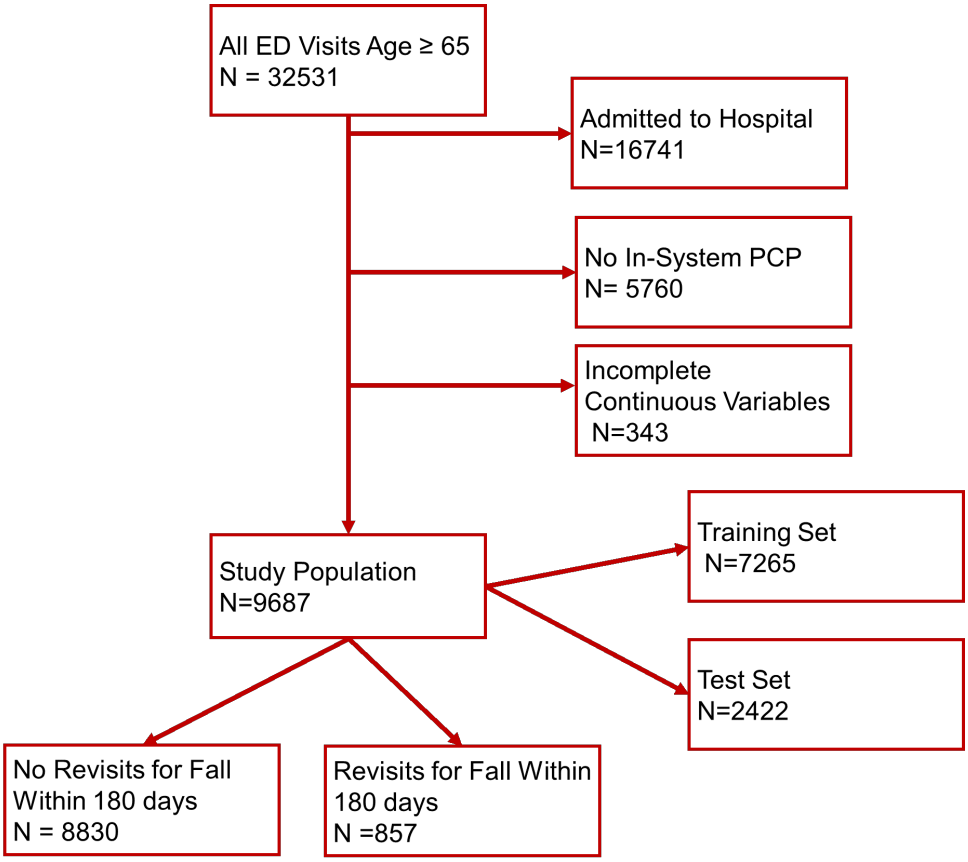


Figure 4.1: Patient allocation.

When comparing models based on AUC, the random forest model achieved an AUC of 0.78 (95%CI 0.74-0.81), and AdaBoost also had an AUC of 0.78 (95%CI 0.74-0.81). These tree-based models were the highest performers, followed by ridge-penalized logistic regression at 0.77 (95%CI 0.73-0.80), lasso-penalized logistic regression at 0.76 (95%CI 0.73-0.80), unpenalized linear regression at 0.74 (95%CI 0.71-0.78), and unpenalized logistic regression at 0.72 (95%CI 0.68-0.76). Figure 4.2 shows AUC plots for all tested machine learning models.

Models were further characterized by estimating both number of referrals per week from the study ED, and NNT of referred patients. Figure 4.3

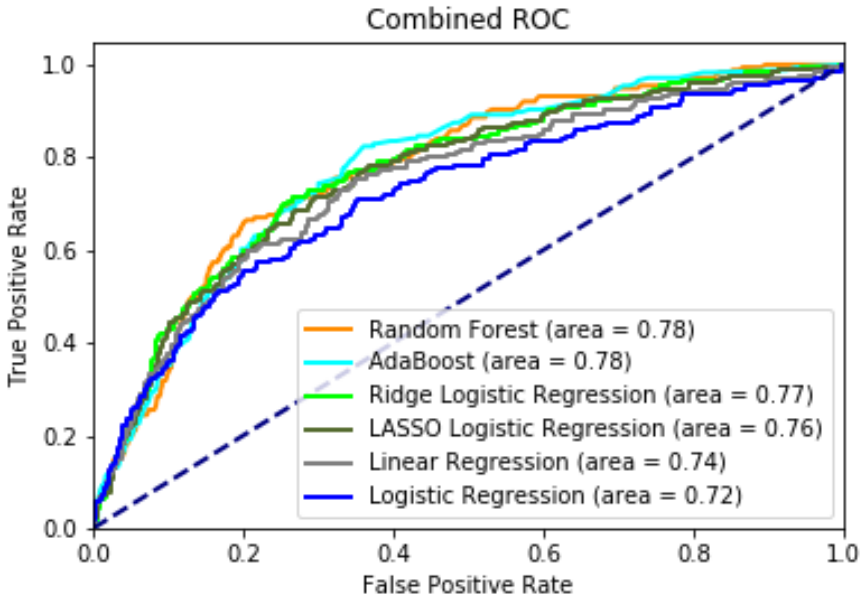


Figure 4.2: Area under Receiver Operating Characteristic Curves (AUC) for models used.

shows these plots. In this analysis, we present the relationship between increasing the number of patients referred, and the decrease in effectiveness per referral as the threshold for defining "high risk" is lowered. The plots additionally contain two fixed points for reference: a "refer all patients" scenario in which all patients are marked as high risk, and a "perfect model" scenario, in which the model predicts with 100% accuracy which patients would go on to fall without the intervention and refers only these patients. In our case, the maximum achievable NNT is 2.6, in the case where a 38% relative risk reduction is applied to a population at 100% risk of falling. Table 4.2 illustrates model performance in terms of predicted NNT at various referrals per week. At the predefined threshold of 10 referrals per week (setting a high risk threshold), the random forest model outperformed the other models, generating an NNT of 12.4. At other thresholds, ridge regression and AdaBoost outperformed the Random

Forest model. The lasso and non-penalized regression models had poorer performance across the spectrum of anticipated referrals.

Referrals per Week Threshold	Random Forest	AdaBoost	Ridge Logit.	Lasso Logit.	Linear Reg.	Logit.
5	12.74	11.94	10.03*	10.70	10.70	11.08
10	12.41*	13.82	13.13	13.13	13.70	14.01
15	15.36	15.49	15.24*	15.75	17.18	17.50
20	18.65	17.40*	18.52	18.38	18.79	20.15
25	21.28	20.76	21.27*	21.71	22.32	22.97
30	23.60*	24.00	24.68	24.52	25.52	26.22
35	26.96*	27.21	27.19	27.02	28.06	28.79
40	29.91	29.44*	29.79	30.14	30.51	31.27

Table 4.2: Model performance at various referrals per week thresholds. Asterisks indicate the best performing model (lowest NNT) at each referral per week threshold.

4.4 Discussion

The various machine learning models tested in this study differed in their ability to predict falls, with the random forest and AdaBoost models offering the best overall performance with an AUC of 0.78. Based on AUC alone, penalized regression-based models including ridge-penalized logistic regression offered similar performance with an AUC of 0.77. This result is consistent with other studies evaluating the performance of tree-based algorithms alongside regression-based methodologies (Kalscheur et al., 2018; Karnik et al., 2012; Philip et al., 2014; Churpek et al., 2016; Li et al., 2016). As opposed to traditional methods, tree-based methodologies have an improved ability to deal with complex variable interactions and nonlinear effects in large databases, which may explain their advantage in these instances (Cairney et al., 2014).

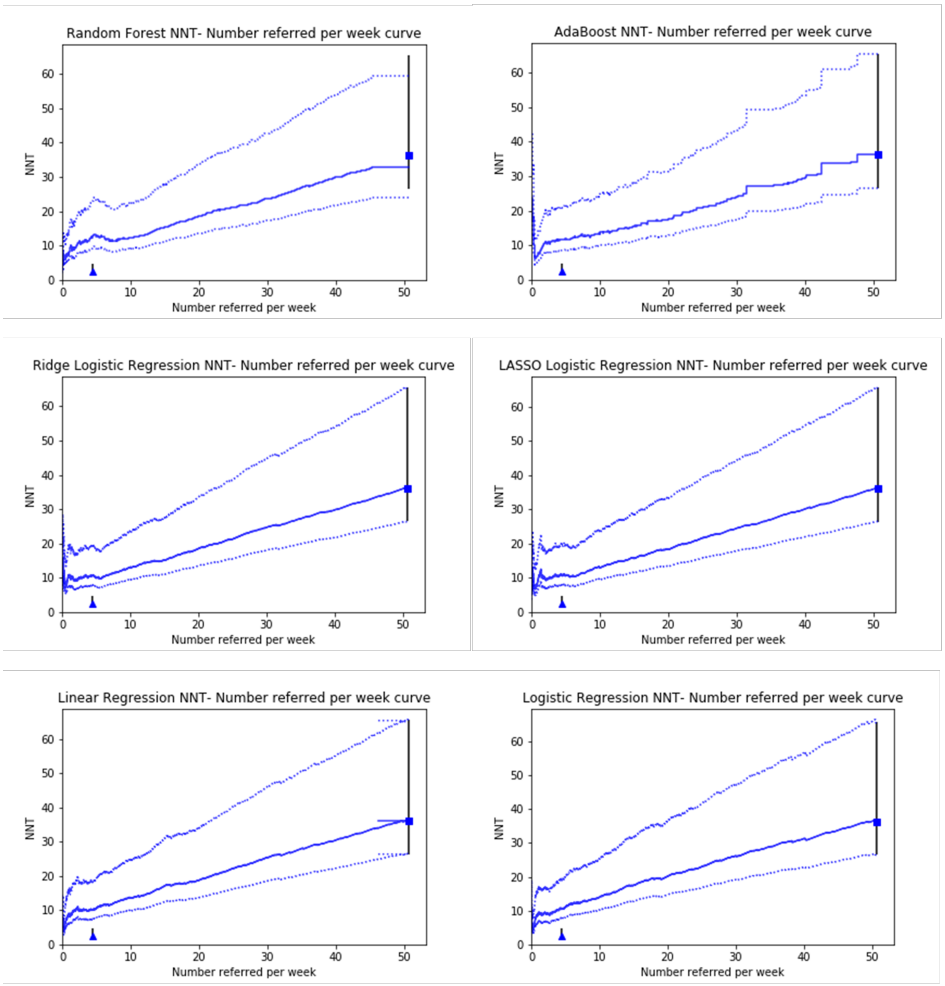


Figure 4.3: NNT vs. Anticipated Referrals per week.

When translating the models into potential deliverable performance at individual thresholds, the random forest-based approach offers the best performance in terms of NNT versus Referrals in the proposed operational scenario, offering the ability to refer 10 patients per week at an NNT of 12.4 referrals to reduce the risk of an ED revisit for fall. While these data

are technically inferable based on the shape of the ROC curves, the degree of distinction between the models would likely not be apparent based on visual inspection alone to a reader not already expert in machine learning or statistics.

Algorithms derived by machine learning have become increasingly common in medicine, with significant excitement surrounding their potential to improve the ability to risk stratify patients (Chen and Asch, 2017; Deo, 2015). Unfortunately, gaps still exist between the ability to predict a potentially avoidable event and specific actionable interventions (Bates et al., 2014). In the majority of studies evaluating machine learning techniques, model performance is reported based on AUC or test characteristics such as sensitivity and specificity (Alanazi et al., 2017). These test characteristics may be useful for establishing predictive performance generally, but may be misleading when not set into clinical context (Lobo et al., 2008). Once AUC curves have been generated for a given risk stratification model in test data, calculating additional information including NNT and anticipated referrals requires only an algebraic transformation of the data, as long as a proposed intervention has been identified along with an estimated effectiveness. The curves generated for this study communicate this trade-off to policymakers, and provide a basis for comparison of anticipated real-world effects of model performance.

For any particular harm-reduction intervention, there is a trade-off when choosing a risk cutoff for referral. The most total harm-reduction would be accomplished by simply referring all patients in a given population; however, such nonspecific referral would be costly in terms of time and resource use, and inefficient as many low risk patients would receive minimal benefit, or potentially be exposed to risks of an intervention. At the same time, selecting only those patients who are at extremely high risk of harm reduces the overall potential benefit of a risk-reduction strategy by not offering it to a large proportion of patients who will go

on to have the outcome of interest. In our example, where a set number of referral slots per week was available, and the task was to select the highest risk patients to fill those slots, the random forest algorithm was the best performer. If there had been only 5 referral spots available, however, the ridge penalized logistic regression model would have been the top performer, despite an overall slightly lower AUC, had better performance in selecting those 5 patients at highest risk, achieving an NNT of roughly 10 vs. roughly 12 for the tree-based models. If the intervention tied to the algorithm were a referral to a less resource-intensive community-based falls prevention program with more availability, policymakers may be looking in a region of higher referrals per week and higher NNT; in this region, model performance was generally similar between the various models.

The projections of performance generated in this study were based on model performance on a set of test data which immediately followed the training data chronologically. While these projections are expected to help policymakers envision potential operational performance, they are not intended to replace evaluation of performance during and after implementation. Machine learning models are tuned to specific population parameters, and subject to calibration drift as patient and data characteristics change over time (Davis et al., 2017), necessitating continued post-implementation monitoring to ensure effective results. To our knowledge, three ED-specific fall screening instruments have been examined: Carpenter et al examined a number of factors for association with future falls, proposing a screen of 4 independent factors, reporting a 4% probability of falling in their lowest risk group and 42% among the highest (Carpenter et al., 2009). Tiedemann et al developed and externally validated a screening instrument with an AUC of 0.70 (Tiedemann et al., 2013), and Greenberg et al utilized a modified CAGE criteria but did not report fall outcomes in their pilot (Greenberg et al., 2013). As compared

to these prior efforts, the machine learning-derived algorithms here offer improved performance in terms of test characteristics, and the advantage of not requiring the devotion of scant ED resources to in-person screening (Bates et al., 2014).

4.4.1 Limitations

When generating our NNT, we assumed that the relative risk reduction generated by our proposed intervention would remain constant across varying absolute risks. This assumption, while broadly made in medical decision making literature, is a simplification that is often, but not always, true (Furukawa et al., 2002; Barratt et al., 2004). Furthermore, for the sake of simplifying our calculations, we assumed that all patients referred for fall intervention would attend the required intervention. If an estimate of likelihood of completed referral were available, it could be taken into account in the NNT calculation.

We presented our NNT vs. Anticipated referrals per week curves with error bars based on the effectiveness estimate from the PROFET trial. PROFET measured the effectiveness of an intervention similar to our own falls clinic, but on a somewhat different outcome (any reported fall vs. ED visit for fall) and with somewhat different inclusion criteria (only selected older adults reporting to the ED for fall as opposed to all older adults). Given the relatively wide confidence interval of the PROFET results, we feel the included error bars provide a reasonable estimate of uncertainty; however, these could be widened to incorporate estimated impact of other sources of potential variation in predicted effectiveness.

During model development, we chose to censor visits which were missing data features encoded as continuous variables. (Categorical variables were encoded to allow a "missing" category.) While the inclusion of only complete records has the potential to introduce bias (Rusanov et al., 2014), only 343 (3%) of records were dropped for incompleteness, suggesting

minimal potential for change in algorithm performance if this data were imputed.

Our approach was based on a per-visit basis. Since many patients visited the ED more than once, there were multiple visits per patient. Since our data was split chronologically into training, tuning, and test sets, there is, therefore, a possibility that the same patient appeared in both the training and testing phases of our study. As discussed in Chapter 3, having two replicates of the same (or very similar, in this case) instance in both a training and test set may introduce overly-optimistic results. We recognize this as one drawback to our per-visit approach that could be mitigated by transitioning to a *per-patient* method.

Our model was trained on an outcome of return visits to our emergency department for falls. Patients who fell may in some instances have presented to other emergency departments, in which case they were not captured by our definition. We limited our analysis to patients with a PCP in our system, and only analyzed patients who presented to our emergency department in an index visit in an attempt to minimize this risk.

4.5 Conclusion

In this analysis, we developed an algorithm which had an AUC of 0.78 for prediction of return visit to the ED for fall within 6 months of an index visit. Placed in the clinical context of harm reduction, this offered the ability to refer 10 patients per week to our fall clinic with a predicted NNT of 12 referrals to reduce the risk of a single fall. Our ability to translate the results of our analysis to the potential trade-off between referral numbers and NNT offers decision makers the ability to envision the effects of a proposed intervention prior to implementation.

For this reason, this study exemplifies the concept of *interpretation* given in Chapter 1. Here, all models performed well in terms of AUC, with a

slight advantage going to the random forest and AdaBoost models. In terms of clinical utility, however, the answer of which model is "best" depends on what is desired. Since the number of referrals per week this study focused on was 10, the random forest's performance once again comes out on top in terms of NNT. On the other hand, if the scenario had dictated a number referred per week value of anything other than 10, 30, or 35, random forests would have been surpassed by one or more of the other models when considering NNT as the objective.

This is not the only take-away point this study illustrates. The models discussed in this chapter are currently being transitioned into production at UW Health. In the course of this transition, the complexity of dealing with models in production has entailed making simplifications to these models. Of particular note is that the set of features was reduced from 725 to 15. This was to alleviate the burden of manually mapping all features between datasets. The real-world concern paired with the simplification made underpins another observation made in Chapter 1: the need to adhere to *translational considerations* when handling models in production.

Finally, transferring these models to production in a healthcare organization has involved collaborating with researchers on the clinical and organizational side to ensure a smooth instantiation of these models (i.e., to ensure that there are no deleterious effects on day-to-day operations of the Emergency Department at the UW Hospital). This is embodied in the principle of observing *workflow* when implementing models.

5 OPTIMIZING ON NNT

Chapter 4 discussed the need for an automatic approach to risk stratifying patients for referral to the UW Mobility and Falls Clinic. Of particular interest was how model performance could be understood in terms of NNT, along with the traditional AUROC metric. With NNT as the performance metric, referring patients among varying risk thresholds is a straightforward task. At the same time, the models themselves were not optimized directly on NNT, but rather on traditional metrics (e.g., hinge loss for SVMs). Ideally, then, what is desired is a means of having an NNT-centric optimization directly "baked in" to the machine learning training process.

This chapter explores one approach to doing this, demonstrating that the precision metric as a focus for optimization simultaneously optimizes NNT. This work is being prepared for submission to JAMIA. As with Chapter 4, machine learning analysis was done by authors Collin Engstrom, Varun Sah, David Page, and Brian Patterson. The remaining authors were involved in other aspects of the project relating to dataset curation, background material, etc.

5.1 Introduction

Recently, healthcare has seen elevated interest in the use of machine learning techniques for the development of algorithms aimed at interpreting existing data from the electronic health record (EHR) for clinical risk prediction tasks. Such machine learning models have been applied in diverse applications, including patient falls (Patterson et al., 2019) and sepsis prediction (Seymour et al., 2019).

In the emergency department, return visits have been utilized as a quality metric in several studies (McCusker et al., 2000; Hu et al., 2012; Pat-

terson et al., 2015, 2016; Jorgensen et al., 2018; Rising et al., 2014). Several interventions have been proposed to reduce return risk among high risk patients (Seaberg et al., 2017; Barksdale et al., 2014). This application is typical of risk stratification for many medical scenarios–a fixed number of resources are available for reducing patient risk of return, and patients must be risk stratified to match these resources to the highest risk patients. In such cases as these, ideally machine learning models will flag those patients who would benefit most from the intervention in question. Typically, these machine learning models used in risk stratification are optimized on some objective function (e.g., hinge loss in the case of support vector machines (SVMs)) (Cortes and Vapnik, 1995a).

While such machine learning modeling techniques have shown great predictive value, significant hurdles exist in bridging the gap between theoretically operant models and clinically effective interventions (Chen and Asch, 2017). Optimization and analysis of machine learning algorithms have, in most cases, focused on traditional metrics of error rate and area under receiver operating curve (AUROC). In many cases, optimizing on alternative metrics may produce results better suited to specific scenarios in which operational and clinical constraints need to be taken into account in addition to overall classification performance.

Number needed to treat (or NNT) (Porta, 2016) is one such metric, and is often used by physicians to communicate the efficacy of a given treatment. NNT can be intuitively thought of as the number of patients to which some intervention must be applied before a positive response is elicited in a single patient. As discussed in Chapter 1, NNT is calculated by the formula:

$$\mathrm{NNT} = \frac{1}{\mathit{AbsoluteRiskReduction}}$$
$$= \frac{1}{\mathit{RelativeRiskReduction} * \mathit{AbsoluteRisk}}$$

where the relative risk reduction factor is the fraction of patients for which an intervention improves on some condition, and the absolute risk is the initial fraction of patients affected by this condition in some population.

From a medical and operational standpoint, characterizing an algorithm's performance based on NNT at a given number of referrals represents an easily understandable projection of interventional effectiveness. Past work has shown that a projected NNT for a clinical intervention can be derived from a model's performance in test data (Patterson et al., 2019). A shortcoming of the models developed in this approach, however, is that the underlying models are optimized on an error function, as machine learning models typically are. While this optimization maximizes overall effectiveness across a range of potential referral thresholds, the metric of interest (NNT at a particular number of referrals per week), is not directly optimized. Ideally, one would optimize directly on NNT. In practice, direct optimization can be achieved by making a slight modification to the modeling process. If we treat the above relative risk reduction for a given intervention as a fixed constant, we need only optimize on absolute risk to optimize on NNT. This absolute risk is calculated based on precision (Patterson et al., 2019). More specifically, since k patients are referred during some time period for some intervention following an initial ED visit, optimizing on precision over k for NNT would be ideal.

In fact, prior research has been done on this metric, called precision@k (or pre@k for short) and, more generally, optimizing on ranking for information retrieval and related tasks (Burges et al., 2007; Xu and Li, 2007;

Järvelin et al., 2000; Taylor et al., 2008; Qin et al., 2010; Joachims, 2005; Boyd et al., 2012; Le et al., 2010; Xu et al., 2008). Much of this work has focused on an SVM approach.

In this study, we assume a theoretical intervention exists to reduce return visits to the emergency department for a set number of high risk patients per month, and we train three models with varying optimization thresholds (precision at 5 predictions, overall precision, and hinge loss) on actual emergency department data to predict patient risk of return. We then compare the performance of models optimized traditionally with those explicitly optimized on precision in a hold out test set at prespecified patient referral thresholds using NNT.

5.2 Methods

5.2.1 Setting and Population

We performed a retrospective observational study using patient electronic health records (EHR) data at a single academic medical center ED with level 1 trauma center accreditation and approximately 60,000 patient visits per year. Patient visits were included in this study if they visited the UW Hospital ED between 1/1/2017 and 1/1/2019 and were 65 years of age or older.

5.2.2 Modeling

Instances in the machine learning model were based on patient visits encountered by those included in the study. A visit was considered to be a case if that patient had a return visit within 60 days of an index visit for any reason. Overall, there were 4050 of these patient visits flagged as being return cases, while 11,079 patients were determined to be controls (i.e., no return visit). Since the outcome of interest was return status

after 60 days, the last 60 days of the data were dropped from the set after all instances were labeled. After labeling, the feature vector used for prediction consisted of 15 fields from the patients' medical records. These included demographic data (e.g., age, sex, and race), as well as information about the visit (e.g., insurance status and mode of arrival). To prevent high-magnitude features from skewing results, continuous features were normalized.

The final patient set was ultimately divided into training and test sets. To ensure results were as resistant as possible to data drift over time, our patient population was chronologically divided into a 75/25 train/test split, with the training set being the older 75% of the patient visits and the test set being the newer 25%. All models were first trained on the same training set before testing on the same held-aside test set. Tuning was also performed on the last 25% of the test set, held aside for parameter tuning.

For all of our models, we assumed a theoretical treatment program that has a relative risk reduction factor of 0.2, and we tested these models at a specific level of k (in this case 5), corresponding to the number of patients on a weekly basis to which our theoretical treatment program could be applied in the ED. We also tested the models over the whole test set.

Since SVMs are one of the best studied examples outside of the medical literature, we used SVM-perf (Joachims, 2005), a variant of the SVM-light package (Joachims, 1999, 2006; Joachims and Yu, 2009), as our experimental model. As opposed to traditional SVMs, SVM-perf can optimize on various metrics like AUROC, precision@k, and others. Since SVM-perf has the ability to optimize on precision@k, it is well suited for the task of simultaneously optimizing NNT. For the baseline approach, we used SVM-light, a package designed to use traditional SVM modeling, based on hinge loss.

To keep training consistent, all hyperparameters in these packages were kept constant, converting them, where necessary, to their equivalent

between packages (Arana-Daniel et al., 2016). Within SVM-perf, we tested on two experimental approaches: precison@5 (i.e., the top 5 highest risk patients per week) and precision@max (or precision on all cases).

For evaluation, ROC curves were constructed and area under the ROC curve (AUROC) was examined on the whole test set for each of these models to show how the models performed on all patients in the test set. For each of the three models, the NNT values were also calculated at varying thresholds (i.e., values of k) for all these models.

5.2.3 Statistical Analysis

ROC: Each of the techniques discussed yielded a list of prediction confidences that revealed how confident the model was that the instance was a case. Ranking these predictions produced a list of most confident to least confident of the instances being positive. We used this list to construct our ROC curves for the whole test set.

NNT: NNT was calculated by selecting the top subset of instances in question and calculating the precision on these flagged instances. The resulting NNT can be calculated from the above formula. This was done for k-values ranging from 4 to 50.

Statistical significance: To test for statistical significance, we sought to calculate an empirical confidence interval. This was done by bootstrapping the dataset 1000 times, each time selecting instances with replacement. After this process was repeated, the NNT and AUC measures from each of the 1000 bootstrap samples were ranked in a list. From this list, the top 2.5% and bottom 2.5% were dropped, leaving the middle 95% for our confidence intervals.

5.3 Results

Visits: Visit data for patients are presented in Table 5.1.

	All Analyzed Visits	Visits without 60-Day Return	Visits with 60-Day Return
N (%)	15129	11079	4050
Mean Age (sd)	75.99 (8.43)	75.6 (8.35)	77.04 (8.56)
Sex			
Female	8997	6631	2366
Other	6132	4448	1684
Race			
White	13943	10254	3689
Other	1186	825	361
Insurance Status			
Medicare/Medicaid	13482	9815	3667
Commercial/ Worker's Comp/ Other	1647	1264	383
Mode of Arrival			
Family or Self	10341	7896	2445
EMS or Police	4752	3155	1597
Police	36	28	8

Table 5.1: Characteristics of analyzed visits.

NNT: Table 5.2 describes NNT at various thresholds by model. When comparing models based on NNT, the precision@5 model outperformed all other models at the specified threshold of 5 patients per week, as well as for k-values of 6 and 7. At higher values of k (8 - 50), the precision@max model offered the highest precision, with all three models having similar precision in the highest k ranges. Standard hinge loss only beat the other two models at the threshold k = 4.

AUROC: Over the whole test set, precision@5 had an AUROC of 0.52 (95%CI 0.38-0.66), hinge loss had an AUROC of 0.59 (95%CI 0.3-0.64), and

k-val	pre@5	pre@max	hinge loss
4	13.59 (95%CI 9.19-24.17)	14.03 (95%CI 9.6-30.35)	13.18* (95%CI 8.42-37.29)
5	11.28* (95%CI 9.51-23.7)	15.42 (95%CI 9.62-30.85)	13.19 (95%CI 9.29-38.02)
6	11.2* (95%CI 9.7-22.53)	13.07 (95%CI 10.0-28.0)	13.42 (95%CI 9.66-36.98)
7	11.74* (95%CI 10.18-21.4)	11.93 (95%CI 10.0-27.59)	13.63 (95%CI 9.96-34.7)
8	12.33 (95%CI 10.1-22.35)	12.0* (95%CI 10.18-28.12)	13.48 (95%CI 9.76-35.82)
9	12.62 (95%CI 10.31-22.27)	11.76* (95%CI 10.46-27.48)	13.61 (95%CI 9.74-34.59)
10	13.08 (95%CI 10.51-22.09)	11.93* (95%CI 10.62-27.25)	13.63(95%CI 10.38-33.37)
11	13.57 (95%CI 10.76-22.47)	12.31* (95%CI 10.93-26.83)	13.41(95%CI 10.67-32.39)
12	14.05 (95%CI 10.8-22.15)	12.44* (95%CI 11.01-27.03)	13.47(95%CI 10.68-32.94)
13	14.55 (95%CI 10.98-22.49)	12.14* (95%CI 11.1-26.07)	13.41(95%CI 10.81-32.95)
14	15.15 (95%CI 11.05-22.54)	12.01* (95%CI 11.21-25.85)	13.82(95%CI 10.76-32.45)
15	15.26 (95%CI 11.21-22.37)	12.1* (95%CI 11.26-26.2)	13.76(95%CI 10.89-32.45)
16	15.61 (95%CI 11.3-22.74)	12.0* (95%CI 11.32-26.02)	13.91 (95%CI 10.94-30.95)
20	16.63 (95%CI 11.55-21.78)	12.26* (95%CI 11.53-25.53)	14.72 (95%CI 11.29-28.92)
25	16.99 (95%CI 12.05-21.44)	12.61* (95%CI 12.1-24.39)	14.81 (95%CI 11.64-27.79)
29	17.45 (95%CI 12.58-21.1)	13.07* (95%CI 12.52-23.28)	14.47 (95%CI 12.24-26.1)
35	16.44 (95%CI 13.43-20.04)	13.64* (95%CI 13.36-21.3)	14.25 (95%CI 13.24-23.35)
40	16.8 (95%CI 14.19-18.93)	14.36* (95%CI 14.21-20.78)	15.13 (95%CI 14.25-21.64)
45	17.42 (95%CI 14.84-18.93)	15.35* (95%CI 14.93-19.63)	15.9 (95%CI 14.9-20.09)
50	17.09 (95%CI 15.59-18.67)	16.11* (95%CI 15.49-18.85)	16.32 (95%CI 15.52-18.93)
all test set	17.18 (95%CI 16.39-18.08)	17.18* (95%CI 16.39-18.08)	17.18 (95%CI 16.39-18.08)

Table 5.2: Model performance at various referrals per week thresholds. Asterisks indicate the best performing model (lowest NNT) at each referral per week threshold.

the precision@max followed with an AUROC of 0.63 (95%CI 0.36-0.63). These results are graphically shown in Figure 5.1.

5.4 Discussion

As outlined above, the process of tailoring machine learning models to clinically relevant scenarios hinges on the ability to use some alternate objective function in the training phase. In our case, NNT and, by extension, precision was the measure of interest. By incorporating this into the training phase, we endeavored to have the training algorithms intrinsically consider NNT when selecting the best parameters for the model. This, ultimately, should theoretically allow for greater performance within a specific clinical scenario. In our hypothetical scenario, 5 slots were assumed to be available for intervention on a weekly basis. As Table 5.2 underlines,

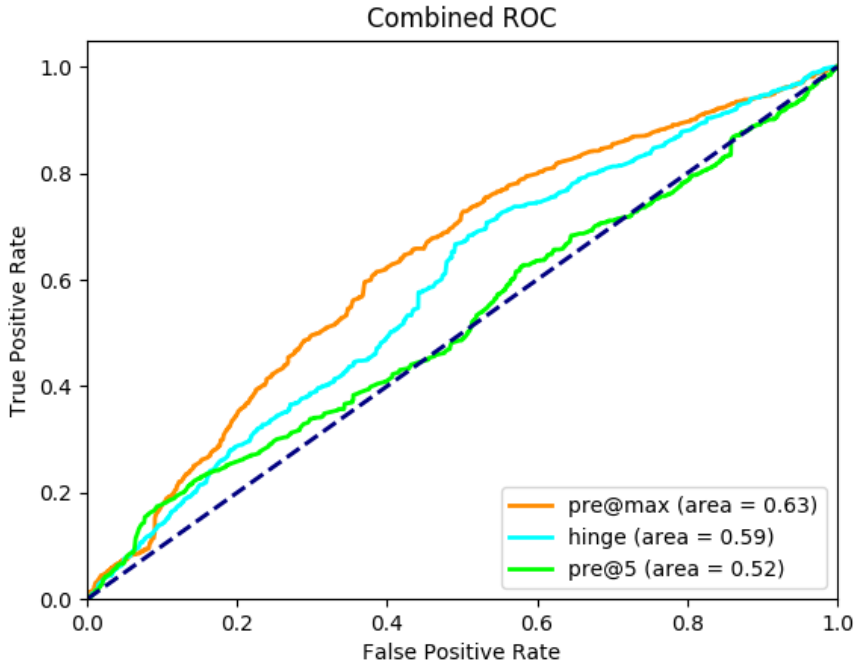


Figure 5.1: Comparison of hinge loss, precision@max (precision over whole test set), and precision@5.

pre@5 outperforms the standard SVM with hinge loss approach and precision@max in NNT at this threshold. Intuitively, this makes sense, because the model was trained to focus on those top 5 most at-risk patients. Since the precision of the model is the primary concern in training, one might reasonably expect a hit in AUROC when compared to the other models. When the models are tested on the whole test set, precision@5 does, in fact, show a lower AUROC than hinge loss and precision@max.

In short, as anticipated, the machine learning models that were optimized on precision demonstrated better performance in NNT values than the traditional machine learning approach on certain intervals of interest. In particular, the precision@5 model did better relative to the other two models in terms of NNT in and around the threshold of k = 5,

the number of patients on which it was optimized. Outside of this range, precision@max beat both precision@5 and the traditional SVM model optimized on hinge loss (with the exception of at $k = 4$), illustrating that optimizing on precision, in general, boosted the NNT for our models. When taking the confidence intervals into consideration, one can see that the lower bounds for the standard SVM with hinge loss model tended to be slightly lower than the precision@5 and precision@max counterparts. On the other hand, though, the upper bounds for hinge loss tended to be much higher than those for the precision models, revealing a much larger variance in hinge loss NNT values.

While machine learning tasks like those cited have been explored in the medical literature for other prediction tasks (Patterson et al., 2019; Seymour et al., 2019; Weng et al., 2017; Deo, 2015; Li et al., 2016; Obermeyer and Emanuel, 2016; Kruppa et al., 2012; Ting et al., 2017), less has been done to adapt machine learning techniques to clinical scenarios as we have done. One case we are aware of where a traditional machine learning algorithm was adapted to a medical context focused on optimizing differential prediction, or cases where one must optimize differently on distinct subgroups in a population (Kuusisto et al., 2014). Additionally, as noted in Chapter 1, another study explored modifications to lasso logistic regression models to consider domain expertise in regularization (Wang et al., 2018). As with the other works, however, NNT was not the target for optimization in either of these two studies. NNT is an attractive target, as it allows clinicians to grasp the utility of the combination of a screening algorithm paired with a resultant intervention (Liu et al., 2019). Optimizing directly on precision@k (which is directly linked to NNT at a given threshold) offers the opportunity to better align the machine learning science with the clinical prediction tasks.

Aiding in this pursuit are techniques that have been developed for optimizing on ranking measures (e.g., precision, precision@k, AUC, etc.).

Historically, this has typically been in the context of search engine queries (Xu and Li, 2007; Järvelin et al., 2000; Taylor et al., 2008; Qin et al., 2010; Boyd et al., 2012; Le et al., 2010; Xu et al., 2008), but they can be adapted to other domains. In our case, results suggest that by varying nothing but the optimization targets, these existing techniques can be extended into the medical domain for risk stratification tasks.

5.4.1 Limitations:

While the precision@5 model tended to outperform the other two on the region for which it was optimized, the overall performance in terms of AUROC was not on par with the other two methods, clocking in at 0.52. The other limitation of our methods are the confidence intervals, which reveal great variance in the performance measures. This could be due to the low AUROC performance on the precision@5 or, potentially, the small number of data points with which the models were trained.

5.5 Conclusion

The application of machine learning models to clinical tasks like risk stratification has seen a sharp increase in recent years. Tasks like anticipating falls among elderly ED patients (Patterson et al., 2019) and sepsis (Seymour et al., 2019) are two instances that have demonstrated promise in automatically flagging patients for further intervention where necessary.

This elevated interested in machine learning models, though, comes with one caveat: the metrics the models have historically been trained on may be less meaningful in a clinical setting. Other metrics like NNT are more readily understood in scenarios where machine learning can aid clinical interventions. In an effort to make the output of machine learning models more accessible, then, one possibility is to change the metric on

which the models are trained (e.g., NNT or precision), thereby unifying what the model returns with what a physician might reasonably expect.

We have shown that one approach to doing so is to use a package that natively supports alternative optimization measures. By training on precision@k, we illustrate how NNT can also be improved in the output.

Ultimately, what this process exemplifies is the need for real-world considerations in modeling. Often, traditional approaches to doing so can be of use, as we saw in Chapter 4. Still other times, however, typical machine learning techniques fall short. In this case, the training phase of machine learning usually does not involve NNT. In actual scenarios, however, it may be desirable to have NNT as objective in optimization. We demonstrate in this chapter how the principle of *adaptation* allows for this. By choosing precision as the metric of interest, NNT is also optimized.

6 DISCUSSION

6.1 Background

In Chapter 4, we were interested in reducing patient falls following visits to the emergency department (ED) for older adults (i.e., 65 years and older). In this scenario, the ED has a set number of potential referral slots for a falls risk reduction clinic. We endeavored to risk stratify patients such that those most at risk would be flagged for intervention. The study was conducted on retrospective data from the UW Health ED, and we trained and optimized six machine learning models to predict fall risk. The performance of the models ranged from an area under the receiver operator characteristic curve (AUROC) of 0.72 for logistic regression to 0.78 for random forests. As was further discussed, models are now being implemented in the ED.

While such machine learning modeling techniques have shown great predictive value, significant hurdles still exist in bridging the gap between theoretically operant models and clinically effective interventions (Chen and Asch, 2017). Moving forward, we plan to investigate open issues relating to *contextual challenges and needs*, *dataset limitations*, *model choice*, and other *general domain challenges*. We discuss these in the following sections.

6.2 Contextual Challenges and Needs

A key question when applying machine learning algorithms to any domain is what the challenges and needs are for the context in which the model will be used. The work presented in Chapters 4 and 5 was focused on the healthcare domain. In this case, clinical needs must drive the machine learning process.

Traditionally, optimization and analysis of machine learning algorithms has tended to focus on metrics of error rate and area under receiver operating curve (AUROC). In many cases, however, optimizing on alternative metrics may produce results better suited to specific scenarios in which operational and clinical constraints need to be taken into account in addition to overall classification performance.

From a medical and operational standpoint, characterizing an algorithm's performance based on NNT at a given number of referrals represents an easily understandable projection of interventional effectiveness. In Chapter 4, we developed methodologies for translating the output of traditionally optimized models into NNTs at given referral thresholds (Patterson et al., 2019). A shortcoming of the models developed in this approach is that they are optimized on traditional objective functions, as machine learning models typically are. While this optimization maximizes overall effectiveness across a range of potential referral thresholds, the metric of interest (NNT at a particular number of referrals per week), is not directly optimized. Ideally, we would optimize directly on NNT, as discussed in Chapter 5. This can be accomplished by treating the relative risk reduction for a given intervention as a fixed constant; then, to optimize on NNT, one need only optimize on absolute risk, which is calculated based on precision (Patterson et al., 2019). More specifically, since k patients are referred per week for intervention following their initial ED visit, optimizing on precision over the k most at-risk patients would be ideal.

In fact, prior research has been done on this metric, called precision@k (or pre@k for short) and, more generally, optimizing on ranking for information retrieval and related tasks (Burges et al., 2007; Xu and Li, 2007; Järvelin et al., 2000; Taylor et al., 2008; Qin et al., 2010; Joachims, 2005; Boyd et al., 2012; Le et al., 2010; Xu et al., 2008). Much of this work has focused on an SVM approach (Joachims, 2005). To encompass other models, we

endeavor to use previous methodologies for precision@k with our fall risk sample. We will start with an SVM approach and further expand the precision@k optimization for other machine learning methods like penalized regression and tree-based methodologies. These latter models are more commonly used and generally better performing based on prior literature in EHR-based risk stratification, as well as our own work in Chapter 4. One way this can be achieved is by modifying the objective function used to optimize a model. Some of these alternative objective functions are summarized in Figure 6.1.

Model Type	SVM	Logistic Regression
Baseline	Standard SVM via SVM-light	Standard logistic regression
Default Objective Function	Hinge loss	Cross entropy
Modification	Use SVM-perf to optimize on precision@k	Use objective function derived in literature for ranking and precision
Hypothesis	NNT with SVM-perf beats SVM-light	NNT with new objective function beats standard logistic regression

Figure 6.1: Comparison of objective functions.

Since SVMs are one of the best studied examples outside of the medical literature, we have used SVM-perf (Joachims, 2005), a variant of the SVM-light package (Joachims, 1999, 2006; Joachims and Yu, 2009), in our work thus far. As opposed to traditional SVMs, SVM-perf can optimize on various metrics like AUROC, precision, pre@k, and others. Since SVM-perf has the ability to optimize on pre@k, it is well suited for the task of simultaneously optimizing NNT. Optimizing on different objective functions for a logistic regression model would entail modifying an existing logistic regression package to use a ranking objective function to accommodate NNT or any other metric of interest.

This approach, however, is specific to SVMs and logistic regression. Ideally, we would like a more general approach that can be used for any

other model (in our case, penalized regression and tree-based methodologies). To accomplish this, we propose creating a wrapper similar to AdaBoost (Schapire and Freund, 1995). As with AdaBoost, we will use a family of weak learners. The difference, however, is that since we are trying to optimize on precision, we specifically want to penalize false positives, instead of prediction error. This is graphically shown in Figure 6.2 in a series of iterations that gradually place more emphasis on incorrectly classified instances. One potential caveat to this approach is that it drives down the number of true positives. To account for this, another constraint like recall must be included to ensure the algorithm does not simply classify as many instances negative as possible.

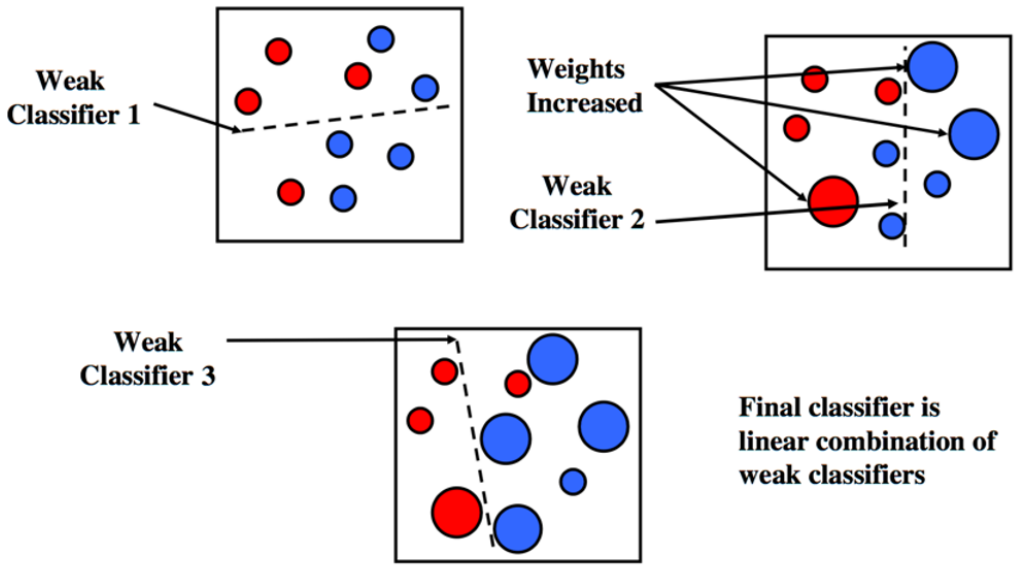


Figure 6.2: AdaBoost visual. Image courtesy of `http://www.vinsol.com`.

6.3 Dataset Limitations

A corollary to Section 6.2 is that with unique domains of applications come unique datasets that must handled appropriately. Often, even within an institution, there may not be a unified view of an underlying dataset. For this reason, feature translation between the differing views may be necessary.

Broadly, this concern of translating models into real-world scenarios has been discussed in the literature (Chen and Asch, 2017; Sendak et al., 2019). This existing work has served to motivate and guide our own research as we move into the production phase. We have obtained operational approval to pilot our work from Chapter 4 within the UW Emergency Department (ED), and are working with operational leadership in the hospital to translate our algorithms from retrospective data to real time operation within our EHR environment. Partnering with the geriatric falls clinic, we aim to operationalize and pilot a screening program incorporating a version of the approach we previously developed (Patterson et al., 2019). This will allow us the opportunity to further investigate the process of translating these algorithms from retrospective data to real time operational functionality, an area with considerable technical challenges not widely explored in the literature.

On the issue of model portability, other groups like the Observational Outcomes Medical Partnership (OMOP) (Stang et al., 2010) and, subsequently, Observational Health Data Sciences (OHDSI) (Hripcsak et al., 2015) have sought to create a common data model (CDM) that lays out a set of standards and best practices across institutions to facilitate the transfer and analysis of EHR-related data. Our approach, however, will focus on model adaptation within the UW Health system between retrospective data warehouses and the transactional database used in real time by the EHR.

As a first step in this, one must determine how features map from

our retrospective dataset derived from EPIC Clarity and stored at the Health Innovation Program (HIP) (i.e., the academic dataset) to those in the "online" environment at UW Hospital, since the variables in these sets differ in nomenclature. This is shown in Figure 6.3. Some variables may also be a mapping derived from some underlying function (e.g., comorbidity scores such as the Center for Medicare & Medicaid Services Hierarchical Condition Category score (Anumula and Sanelli, 2011), which is calculated based on ICD codes in patient charts). We seek a semi-automated process that aids in discovering such mappings. With respect to variable mapping, a technique from databases called schema mapping will likely be useful. In this domain, various metrics of similarity can be used (Bilenko, 2006; Bigi, 2003) to ascertain the likelihood two variables from different datasets are related. We will employ one or more of these as our first step in moving our model to the real time ED environment. (See Figure 6.4.) After these variable mappings have been established, the models can be trained and tested on the online dataset.

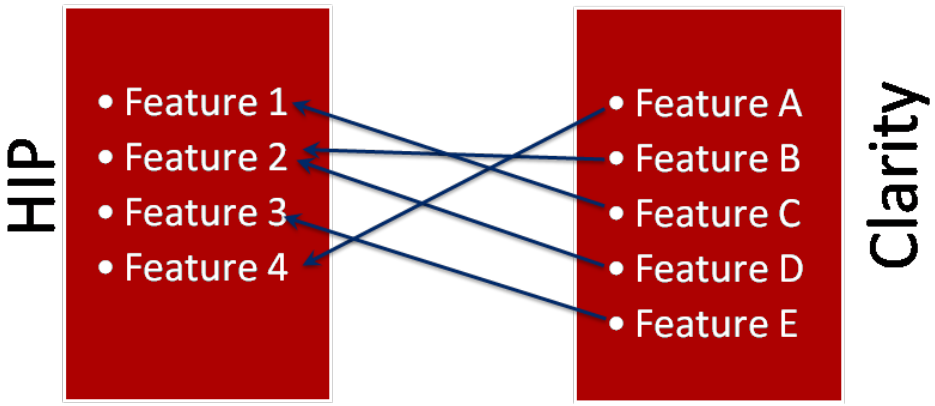


Figure 6.3: HIP features mapped to Clarity.

As a second comparative step, we would like to determine how well

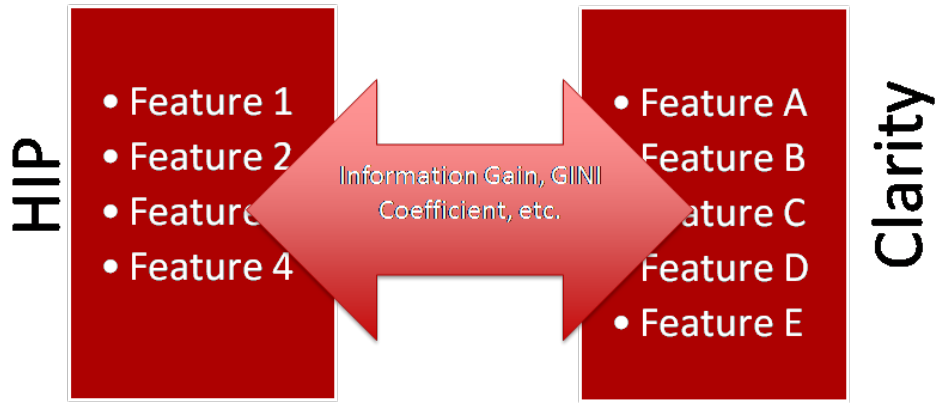


Figure 6.4: Schema mapping used for model translation.

simply retraining models on the new dataset works. This approach is agnostic of any variable mappings and will allow for a de novo selection of features for use in modeling. After training and testing on the (non-restricted set of) variables is done, the end results can be compared to the first approach of a mapped feature set. In the end, this will demonstrate if there is, in fact, any merit in re-selecting variables for model training.

6.4 Model Choice

As mentioned in Section 6.2, the issue of model selection was addressed in that a wrapper-based approach to optimizing NNT would allow for the ability to use any model. One particular advantage of this is that it facilitates using a more transparent model than SVMs–one that may be more appropriate based on the scenario and end users.

When the scenario is of a clinical nature, as discussed in Chapter 2, there are certain models that serve clinicians best. In particular, regression- and tree-based ones have typically seen widespread medical application

due to the issue of *model transparency*, or the ability to intuit how and why features are chosen. We seek to investigate other factors relating to model trustworthiness by clinicians and what role this may play moving forward when less intuitive models (e.g., deep learning) must be employed.

6.5 General Domain Challenges

In this section, other general domain challenges are explored.

6.5.1 Model Drift

In Chapter 4, the issue of model (or concept) drift was alluded to in our choice of how to divide the dataset. To understand this, we note that when models are used over a long period of time, shifts in the population characteristics of the underlying distribution may be observed (Klinkenberg and Joachims, 2000; Klinkenberg, 2003). To account for this, we used a chronological split in our dataset in Chapter 4. This is graphically depicted in Figure 6.5. In this arrangement, the earliest visits fall into the training set (January 2013), while the most recent ones belong to the test set (July 2016). The goal of a chronological split is to capture population shifts that have occurred in a dataset over the time period measured.

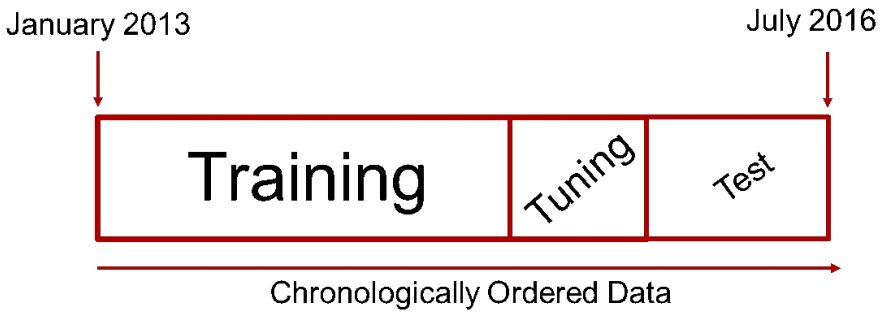


Figure 6.5: Chronological train/tune/test split to account for model drift.

One should observe that, even after drift has been accounted for in the model, the model may still need to be updated, or "refreshed," from time to time as the population changes. In addition to using the chronological split, an open question is with what frequency the models should be updated. These matters can be explored in a production environment like the one at UW Health.

6.5.2 Workflow

Another consideration is that even after models have been trained and optimized, they must be transitioned into production. This process is currently underway for the models described in Chapter 4. As was pointed out, this transition process has included a collaboration between the UW Emergency Department, the Industrial Engineering Department, and the Enterprise Analytics team at UW Health. As part of this, the SEIPS model and PDCA process have been critical to ensuring the day-to-day operations are not disrupted upon model deployment.

In the future, we will endeavor to include the techniques from Sections 6.2 and 6.3 in production as well. This will likely necessitate further PDCA iterations, along with additional consideration of the SEIPS model.

6.6 Conclusion

These issues of *contextual challenges and needs*, *dataset limitations*, *model choice*, and other *general domain challenges* are central concerns to bringing our methodologies to fruition. Moving forward in others' work, along with our own, they will ultimately play a huge role in successful application of machine learning methods.

7 CONCLUSION

This dissertation has explored the practical side of machine learning methods with respect to biomedical and clinical applications. In terms of the former, this has translated into a need for new ways to run established machine learning methodologies; that is, *adapting* old techniques to new scenarios. In terms of clinical applications, we have demonstrated that real-world needs drive machine learning methodologies. Each of these has provided take-away lessons to help guide future work in the healthcare setting.

7.1 Predicting Neurotoxins

In Chapter 3, we explored a biomedical application for linear support vector machines (SVMs): delineating neural toxic compounds from non-toxic ones. The features used for the model were gene expression levels generated by RNA-Seq technology applied to stem cell-derived neural constructs. The machine learning portion of this project focused on achieving high predictive accuracy not only on a held-aside test set, but also on a blinded set of compounds, the identities of which were not known to the machine learning researchers.

Ultimately, the AUROC values for days 16 and 21 were 0.86 and 0.88, respectively. When the data were averaged together for both days, this rose to 0.91. The high predictive accuracy also carried over to the held-aside blind test set, where 90% of the compounds were predicted correctly.

In this study, the modeling process was bolstered by modifying the traditional leave-one-out cross-validation strategy to one that held out *both* replicates for each compound. Since there were 60 compounds, this translated into 60 iterations of training and testing, where each compound was tested once. In this case, doing so may have resulted in more conservative

results, but ultimately, this allowed for greater predictive ability.

7.2 Risk Stratifying ED Patients for Falls

Chapter 4 examined the use case of falls among elderly patients in the UW Hospital's emergency department (ED). As noted in that chapter, falls are of significant public health interest. UW Hospital has a Mobility and Falls Clinic that offers interventions to reduce the likelihood of such a fall event. The limiting step in referring patients is ascertaining *which* patients are at high risk and should be referred. In Chapter 4, we provided an automated means by which elderly patients can be risk stratified for potential referral to the Mobility and Falls Clinic.

From this study, results ranged from an AUROC measure of 0.72 for non-regularized logistic regression to 0.78 for random forests and AdaBoost. Of particular interest to physicians, the number needed to treat (NNT) values were all in the range of 12-14 for the referral region of interest (i.e., 10 patients referred per week). One benefit to using NNT to prioritize referrals is that it allows end users to examine model performance at a given threshold determined by the number of referral slots available. For this reason, should another 5 slots per week become available, one would need only consider an NNT threshold of 15 for the number of referrals.

7.3 Optimizing on NNT

A consequence of using alternate measures like the NNT mentioned in Section 7.2 has been a new push in the medical field toward modifying existing machine learning methodologies to intrinsically consider some alternate optimization target, as opposed to traditional metrics like hinge loss. As demonstrated in Chapter 5, this is entirely achievable. By considering the mathematical expression for NNT, one will see that once the

relative risk reduction is fixed as a constant, the only other parameter to be varied is the absolute risk. The relative risk reduction factor can be found in the literature based on how effective an intervention is; once determined for a given scenario, it becomes a fixed constant. In the case of our hypothetical intervention, this was 0.2. With this fixed, the absolute risk was shown to be TP / (TP + FP), which is congruent to the mathematical formulation for precision. Thus, optimizing on precision was shown to optimize NNT.

Upon implementation, this theory matched empirical results, with the precision-based model outperforming the other methodologies (namely a traditional SVM optimized on hinge loss). This demonstrates that alternative metrics used in medicine can lead to novel ways of performing machine learning.

7.4 Summary of Take-Aways

This dissertation has endeavored to show how machine learning methods and models can be used in the biomedical and clinical environments. Doing so is often advantageous for researchers in these fields, but it raises unique challenges that must be overcome for successful results. As summarized in the three sections above, we have explored machine learning in the context of neurotoxin prediction, risk stratification for patient falls in the Emergency Department at UW Health, and finally in optimizing on NNT as an objective. In this section, we summarize the lessons learned from these applications.

7.4.1 Model Choice

One of the most important decisions a researcher must make is what model to use for the given problem. While many factors must be considered in

this choice, a model must ultimately be selected based on the nature of research-related desiderata.

As discussed in Chapter 2, tree-based methods carry the benefit of better capturing non-linear relationships in the data than other models. Another benefit of this type of model is the ease of interpretation. These properties make tree-based methods an attractive choice for a physician who is interested in the transparency of what features a model is selecting and how it is selecting them. In the end, our results in Chapter 4 showed the top performing models to be AdaBoost and random forests, both tree-based methodologies.

Chapter 5 also touches on the concept of model choice. In this case, the trend toward clinical use of machine learning models has raised the issue of how existing machine learning techniques may be modified to incorporate metrics from outside of machine learning. For this project, SVMs and modified SVMs were chosen, because SVMs constitute a family of models that have a solid basis in the literature and tend to do well in prediction tasks like this one.

7.4.2 Translational Considerations

While real-world needs drive initial model choice, an important part of using machine learning models is knowing what the *translational considerations* are. This was epitomized by the work in Chapter 4.

In this chapter, we noted that the feature set used in our study of at-home falls risk was prepared by researchers on the academic end of the UW Health–SMPH partnership. (I.e., they came from a dataset that had been cleaned, and the cleaning steps were not readily available.) The academic dataset could not be used at UW Health, due to the production data not matching the academic data. For these reasons, the implementation of our models proceeded by generating features de novo at UW Health's end. Aiding us was the discovery that using a pared-down model (i.e.,

15 features) did not cause a dramatic hit to performance, at least on the academic side. While we found similar AUC performance measures for these "parsimonious" models in production, the top-performing methods were now the regression-based ones.

In this way, the process of model translation demonstrates that even within the same institution, there may be disparate datasets and results. Often, to get a working final model, simplifications must be made. In short, this illustrates how central *translational considerations* are to the machine learning application process.

7.4.3 Model Placement in the Workflow

Our first two principles deal with model design and implementation. Once a final version of a model has been constructed, the issue arises as to how the model will be situated in the organization's operations.

The models developed in Chapter 4 are now in the process of being pushed into production. As we have begun to implement the models at UW Health, we have been aware of the role they will play in the workflow. This role can be better understood if contextualized by the PDCA cycle discussed in Chapter 2. The placement and results of these models, if properly understood and used, should lead to an improved quality of care for patients.

7.4.4 Interpreting Machine Learning Models

Of paramount importance to the workflow step (expounded on in the last section) is that researchers do not draw incorrect conclusions from models. In order to observe this caveat, clinicians may also need to expand on metrics traditionally used by machine learning.

This was the case in Chapter 4, where the objective was to predict patient falls following a visit to the UW Hospital Emergency Department.

In this scenario, machine learning models were used to first assign a risk score to each patient. As discussed in Section 7.2, NNT (a clinical measure) was ultimately used for the final risk stratification. For this task, using AUC would not have made as much sense from the standpoint of trying to refer patients for follow-up care. This shows how crucial a role correct metric choice and *model interpretation* play in applying machine learning models.

7.4.5 Adapting Machine Learning Models

Correct interpretation and understanding of models is of paramount importance, but what about instances where existing methodologies do not suffice? In these cases, one must be prepared to *adapt machine learning models and methodologies*.

One incarnation of this principle was noted from work in Chapter 4 and became the impetus behind Chapter 5. As we demonstrated, traditional error functions (like hinge loss for SVMs) need not be the only objective. Instead, one can optimize on precision to generate final results that bolster other metrics like NNT.

The concept of *adaptation* was also explored in Chapter 3 in modifying the typical leave-one-out cross-validation strategy to leave-one-compound-out. For this application, to leave out only one of the two replicates of a compound would produce overly optimistic results.

In both of these cases, we see a need to exercise flexibility when leveraging existing techniques. One must carefully think through the problem being explored and determine how best to proceed. Often, well-established methodologies can serve as a starting point for further adaptations that better suit the needs of the research being done.

REFERENCES

Abuhasel, Khaled A, Abdullah M Iliyasu, and Chastine Fatichah. 2015. A combined AdaBoost and NEWFM technique for medical data classification. In *Information science and applications*, 801–809. Springer.

Aday, Lu Ann, and Ronald Andersen. 1974. A framework for the study of access to medical care. *Health services research* 9(3):208–220.

Adegoke, Vincent F, Daqing Chen, Ebad Banissi, and Safia Barikzai. 2017. Prediction of breast cancer survivability using ensemble algorithms. In *2017 international conference on smart systems and technologies (SST)*, 223–231. IEEE.

Adelaine, S, F Liao, and MA Smith. 2019. Predictive models: A toolkit to guide implementation. `http://www.hipxchange.org/ImplementPredictiveModels`.

Alam, Md Zahangir, M Saifur Rahman, and M Sohel Rahman. 2019. A random forest based predictor for medical data classification using feature ranking. *Informatics in Medicine Unlocked* 15:100180.

Alanazi, Hamdan O, Abdul Hanan Abdullah, and Kashif Naseer Qureshi. 2017. A critical review for developing accurate and dynamic predictive models using machine learning methods in medicine and health care. *Journal of medical systems* 41(4):69.

American Geriatrics Society, Geriatric Emergency Department Guidelines Task Force, and American College of Emergency Physicians and Emergency Nurses Association. 2014. Geriatric emergency department guidelines.

Anand, Rajsavi S, Paul Stey, Sukrit Jain, Dustin R Biron, Harikrishna Bhatt, Kristina Monteiro, Edward Feller, Megan L Ranney, Indra Neil

Sarkar, and Elizabeth S Chen. 2018. Predicting mortality in diabetic ICU patients using machine learning and severity indices. *AMIA Summits on Translational Science Proceedings* 2018:310.

Andersen, Ronald M. 1995. Revisiting the behavioral model and access to medical care: does it matter? *Journal of health and social behavior* 1–10.

Andersen, Ronald M, Thomas H Rice, and Gerald F Kominski. 2011. *Changing the US health care system: key issues in health services policy and management*. John Wiley & Sons.

Anumula, N, and PC Sanelli. 2011. Physician quality reporting system. *American Journal of Neuroradiology* 32(11):2000–2001.

Arana-Daniel, Nancy, Alberto A Gallegos, Carlos López-Franco, Alma Y Alanís, Jacob Morales, and Adriana López-Franco. 2016. Support vector machines trained with evolutionary algorithms employing kernel adatron for large scale classification of protein structures. *Evolutionary Bioinformatics* 12:EBO–S40912.

Bader, Mary Kay, Sylvain Palmer, Connie Stalcup, and Thomas Shaver. 2002. Using a FOCUS-PDCA quality improvement model for applying the severe traumatic brain injury guidelines to practice: process and outcomes. *Worldviews on Evidence-based Nursing presents the archives of Online Journal of Knowledge Synthesis for Nursing* 9(1):97–100.

Baker, Alastair. 2001. Crossing the quality chasm: a new health system for the 21st century.

Bal-Price, Anna, Kevin M Crofton, Marcel Leist, Sandra Allen, Michael Arand, Timo Buetler, Nathalie Delrue, Rex E FitzGerald, Thomas Hartung, Tuula Heinonen, et al. 2015. International STakeholder NETwork (ISTNET): creating a developmental neurotoxicity (DNT) testing road map for regulatory purposes.

Ball, John R, and Erin Balogh. 2016. Improving diagnosis in health care: highlights of a report from the national academies of sciences, engineering, and medicine. *Annals of internal medicine* 164(1):59–61.

Ban, Hyo-Jeong, Jee Yeon Heo, Kyung-Soo Oh, and Keun-Joon Park. 2010. Identification of type 2 diabetes-associated combination of SNPs using support vector machine. *BMC genetics* 11(1):26.

Barksdale, Aaron, Jeff Hackman, Aaron Bonham, and Matt Gratton. 2014. Cardiology clinic follow-up did not decrease return visits to the ED for chest pain patients. *The American journal of emergency medicine* 32(10): 1208–1211.

Barratt, Alexandra, Peter C Wyer, Rose Hatala, Thomas McGinn, Antonio L Dans, Sheri Keitz, Virginia Moyer, et al. 2004. Tips for learners of evidence-based medicine: 1. Relative risk reduction, absolute risk reduction and number needed to treat. *Cmaj* 171(4):353–358.

Bates, David W, Suchi Saria, Lucila Ohno-Machado, Anand Shah, and Gabriel Escobar. 2014. Big data in health care: using analytics to identify and manage high-risk and high-cost patients. *Health Affairs* 33(7):1123–1131.

Bennett, James, Stan Lanning, et al. 2007. The netflix prize. In *Proceedings of KDD cup and workshop*, vol. 2007, 35. New York, NY, USA.

Benson Edwin Raj, S., and A. Annie Portia. 2011. Analysis on credit card fraud detection methods. In *2011 international conference on computer, communication and electrical technology (ICCCET)*, 152–156.

Bigi, Brigitte. 2003. Using Kullback-Leibler distance for text categorization. In *European Conference on Information Retrieval*, 305–319. Springer.

Bilenko, Mikhail Yuryevich. 2006. Learnable similarity functions and their application to record linkage and clustering. Ph.D. thesis.

Boyd, Stephen, Corinna Cortes, Mehryar Mohri, and Ana Radovanovic. 2012. Accuracy at the top. In *Advances in neural information processing systems*, 953–961.

Breiman, L, J Friedman, R Olshen, and C Stone. 1984. Classification and regression trees–crc press. *Boca Raton, Florida*.

Breiman, Leo. 2001. Random forests. *Machine learning* 45(1):5–32.

Burges, Christopher J, Robert Ragno, and Quoc V Le. 2007. Learning to rank with nonsmooth cost functions. In *Advances in neural information processing systems*, 193–200.

Burges, Christopher J.C. 1998. A tutorial on support vector machines for pattern recognition. *Data Mining and Knowledge Discovery* 2(2):121–167.

Cairney, John, Scott Veldhuizen, Simone Vigod, David L Streiner, Terrance J Wade, and Paul Kurdyak. 2014. Exploring the social determinants of mental health service use using intersectionality theory and CART analysis. *J Epidemiol Community Health* 68(2):145–150.

Capobianchi, MR, E Giombini, and G Rozera. 2013. Next-generation sequencing technology in clinical virology. *Clinical Microbiology and Infection* 19(1):15–22.

Carayon, Pascale, A Schoofs Hundt, BT Karsh, Ayse P Gurses, CJ Alvarado, M Smith, and P Flatley Brennan. 2006. Work system design for patient safety: the SEIPS model. *BMJ Quality & Safety* 15(suppl 1):i50–i58.

Carpenter, Christopher R, Michael S Avidan, Tanya Wildes, Susan Stark, Susan A Fowler, and Alexander X Lo. 2014. Predicting geriatric falls following an episode of emergency department care: a systematic review. *Academic emergency medicine* 21(10):1069–1082.

Carpenter, Christopher R, Richard T Griffey, Susan Stark, Craig M Coopersmith, and Brian F Gage. 2011. Physician and nurse acceptance of technicians to screen for geriatric syndromes in the emergency department. *Western Journal of Emergency Medicine* 12(4):489–495.

Carpenter, Christopher R, and Alexander X Lo. 2015. Falling behind? Understanding implementation science in future emergency department management strategies for geriatric fall prevention. *Academic emergency medicine: official journal of the Society for Academic Emergency Medicine* 22(4):478–480.

Carpenter, Christopher R, Mark D Scheatzle, Joyce A D'Antonio, Paul T Ricci, and Jeffrey H Coben. 2009. Identification of fall risk factors in older adult emergency department patients. *Academic emergency medicine* 16(3): 211–219.

Centers for Medicare & Medicaid Services, and others. 2016. Physician quality reporting system (PQRS) measures groups specifications manual. 2016.

Chatterjee, Satabdi, Hua Chen, Michael L Johnson, and Rajender R Aparasu. 2012. Risk of falls and fractures in older adults using atypical antipsychotic agents: a propensity score–adjusted, retrospective cohort study. *The American journal of geriatric pharmacotherapy* 10(2):83–94.

Chen, Jonathan H, and Steven M Asch. 2017. Machine learning and prediction in medicine–beyond the peak of inflated expectations. *The New England journal of medicine* 376(26):2507–2509.

Chen, Tianqi, and Carlos Guestrin. 2016. XGBoost: A scalable tree boosting system. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, 785–794. KDD '16, New York, NY, USA: ACM.

Chen, Yao, Xiao Wang, Yonghan Jung, Vida Abedi, Ramin Zand, Marvi Bikak, and Mohammad Adibuzzaman. 2018. Classification of short single-lead electrocardiograms (ECGs) for atrial fibrillation detection using piecewise linear spline and XGBoost. *Physiological Measurement* 39(10): 104006.

Churpek, Matthew M, Trevor C Yuen, Christopher Winslow, David O Meltzer, Michael W Kattan, and Dana P Edelson. 2016. Multicenter comparison of machine learning methods and conventional regression for predicting clinical deterioration on the wards. *Critical care medicine* 44(2):368–374.

Clegg, Alex, Collin Engstrom, Gwen Jacobsohn, Manish Shah, Maureen Smith, and Brian Patterson. Using electronic health record data and machine learning to predict outpatient falls after emergency department visits. Pending submission.

Close, Jacqueline, Margaret Ellis, Richard Hooper, Edward Glucksman, Stephen Jackson, and Cameron Swift. 1999. Prevention of falls in the elderly trial (PROFET): a randomised controlled trial. *The Lancet* 353(9147): 93–97.

Colombet, Isabelle, Alan Ruelland, Gilles Chatellier, François Gueyffier, Patrice Degoulet, and Marie-Christine Jaulent. 2000. Models to predict cardiovascular risk: comparison of CART, multilayer perceptron and logistic regression. In *Proceedings of the AMIA symposium*, 156–160. American Medical Informatics Association.

Cook, Richard J, and David L Sackett. 1995. The number needed to treat: a clinically useful measure of treatment effect. *Bmj* 310(6977):452–454.

Cortes, Corinna, and Vladimir Vapnik. 1995a. Support-vector networks. *Machine Learning* 20(3):273–297.

———. 1995b. Support-vector networks. *Machine learning* 20(3):273–297.

Craven, Mark, Andrew McCallum, Dan PiPasquo, Tom Mitchell, and Dayne Freitag. 1998. Learning to extract symbolic knowledge from the world wide web. Tech. Rep., Carnegie-Mellon Univ. Pittsburgh PA School of Computer Science.

Crofton, Kevin M, William R Mundy, Pamela J Lein, Anna Bal-Price, Sandra Coecke, Andrea EM Seiler, Holger Knaut, Leonora Buzanska, and Alan Goldberg. 2011. Developmental neurotoxicity testing: recommendations for developing alternative methods for the screening and prioritization of chemicals. *ALTEX-Alternatives to animal experimentation* 28(1):9–15.

Davis, Sharon E, Thomas A Lasko, Guanhua Chen, Edward D Siew, and Michael E Matheny. 2017. Calibration drift in regression and machine learning models for acute kidney injury. *Journal of the American Medical Informatics Association* 24(6):1052–1061.

Deming, WE, Productivity Quality, and MIT Competitive Position. 1986. MIT, center for advanced engineering study.

Deo, Rahul C. 2015. Machine learning in medicine. *Circulation* 132(20): 1920–1930.

Devi, M Renuka, and J Maria Shyla. 2016. Analysis of various data mining techniques to predict diabetes mellitus. *International Journal of Applied Engineering Research* 11(1):727–730.

Dietterich, Thomas G. 2000. An experimental comparison of three methods for constructing ensembles of decision trees: bagging, boosting, and randomization. *Machine learning* 40(2):139–157.

Dimitrakopoulos, Georgios N, Aristidis G Vrahatis, Vassilis Plagianakos, and Kyriakos Sgarbas. 2018. Pathway analysis using XGBoost classification in biomedical data. In *Proceedings of the 10th hellenic conference on artificial intelligence*, 46. ACM.

Engstrom, Collin, Frank Liao, David Page, Varun Sah, Manish Shah, Maureen Smith, and Brian Patterson. Optimizing machine learning models for clinical applications. Under submission.

Fabre, Kristin M, Christine Livingston, and Danilo A Tagle. 2014. Organs-on-chips (microphysiological systems): tools to expedite efficacy and toxicity testing in human tissue. *Experimental biology and medicine* 239(9): 1073–1077.

Fletcher, Robert H, Suzanne W Fletcher, and Grant S Fletcher. 2012. *Clinical epidemiology: the essentials*. Lippincott Williams & Wilkins.

Flynn, Allen J, Charles P Friedman, Peter Boisvert, Zachary Landis-Lewis, and Carl Lagoze. 2018. The knowledge object reference ontology (KORO): a formalism to support management and sharing of computable biomedical knowledge for learning health systems. *Learning Health Systems* 2(2): e10054.

Freund, Yoav, and Robert E Schapire. 1997. A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of computer and system sciences* 55(1):119–139.

Friedman, Jerome, Trevor Hastie, and Rob Tibshirani. 2010. Regularization paths for generalized linear models via coordinate descent. *Journal of statistical software* 33(1):1–22.

Friedman, Jerome, Trevor Hastie, Robert Tibshirani, et al. 2000. Additive logistic regression: a statistical view of boosting (with discussion and a rejoinder by the authors). *The annals of statistics* 28(2):337–407.

Furey, Terrence S, Nello Cristianini, Nigel Duffy, David W Bednarski, Michel Schummer, and David Haussler. 2000. Support vector machine classification and validation of cancer tissue samples using microarray expression data. *Bioinformatics* 16(10):906–914.

Furukawa, Toshiaki A, Gordon H Guyatt, and Lauren E Griffith. 2002. Can we individualize the 'number needed to treat'? An empirical study of summary effect measures in meta-analyses. *International journal of epidemiology* 31(1):72–76.

Genuer, Robin, Jean-Michel Poggi, and Christine Tuleau-Malot. 2010. Variable selection using random forests. *Pattern Recognition Letters* 31(14): 2225–2236.

Goldstein, Benjamin A, Ann Marie Navar, Michael J Pencina, and John Ioannidis. 2017. Opportunities and challenges in developing risk prediction models with electronic health records data: a systematic review. *Journal of the American Medical Informatics Association* 24(1):198–208.

Golub, Todd R, Donna K Slonim, Pablo Tamayo, Christine Huard, Michelle Gaasenbeek, Jill P Mesirov, Hilary Coller, Mignon L Loh, James R Downing, Mark A Caligiuri, et al. 1999. Molecular classification of cancer: class discovery and class prediction by gene expression monitoring. *Science* 286(5439):531–537.

Gómez-Ríos, Anabel, Julián Luengo, and Francisco Herrera. 2017. A study on the noise label influence in boosting algorithms: AdaBoost, GBM and XGBoost. In *Hybrid artificial intelligent systems*, ed. Francisco Javier Martínez de Pisón, Rubén Urraca, Héctor Quintián, and Emilio Corchado, 268–280. Cham: Springer International Publishing.

Grandjean, Philippe, and Philip J Landrigan. 2014. Neurobehavioural effects of developmental toxicity. *The lancet neurology* 13(3):330–338.

Greenberg, M, Michael Nguyen, B Porter, R Barracco, Brian Stello, A Goldberg, C Lenhart, A Kurt, and B Kane. 2013. 298 modified CAGE as a screening tool for mechanical fall risk assessment: A pilot survey. *Annals of Emergency Medicine* 62(4):S107–S108.

Griffith, Malachi, Jason R Walker, Nicholas C Spies, Benjamin J Ainscough, and Obi L Griffith. 2015. Informatics for RNA sequencing: a web resource for analysis on the cloud. *PLoS computational biology* 11(8): e1004393.

Grumbach, Kevin, Catherine R. Lucey, and S. Claiborne Johnston. 2014. Transforming from centers of learning to learning health systems: the challenge for academic health centers. *JAMA* 311(11): 1109–1110. `https://jamanetwork.com/journals/jama/articlepdf/1841977/jvp140015.pdf`.

Hardin, Johanna, Michael Waddell, C David Page, Fenghuang Zhan, Bart Barlogie, John Shaughnessy, and John J Crowley. 2004. Evaluation of multiple models to distinguish closely related forms of disease using DNA microarray data: an application to multiple myeloma. *Statistical applications in genetics and molecular biology* 3(1):1–21.

Hay, Michael, David W Thomas, John L Craighead, Celia Economides, and Jesse Rosenthal. 2014. Clinical development success rates for investigational drugs. *Nature biotechnology* 32(1):40.

Hill, Brian L, Robert Brown, Eilon Gabel, Nadav Rakocz, Christine Lee, Maxime Cannesson, Pierre Baldi, Loes Olde Loohuis, Ruth Johnson, Brandon Jew, et al. 2019. An automated machine learning-based model predicts postoperative mortality using readily-extractable preoperative electronic health record data. *British Journal of Anaesthesia*.

Hoerl, Arthur E, and Robert W Kennard. 1970. Ridge regression: biased estimation for nonorthogonal problems. *Technometrics* 12(1):55–67.

Holden, Richard J, Pascale Carayon, Ayse P Gurses, Peter Hoonakker, Ann Schoofs Hundt, A Ant Ozok, and A Joy Rivera-Rodriguez. 2013. SEIPS 2.0: a human factors framework for studying and improving the work of healthcare professionals and patients. *Ergonomics* 56(11):1669–1686.

Hripcsak, George, Jon D Duke, Nigam H Shah, Christian G Reich, Vojtech Huser, Martijn J Schuemie, Marc A Suchard, Rae Woong Park, Ian Chi Kei Wong, Peter R Rijnbeek, et al. 2015. Observational health data sciences and informatics (OHDSI): opportunities for observational researchers. *Studies in health technology and informatics* 216:574.

Hsu, Jia-Lien, Ping-Cheng Hung, Hung-Yen Lin, and Chung-Ho Hsieh. 2015. Applying under-sampling techniques and cost-sensitive learning methods on risk assessment of breast cancer. *Journal of Medical Systems* 39(4):40.

Hu, Keng-Wei, Yu-Hui Lu, Hung-Jung Lin, How-Ran Guo, and Ning-Ping Foo. 2012. Unscheduled return visits with and without admission post emergency department discharge. *The Journal of emergency medicine* 43(6):1110–1118.

Imielinski, Marcin, Robert N Baldassano, Anne Griffiths, Richard K Russell, Vito Annese, Marla Dubinsky, Subra Kugathasan, Jonathan P Bradfield, Thomas D Walters, Patrick Sleiman, et al. 2009. Common variants at five new loci associated with early-onset inflammatory bowel disease. *Nature genetics* 41(12):1335.

IWPC. 2009. Estimation of the warfarin dose with clinical and pharmacogenetic data. *New England Journal of Medicine* 360(8):753–764. PMID: 19228618, `https://doi.org/10.1056/NEJMoa0809329`.

Janes, Holly, Gary Longton, and Margaret S Pepe. 2009. Accommodating covariates in receiver operating characteristic analysis. *The Stata Journal* 9(1):17–39.

Janizek, Joseph D., Safiye Celik, and Su-In Lee. 2018. Explainable machine learning prediction of synergistic drug combinations for precision cancer medicine. *bioRxiv*. `https://www.biorxiv.org/content/early/2018/05/27/331769.full.pdf`.

Järvelin, Kalervo, Kalervo Järvelin, and Jaana Kekäläinen. 2000. IR evaluation methods for retrieving highly relevant documents. In *Proceedings of the 23rd annual international acm sigir conference on research and development in information retrieval*, 41–48. ACM.

Joachims, Thorsten. 1999. In *Advances in kernel methods*, ed. Bernhard Schölkopf, Christopher J. C. Burges, and Alexander J. Smola, chap. Making Large-scale Support Vector Machine Learning Practical, 169–184. Cambridge, MA, USA: MIT Press.

———. 2005. A support vector method for multivariate performance measures. In *Proceedings of the 22nd international conference on machine learning*, 377–384. ACM.

———. 2006. Training linear SVMs in linear time. In *Proceedings of the 12th ACM SIGKDD international conference on knowledge discovery and data mining*, 217–226. ACM.

Joachims, Thorsten, and Chun-Nam John Yu. 2009. Sparse kernel SVMs via cutting-plane training. *Machine Learning* 76(2-3):179–193.

Jorgensen, Sarah, Mira Zurayk, Samantha Yeung, Jill Terry, Maureen Dunn, Paul Nieberg, and Annie Wong-Beringer. 2018. Risk factors for early return visits to the emergency department in patients with urinary tract infection. *The American journal of emergency medicine* 36(1):12–17.

Judson, Richard, Keith Houck, Matt Martin, Thomas Knudsen, Russell S Thomas, Nisha Sipes, Imran Shah, John Wambaugh, and Kevin Crofton. 2014. In vitro and modelling approaches to risk assessment from the US Environmental Protection Agency ToxCast programme. *Basic & clinical pharmacology & toxicology* 115(1):69–76.

Kalscheur, Matthew M, Ryan T Kipp, Matthew C Tattersall, Chaoqun Mei, Kevin A Buhr, David L DeMets, Michael E Field, Lee L Eckhardt, and C David Page. 2018. Machine learning algorithm predicts cardiac resynchronization therapy outcomes: lessons from the companion trial. *Circulation: Arrhythmia and Electrophysiology* 11(1):e005499.

Karnik, Shreyas, Sin Lam Tan, Bess Berg, Ingrid Glurich, Jinfeng Zhang, Humberto J Vidaillet, C David Page, and Rajesh Chowdhary. 2012. Predicting atrial fibrillation and flutter using electronic health records. In *2012 annual international conference of the ieee engineering in medicine and biology society*, 5562–5565. IEEE.

Kaur, Pavleen, Ravinder Kumar, and Munish Kumar. 2019. A healthcare monitoring system using random forest and internet of things (IoT). *Multimedia Tools and Applications* 78(14):19905–19916.

Khoury, Joseph D, Nizar M Tannir, Michelle D Williams, Yunxin Chen, Hui Yao, Jianping Zhang, Erika J Thompson, Funda Meric-Bernstam, L Jeffrey Medeiros, John N Weinstein, et al. 2013. Landscape of DNA virus associations across human malignant cancers: analysis of 3,775 cases using RNA-Seq. *Journal of virology* 87(16):8916–8926.

Kim, Soo Yeon, Saehoon Kim, Joongbum Cho, Young Suh Kim, In Suk Sol, Youngchul Sung, Inhyeok Cho, Minseop Park, Haerin Jang, Yoon Hee Kim, et al. 2019. A deep learning model for real-time mortality prediction in critically ill children. *Critical Care* 23(1):279.

King, Ross D, Mohammed Ouali, Arbra T Strong, Alaaeldin Aly, Adel Elmaghraby, Mehmed Kantardzic, and David Page. 2000. Is it better to combine predictions? *Protein Engineering* 13(1):15–19.

Kingsford, Carl, and Steven L Salzberg. 2008. What are decision trees? *Nature biotechnology* 26(9):1011–1013.

Klinkenberg, Ralf. 2003. Predicting phases in business cycles under concept drift. In *Proc. of llwa*, 3–10.

Klinkenberg, Ralf, and Thorsten Joachims. 2000. Detecting concept drift with support vector machines. In *Icml*, 487–494.

Kohavi, Ron, et al. 1995. A study of cross-validation and bootstrap for accuracy estimation and model selection. In *Ijcai*, vol. 14, 1137–1145. Montreal, Canada.

Kruppa, Jochen, Andreas Ziegler, and Inke R König. 2012. Risk estimation and risk prediction using machine-learning methods. *Human genetics* 131(10):1639–1654.

Kukar, Matjaž. 2003. Drifting concepts as hidden factors in clinical studies. In *Artificial intelligence in medicine*, ed. Michel Dojat, Elpida T. Keravnou, and Pedro Barahona, 355–364. Berlin, Heidelberg: Springer Berlin Heidelberg.

Kuusisto, Finn, Vitor Santos Costa, Houssam Nassif, Elizabeth Burnside, David Page, and Jude Shavlik. 2014. Support vector machines for differential prediction. In *Joint european conference on machine learning and knowledge discovery in databases*, 50–65. Springer.

Lai, Hang, Huaxiong Huang, Karim Keshavjee, Aziz Guergachi, and Xin Gao. 2019. Predictive models for diabetes mellitus using machine learning techniques. *BMC Endocrine Disorders* 19(1):1–9.

Landis, Suzanne E, and Shelley L Galvin. 2014. Implementation and assessment of a fall screening program in primary care practices. *Journal of the American Geriatrics Society* 62(12):2408–2414.

Le, Quoc V, Alex Smola, Olivier Chapelle, and Choon Hui Teo. 2010. Optimization of ranking measures. *Journal of Machine Learning Research* 1:1–48.

Lewis, Roger J. 2000. An introduction to classification and regression tree (CART) analysis. In *Annual meeting of the society for academic emergency medicine in san francisco, california*, vol. 14.

Li, Xiang, Haifeng Liu, Xin Du, Ping Zhang, Gang Hu, Guotong Xie, Shijing Guo, Meilin Xu, and Xiaoping Xie. 2016. Integrated machine learning approaches for predicting ischemic stroke and thromboembolism in atrial fibrillation. In *Amia annual symposium proceedings*, vol. 2016, 799–807. American Medical Informatics Association.

Liu, Vincent X, David W Bates, Jenna Wiens, and Nigam H Shah. 2019. The number needed to benefit: estimating the value of predictive analytics in healthcare. *Journal of the American Medical Informatics Association*.

Lobo, Jorge M, Alberto Jiménez-Valverde, and Raimundo Real. 2008. AUC: a misleading measure of the performance of predictive distribution models. *Global ecology and Biogeography* 17(2):145–151.

Luo, Wei, Dinh Phung, Truyen Tran, Sunil Gupta, Santu Rana, Chandan Karmakar, Alistair Shilton, John Yearwood, Nevenka Dimitrova, Tu Bao Ho, Svetha Venkatesh, and Michael Berk. 2016. Guidelines for developing and reporting machine learning predictive models in biomedical research: a multidisciplinary view. *J Med Internet Res* 18(12):e323.

Maglogiannis, Ilias, Euripidis Loukis, Elias Zafiropoulos, and Antonis Stasis. 2009. Support vectors machine-based identification of heart valve

diseases using heart sounds. *Computer methods and programs in biomedicine* 95(1):47–61.

Mangal, Ankita, and Nishant Kumar. 2016. Using big data to enhance the bosch production line performance: a Kaggle challenge. In *2016 IEEE international conference on big data (big data)*, 2029–2035. IEEE.

Marill, Keith A. 2004a. Advanced statistics: linear regression, part I: simple linear regression. *Academic emergency medicine* 11(1):87–93.

———. 2004b. Advanced statistics: linear regression, part II: multiple linear regression. *Academic emergency medicine* 11(1):94–102.

Mausner, Judith S, and Shira Kramer. 1985. *Epidemiology: An introductory text*. W. B. Saunders Company.

McCusker, Jane, Sylvie Cardin, Franc¸ois Bellavance, and Eric Belzile. 2000. Return to the emergency department among elders: patterns and predictors. *Academic Emergency Medicine* 7(3):249–259.

McDonald, Ross A, David J Hand, and Idris A Eckley. 2003. An empirical comparison of three boosting algorithms on real data sets with artificial class noise. In *International workshop on multiple classifier systems*, 35–44. Springer.

Mortazavi, Bobak J, Nicholas S Downing, Emily M Bucholz, Kumar Dharmarajan, Ajay Manhapra, Shu-Xia Li, Sahand N Negahban, and Harlan M Krumholz. 2016. Analysis of machine learning techniques for heart failure readmissions. *Circulation: Cardiovascular Quality and Outcomes* 9(6): 629–640.

Obermeyer, Ziad, and Ezekiel J Emanuel. 2016. Predicting the future–big data, machine learning, and clinical medicine. *The New England journal of medicine* 375(13):1216.

Olsen, LeighAnne, Dara Aisner, J Michael McGinnis, et al. 2007. *The learning healthcare system: workshop summary*. Natl Academy Pr.

Olson, Harry, Graham Betton, Denise Robinson, Karluss Thomas, Alastair Monro, Gerald Kolaja, Patrick Lilly, James Sanders, Glenn Sipes, William Bracken, et al. 2000. Concordance of the toxicity of pharmaceuticals in humans and in animals. *Regulatory Toxicology and Pharmacology* 32(1): 56–67.

Panel on Prevention of Falls in Older Persons, American Geriatrics Society, and British Geriatrics Society. 2011. Summary of the updated American Geriatrics Society/British Geriatrics Society clinical practice guideline for prevention of falls in older persons.

Patel, Shilpa J, Daniel B Chamberlain, and James M Chamberlain. 2018. A machine learning approach to predicting need for hospitalization for pediatric asthma exacerbation at the time of emergency department triage. *Academic Emergency Medicine* 25(12):1463–1470.

Patterson, Brian W, Collin J Engstrom, Varun Sah, Maureen A Smith, Eneida A Mendonça, Michael S Pulia, Michael D Repplinger, Azita G Hamedani, David Page, and Manish N Shah. 2019. Training and interpreting machine learning algorithms to evaluate fall risk after emergency department visits. *Medical care* 57(7):560–566.

Patterson, Brian W, Peter S Pang, Lora AlKhawam, Azita G Hamedani, Eneida A Mendonca, Ying-Qi Zhao, and Arjun K Venkatesh. 2016. The association between use of brain CT for atraumatic headache and 30-day emergency department revisitation. *American Journal of Roentgenology* 207(6):W117–W124.

Patterson, Brian W, Arjun K Venkatesh, Lora AlKhawam, and Peter S Pang. 2015. Abdominal computed tomography utilization and 30-day re-

visitation in emergency department patients presenting with abdominal pain. *Academic Emergency Medicine* 22(7):803–810.

Pedregosa, Fabian, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, et al. 2011. Scikit-learn: machine learning in Python. *Journal of machine learning research* 12(Oct):2825–2830.

Phelan, Elizabeth A, Jane E Mahoney, Jan C Voit, and Judy A Stevens. 2015. Assessment and management of fall risk in primary care settings. *Medical Clinics* 99(2):281–293.

Philip, Femi, Heather L Gornik, Jeevanantham Rajeswaran, Eugene H Blackstone, and Mehdi H Shishehbor. 2014. The impact of renal artery stenosis on outcomes after open-heart surgery. *Journal of the American College of Cardiology* 63(4):310–316.

Polo-Hernández, Erica, Fernando De Castro, Alejandro G García-García, Arantxa Tabernero, and José M Medina. 2010. Oleic acid synthesized in the periventricular zone promotes axonogenesis in the striatum during brain development. *Journal of neurochemistry* 114(6):1756–1766.

Porta, Miquel. 2016. *A dictionary of epidemiology*. Oxford University Press.

Provost, Foster, and Ron Kohavi. 1998. Guest editors' introduction: on applied research in machine learning. *Machine learning* 30(2):127–132.

Qin, Tao, Tie-Yan Liu, and Hang Li. 2010. A general approximation framework for direct optimization of information retrieval measures. *Information retrieval* 13(4):375–397.

Radford, Alan D, David Chapman, Linda Dixon, Julian Chantrey, Alistair C Darby, and Neil Hall. 2012. Application of next-generation sequencing technologies in virology. *The Journal of general virology* 93(Pt 9): 1853.

Ricketts, Thomas C, and Laurie J Goldsmith. 2005. Access in health services research: the battle of the frameworks. *Nursing outlook* 53(6): 274–280.

Rising, Kristin L, Timothy W Victor, Judd E Hollander, and Brendan G Carr. 2014. Patient returns to the emergency department: the time-to-return curve. *Academic Emergency Medicine* 21(8):864–871.

Rusanov, Alexander, Nicole G Weiskopf, Shuang Wang, and Chunhua Weng. 2014. Hidden in plain sight: bias towards sick patients when sampling patients with sufficient electronic health record data for research. *BMC medical informatics and decision making* 14(1):51.

Russell, Stuart J, and Peter Norvig. 2016. *Artificial intelligence: a modern approach*. Malaysia; Pearson Education Limited.

Schapire, RE, and Y Freund. 1995. A decision-theoretic generalization of on-line learning and an application to boosting. In *Second European conference on computational learning theory*, 23–37.

Schoenman, Julie A, and Nancy Chockley. 2012. The concentration of health care spending. *National Institute for Health Care Management Research Educational Foundation (NIHCM) Foundation Data Brief*.

Schwartz, Michael P, Zhonggang Hou, Nicholas E Propson, Jue Zhang, Collin J Engstrom, Vitor Santos Costa, Peng Jiang, Bao Kim Nguyen, Jennifer M Bolin, William Daly, et al. 2015. Human pluripotent stem cell-derived neural constructs for predicting neural toxicity. *Proceedings of the National Academy of Sciences* 112(40):12516–12521.

Seaberg, David, Stanton Elseroad, Michael Dumas, Sudave Mendiratta, Jessica Whittle, Cheryl Hyatte, and Jan Keys. 2017. Patient navigation for patients frequently visiting the emergency department: a randomized, controlled trial. *Academic Emergency Medicine* 24(11):1327–1333.

Sendak, Mark, Michael Gao, Marshall Nichols, Anthony Lin, and Suresh Balu. 2019. Machine learning in health care: A critical appraisal of challenges and opportunities. *eGEMs* 7(1).

Seymour, Christopher W, Jason N Kennedy, Shu Wang, Chung-Chou H Chang, Corrine F Elliott, Zhongying Xu, Scott Berry, Gilles Clermont, Gregory Cooper, Hernando Gomez, et al. 2019. Derivation, validation, and potential treatment implications of novel clinical phenotypes for sepsis. *Jama* 321(20):2003–2017.

Stang, Paul E, Patrick B Ryan, Judith A Racoosin, J Marc Overhage, Abraham G Hartzema, Christian Reich, Emily Welebob, Thomas Scarnecchia, and Janet Woodcock. 2010. Advancing the science for active surveillance: rationale and design for the observational medical outcomes partnership. *Annals of internal medicine* 153(9):600–606.

Stephens, Caroline E, Robert Newcomer, Mary Blegen, Bruce Miller, and Charlene Harrington. 2011. Emergency department use by nursing home residents: effect of severity of cognitive impairment. *The Gerontologist* 52(3):383–393.

Sterling, Daniel A, Judith A O'Connor, and John Bonadies. 2001. Geriatric falls: injury severity is high and disproportionate to mechanism. *Journal of Trauma and Acute Care Surgery* 50(1):116–119.

Stilgoe, Jack. 2018. Machine learning, social learning and the governance of self-driving cars. *Social Studies of Science* 48(1):25–56. PMID: 29160165, `https://doi.org/10.1177/0306312717741687`.

Stoltzfus, Jill C. 2011. Logistic regression: a brief primer. *Academic Emergency Medicine* 18(10):1099–1104.

Struyf, Jan, Seth Dobrin, and David Page. 2008. Combining gene expression, demographic and clinical data in modeling disease: a case study of bipolar disorder and schizophrenia. *Bmc Genomics* 9(1):531.

Svetnik, Vladimir, Andy Liaw, Christopher Tong, and Ting Wang. 2004. Application of Breiman's random forest to modeling structure-activity relationships of pharmaceutical molecules. In *Multiple classifier systems*, ed. Fabio Roli, Josef Kittler, and Terry Windeatt, 334–343. Berlin, Heidelberg: Springer Berlin Heidelberg.

Taylor, Michael, John Guiver, Stephen Robertson, and Tom Minka. 2008. Softrank: optimizing non-smooth rank metrics. In *Proceedings of the 2008 international conference on web search and data mining*, 77–86. ACM.

Taylor, Michael J, Chris McNicholas, Chris Nicolay, Ara Darzi, Derek Bell, and Julie E Reed. 2014. Systematic review of the application of the plan–do–study–act method to improve quality in healthcare. *BMJ Qual Saf* 23(4):290–298.

Taylor, R Andrew, Joseph R Pare, Arjun K Venkatesh, Hani Mowafi, Edward R Melnick, William Fleischman, and M Kennedy Hall. 2016. Prediction of in-hospital mortality in emergency department patients with sepsis: a local big data–driven, machine learning approach. *Academic emergency medicine* 23(3):269–278.

Teubert, Annekatrin, Johannes Thome, Andreas Büttner, Jörg Richter, and Gisela Irmisch. 2013. Elevated oleic acid serum concentrations in patients suffering from alcohol dependence. *Journal of molecular psychiatry* 1(1): 13.

Tibshirani, Robert. 1996. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society: Series B (Methodological)* 58(1): 267–288.

———. 2011. Regression shrinkage and selection via the lasso: a retrospective. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* 73(3):273–282.

Tiedemann, Anne, Catherine Sherrington, Teresa Orr, Jamie Hallen, Donna Lewis, Ann Kelly, Constance Vogler, Stephen R Lord, and Jacqueline CT Close. 2013. Identifying older people at high risk of future falls: development and validation of a screening tool for use in emergency departments. *Emerg Med J* 30(11):918–922.

Ting, Daniel Shu Wei, Carol Yim-Lui Cheung, Gilbert Lim, Gavin Siew Wei Tan, Nguyen D Quang, Alfred Gan, Haslina Hamzah, Renata Garcia-Franco, Ian Yew San Yeo, Shu Yen Lee, et al. 2017. Development and validation of a deep learning system for diabetic retinopathy and related eye diseases using retinal images from multiethnic populations with diabetes. *Jama* 318(22):2211–2223.

Trapnell, Cole, David G Hendrickson, Martin Sauvageau, Loyal Goff, John L Rinn, and Lior Pachter. 2013. Differential analysis of gene regulation at transcript resolution with RNA-seq. *Nature biotechnology* 31(1): 46.

UW Health. 2019. About UW Health. `https://www.uwhealth.org/about-uwhealth/uw-health/11012`.

Wang, Jiaxuan, Jeeheh Oh, Haozhu Wang, and Jenna Wiens. 2018. Learning credible models. In *Proceedings of the 24th ACM SIGKDD international conference on knowledge discovery & data mining*, 2417–2426. ACM.

Weigand, John V, and Lowell W Gerson. 2001. Preventive care in the emergency department: should emergency departments institute a falls prevention program for elder patients? A systematic review. *Academic Emergency Medicine* 8(8):823–826.

Weng, Stephen F, Jenna Reps, Joe Kai, Jonathan M Garibaldi, and Nadeem Qureshi. 2017. Can machine-learning improve cardiovascular risk prediction using routine clinical data? *PloS one* 12(4):e0174944.

Wu, Jionglin, Jason Roy, and Walter F Stewart. 2010. Prediction modeling using EHR data: challenges, strategies, and a comparison of machine learning approaches. *Medical care* S106–S113.

Xie, Zidian, Olga Nikolayeva, Jiebo Luo, and Dongmei Li. 2019. Peer reviewed: building risk prediction models for type 2 diabetes using machine learning techniques. *Preventing chronic disease* 16.

Xu, Jun, and Hang Li. 2007. Adarank: a boosting algorithm for information retrieval. In *Proceedings of the 30th annual international acm sigir conference on research and development in information retrieval*, 391–398. ACM.

Xu, Jun, Tie-Yan Liu, Min Lu, Hang Li, and Wei-Ying Ma. 2008. Directly optimizing evaluation measures in learning to rank. In *Proceedings of the 31st annual international acm sigir conference on research and development in information retrieval*, 107–114. ACM.

Yu, Wei, Tiebin Liu, Rodolfo Valdez, Marta Gwinn, and Muin J Khoury. 2010. Application of support vector machine modeling for prediction of common diseases: the case of diabetes and pre-diabetes. *BMC medical informatics and decision making* 10(1):16.