## <1>Chapter 9: Big data and machine learning

Precision medicine research designed to reduce health disparities often involves studying multi-level datasets to understand how diseases manifest disproportionately in one group over another, and how scarce healthcare resources can be directed precisely to those most at risk for disease.[1] For example, a precision medicine approach to address disparities in environmental health is currently instituted in my clinical practice at an academic medical center. The approach attempts to reduce pulmonary, cardiovascular, and autoimmune diseases that may be related to environmental contamination in patient homes. A comprehensive home scan for key environmental contaminants would not be economically or practically feasible; hence, clinicians must judiciously choose which patients should be referred for a home visit and environmental contamination scan. Numerous researchers are interested in identifying which biomarkers may be indicative of specific environmental contaminants in the home and could therefore be used to direct home visits and environmental scans to the highest-risk patients. A number of alternative biomarkers have been proposed to screen patients, because simple verbal surveys have limited discriminatory ability. But each biomarker interacts as part of a complex biological pathway, and has multi-level interactions with clinical, demographic, social, economic, and community-level factors such as a person's co-morbid conditions, age, housing conditions, and the community's history of environmental contamination in certain neighborhoods. A biomarker may be indicative of contamination only among certain people with a key residential housing background, and may only be useful for screening disease among those; furthermore, a single biomarker is often triggered to be high through multiple mechanisms, such that an environmental scan is only appropriate for those with other factors potentially leading to a high biomarker level (e.g., C-reactive protein may be high from inflammatory atherosclerotic processes among people with coronary artery disease, or due to specific environmental pollutants).[2]

This resource allocation problem is typical of precision medicine research designed to reduce health disparities: it involves identifying which patients should be triaged or treated in a way that may help reduce disparities between the most vulnerable people and those with less disease burden; it involves a problem of interactions between subcellular, clinical, and macro-social factors; and it presents a complex methodological problem: how can we decipher a complex interaction between multiple interdependent factors at multiple levels of a person's life, and how can we ensure that we maximize the clinical utility of a tool that we develop to identify who to offer a resource-limited service to?

In this manuscript, I seek to address the latter two methodological questions. I specifically provide a structured tutorial for medical and public health researchers on the application of machine learning methods to conduct precision medicine research designed to reduce health disparities. "Machine learning" refers to the development of a set of algorithmic approaches—commonly referred to as learners, predictive models, or estimators—that seek to categorize data or predict an outcome. The term "machine learning" encompasses a wide array of methods that have a common set of principles; most importantly, the methods seek to link inputs to an output by repeatedly refining rules that govern how input data relate to the output result, by analyzing multiple subsets of data and sequentially improving the rules being learned. For example, a complex constellation of inputs (biomarkers, clinical features, and social circumstances) may be predictive of the output (whether a person will experience a particular disease complication). A machine learning method will seek to distinguish that unique combination of inputs from other possible complications of markers, features and circumstances, using a systematic approach that allows a learner to become more accurate as more data are available to train it.[3] In this tutorial, I walk the reader through some of the most common machine learning approaches in use today, highlighting their

some of the most common machine learning approaches in use today, highlighting their advantages and disadvantages, their potential uses and situations in which they may need to be avoided, and demonstrating their application in an example dataset with open-source statistical code that accompanies this article.

## <2>Example research problem

Suppose we wish to identify which patients in a clinical setting we would want to refer to an environmental home scan service—a service that tests for, and removes, potentially hazardous contaminants from a patient's home that are related to a disease of interest. Also suppose that we are practically and economically constrained, such that we only want to refer those patients who are likely to benefit from the service. We have a historical database of patient features, including a series of biomarkers suspected of being potentially related to the contaminants of interest, data about patient demographics and clinical features, and outcomes data revealing which patients actually had contamination in their homes and which patients did not. Our goal is to design a machine learning algorithm that will learn from this historical data and accurately direct referrals for future patients, optimizing the referral process to precisely refer those patients most likely to benefit from the intervention.

To guide our approach to studying this problem, I accompany this article with a simulated dataset and statistical code that can be opened and evaluated in the common statistical program *R*.

## <2>Key terms and concepts

*Supervised versus unsupervised learning.* There are two major classes of machine learners (though purists will point out that some learners span both categories, or fail to fit perfectly into either). "Supervised" learners are those for which we have a dataset with both input data and outcomes data, such as patient features as inputs and a disease outcome. Standard logistic regression is an example of a supervised learning method, as it maps covariates onto a probability of an outcome. The goal of supervised learners is to learn the relations between input and outcomes data, then predict future outcomes based on future patient's input data. Supervised learners will be the exclusive focus of this article. "Unsupervised" learners, by contrast, are those that learn how to cluster or categorize only input data, providing natural groupings of similar data types, such as through factor analysis.[3] In the above example research problem, a supervised learner would be one that uses patient features to predict who will have contamination in their home, while an unsupervised learner would be one that clusters patients into groups based on having similar features (e.g., similar demographics, similar biomarkers, etc.).

*Overfitting, regularization and cross-validation.* "Overfitting" refers to the process by which a learner not only captures the general relationships between input and output data, but wrongly captures random noise in the dataset (**Figure 1**), which prevents the learner from making accurate and generalizable predictions when applied in the future. In the above example research problem, overfitting might occur if a learner only learns that people who are between 4.1 and 6.9 years old with high values on three biomarkers are at risk for having contamination, and fails to identify that a 4.0 year old or a 7.0 year old with similarly elevated biomarkers are also at risk.

To increase generalizability and prevent overfitting, we try to follow Occam's Razor, which reminds us that we don't want to produce a complicated algorithm to do something if we can have a simpler algorithm that does our chosen task well. In addition to being more generalizable, a simpler algorithm may require less input data or complicated data transformations, may be more interpretable, and may be more easily fixed if predictions turn out to be erroneous.

"Regularization" refers to processes that help enforce Occam's Razor when we're training a machine learner, particularly when there are many correlated variables in the input data, as is often the case with patient clinical data (as biomarkers are often highly correlated to each other, as are clinical diagnoses and social variables). The two most common regularization processes are called: (i) L1 regularization, or "LASSO" (which stands for least absolute shrinkage and selection operator), and (ii) L2 regularization, or

"ridge" regularization. LASSO tends to select one correlated variable among others, like choosing a representative variable for many related variables. In a standard logistic regression, for example, LASSO will include one correlated predictor variable in the final equation and set the regression coefficients for its related variables to zero; this is operationalized by penalizing the absolute sum of the regression coefficients. By contrast, ridge regularization of a logistic regression tends to set the regression coefficients of correlated variables to an equal value, under the premise that correlated variables should be similar and extreme values may be based on outliers; this is operationalizing by penalizing the squared sum of the regression coefficients.[4]

In practice, we split their data into two sets. We use the "validation" subset only once, after the learner has been fine-tuned, to assess the generalizability of the estimator. The validation subset is treated like a reserve wine, opened only once and drunk after all preparations have been made. But we repeatedly sample the other subset, known as the "training" subset, for a process called "cross-validation" (**Figure 1**). The training subset of the data is our workhorse, our table wine for daily consumption and utility. "Cross-validation" involves repeatedly sampling from a training subset of the data to ensure that our learner is generalizable across multiple subsamples, and to choose how much we want to regularize the learner by selecting a penalty parameter that determines how much to choose one correlated variable over others (via LASSO) or restrict correlated variables' coefficients to be similar (via ridge). After each sample of training data is obtained, we train the learner by finding the penalty parameter value that minimizes the mean-squared error between the predictions from the learner and the observed training data subset. We often select a combination of both LASSO and ridge regularization, finding what combination of the two processes minimizes the error overall across all cross-validations. Combining a bit of both LASSO and ridge is called elastic net regularization, and the statistical code accompanying this tutorial illustrates how to implement this regularization approach through cross-validation in *R*.[4]

*Bagging and boosting.* A central principal of machine learning is to improve our predictions by making multiple repeated comparisons between our learner's predictions and subsamples of our training dataset. There are two general strategies to maximize our learner's predictive performance. One strategy is to train lots of learners on lots of subsamples of the training data, assuming that doing the process over and over again will help weed out extreme outlier predictions and help us settle to a good average prediction. This averaging strategy is known as "bagging", as in taking the average of a big bag filled with learners that we each treat equally.[5] A common alternative strategy is to train one learner, then learn from the first learners' errors to make a more optimal learner in the next round of sampling, and so on, which is called a "boosting" strategy.[6,7] Bagging can help us avoid getting stuck down a wrong path of assuming that slight improvements in our evaluation metrics (discussed below) signal the best path to follow; sometimes the path that seems like an incremental improvement is just a random deviation from a bigger improvement that could be found by looking more broadly across a big set of possible ways to model the data through bagging. Conversely, simpler problems such as predicting an outcome with few classes (e.g., just a 0/1 for absence/presence of the outcome, rather than a multi-class or continuous outcome variables) can often be modeled best through a boosting strategy.[8,9] Our statistical code illustrates both approaches, which are discussed further below under the "major families of machine learning methods" heading.

## <2>Metrics for rigorous evaluation

Before we go about training a learner, we should pre-specify the answer to a key study design question: what will we consider a metric of a "good" learner, and how will we fairly compare our learner to alternatives? The most common machine learning evaluation metric is the C-statistic, also known as the "area under the receiver operating characteristic (ROC) curve" (**Figure 2**).

The C-statistic captures how often a learner will correctly select the higher-risk person among a pair of people, where one member of the pair will have the outcome and the other will not. The C-statistic is therefore called a metric of "discrimination".[10] A common rule of thumb is that C-statistic values over 0.7 are considered acceptable, while

common rule of thumb is that C-statistic values over 0.7 are considered acceptable, while values over 0.8 are desirable. Yet it is better to understand the intuition behind C-statistics rather than simply use a rule of thumb to determine whether a learner is "good" at prediction. In clinical medicine, the x-axis of the ROC plot corresponds to 1 minus the specificity of a test, and the y-axis to the sensitivity of a test. Hence, if we desire a machine learner that we want to use for reassuring people that they do not have a disease, (being most interested in avoiding false negative values), we would desire a learner with a high value on the y-axis of the ROC curve (sensitivity) even if we had a mediocre value on the x-axis (specificity). In the above example research problem of finding children with contamination, we may want to have a highly sensitive learner because we wouldn't want to miss a case of contamination, even if we have a few false positive cases of investigations that didn't find contamination. Conversely, if we desire a machine learner that we want to use for confirming the presence of a disease (being most interested in avoiding false positives), we may tolerate a learner with a low value on the x-axis (high specificity) even if it has mediocre y-axis values (sensitivity). In these settings, pre-specifying a sensitivity or specificity evaluation metric, rather than just the composite C-statistic, can be important.

In many clinical cases, we don't simply want to distinguish between higher-risk and lower-risk people, but actually want a correct estimate for the absolute risk a person might have for an outcome. In cardiovascular disease prevention, for example, algorithms to treat patients with primary prevention medications (such as statins) often direct therapy depending on whether a person's absolute risk of a heart attack or stroke is considered low, moderate, high, or very high. Hence, we don't simply want to know which of two people is at higher risk for a heart attack or stroke; we need to know whether the learner correctly predicts a person's risk as 6% or 8% or 10% (each of which confers different therapeutic recommendations), and how far that estimate is from the truth. In such settings, we want to assess the learner's "calibration", and a key additional statistic beyond the C-statistic: the calibration curve. The calibration curve (**Figure 2**) is a plot of the predicted rate of outcomes among centiles of the validation dataset population (x-axis) against the observed rate of outcomes among those centiles of the validation dataset population (y-axis); a perfect learner will have a 45-degree line between the predicted and observed outcome rates. The Hosmer-Lemeshow test is a common statistical test for evaluating the degree of error between the predicted and observed rates of the outcomes plotted in the calibration curve.[11]

Other metrics are also commonly used in the machine learning literature. These include a "confusion matrix", which epidemiologists commonly call a contingency table —a 2-by-2 table of true and false positive and negative outcomes in the validation dataset (**Appendix Table 1**). Some machine learning papers also use the metric of "accuracy", which is the sum of true positives plus true negatives, divided by the total number of predictions. A third common metric is "precision", which is the sum of true positives divided by the sum of true and false positives, which clinical epidemiologists call the positive predictive value. In all cases, the pre-specified metric that is chosen for evaluation should correspond to the ultimate application that the learner will be used for, by selecting a metric that corresponds most to the goals, time, effort, and budget of end-users.

## <2>Major families of machine learning methods

There are numerous strategies for training a learner. Here, I provide a simplified rubric that covers the major current approaches for training a machine learner.

*Tree-based learners.* Two of the most common approaches in machine learning are essentially strategies for building "decision trees" from data. Many clinicians and epidemiologists will be familiar decision trees, which are essentially flowcharts that guide users about how to categorize or treat a particular patient or population (**Figure 3**). In a typical decision tree, each branch of the tree divides the sampled study population into increasingly-smaller subgroups that differ in their probability of an outcome of interest or their likelihood of benefitting from a particular intervention.[12] A good decision tree will separate the sampled population into groups that have low within-group

variability, but high between-group variability in the probability of the outcome. In the above example research problem, where we are trying to predict whether a patient has home environment contamination, a decision tree might first separate our patient population by neighborhood, then identify which features in the first neighborhood define patients into high-risk versus low-risk groups; those features may differ in the second neighborhood, and the third, and so on.

An advantage of a decision tree is the ability to consider multiple covariates at once, potentially capturing complex interactions between covariates (such as between neighborhood and other factors) and nonlinearities (since covariates can have different cut-points defining branches, and different outcome rates in one branch than another). However, a limitation of decision trees is that they are prone to overfitting, such that a subgroup may be identified because the decision tree has over-interpreted noise or random outliers in the data as reflecting a real phenomenon or a real subgroup; even cross-validation may not detect the over-fitting.[13]

Two common methods for constructing decision trees from data, while minimizing the risk of overfitting, are gradient boosting machines (GBM) and random forests (RF). In both methods, many trees are grown to subsamples of the training dataset. GBMs average many trees where errors made by the first tree contribute to learning of a more optimal tree in the next iteration (a boosting strategy).[6,7] RF also builds numerous decision trees, but averages a forest composed of many trees, where each tree is independently fitted (a bagging strategy) with a random subset of covariates selected to be eligible to define the branches.[5] As noted above, the bagging strategy will often produce higher discrimination than the boosting strategy in situations where there is a complex outcome being predicted (such as a categorical or continuous outcome), rather than a simple dichotomous outcome (such as the absence or presence of a disease or condition).[8,9]

Training learners using tree-based methods can be particularly helpful to researchers who are trying to identify how different risk factors—in isolation or in combination—contribute to differential risk of an outcome, as is often the case in health disparities research. Additionally, tree-based methods can be particularly helpful when researchers believe that complex combinations of factors at multiple levels—such as among subcellular, individual, and neighborhood-level factors—interact to heighten the risk of an outcome of interest. Trees can also be more useful than standard logistic regression when researchers are trying to predict a rare outcome (such as a high-cost hospitalization, or a rare but severe disease complication) that is predictable from a constellation of complex interacting factors.[14]

An important caveat when considering a tree-based approach to machine learning is that the GBM approach typically requires researchers to decide several factors that can influence how well a learner performs (known as "tuning"): how many trees to average, how deep the trees should be (how many "layers" of branches to have, which determines how many subpopulations to divide the population into), and how quickly the trees should adapt to initial error (the "learning rate") to optimize a performance metric such as the C-statistic. On the other hand, the RF approach generally produces a reproducible result with maximum discrimination across a wide range of specifications, thereby not requiring extensive tuning. Hence, the RF approach is often reasonably favored by researchers new to machine learning. The statistical code accompanying this article provides examples of both GBMs and RFs applied to the prediction of environmental contamination in a mock dataset, and compares their performance to a standard logistic regression.

*Deep learning.* Deep learning refers to the training of "neural networks" to predict outcomes; a neural network is a series of data transformations, where the outputs from one series of transformations informs the inputs to the next series of transformations (**Figure 3**).[15] Each transformer (or neuron) in the network takes a weighted combination of inputs reflecting a weighted sum of the observed data (for the first layer of neurons) or from a previous layer of neurons (for the second and subsequent layers of neurons), and produces an output based on a nonlinear transformation function known as an activation function, which can be thought of as analogous to a link function in statistical regression

function, which can be thought of as analogous to a link function in statistical regression. The weights for each sum, and activation function values, determine the output from the network. A standard logistic regression is simply a neural network with a single layer of neurons, in which each transformer multiplies the input covariate by a regression coefficient, and a logistic function is the activation function. Regularization (for instance, LASSO or ridge) and cross-validation techniques are typically applied to neural networks to prevent overfitting.

There are many ways to adapt a deep learning neural network to complex problems. The simplest deep learning neural network structure, visualized in **Figure 3**, is known as a "feedforward" neural network, in which each layer of neurons is fully connected to the next layer, and information only flows in one direction. By contrast, "recurrent" neural networks have backwards feedback from one layer to a prior layer, which allows for more learning across time series, particularly for problems where knowledge of one prediction affects the next prediction (such as in speech recognition, where the prior word informs the choice of the next word; or in predicting sequential disease events for an individual, where prior diagnoses inform the probability of a subsequent diagnosis).[16] Additional common types of supervised deep learners include: (i) "convolutional" neural networks, which capture local features of the training dataset, then combine locally-learned networks to gather a global understanding (such as for image recognition, where cluster of pixels form one part of a picture, and these parts are combined to recognize the overall image); (ii) "autoencoders", which take noisy data and try to reconstruct the original data, a process known as "denoising" (also useful for image processing and sound processing); and (iii) "word2vec", which is a series of two-layer neural networks trained to detect words in their context and thereby aid in natural language processing.[16–19]

At the time of this writing, deep learning neural networks in the medical and public health context remain relatively limited to image recognition (e.g., radiologic detection of abnormalities on X-rays), disease classification problems (e.g., "reading" electronic medical record notes to categorize and classify disease phenotypes in a data-driven manner), and outcome prediction (predicting a clinical outcome based on complex features).[16,19,20] Deep learning may be more effective than tree-based methods for outcome prediction when complex context in text (such as in clinical notes or other written assessments) must be processed and included as predictors, when extremely complex interactions are thought to exist between covariates that predict an outcome (such as between multiple -omics markers), or when complex sequential processes must be predicted rather than just a single outcome (e.g., hospital admission, discharge diagnosis, then readmission, and mortality). Deep learning can, however, be difficult to implement because researchers have to choose activation functions, network depths (number of layers), and degree of regularization, among other choices in structuring the model to customize it for the task being accomplished. The statistical code accompanying this article provides an example of constructing and tuning a standard feedforward neural network, and includes multiple common options for activation functions, network depths, and regularization processes; the code also enables comparison of network learners to tree-based estimators and standard logistic regression for the example problem of predicting home environmental contamination. Although the statistical code is in *R* to be familiar to epidemiologists and health services researchers, it should be noted that deep learning neural networks are more commonly programmed in Python due to speed and scaling limitations of *R*, and online tutorials for more advanced deep learning for medical applications using Python are recommended.

*Ensembles.* A key insight from recent machine learning research is that a combination of learners can often improve prediction over any single learner. Just as a multi-center clinical trial can help us identify generalizable insights better than a single-center trial, an "ensemble" of learners can help more closely approximate the truth, even when none of the underlying "base learners" is fully correct.[21] Researchers who have little *a priori* theory to favor one type of learner over another can particularly benefit from training an ensemble of learners. To train an ensemble, a researcher will first individually develop learners on the dataset, then use a "meta-learner" (sometimes called a "super learner" method) to combine the predictions of the underlying

a "super learner" or stacking method) to combine the predictions of the underlying base learners.[22] The meta-learner gives a weight to each underlying base learner, typically using a strategy such as elastic net regularization to estimate what combination of weights across the base learners minimizes the overall error between the weighted predictions from the base learners and the observed outcomes in repeated samples from the training data. Technically, both GBM and RF are ensemble learners because they combine individual decision trees to produce a composite prediction.

## <2>Performance comparison among methods for the example problem

Here, I analyze the example problem described under Methods, which uses a simulated dataset containing 100 variables that are correlated and inter-dependent in complicated ways. The statistical code accompanying this article allows readers to reproduce the simulated dataset, within which there are 40 binary variables, 30 secondary dependent variables correlated with or conditioned upon some of the first 40 variables (reflecting, for example, co-morbidities that often occur together), 20 categorical variables, and 10 continuous variables, some a subset of variables predicting the outcome of home environmental contamination (a dichotomous outcome) with complex interactions and dependencies.

Using the statistical code and approaches described under Methods, I find that a standard logistic regression—choosing covariates using elastic net regularization to find a balance between LASSO and ridge regression, and thereby address collinearity and reduce overfitting—performed relatively poorly in terms of discrimination, with a C-statistic of 0.64. A RF learner provided slight improvement with a C-statistic of 0.69, which was also produced by a GBM learner. A deep learner only produced a slight further improvement with a C-statistic of 0.70. An ensemble of these learners, however, dramatically improved discrimination with a C-statistic of 0.86. To further characterize the sensitivity and specificity of the learners, I have included a confusion matrix (contingency table) in the Appendix. Note that all evaluation metrics are calculated on the validation dataset, not on the training data.

## <2>Interpreting machine learners: visualizing inside the black box

Two key strategies are available to researchers trying to peer inside the "black box" of machine learners, and understand how the learners have interpreted the data to produce useful predictions. First, a variable importance table or plot is a standard method that identifies how often different variables are included in the underlying learners, and therefore clarifies which variables are most influential in prediction (**Figure 4**). The plot displays standardized coefficient magnitudes (i.e., Z-scores) across all variables.[22]

Second, partial dependence plots allow us to visualize how learners relate covariates to outputs.[23] In particular, when covariates go through complex transformations, a partial dependence plot effectively reveals to us how a learner has related values of an individual covariate with the probability of the outcome, revealing important non-linearities for example (**Figure 4**).

Machine learning methods can be challenging to learn, but may ultimately provide superior results to traditional regression for some complex research problems at the intersection of precision medicine and health disparities. Here, I reviewed key terms and concepts in machine learning, critical methods for evaluation and interpretation of machine learners, and major common types of learners, with accompanying statistical code to demonstrate their application.

As with the presentation of many other types of research, machine learning papers are recommended to follow key principles of good research practice. In closing this manuscript, I review some of the principles considered most critical for high-quality machine learning papers.[24–26]

First, an important standard for reproducibility and extension in machine learning literature is the sharing of statistical code and underlying data. De-identifying data and sharing the raw statistical code is particularly important given the problem that many researchers papers have been found to not be reproducible.[27]

Second, it is important for machine learning problems to be pre-specified, so that researchers are not tempted to use the approach purely to produce (potentially false-positive) associations. Choosing a pre-specified metric for evaluation is an important part of having a pre-specified design for a machine learning project.

Third, the end-user of a machine learner must be kept in mind. Different audiences need to either be able to interpret a learner, or just use the learner by inputting data and having the learner automatically provide results (e.g., at the backend of an electronic medical record). Some covariates may be harder to collect than others, and end-users may prefer some metrics (e.g., sensitivity) over others (e.g., specificity).

Finally, it is critical that researchers using machine learning methods have "data empathy," or the perspective that the quality and type of data must correspond well to the type of question being asked and the future utilization of the method. If a particular question cannot be answered well with a small dataset, it is unlikely that the question will be better answered with a larger dataset of the same type and data quality.[28] For example, abundant use of insurance claims or electronic medical record data, which are large and widely available in the medical literature, is problematic for clinical studies, as prediction models will not actually predict the presence of disease, but rather of diagnostic billing codes that may poorly correlate to actual disease (and suffer from selection biases and misclassification errors). Hence, it is vital for a researcher to prospectively collect data for a given question, rather than treat machine learning as a "hammer" that can hit any available nail.

Ultimately, machine learning methods are potentially useful for precision health and health disparities researchers, if prediction is an important and meaningful endeavor for such research. As machine learning methods evolve, abiding by key principles of good machine learning practice will likely serve to help improve the utility, trustworthiness, and ultimately impact of machine learning.