What Is Machine Learning: a Primer for the Epidemiologist

Qifang Bi*, Katherine E. Goodman*, Joshua Kaminsky, and Justin Lessler

* - these authors contributed equally to this paper

Correspondence to Dr. Justin Lessler, Department of Epidemiology, Johns Hopkins Bloomberg School of Public Health, 615 N Wolfe St, Baltimore, MD 21231 (email: justin@jhu.edu; work phone: 410-955-3551)

Author Affiliations: Department of Epidemiology, Johns Hopkins Bloomberg School of Public Health, Baltimore, Maryland (Qifang Bi, Katherine E. Goodman, Joshua Kaminsky, and Justin Lessler)

The authors declared no conflict of interest.
This work received no funding.

Running head: Machine Learning for Epidemiologists

Abstract

Machine learning is a branch of computer science that has the potential to transform epidemiological sciences. Amid a growing focus on “Big Data,” it offers epidemiologists new tools to tackle problems for which classical methods are not well-suited. In order to critically evaluate the value of integrating machine learning algorithms and existing methods, however, it is essential to address language and technical barriers between the two fields that can make it difficult for epidemiologists to read and assess machine learning studies. Here, we provide an overview of the concepts and terminology used in machine learning literature, which encompasses a diverse set of tools with goals ranging from prediction, to classification, to clustering. We provide a brief introduction to five common machine learning algorithms and four ensemble-based approaches. We then summarize epidemiological applications of machine learning techniques in the published literature. We recommend approaches to incorporate machine learning in epidemiological research and discuss opportunities and challenges for integrating machine learning and existing epidemiological research methods.

Key words: Machine learning, Big Data, ensemble models

Abbreviations

BMI Body mass index

ANN Artificial neural networks

CART Classification and Regression Tree

SVMs Support vector machines

BMA Bayesian model averaging

GWAS Genome-wide association studies)

SNPs single nucleotide polymorphisms

Machine learning is a branch of computer science that broadly aims to enable computers to “learn” without being directly programmed.(1) It has origins in the artificial intelligence movement of the 1950s and emphasizes practical objectives and applications, in particular prediction and optimization. Computers “learn” in machine learning by improving their performance at tasks through “experience”.(2, p. XV) In practice, “experience” usually means fitting to data, hence there is not a clear boundary between machine learning and statistical approaches. Indeed, whether a given methodology is considered “machine learning” or “statistical” often reflects its history as much as genuine differences, and many algorithms (e.g. LASSO, stepwise regression) may or may not be considered machine learning depending on who you ask. Still, despite methodological similarities, machine learning is philosophically and practically distinguishable. At the liberty of (considerable) over-simplification, machine learning generally emphasizes predictive accuracy over hypothesis-driven inference, usually focusing on large, high-dimensional (i.e., having many covariates) datasets.(3,4) Regardless of the precise distinction between approaches, in practice, machine learning offers epidemiologists important tools. In particular, a growing focus on “Big Data” emphasizes problems and datasets where machine learning algorithms excel while more commonly used statistical approaches struggle.

This primer provides a basic introduction to machine learning with the aim of providing readers a foundation to critically read studies based on these methods and a jumping off point for those interested in using machine learning techniques in epidemiological research. The Concepts and Terminology section presents concepts and terminology

used in the machine learning literature. The Machine Learning Algorithms section provides a brief introduction to five common machine learning algorithms: artificial neural networks, decision trees, support vector machines, Naive Bayes, and k-means clustering. These are important and commonly used algorithms that epidemiologists are likely to encounter in practice, but these are by no means comprehensive of this large and highly diverse field. The following two sections, Ensemble Methods and Epidemiological Applications, extend this examination to ensemble-based approaches and epidemiological applications in the published literature. Brief Recommendations provides some recommendations for incorporating machine learning into epidemiological practice, and the last section discusses opportunities and challenges.

CONCEPTS AND TERMINOLOGY

For epidemiologists seeking to integrate machine learning techniques into their research, language and technical barriers between the two fields can make reading source materials and studies challenging. Some machine learning concepts lack statistical or epidemiological parallels, and machine learning terminology often differs even where the underlying concepts are the same. Here we briefly review basic machine learning principles and provide a glossary of machine learning terms and their statistical/epidemiological equivalents (**Table 1**).

Supervised, unsupervised, and semi-supervised learning

Machine learning is broadly classifiable by whether the computer's learning (i.e., model fitting) is "supervised" or "unsupervised." *Supervised learning* is akin to the type of model fitting that is standard in epidemiologic practice: the value of the outcome (i.e., the dependent variable), often called its "label" in machine learning, is known for each observation. Data with specified outcome values is called "labeled data". Common supervised learning techniques include standard epidemiologic approaches such as linear and logistic regression, as well as many of the most popular machine learning algorithms (e.g., decision trees, support vector machines).

In *unsupervised learning* the algorithm attempts to identify natural relationships and groupings within the data without reference to any outcome or the "right answer".(5, p.517) Unsupervised learning approaches share similarities in goals and structure with statistical approaches that attempt to identify unspecified subgroups with similar characteristics (e.g.,'latent' variables or classes).(6) Clustering algorithms, which group observations based upon similar data characteristics (e.g., both oranges and beach balls are round), are common unsupervised learning implementations. Examples may include k-means clustering and Expectation–Maximization clustering using Gaussian mixture models.(7,8)

Semi-supervised learning fits models to both labeled and unlabeled data. Labeling data (outcomes) is often time-consuming and expensive, particularly for large datasets. Semi-supervised learning supplements limited labeled data with an abundance of

unlabeled data with the goal of improving model performance (studies show that unlabeled data can help build a better classifier, but appropriate model selection is critical).(9) For example, in a study of webpage classification, Nigam *et al.* fit a Naive Bayes classifier to labeled data and then used the same classifier to probabilistically label unlabeled observations (i.e., fill in missing outcome data). (10) They then retrained a new classifier on the resulting, fully labeled dataset, thereby achieving a 30 percent increase in webpage classification accuracy on data outside the training set. Semi-supervised learning can bear some similarity to statistical approaches for missing data and censoring (e.g., multiple imputation), but as an approach that focuses on imputing missing outcomes rather than missing covariates.

Classification versus regression algorithms

Within the domain of supervised learning, machine learning algorithms can be further divided into classification or regression applications depending upon the nature of the response variable. In general, in the machine learning literature classification refers to prediction of categorical outcomes, while regression refers to prediction of continuous outcomes. We use this terminology throughout this primer and are explicit when referring to specific regression algorithms (e.g., logistic regression). Many machine learning algorithms that were developed to perform classification have been adapted to also address regression problems, and vice versa.

Generative versus discriminative algorithms

Machine learning algorithms, both supervised and unsupervised, can be discriminative or generative (11,12). *Discriminative algorithms* directly model the conditional probability of an outcome, Pr(*y*|*x*) ("the probability of 'y' given 'x'"), in a set of observed data -- for example, the probability that a subject has type II diabetes given a certain body mass index (BMI). Most statistical approaches familiar to epidemiologists (e.g., linear and logistic regression) are discriminative, as are most of the algorithms discussed in this primer.

In contrast, while *generative algorithms* can also compute the conditional probability of an outcome, this computation occurs indirectly. Generative algorithms first model the joint probability distribution, Pr(x,y) (the probabilities associated with all possible combinations of 'x' and 'y'), or, continuing our example, a probabilistic model that accounts for all observed combinations of BMIs and diabetes outcomes (Table 2). This joint probability distribution can be transformed into a conditional probability distribution in order to classify data, as Pr(*y*|x) = Pr(x,y) / Pr(x). Because the joint probability distribution models the underlying data-generating process, generative models can also be used, as their name suggests, for directly generating new simulated data-points reflecting the distribution of the covariates and outcome in the modeled population.(11) However, as they model the full joint distribution of outcomes and covariates, generative models are generally more complex and require more assumptions to fit than discriminative algorithms.(12,13) Examples of generative algorithms include Naive Bayes and hidden Markov models.(11)

Reinforcement learning

In reinforcement learning, systems learn to excel at a task over time through trial and error.(14) Reinforcement learning techniques take an iterative approach to learning by obtaining positive or negative feedback based on performance of a given task on some data (whether prediction, classification, or another action) and then self-adapting and attempting the task again on new data (though old data may be re-encountered).(15) Depending on how it is implemented, this approach can be akin to supervised learning, or it may represent a semi-supervised approach (as in generative adversarial neural networks (16)). Reinforcement learning algorithms often optimize the use of early, 'exploratory' versions of a model -- i.e., task attempts -- that perform poorly to gain information to perform better on future attempts, and then become less labile as the model "learns" more.(15) Medical and epidemiological applications of reinforcement learning have included modeling the effect of sequential clinical treatment decisions on disease progression (17) (e.g., optimizing first- and second-line therapy decisions for schizophrenia management (18)) and personalized, adaptive medication dosing strategies. For example, Nemati *et al*. used reinforcement learning with artificial neural networks in a cohort of intensive care unit patients to develop individualized heparin-dosing strategies that evolve as a patient's clinical phenotype changes, in order to maximize the time that blood drug levels remain within the therapeutic window (19).

MACHINE LEARNING ALGORITHMS

In this section we introduce five common machine learning algorithms: artificial neural networks, decision trees, support vector machines, Naive Bayes, and k-means clustering. For each, we include a brief description, summarize strengths and limitations, and highlight implementations available on common statistical computing platforms. This section is intended to provide a high-level introduction to these algorithms, and we refer interested readers to the cited references for further information.

Artificial neural networks

Artificial neural networks (ANNs) are inspired by the signaling behavior of neurons in biological neural networks. ANNs, which consist of a population of neurons interconnected through complex signaling pathways, use this structure to analyze complex interactions between a group of measurable covariates in order to predict an outcome. ANNs possess layers of "neurons" connected by "axons" (20) (Figure 1A). These layers are grouped into: 1) an input layer; 2) one or more middle "hidden" layers; and 3) an output layer. The neurons in the input and output layers correspond to the independent and dependent variables, respectively. Neurons in adjacent layers communicate with each other through activation functions, which convert the weighted sum of a neuron's inputs into an output (Figure 1B). Depending upon the type of activation function, the output can be dichotomous ("1" when the weighted sum exceeds a given threshold, and "0" otherwise) or continuous. The weighted sum of a neuron's inputs is somewhat analogous to coefficients in linear or logistic regression.

Figure 1 illustrates a simple neural network with a single hidden layer and a feed-forward structure (i.e., signals progress unidirectionally from input to output layers). For supervised learning applications, once the number of layers and neurons are selected, the connection weights of the ANN are fit on a training set of labeled data through a reinforcement learning approach. Initial connection weights are generally selected randomly, and network output is compared to the correct output (class labels) using a loss function, which is based on the difference between the predicted and the true values of the outcome. The goal is to reduce the loss function to zero, i.e., to make the ANN's predicted output match truth as closely as possible, albeit while also protecting against overfitting. In response, 1) resulting error values are distributed backwards through the network, from output to input, in order to assign an error value contribution to each hidden and input layer neuron ("back-propagation"; for additional technical information on this process, *see, e.g.,* (21)), and 2) connection weights are updated in order to minimize the loss function ("weight adjustment"). This two-fold optimization process repeats for a number of "epochs" or iterations until the network meets a pre-specified stopping rule or error rate threshold.(22,23)

Strengths and limitations: Strengths of ANNs include their ability to accommodate variable interactions and non-linear associations without user specification.(22) The primary limitation of ANNs is that, although arguably not completely a "black box", (23, p.1112) the underlying model nevertheless remains largely opaque. Effects are mediated exclusively through hidden layer(s), making interpreting relationships between

input and output layers challenging, especially for “deep” ANNs, which include multiple hidden layers. This lack of transparency complicates commonsense or etiological interpretation of individual variable effects and connection weights, although there are continuing efforts to enhance ANN interpretability (see, e.g., (20,24,25). ANN training parameters can also be complex, and setting and tuning these parameters generally necessitates technical expertise. Moreover, complex ANNs, including deep networks, can require large datasets (potentially in the tens or hundreds of thousands, although there is no hard and fast rule) in order to achieve optimal model performance, which may be prohibitive for some epidemiological applications (26).

Sample statistical packages and modules: *R* (R Foundation for Statistical Computing, Vienna, Austria) - neuralnet, nnet, deepnet, tensorflow; *Stata* (StataCorp, College Station, TX) - N/A; *SAS* (SAS Institute Inc, Cary, NC) - Enterprise Miner Neural Network and AutoNeural; *Python* (Python Software Foundation, Wilmington, DE) - sklearn, tensorflow

Decision trees

Decision trees (i.e., classification and regression trees) create a series of decision rules based on continuous and/or categorical input variables to predict an outcome.(5,27) Classification trees predict categorical outcomes, and regression trees predict continuous outcomes. Classification and Regression Tree (CART) analysis has been popularized as an umbrella term for any decision tree learning method.(27) However, ‘CART’ is also a common implementation algorithm in the epidemiologic and medical

literature, although a number of other decision tree algorithms have also been developed (e.g., ID3, CHAID).(28–30)

Figure 2 presents a hypothetical classification tree for a binary outcome, diabetes. To derive a decision tree, the algorithm applies a splitting rule on successively smaller partitions of data, with each partition being a node on the tree. The partition consisting of all data is the root node; in Figure 2 this node is split based on BMI. Splits are selected to minimize some measure of node impurity (i.e., diversity of classes) or heterogeneity (i.e., variance) in each resulting partition (the "daughter nodes"). (5,27) The splitting process repeats on each branch of the tree until additional splits yield no further reductions in node impurity, or some other stopping criteria is reached (e.g., a specified minimum number of observations in terminal nodes or the value at which error is minimized in cross-validation (31)). In many algorithms, this splitting is often followed by a "pruning" step in which partitions are re-merged (i.e., some bottom nodes are removed, making the final tree smaller) based on some criteria designed to increase generalizability.(32)

Strengths and limitations: Decision trees are generally easy to understand -- having been described that "[o]n interpretability, trees rate an A+" (4) -- making their output ideal for a range of target audiences. They are also flexible to non-linear covariate effects and can incorporate higher-order interactions between covariates.(27,33) Trees may lose information by dichotomizing or categorizing variables where associations are continuous, and they can be unstable to even small data changes. Because most

decision tree algorithms are “greedy” (splitting decisions are locally-optimized at nodes), through a domino-effect dramatically different trees can result if even a single higher-level node shifts to a different variable.(34) Hence, decision trees can be highly sensitive to small perturbations in data. Perhaps most fundamentally, decision trees are prone to overfitting, and their ultimate utility depends heavily on appropriately implemented pruning and/or stopping criteria. Ensemble-based decision trees (e.g., random forests) can address some of these concerns (*see* Section Ensemble Methods), but they do not produce a single, easily interpretable tree.

Sample statistical packages and modules: *R* - rpart, caret, ctree, randomForest (ensemble decision trees); *Stata* - cart (failure-time data only), chaid, chaidforest (ensemble decision trees); *SAS* - Enterprise Miner Decision Tree; *Python* - sklearn

Support vector machine

Support vector machines (SVMs) are a set of supervised learning methods used for classification and regression problems (35,36). SVMs construct an optimal boundary, called a hyperplane, that best separates observations of different classes. In one dimension, this boundary is a point, in two, a line, and in three, a plane (**Figure 3**). However, many observations often need to be transformed before they can be separated by a hyperplane. SVMs address this problem by applying a data transformation called a “kernel function” to the data (3). Kernel functions project the data into a higher-dimensional space where the input variables are separable (**Figure 3**). The optimal kernel function is usually chosen from a set of commonly used kernel

functions selected through cross-validation. Popular kernel functions include polynomial kernel, gaussian kernel, and sigmoid kernel. Following kernel function transformation, the best hyperplane maximizes the separation between the different classes (i.e., the margin, defined as the distance from the hyperplane to the closest data point), while tolerating a specified level of misclassification. SVMs are traditionally used for binary classification, but multiple pairwise comparison can be applied for multi-class classification (36). Extensions to SVM techniques have also been developed that can be used to predict continuous outcomes (called support vector regression) (37).

In Figure 3, individuals with and without diabetes cannot be separated by a line in the two-dimensional space based upon the predictors, age and BMI (**Figure 3, A**). However, when we project the data into a three-dimensional space by applying a kernel given by φ((*age*, *bmi*)) = (*age*, *bmi*, (*bmi*-*a*)*(*age*-*b*)), where *a* and *b* are fixed parameters estimated from the data, the data is now separable in the three-dimensional space by a plane (**Figure 3, B**).

Strengths and limitations: SVMs generally demonstrate low misclassification error and scale well to high-dimensional data (38). SVMs have reasonable interpretability, especially when a kernel function is not used. Where a kernel function is necessary, however, selecting the optimal kernel function typically requires experimenting with a set of standard functions. This approach can be time-consuming and does not guarantee that the set of standard kernel functions that were evaluated included the optimal function, and in some cases hand-crafted kernel functions are used instead.

Sample statistical packages and modules: *R* - e1071, kernlab, caret; *Stata* – svmachines (39); *SAS* - PROC SVM; *Python* - sklearn

Naive Bayes

Naive Bayes is simple probabilistic classification algorithm based upon Bayes' theorem that makes the "naive" assumption of independence between predictive variables (40). Naive Bayes calculates the probability associated with each possible class conditional on a set of covariates -- i.e., the product of the prior probability and the likelihood function. The classifier then selects the class with the highest probability as the "correct" class (**Figure 4**). The prior probability typically reflects one's belief about the outcome, either based on the study itself or from other published literature. The independence assumption in Naive Bayes greatly simplifies the calculation by decomposing the likelihood function into a product of likelihood functions, one for each covariate. Though adaptations of Naïve Bayes for regression exist (41), the algorithm is most commonly used for classification.

Continuing our diabetes example, a Naive Bayes classifier would calculate the likelihood of each observation (e.g., BMI>32, Age>55, Female) among people who are and are not diabetic (**Figure 4**). Assuming equal prior probability for diabetes, an individual would be assigned to the class (i.e., diabetic vs. not diabetic) that had the highest likelihood of independently producing each observation.

Strengths and limitations: The simplicity of Naive Bayes contributes to the popularity of these algorithms. It has been shown to perform relatively well in the presence of noise, missing data, and irrelevant features (42). Because of the independence assumption, Naive Bayes requires estimating fewer parameters, and thus a smaller training set, than more complex algorithms (43,44).

Arguably the most important limitation of Naive Bayes is that its independence assumption is often violated in the real-world. In addition, the most probable class may weigh heavily on the chosen prior. Thus, proper adjustment for underlying class frequencies is necessary when prior probability in the training set is not representative of the general population. In addition, when data are correlated, Naive Bayes gives more influence to the likelihood function of highly correlated features and may bias the prediction (43). These limitations will, however, not impact classification performance so long as the ordering of the biased probabilities is the same as that of the correct ones. Naive Bayes probability outputs nevertheless should **never** be interpreted as actual probabilities of class membership.

Sample statistical packages and modules: *R* - e1071, klaR, bnlearn, h2o, naivebayes; *Stata* - use Multinomial mixture models in StataStan (45); *SAS* - PROC HPBNET; *Python* - sklearn

K-means clustering

K-means clustering is one of the simplest unsupervised learning algorithms (46). It partitions observations into a pre-specified number of distinct clusters (*k*), such that within-cluster variation (e.g., squared Euclidean distance) is as small as possible (47). K-means clustering first randomly selects *k* centroids, with each centroid defining one cluster (i.e., each observation is assigned to its closest centroid). Following *k* selection, the algorithm iteratively alternates between two steps until classification remains unchanged: 1) assign each observation to its nearest centroid, typically defined by squared Euclidean distance, and 2) move the location of the centroid to the mean of all data points assigned to that centroid's cluster (Figure 5). There are a variety of methods for selecting *k*. Often investigators pre-specify *k* based on background knowledge or visual examination of the data; however, likelihood and error-based approaches to selecting *k* have been developed (48) .

Strengths and limitations: K-means clustering is simple, easy-to-interpret, and computationally efficient. However, one important limitation is that the number of clusters need to be pre-specified. A slight difference in *k* can produce very different results, and methods for estimating *k* (49) don't necessarily agree with each other (50). In addition, when the distance between observations and cluster centroids is calculated with Euclidean metrics, the algorithm assumes clusters have the same within-cluster variance. If some clusters are much larger than others, k-means can produce non-intuitive results (50) (Figure 6).

Sample statistical packages and modules: *R* **-** ClusterR, fpc, akmeans, kmeans in base R; *Stata* - cluster kmeans; *SAS* - FASTCLUS and HPCLUS; *Python* - sklearn

ENSEMBLE METHODS

Ensemble methods utilize information from multiple models to improve predictive performance compared to a single model. The idea is that even though any individual model within an ensemble is not adequate to capture the characteristics of the entire phenomenon, so long as they perform better than at random, once combined they can borrow strength from each other and achieve high predictive accuracy. Broadly, ensemble methods improve performance by creating a population of models through either: 1) training the same underlying algorithm to different versions of a dataset (e.g., bagging and boosting), or 2) training qualitatively different models on the same dataset (e.g., Bayesian model averaging, Super Learner) (Web Figure 1), and then combining results across these models based upon a defined algorithm. While the primary objective of bagging and boosting is to minimize overfitting, multiple algorithm ensembles capitalize on different models' strengths and avoid the need for model pre-selection. These alternative ensemble approaches are often used in combination, either as part of the same algorithm, or through nested approaches.

Bagging

Bagging (or bootstrap aggregating) fits the same underlying algorithm to each bootstrapped copy of the original training data and then creates a final prediction based upon outputs from the resulting, parameterized, models (51). The final prediction for a quantitative outcome is obtained by averaging the predictions. For a qualitative outcome, the final prediction either takes the majority vote among the classifiers or averages probabilities across the number of bootstrap fits. Bagging reduces model variance significantly without affecting bias (52,53).

Feature bagging attempts to further reduce overfitting. It trains models on random subsets of variables/features instead of all variables in an attempt to reduce correlation between models in an ensemble. When applied to tree-based methods, the resulting models are called *random forests*, which force each split to consider a random subset of predictors (54), giving other weak predictors a greater chance to be selected as split candidates. Otherwise, when there is a strong predictor for the outcome, many trees would choose to first split on that predictor, creating highly correlated predictions regardless of the variables chosen at the subsequent splits.

Aside from k-fold cross validation, one way to estimate prediction errors specifically for random forests is to compute *out-of-bag error* (55). Out-of-bag error is the mean prediction error for each observation, using only the models that did not include the observation in their bootstrapped samples. *Variable importance rankings* summarize the relative importance of each predictor across all fitted trees. These rankings reflect the

importance of a variable for predicting outcomes by averaging the impurity decrease for all nodes where the variable is used across all trees in the forest. (51) Impurity decrease measures changes in accuracy of a tree, and can be described by, for example, Gini impurity (a measure of the probability of mistaken categorization within a node) for bagging classification trees or the Residual Sum of Squares for bagging regression trees. “Important” variables change the accuracy of the trees the most. Importance rankings can be used to assess the relative impact of individual predictors, as well as the interaction between predictors, in predicting the outcome (56,57).

Sample Statistical Packages and Modules: *R* - caret, randomForest, adabag; *Stata* - ‘chaidforest’ for random forests; *SAS* – see ref (58); *Python* - sklearn.ensemble

Boosting

Like bagging, *boosting* also trains models on subsets of data, but it does it in a sequential fashion and improves the classifiers by analyzing prediction errors (59,60). *Adaboost* is a well-known boosting method that sets weights to both observations and classifiers (61,62). Observations are given weights, initially equal, that increase if incorrectly classified by the last iteration of the classifier; hence, subsequent iterations will prioritize correctly classifying these observations. The final output classifier is a weighted average from the classifier built in each iteration, with higher weight given to classifiers with higher predictive accuracy (i.e., lower error rates on training data). *Gradient boosting* is a generalization of AdaBoost that uses gradient descent to

optimize any differentiable loss function (i.e., a measure of classifier performance other than simple classification error) (63,64).

Sample statistical packages and modules: *R* - gbm, adabag, fastAdaboost, xgboost, ada, caret; *Stata* – see ref (65); *SAS* – see ref (66); *Python* - sklearn.ensemble

Bayesian model averaging

Bayesian model averaging (BMA) estimates the posterior distribution of a predicted value (or the parameters defining a parametric relationship) by calculating the weighted average of model-specific estimates, where the weights are driven by how much the data supports each competing model (67). BMA has been applied to many statistical models including linear regression, generalized linear models, and cox proportional hazards models, and provides better predictive ability than using any single model (67). Its variants, such as *Bayesian model combination*, have emerged to further tackle the issue of overfitting, as BMA has a tendency of placing too much weight on the most probable model (68). *Bayesian model combination* creates a set of ensembles, each representing a combination of individual models, and weights the ensemble-specific estimate of the effect size (as opposed to estimates based on the most probable model in BMA) by the probability that the ensemble is correct given the data (68,69).

Sample statistical packages and modules: *R* - BMS/BAS/BMA; *Stata* - NA; *SAS* - see module (70); *Python* - pyBMA

Super Learner

Super Learner is a prediction algorithm that uses cross-validation to determine the optimal weighted combination of predictions from a group of candidate learners. (71-73) Building on the "stacked generalization" approach proposed by Wolpert, this approach allows the use of machine learning algorithms (e.g., random forests) in addition to standard parametric algorithms (e.g., logistic regression). K-fold cross validation (Web Figure 1) is used to assign weights to each of a user-defined pool of component algorithms based on out-of-training set performance, and then the component models are fit to the entire data set. Model outputs are based on the predictions of these candidate models weighted by the cross validation-derived weights. It has been applied to predict the fitness of the HIV virus as a function of its mutations (72), and is has been used as part of procedures to estimate causal effects (see Section Epidemiological Applications).

Sample statistical packages and modules: *R* - SuperLearner; *Stata* - NA; *SAS* - NA; *Python* - Scikit-learn

EPIDEMIOLOGICAL APPLICATIONS

In this section we give an overview of the way in which machine learning algorithms have been used in various applications related to epidemiological practice. This is not a comprehensive review or intended to discuss every limitation and nuance of these approaches, but to serve to direct readers to areas of active research in the literature.

Causal inference

Relative to classical statistical or epidemiological approaches, machine learning algorithms have historically placed less emphasis on causal inference. Indeed, machine learning has been described as a "black box" method because it is difficult to draw etiological inferences from the output of some algorithms (e.g., ANNs). However, machine learning techniques can still be an important component of approaches to estimating causal effects in observational studies, with sometimes superior performance for reducing bias and controlling for confounding.(74)

Propensity score weighting is a common approach for estimating causal effects in observational studies.(75) Propensity scores have been traditionally estimated with logistic regression, but this approach requires assumptions that, if unmet, may render biased effect estimates despite propensity score conditioning. Machine learning algorithms often deal implicitly with interactions and non-linearities, whereas such high-order terms must be explicitly specified (and are commonly ignored) in logistic regression. Machine learning algorithms also perform well in estimating propensity scores in the presence of high-dimensional data and can reduce underlying model misspecification.(76) Although these machine learning benefits may be at the expense of easy interpretability, these concerns are not pertinent to propensity score estimation, as the interpretability of propensity scores is not relevant to their performance. Multiple studies have empirically demonstrated bias reductions where propensity scores are generated with machine learning methods, in particular ensemble-based approaches.(76–79) Under certain conditions, however, bias may persist or be exacerbated by machine learning methods (80-82). Publications that calculated

propensity scores with machine learning approaches have included studies assessing the effect of early sexual initiation on young adult health (83), vaccination on birth outcome (84), and combination antibiotic treatment on Gram-negative bacteremia (85).

Likewise, machine learning algorithms can be used as a component of any causal inference framework where an estimate of the likelihood (or distribution) of an outcome is an important component of the inferential process, but need not be directly interpretable. For example, the Super Learner algorithm has been used as a component of targeted maximum likelihood estimation (TMLE) and marginal structural model approaches (86). Examples include an investigation the effect of alcohol outlet density on patterns of alcohol consumption (87) and the relationship between childhood adversities and mental disorders by race/ethnicity (88).

Machine learning methods have been used more directly to attempt to understand heterogeneity in treatment effects across sub-populations. For example, Athey and colleagues have developed an approach to building “casual trees” that create decision trees where groupings are based on treatment effect and provide principled estimates of treatment effects within these strata using an approach they call “honest estimation”(89). This approach has been extended to apply the random forest algorithm to these trees, creating so called “casual forests” that can be used to estimate treatment effects in individuals with particular covariate profiles (90).

Another application of machine learning more directly related to problems of causal inference is causal structure learning, which has grown as a distinct branch of machine learning. Causal structure learning encompasses a group of exploratory techniques that identify an optimal directed acyclic graph consistent with conditional independence

relationships in the data and provided background knowledge. Approaches to causal structure learning include Bayesian network approaches (*see* Scutari and Denis for an overview (91)) and Linear, Non-Gaussian, Acyclic Models (LiNGAMs) (92-94). The former have been applied to derive causal influences in cellular signaling pathways (95) and to infer causal associations between gene expression and disease (96), while LiNGAMs have been used to estimate causal directionality between sleep disorders and depression (97) and to explore causality between TV viewing habits and weight change.(98)

Diagnostics, prognostic predictive models, and other clinical decision support tools

Disease diagnosis and prognosis are perhaps the oldest clinical utilizations of machine learning techniques (99) and remain common applications in the epidemiological literature. Machine learning is particularly well-suited to certain diagnostic questions (e.g., those that involve imaging and/or high-dimensional data), and it can enhance prognostic models and clinical decision support tools through, for example, automation and ease-of-use.

Diagnostics that involve imaging, where each pixel can be conceptualized as a feature, and other high-dimensional data are problems well suited for machine learning approaches. SVMs have been utilized extensively in oncology for diagnosis and disease staging from radiological and tissue data.(100–108) They have also been utilized for tumor typing from tissue microarray gene expression data, which, due to their high

dimensionality, can be problematic for traditional statistical models.(109–112) Outside of oncology, SVMs have shown promise for neuro-imaging diagnostics, including for dementia (113) and autism spectrum disorder.(114–116)

Machine learning techniques are also well-suited to prognostic models and other clinical decision support tools where accurate diagnosis (i.e., low classification error) is the primary objective, or where automation is desired. For example, Palaniappan and Awang employed a combination of ANNs, Naive Bayes, and decision trees in order to develop an automated, web-based prediction tool, the *Intelligent Heart Disease Prediction System* (IHDPS).(117) Incorporation into hospital and emergency room operations research is also common. ANNs have been used in emergency room populations, for example, to predict death among sepsis patients (118) and prolonged hospital stays among the elderly (119), and random forests have been used to build electronic triage models for risk-stratifying patients.(120,121) Because many machine learning methods can accommodate complex variable interactions without *a priori* specification, they may also uncover previously unknown prognostic sub-groups.(122) For example, Brims *et al*. application of CART algorithms to a dataset of malignant pleural mesothelioma cases revealed four distinct prognostic groups based upon clinical characteristics.(122)

Conversely, machine learning methodologies can also be helpful where manual use, rather than automation, is contemplated. In particular, decision trees are popular clinical prediction tools for both diagnosis and prognosis due to their simplicity and interpretability. Because their output uses branching logic rather than calculations, decision trees are generally user-friendly for clinicians to apply at the bedside (e.g., to

predict the likelihood that an infection is drug-resistant while awaiting microbiological confirmation (123)).(124)

Genome-wide association studies (GWAS)

Genome-wide association studies seek to identify genetic variants that influence disease risk. Genome datasets generally contain large numbers of genes and single nucleotide polymorphisms (SNPs) of interest but, due to sequencing costs and other practical constraints, are of limited sample size. These high-dimensional data are the types of data on which machine learning algorithms perform well. Hence, ensemble machine learning approaches such as random forests are commonly used. Random forests can rank the most important SNPs for disease outcome, and they have been used, for example, to predict drug response in epilepsy patients based on clinical and genetic information(125); to identify genetic variants associated with Parkinson's disease and other neurological disorders(126); and to "data mine" high density genetic data to predict Alzheimer's disease risk.(127)

Other, non-ensemble algorithms are also popular in the GWAS literature. Researchers have applied SVMs with bayesian model averaging to genome-wide data to predict late onset Alzheimer's disease (128), and *k*-nearest neighbors (a relatively simple unsupervised classification algorithm (129)) to predict the heritable genetic susceptibility of common cancers.(130) Microbiome studies, which also involve high-dimensional (albeit bacterial) genetic data, have likewise utilized machine learning to identify disease risk factors among microbiota/microbiome signatures.(131) Moreover, because

interactions do not require *a priori* specification under many machine learning algorithms, machine learning approaches are well-suited to identify complex gene-gene (132) and genetic-environmental interactions that may modulate disease risk (e.g., use of ANNs to explore interactions between nutrient intake and metabolic pathway polymorphisms on breast cancer susceptibility (133)).

Geospatial applications

Machine learning can help to predict and map disease occurrence and health indicators in areas where data are limited. Its ability to efficiently process high-dimensional datasets from heterogeneous contexts and multiple geographic scales makes it particularly suitable for this task. A major focus is the development of the WorldPop project, which is an open-source archive of demographic parameters on fine spatial scales. It uses random forests to map global population distribution on a per-pixel scale by combining remote sensing data (e.g., satellite) across multiple geospatial scales (134). Beyond WorldPop's use of random forests, another type of ensemble machine learning algorithm, boosted regression trees, has also been widely used to map environmental suitability for disease transmission, including dengue (135), leishmaniasis (136), Ebola (137), Crimean-Congo Haemorrhagic Fever (138), and Zika virus (139,140). In general, these studies: 1) chose a set of known or proposed environmental and socioeconomic covariates, 2) incorporated global assessments regarding whether the disease(s) of interest is circulating in the country or region, and 3) with these data, built boosted regression tree models. The resulting models were used to predict infection probabilities on a pixel-by-pixel scale.

Text mining

Electronic health records provide an unprecedented amount of clinical information for research, but to utilize these data sources effectively in studies or for surveillance is generally cost-prohibitive without some form of automated data extraction. Machine learning offers automated tools for extracting unstructured information from textual clinical documents. For example, i2b2 (Informatics for Integrating Biology and the Bedside) Challenges address a range of projects aiming to develop and evaluate information extraction methods for clinical text.(141) The 2009 Medication Challenge focused on providing a schema to extract information including medications, dosages, modes (routes) of administration, frequencies, durations, and reasons for administration from discharge summaries (142). Other applications include de-identifying personal health information, research subject recruitment, coding, and surveillance. Machine learning has been used to remove personal health information from clinical records, such that de-identified records may be made public for research purposes without obtaining individual informed consent (143). Studies have also used textual data and machine learning algorithms to identify patients who may qualify and benefit from participation in clinical studies (144). Furthermore, text mining can improve efficiency of systematic reviews by facilitating the identification, rapid categorization, and summarization of relevant literature (145). Finally, natural language processing of clinical documents can supplement manual surveillance, and has been used to identify a range of reportable post-operative complications (146).

In addition to clinical settings, text mining algorithms have been incorporated into automated infectious disease surveillance systems that acquire, classify, and process

web-accessible data. These algorithms can improve detection of early outbreaks and complement traditional surveillance efforts performed by government and international organizations. For example, HealthMap graphically displays areas where diseases are circulating by combining search query data, social media data, validated official reports, and expert-curated accounts (e.g., ProMED emails) (147,148). Similarly, the BioCaster system tracks infectious disease outbreaks on Google maps based upon residual sum of squares feeds (149).

Prediction and forecasting of infectious disease

Machine learning methods have been incorporated into prediction and forecasting models for infectious disease. For example, SVMs have been used to predict whether dengue incidence exceeded a chosen threshold using google search terms (150). Researchers have also used SVMs to predict levels of influenza-like illness from Twitter data 1-2 weeks before official reports (151). In addition, infectious disease forecasters have adopted ensemble-based methods traditionally used for meteorological and oceanographic predictions. For example, climate forecasting from multi-model ensembles has been adapted to produce early malaria warning systems (152). Moreover, ensemble-based forecasting methods based on sequential data assimilation approaches are increasingly common infectious disease forecasting tools, due to their ability to correct for various sources of uncertainty in mathematical simulations compared to traditional linear time-series models such as negative binomial models and ARIMA. One type of sequential ensemble filtering, Ensemble Adjustment Kalman Filter (EAKF), has been used to forecast seasonal outbreaks of influenza (153), to reconstruct

the transmission network of the 2014–2015 Ebola epidemic in Sierra Leone (154), and to retrospectively forecast West Nile virus (155) and respiratory syncytial virus cases (156).

BRIEF RECOMMENDATIONS

In this primer we discussed several important algorithms, but this is only the tip of the iceberg. We refer the readers to the herein-referenced machine learning textbooks for a more comprehensive review (2,5,31). Choice of algorithm is highly tied to research goals of its use, and there is no single recommendation for all projects. However, epidemiologists interested in adopting machine learning methodologies will often be most interested in accurate prediction in the context of a large number of covariates. In these cases, we encourage them to start with ensemble-based boosting or bagging approaches. Through refitting the same underlying model to different versions of a dataset, these ensembles are less susceptible to overfitting and less sensitive to tuning parameters. They are also easy to implement with many commonly available tools and packages, with random forests analysis being a popular choice. The Super Learner approach, which fits many different models to a set data, is also attractive as it allows simultaneous consideration of multiple algorithms and automates many of the best practices for fitting and validating machine learning models. However, as with traditional epidemiological or statistical approaches, a rigorous approach to assessing performance and appropriate matching of model to use are more important than the specific algorithm used.

Despite the benefits of boosting and bagging, as a general rule, these ensemble approaches add another stage to modeling, making them harder to interpret. Investigators should carefully consider their primary objective: is it predictive accuracy or interpretability? Where interpretability is important, as in many clinical applications, researchers might consider single, more easily understandable algorithms such as decision trees. However, many machine learning algorithms, particularly non-ensemble approaches, are prone to overfitting. Measures of fit alone (e.g., R-squared) should be interpreted with caution, as they can be effectively meaningless for some machine learning applications (157). Without a likelihood function, techniques such as AIC evaluation are not available metrics for assessing the generalizability of machine learning models, hence, cross-validation (whether *k*-fold, leave-one-out, or another approach) is a critical tool for evaluating model performance. These methods must be used appropriately, however, or they can fool the researcher. The testing and validation plan should be specified *a priori* and must be run on the full algorithm: e.g., if there is a data-based variable selection step, it should be run on each data partition used in cross-validation, not on the full dataset prior to the cross-validation. It is important that researchers are clear that these cross-validation approaches give expected out-of-dataset performance given the algorithm used, not an assessment of the particular fitted model, and that they recognize that the quality of these measures depends on the representativeness of the population and the correlation between observations in the training and testing sets (i.e., if there is high correlation, cross-validated performance will be deceptively high).

OPPORTUNITIES AND CHALLENGES

The field of machine learning is rapidly developing and can make any technical review seem obsolete within months. Growing interest in the field from the general public, as reflected in extensive coverage of self-driving cars and AlphaGo in the mainstream media, is accompanied by efforts from the machine learning community to make advanced machine learning technologies more accessible. Educational companies such as Udacity and Coursera have partnered with companies like Google and academic institutions to create online and freely available courses on machine learning and deep learning. In addition to the growing educational resources, large technological companies, including Google, IBM, and Amazon Web Services, are heavily investing in open-source machine learning that use data flow graphs to build models (e.g., TensorFlow (158)). The use of dataflow graphs in TensorFlow enables developers and data scientists to focus on the high-level overall logic of the algorithms rather than the technical coding details, which greatly increases reproducibility and optimizability of the models. Models built with TensorFlow can be integrated into mobile devices, making on-device/bedside diagnosis practical when combined with mobile sensors. The ability of TensorFlow to build and run models on the cloud also dramatically increases processing power and storage ability, which is particularly helpful for analyzing large datasets with complex algorithms. These machine learning developments continue to ease the entry barriers for epidemiologists interested in using advanced machine learning technologies, and they have the potential to transform epidemiological research.

Yet, there continue to be challenges that impede greater integration of machine learning into epidemiological research. Classically trained epidemiologists often lack the skills to

take full advantage of machine learning technologies, in part due to the continued popularity of closed-source programming (e.g., SAS, STATA) languages in epidemiology. In addition, despite the promise of 'Big Data,' logistical roadblocks to sharing de-identified patient data and amassing large healthcare datasets can make it challenging for epidemiologists to leverage these opportunities, particularly compared to the private sector. Even when data are available, epidemiologists should be mindful of the class-imbalance issue (see Table 1) often inherent in healthcare and surveillance data, which can pose challenges for many standard algorithms (159). Most importantly, a general lack of working knowledge on machine learning algorithms, despite their substantial methodological overlap with statistical methods, reduces the practical uptake of these techniques in the epidemiological literature despite their transformative potential.

Ultimately, advanced machine learning algorithms offer epidemiologists new tools to tackle problems that classical methods are not well-suited for, but they by no means serve as a cure-all for poor study design or poor data quality. Further eroding the cultural and language barriers between machine learning and epidemiology serves as an essential first step to understanding the value of, and achieving greater integration with, machine learning and existing epidemiological research methods.

FURTHER READING: MACHINE LEARNING RESOURCES FOR EPIDEMIOLOGISTS

Many machine learning articles and textbooks are written for an audience with a computer science background, and as a consequence, the language and terminology can be unfamiliar to epidemiologists. In order to help interested readers to further explore these topics, we have selected a sample of relatively easily-accessible articles that introduce the algorithms and ensemble models reviewed in this primer in greater detail:

- Artificial Neural Networks: Jain *et al*. (160); Olden *et al*. (161)
- Decision Trees: Therneau & Atkinson (162); Olden *et al*. (161)
- Support Vector Machines: Noble *et al*. (34)
- Naive Bayes: Lewis (36)
- K-means Clustering: Jain (46)
- Bayesian Model Averaging: Hoeting *et al.* (67)
- Super Learner: Polley & van der Laan (163)
- Boosting and Bagging: Opitz & Maclin (164)

In addition, *An Introduction to Statistical Learning* by James *et al.* provides an accessible overview of popular machine learning algorithms and discusses them in parallel with traditional statistical approaches (31). The supplemental 15-hour online tutorial discusses much of the same material in further detail and offers an alternative learning format. It is available at the website listed below. Both resources are open-access.

- The 15-hour online tutorial is available online. See reference (165).

Acknowledgment

We would like to thank Dr. Maria Glymour for her very helpful suggestions in the preparation of this manuscript. Conflict of interest: none declared

References

1. Samuel AL. Some Studies in Machine Learning Using the Game of Checkers. *IBM J. Res. Dev.* 1959;3(3):210–229.

2. Mitchell TM. *Machine Learning.* 1st ed. New York, NY: McGraw-Hill Education; 1997.

3. Rasmussen CE, Williams CKI. 1st ed. *Gaussian Processes for Machine Learning.* Cambridge, MA: Mit Press; 2006.

4. Breiman L. Statistical Modeling: The Two Cultures (with comments and a rejoinder by the author). *Stat. Sci.* 2001;16(3):199–231.

5. Duda RO, Hart PE, Stork DG. *Pattern Classification.* 2nd Ed. Hoboken, NJ: John Wiley & Sons; 2012 517 p.

6. Bartholomew DJ, Knott M, Moustaki I. *Latent Variable Models and Factor Analysis: A Unified Approach.* 3rd ed. Hoboken, NJ: John Wiley & Sons; 2011

7. Hennig C, Meila M, Murtagh F, et al. *Handbook of Cluster Analysis.* 1st ed. Boca Raton, FL: CRC Press; 2015 34p.

8. Bishop CM. *Pattern Recognition and Machine Learning.* 1st ed New York, NY: Springer; 2006 424 p.

9. Zhu X, Goldberg AB. *Introduction to Semi-supervised Learning.* 1st ed. San Rafael, CA: Morgan & Claypool Publishers; 2009 11 p.

10. Nigam K, Mccallum AK, Thrun S, et al. Text Classification from Labeled and

Unlabeled Documents using EM. *Mach. Learn.* 2000;39(2):103–134.

11. Ng AY, Jordan MI. On Discriminative vs. Generative Classifiers: A comparison of logistic regression and naive Bayes. In: Dietterich TG, Becker S, Ghahramani Z, eds. *Advances in Neural Information Processing Systems 14*. MIT Press; 2002:841–848. http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.19.9829. Accessed July 4th, 2019.
12. Vapnik VN. *Statistical learning theory.* 1st ed. Ed Hoboken, NJ: Wiley-Interscience; 1998 12-21 p.
13. Pernkopf F, Bilmes J. Discriminative versus generative parameter and structure learning of Bayesian network classifiers. In: *Proceedings of the 22nd international conference on Machine learning - ICML '05.* https://dl.acm.org/citation.cfm?id=1102434. Accessed July 4, 2019.
14. Sutton RS, Barto AG. Reinforcement Learning: An Introduction. *IEEE Trans. Neural Netw.* 1998;9(5):1054–1054.
15. Rao RPN. Reinforcement Learning: An Introduction; R.S. Sutton, A.G. Barto (Eds.); MIT Press, Cambridge, MA, 1998, 380 pages, ISBN 0-262-19398-1, *Neural Netw.* 2000;13(1):133–135.
16. Ganguly K. *Learning Generative Adversarial Networks.* 1st ed. Birmingham, United Kingdom: Packt Publishing; 2017.
17. Asoh H, Akaho MSS, Kamishima T, et al. An application of inverse reinforcement learning to medical records of diabetes treatment. In: *ECMLPKDD2013 Workshop*

on Reinforcement Learning with Generalized Feedback. 2013 https://pdfs.semanticscholar.org/5f44/548a3cd1932fc5035236b9be9018df5103c5.pdf. Accessed July 3, 2019

18. Shortreed SM, Laber E, Lizotte DJ, et al. Informing sequential clinical decision-making through reinforcement learning: an empirical study. *Mach. Learn.* 2011;84(1-2):109–136.

19. Nemati S, Ghassemi MM, Clifford GD. Optimal medication dosing from suboptimal clinical examples: a deep reinforcement learning approach. *Conf. Proc. IEEE Eng. Med. Biol. Soc.* 2016;2016:2978–2981.

20. Olden JD, Jackson DA. Illuminating the "black box": a randomization approach for understanding variable contributions in artificial neural networks. *Ecol. Modell.* 2002;154(1-2):135–150.

21. Rumelhart DE, Hinton GE, Williams RJ. Learning representations by back-propagating errors. *Nature.* 1986;323(6088):533–536.

20. McCulloch WS, Pitts W. A logical calculus of the ideas immanent in nervous activity. 1943. *Bull. Math. Biol.* 1990;52(1-2):99–115; discussion 73–97.

23. Duh MS, Walker AM, Ayanian JZ. Epidemiologic interpretation of artificial neural networks. *Am. J. Epidemiol.* 1998;147(12):1112–1122.

24. Papadokonstantakis S, Lygeros A, Jacobsson SP. Comparison of recent methods for inference of variable influence in neural networks. *Neural Netw.* 2006;19(4):500–513.

25. Beckmw. Variable importance in neural networks. *R-Bloggers*. 2013;(https://www.r-bloggers.com/variable-importance-in-neural-networks/). (Accessed June 19, 2018)

26. Hershey S, Chaudhuri S, Ellis DPW, et al. CNN Architectures for Large-Scale Audio Classification. Preprint. 2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), New Orleans, LA, 2017, pp. 131-135. https://ai.google/research/pubs/pub45611. Accessed July 15, 2019.

27. Breiman L, Friedman J, Stone CJ, et al. *Classification and Regression Trees.* 1st ed. London, United Kingdom: Chapman and Hall/CRC; 1984.

28. Kass GV. An Exploratory Technique for Investigating Large Quantities of Categorical Data. *Appl. Stat.* 1980;29(2):119.

29. Biggs D, De Ville B, Suen E. A method of choosing multiway partitions for classification and decision trees. *J. Appl. Stat.* 1991;18(1):49–62.

30. Quinlan JR. Induction of decision trees. *Mach. Learn.* 1986;1(1):81–106.

31. James G, Witten D, Hastie T, et al. *An Introduction to Statistical Learning: with Applications in R.* Ed New York, NY: Springer; 2013.

32. Almuallim H. An efficient algorithm for optimal pruning of decision trees. *Artif. Intell.* 1996;83(2):347–362.

33. Boulesteix A-L, Janitza S, Hapfelmeier A, et al. Letter to the Editor: On the term “interaction” and related phrases in the literature on Random Forests. *Brief. Bioinform.* 2015;16(2):338–345.

34. Aluja-Banet T, Nafria E. Stability and scalability in decision trees. *Comput. Stat.* 2003;18(3):505–520.

35. Boser BE, Guyon IM, Vapnik VN. A Training Algorithm for Optimal Margin Classifiers. In: *Proceedings of the Fifth Annual Workshop on Computational Learning Theory*. New York, NY, USA: ACM; 1992:144–152. http://www.svms.org/training/BOGV92.pdf. Accessed July 3, 2019

36. Noble WS. What is a support vector machine? *Nat. Biotechnol.* 2006;24(12):1565–1567.

37. Smola AJ, Schölkopf B. A tutorial on support vector regression. *Stat. Comput.* 2004;14(3):199–222.

38. Belousov, A. I., S. A. Verzakov, and J. von Frese. 2002. “A Flexible Classification Approach with Optimal Generalisation Performance: Support Vector Machines.” Chemometrics and Intelligent Laboratory Systems 64 (1): 15–25.

39. Guenther N, Schonlau M. Support vector machines. The Stata Journal. 16(4):917-937. http://www.schonlau.net/publication/16svm_stata.pdf. Accessed July 3, 2019

40. Lewis DD. *Naive (Bayes) at forty: The independence assumption in information retrieval.* In: Nédellec C, Rouveirol C, eds. Machine Learning: ECML-98. Berlin, Heidelberg: Springer Berlin Heidelberg; 1998:4–15.

41. Frank E, Trigg L, Holmes G, et al. Technical Note: Naive Bayes for Regression. Mach. Learn. 2000;41(1):5–25.

42. Rish I. An empirical study of the naive Bayes classifier. In IJCAI 2001 Workshop on Empirical Methods in Artificial Intelligence, 3:41–46. IBM New York. http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.330.2788&rep=rep1&type=pdf. Accessed July 3, 2019

43. Russek E, Kronmal RA, Fisher LD. The effect of assuming independence in applying Bayes' theorem to risk estimation and classification in diagnosis. *Comput. Biomed. Res.* 1983;16(6):537–552.

44. Hand DJ. Statistical methods in diagnosis. *Stat. Methods Med. Res.* 1992;1(1):49–67.

45. Stan User's Guide. Stan Development Team. Version 2.19. https://mc-stan.org/docs/2_19/stan-users-guide/naive-bayes-classification-and-clustering.html. Accessed Jul 20, 2019.

46. Jain AK. Data clustering: 50 years beyond K-means. *Pattern Recognit. Lett.* 2010;31(8):651–666.

47. Lloyd S. Least squares quantization in PCM. *IEEE Trans. Inf. Theory*. 1982;28(2):129–137.

48. Pham DT, Dimov SS, Nguyen CD. Selection of K in K-means clustering. Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science. 2005;219(1)

49. Tibshirani R, Walther G, Hastie T. Estimating the number of clusters in a data set via the gap statistic. *J. R. Stat. Soc. Series B Stat. Methodol.* 2001;63(2):411–423.

50. Raykov YP, Boukouvalas A, Baig F, et al. What to Do When K-Means Clustering Fails: A Simple yet Principled Alternative Algorithm. *PLoS One*. 2016;11(9):e0162259.

51. Breiman L. Bagging predictors. *Mach. Learn.* 1996;24(2):123–140.

52. Schapire, Robert E.; Freund, Yoav; Bartlett, Peter; Lee, Wee Sun. Boosting the margin: a new explanation for the effectiveness of voting methods. Ann. Statist. 1998; 26(5): 1651-1686.

53. Breiman L. Bias, variance, and arcing classifiers. 1996;(http://www.stat.berkeley.edu/~breiman/arcall96.pdf). (Accessed July 3, 2019)

54. Breiman L. Random Forests. *Mach. Learn.* 2001;45(1):5–32.

55. Breiman L. Manual on setting up, using, and understanding random forests v3. 1. *Statistics Department University of California Berkeley, CA, USA*. 2002;1. https://www.stat.berkeley.edu/~breiman/Using_random_forests_V3.1.pdf. Accessed July 3, 2019.

56. Friedman J, Hastie T, Tibshirani R. T*he elements of statistical learning: data mining, inference, and prediction.* 2nd ed. New York, NY: Springer; 2009.

57. van der Laan MJ. Statistical Inference for Variable Importance. *Int. J. Biostat.* 2006;2(1), ISSN (Online) 1557-4679.

58. Maldonado M, Dean J, Czika W, Haller S. Leveraging Ensemble Models in SAS Enterprise Miner. SAS Institute Inc.

https://support.sas.com/resources/papers/proceedings14/SAS133-2014.pdf. Accessed July 3, 2019.

59. Schapire RE. The Strength of Weak Learnability. *Machine Learning*. 1990:5, 197-227.

60. Freund Y. Boosting a Weak Learning Algorithm by Majority. *Inform. and Comput.* 1995;121(2):256–285.

61. Schapire RE. A brief introduction to boosting. In: *Ijcai*. 1999:1401–1406. http://rob.schapire.net/papers/Schapire99c.pdf. Accessed July 3, 2019

62. Freund Y, Schapire RE. A desicion-theoretic generalization of on-line learning and an application to boosting. *Journal of Computer and System Sciences.* 1997;55(1):119-139.

63. Friedman JH. Greedy Function Approximation: A Gradient Boosting Machine. *Ann. Stat.* 2001;29(5):1189–1232.

64. Friedman JH. Stochastic gradient boosting. *Comput. Stat. Data Anal.* 2002;38(4):367–378.

65. Schonlau M. Boosted regression (boosting): An introductory tutorial and a Stata plugin. The Stata Journal. 2005: 5(3): 330-354. https://www.stata-journal.com/sjpdf.html?articlenum=st0087. Accessed July 3, 2019.

66. Maidonado M, Dean J, Czika W, Haller S. Leveraging ensemble models in SAS Enterprise Miner.

https://support.sas.com/resources/papers/proceedings14/SAS133-2014.pdf. Accessed July 15, 2019

67. Hoeting JA, Madigan D, Raftery AE, et al. Bayesian model averaging: a tutorial. *Stat. Sci.* 1999:14(4):382-417.

68. Domingos P. Bayesian Averaging of Classifiers and the Overfitting Problem. In: *IN PROC. 17TH INTERNATIONAL CONF. ON MACHINE LEARNING*. 2000 (http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.34.9147). (Accessed August 27, 2017)

69. Monteith K, Carroll JL, Seppi K, et al. Turning Bayesian model averaging into Bayesian model combination. In: *The 2011 International Joint Conference on Neural Networks*. 2011:2657–2663. http://axon.cs.byu.edu/papers/Kristine.ijcnn2011.pdf. Accessed July 3, 2019

70. Whitney M, Ngo L. Bayesian Model Averaging Using SAS Software. SAS SUGI 29: 203-29. http://www2.sas.com/proceedings/sugi29/203-29.pdf. Accessed July 3, 2019

71. van der Laan MJ, Polley EC, Hubbard AE. Super learner. *Stat. Appl. Genet. Mol. Biol.* 2007;6:Article25.

72. Sinisi SE, Polley EC, Petersen ML, et al. Super learning: an application to the prediction of HIV-1 drug resistance. *Stat. Appl. Genet. Mol. Biol.* 2007;6:Article7.

73. Wolpert DH. Stacked generalization. Neural Netw. 1992;5(2):241–259.

74. Hernán MA, Robins JM. Using Big Data to Emulate a Target Trial When a Randomized Trial Is Not Available. *Am. J. Epidemiol.* 2016;183(8):758–764.

75. Rosenbaum PR, Rubin DB. Reducing Bias in Observational Studies Using Subclassification on the Propensity Score. *Journal of the American Statistical Association.* 1984;79(897):516-524.

76. Westreich D, Lessler J, Funk MJ. Propensity score estimation: neural networks, support vector machines, decision trees (CART), and meta-classifiers as alternatives to logistic regression. *J. Clin. Epidemiol.* 2010;63(8):826–833.

77. Lee BK, Lessler J, Stuart EA. Improving propensity score weighting using machine learning. *Stat. Med.* 2010;29(3):337–346.

78. Pirracchio R, Petersen ML, van der Laan M. Improving propensity score estimators' robustness to model misspecification using super learner. *Am. J. Epidemiol.* 2015;181(2):108–119.

79. Watkins S, Jonsson-Funk M, Brookhart MA, et al. An empirical comparison of tree-based methods for propensity score estimation. *Health Serv. Res.* 2013;48(5):1798–1817.

80. Schnitzer ME, Lok JJ, Gruber S. Variable Selection for Confounder Control, Flexible Modeling and Collaborative Targeted Minimum Loss-Based Estimation in Causal Inference. Int. J. Biostat. 2016;12(1):97–115.

81. Moodie EEM, Stephens DA. Treatment Prediction, Balance, and Propensity Score Adjustment. Epidemiology. 2017;28(5):e51–e53.

82. Bahamyirou A, Blais L, Forget A, et al. Understanding and diagnosing the potential for bias when using machine learning methods with doubly robust causal estimators. Stat. Methods Med. Res. 2019 Jun;28(6):1637-1650.

83. Kugler KC, Vasilenko SA, Butera NM, et al. Long-term consequences of early sexual initiation on young adult health: A causal inference approach. *J. Early Adolesc.* 2017;37(5):662–676.

84. Oppermann M, Fritzsche J, Weber-Schoendorfer C, et al. A(H1N1)v2009: a controlled observational prospective cohort study on vaccine safety in pregnancy. *Vaccine*. 2012;30(30):4445–4452.

85. Tamma PD, Turnbull AE, Harris AD, et al. Less is more: combination antibiotic therapy for the treatment of gram-negative bacteremia in pediatric patients. *JAMA Pediatr.* 2013;167(10):903–910.

86. Schuler MS, Rose S. Targeted Maximum Likelihood Estimation for Causal Inference in Observational Studies. *Am. J. Epidemiol.* 2017;185(1):65–73.

87. Ahern J, Balzer L, Galea S. The roles of outlet density and norms in alcohol use disorder. *Drug Alcohol Depend.* 2015;151:144–150.

88. Ahern J, Karasek D, Luedtke AR, et al. Racial/Ethnic Differences in the Role of Childhood Adversities for Mental Disorders Among a Nationally Representative Sample of Adolescents. *Epidemiology*. 2016;27(5):697–704.

89. Athey S, Imbens G. Recursive partitioning for heterogeneous causal effects. *Proc. Natl. Acad. Sci. U. S. A.* 2016;113(27):7353–7360.

90. Wager S, Athey S. Estimation and Inference of Heterogeneous Treatment Effects using Random Forests. *J. Am. Stat. Assoc.* 2017;1–15.

91. Scutari M, Denis JB. *Bayesian networks: with examples in R.* 1st ed. London, United Kingdom: Chapman and Hall/CRC; 2014.

92. Shimizu S, Hoyer PO, Hyvärinen A, et al. A Linear Non-Gaussian Acyclic Model for Causal Discovery. *J. Mach. Learn. Res.* 2006;7(Oct):2003–2030.

93. Hoyer PO, Shimizu S, Kerminen AJ, et al. Estimation of causal effects using linear non-Gaussian causal models with hidden variables. *Int. J. Approx. Reason.* 2008;49(2):362–378.

94. Shimizu S. LINGAM: NON-GAUSSIAN METHODS FOR ESTIMATING CAUSAL STRUCTURES. *Behaviormetrika*. 2014;41(1):65–98.

95. Sachs K, Perez O, Pe'er D, et al. Causal protein-signaling networks derived from multiparameter single-cell data. *Science*. 2005;308(5721):523–529.

96. Schadt EE, Lamb J, Yang X, et al. An integrative genomics approach to infer causal associations between gene expression and disease. *Nat. Genet.* 2005;37(7):710–717.

97. Rosenström T, Jokela M, Puttonen S, et al. Pairwise measures of causal direction in the epidemiology of sleep problems and depression. *PLoS One*. 2012;7(11):e50841.

98. Helajärvi H, Rosenström T, Pahkala K, et al. Exploring causality between TV

viewing and weight change in young and middle-aged adults. The Cardiovascular Risk in Young Finns study. *PLoS One*. 2014;9(7):e101860.

99. Warner HR, Toronto AF, Veasey LG, et al. A mathematical approach to medical diagnosis. Application to congenital heart disease. *JAMA*. 1961;177:177–183.

100. Blumenthal DT, Artzi M, Liberman G, et al. Classification of High-Grade Glioma into Tumor and Nontumor Components Using Support Vector Machine. *AJNR Am. J. Neuroradiol.* 2017;38(5):908–914.

101. Artzi M, Liberman G, Nadav G, et al. Differentiation between treatment-related changes and progressive disease in patients with high grade brain tumors using support vector machine classification based on DCE MRI. *J. Neurooncol.* 2016;127(3):515–524.

102. Zarinabad N, Abernethy LJ, Avula S, et al. Application of pattern recognition techniques for classification of pediatric brain tumors by in vivo 3T (1) H-MR spectroscopy-A multi-center study. *Magn Reson Med. 2018 Apr;79(4):2359-2366*

103. Chang Y, Paul AK, Kim N, et al. Computer-aided diagnosis for classifying benign versus malignant thyroid nodules based on ultrasound images: A comparison with radiologist-based assessments. *Med. Phys.* 2016;43(1):554.

104. El-Naqa I, Yang Y, Wernick MN, et al. A support vector machine approach for detection of microcalcifications. *IEEE Trans. Med. Imaging*. 2002;21(12):1552–1563.

105. Polat K, Güneş S. Breast cancer diagnosis using least square support vector

machine. *Digit. Signal Process.* 2007;17(4):694–701.

106. Akay MF. Support vector machines combined with feature selection for breast cancer diagnosis. *Expert Syst. Appl.* 2009/3;36(2, Part 2):3240–3247.

107. Wang Z-L, Zhou Z-G, Chen Y, et al. Support Vector Machines Model of Computed Tomography for Assessing Lymph Node Metastasis in Esophageal Cancer with Neoadjuvant Chemotherapy. *J. Comput. Assist. Tomogr.* 2017;41(3):455–460.

108. Zhang X-P, Wang Z-L, Tang L, et al. Support vector machine model for diagnosis of lymph node metastasis in gastric cancer with multidetector computed tomography: a preliminary study. *BMC Cancer*. 2011;11(1).

109. Brown MP, Grundy WN, Lin D, et al. Knowledge-based analysis of microarray gene expression data by using support vector machines. *Proc. Natl. Acad. Sci. U. S. A.* 2000;97(1):262–267.

110. Furey TS, Cristianini N, Duffy N, et al. Support vector machine classification and validation of cancer tissue samples using microarray expression data. *Bioinformatics*. 2000;16(10):906–914.

111. Golub TR, Slonim DK, Tamayo P, et al. Molecular classification of cancer: class discovery and class prediction by gene expression monitoring. *Science*. 1999;286(5439):531–537.

112. Pomeroy SL, Tamayo P, Gaasenbeek M, et al. Prediction of central nervous system embryonal tumour outcome based on gene expression. *Nature*.

2002;415(6870):436–442.

113. Orrù G, Pettersson-Yeo W, Marquand AF, et al. Using Support VectorMachine to identify imaging biomarkers of neurological and psychiatric disease: a critical review. *Neurosci. Biobehav. Rev.* 2012;36(4):1140–1152.

114. Costafreda SG, Chu C, Ashburner J, et al. Prognostic and diagnostic potential of the structural neuroanatomy of depression. *PLoS One*. 2009;4(7):e6353.

115. Costafreda SG, Khanna A, Mourao-Miranda J, et al. Neural correlates of sad faces predict clinical remission to cognitive behavioural therapy in depression. *Neuroreport*. 2009;20(7):637–641.

116. Gong Q, Wu Q, Scarpazza C, et al. Prognostic prediction of therapeutic response in depression using high-field MR imaging. *Neuroimage*. 2011;55(4):1497–1503.

117. Palaniappan S, Awang R. Intelligent heart disease prediction system using data mining techniques. In: *2008 IEEE/ACS International Conference on Computer Systems and Applications*. 2008:108–115. http://paper.ijcsns.org/07_book/200808/20080849.pdf. Accessed July 3, 2019.

118. Jaimes F, Farbiarz J, Alvarez D, et al. Comparison between logistic regression and neural networks to predict death in patients with suspected sepsis in the emergency room. *Crit. Care*. 2005;9(2):R150–6.

119. Launay CP, Rivière H, Kabeshova A, et al. Predicting prolonged length of hospital stay in older emergency department users: use of a novel analysis method, the Artificial Neural Network. *Eur. J. Intern. Med.* 2015;26(7):478–482.

120. Demšar J, Zupan B, Aoki N, et al. Feature mining and predictive model construction from severe trauma patient's data. *Int. J. Med. Inform.* 2001/9;63(1–2):41–50.

121. Levin S, Toerper M, Hamrock E, et al. Machine-Learning-Based Electronic Triage More Accurately Differentiates Patients With Respect to Clinical Outcomes Compared With the Emergency Severity Index. *Ann. Emerg. Med.* 2018;71(5):565–574.e2.

122. Brims FJH, Meniawy TM, Duffus I, et al. A Novel Clinical Prediction Model for Prognosis in Malignant Pleural Mesothelioma Using Decision Tree Analysis. *J. Thorac. Oncol.* 2016;11(4):573–582.

123. Goodman KE, Lessler J, Cosgrove SE, et al. A Clinical Decision Tree to Predict Whether a Bacteremic Patient Is Infected With an Extended-Spectrum β-Lactamase-Producing Organism. *Clin. Infect. Dis.* 2016;63(7):896–903.

124. Dias CC, Pereira Rodrigues P, Fernandes S, et al. The risk of disabling, surgery and reoperation in Crohn's disease - A decision tree-based approach to prognosis. *PLoS One*. 2017;12(2):e0172165.

125. Silva-Alves MS, Secolin R, Carvalho BS, et al. A Prediction Algorithm for Drug Response in Patients with Mesial Temporal Lobe Epilepsy Based on Clinical and Genetic Information. *PLoS One*. 2017;12(1):e0169214.

126. Nguyen T-T, Huang J, Wu Q, et al. Genome-wide association data classification and SNPs selection using two-stage quality-based Random Forests. *BMC*

Genomics. 2015;16 Suppl 2:S5.

127. Briones N, Dinu V. Data mining of high density genomic variant data for prediction of Alzheimer's disease risk. *BMC Med. Genet.* 2012;13:7.

128. Wei W, Visweswaran S, Cooper GF. The application of naive Bayes model averaging to predict Alzheimer's disease from genome-wide data. *J. Am. Med. Inform. Assoc.* 2011;18(4):370–375.

129. Altman NS. An Introduction to Kernel and Nearest-Neighbor Nonparametric Regression. *Am. Stat.* 1992;46(3):175.

130. Kim B-J, Kim S-H. Prediction of inherited genomic susceptibility to 20 common cancer types by a supervised machine-learning method. *Proc. Natl. Acad. Sci. U. S. A.* 2018;115(6):1322–1327.

131. Montassier E, Al-Ghalith GA, Ward T, et al. Pretreatment gut microbiome predicts chemotherapy-related bloodstream infection. *Genome Med.* 2016;8(1):49.

132. Upstill-Goddard R, Eccles D, Fliege J, et al. Machine learning approaches for the discovery of gene-gene interactions in disease data. *Brief. Bioinform.* 2013;14(2):251–260.

133. Naushad SM, Ramaiah MJ, Pavithrakumari M, et al. Artificial neural network-based exploration of gene-nutrient interactions in folate and xenobiotic metabolic pathways that modulate susceptibility to breast cancer. *Gene*. 2016;580(2):159–168.

134. Stevens FR, Gaughan AE, Linard C, et al. Disaggregating census data for population mapping using random forests with remotely-sensed and ancillary data. *PLoS One*. 2015;10(2):e0107042.

135. Bhatt S, Gething PW, Brady OJ, et al. The global distribution and burden of dengue. *Nature*. 2013;496(7446):504–507.

136. Pigott DM, Bhatt S, Golding N, et al. Global distribution maps of the leishmaniases. *Elife*. 2014;3:e02851.

137. Pigott DM, Golding N, Mylne A, et al. Mapping the zoonotic niche of Ebola virus disease in Africa. *Elife*. 2014;3:e04395.

138. Messina JP, Pigott DM, Golding N, et al. The global distribution of Crimean-Congo hemorrhagic fever. *Trans. R. Soc. Trop. Med. Hyg.* 2015;109(8):503–513.

139. Messina JP, Kraemer MU, Brady OJ, et al. Mapping global environmental suitability for Zika virus. *Elife*. 2016 Apr 19;5. Pii:15272.

140. Perkins TA, Siraj AS, Ruktanonchai CW, et al. Model-based projections of Zika virus infections in childbearing women in the Americas. *Nat Microbiol*. 2016;1(9):16126.

141. i2b2: Informatics for Integrating Biology & the Bedside. (https://www.i2b2.org/about/index.html). (Accessed May 20, 2018)

142. Uzuner O, Solti I, Cadag E. Extracting medication information from clinical text. *J. Am. Med. Inform. Assoc.* 2010;17(5):514–518.

143. Meystre SM, Friedlin FJ, South BR, et al. Automatic de-identification of textual documents in the electronic health record: a review of recent research. *BMC Med. Res. Methodol.* 2010;10:70.

144. Pakhomov S, Weston SA, Jacobsen SJ, et al. Electronic medical records for clinical research: application to the identification of heart failure. *Am. J. Manag. Care*. 2007;13(6 Part 1):281–288.

145. Thomas J, McNaught J, Ananiadou S. Applications of text mining within systematic reviews. *Res Synth Methods*. 2011;2(1):1–14.

146. Murff HJ, FitzHenry F, Matheny ME, et al. Automated identification of postoperative complications within an electronic medical record using natural language processing. *JAMA*. 2011;306(8):848–855.

147. Brownstein JS, Freifeld CC, Reis BY, et al. Surveillance Sans Frontières: Internet-based emerging infectious disease intelligence and the HealthMap project. *PLoS Med.* 2008;5(7):e151.

148. Freifeld CC, Mandl KD, Reis BY, et al. HealthMap: global infectious disease monitoring through automated classification and visualization of Internet media reports. *J. Am. Med. Inform. Assoc.* 2008;15(2):150–157.

149. Collier N, Doan S, Kawazoe A, et al. BioCaster: detecting public health rumors with a Web-based text mining system. *Bioinformatics*. 2008;24(24):2940–2941.

150. Althouse BM, Ng YY, Cummings DAT. Prediction of dengue incidence using search query surveillance. *PLoS Negl. Trop. Dis.* 2011;5(8):e1258.

151. Signorini A, Segre AM, Polgreen PM. The use of Twitter to track levels of disease activity and public concern in the U.S. during the influenza A H1N1 pandemic. *PLoS One*. 2011;6(5):e19467.

152. Thomson MC, Doblas-Reyes FJ, Mason SJ, et al. Malaria early warnings based on seasonal climate forecasts from multi-model ensembles. *Nature*. 2006;439(7076):576–579.

153. Shaman J, Karspeck A. Forecasting seasonal outbreaks of influenza. *Proc. Natl. Acad. Sci. U. S. A.* 2012;109(50):20425–20430.

154. Yang W, Zhang W, Kargbo D, et al. Transmission network of the 2014–2015 Ebola epidemic in Sierra Leone. *J. R. Soc. Interface*. 2015;12(112):20150536.

155. DeFelice NB, Little E, Campbell SR, et al. Ensemble forecast of human West Nile virus cases and mosquito infection rates. *Nat. Commun.* 2017;8:14592.

156. Reis J, Shaman J. Retrospective Parameter Estimation and Forecast of Respiratory Syncytial Virus in the United States. *PLoS Comput. Biol.* 2016;12(10):e1005133.

157. Mountford MD, Steel RGD, Torrie JH. Principles and Procedures of Statistics with Special Reference to the Biological Sciences. *Biometrics*. 1962;18(1):127.

158. Abadi M, Agarwal A, Barham P, et al. Tensorflow: Large-scale machine learning on heterogeneous distributed systems, 2015. *arXiv preprint arXiv:1603. 04467*. 2015;

159. Khalilia M, Chakraborty S, Popescu M. Predicting disease risks from highly imbalanced data using random forest. *BMC Med. Inform. Decis. Mak.* 2011;11:51.

160. Jain AK, Mao J, Mohiuddin KM. Artificial neural networks: a tutorial. *Computer* . 1996;29(3):31–44.

161. Olden JD, Lawler JJ, Poff NL. Machine learning methods without tears: a primer for ecologists. *Q. Rev. Biol.* 2008;83(2):171–193.

162. Atkinson TM, Therneau EJ. An Introduction to Recursive Partitioning Using the RPART Routines. The Mayo Clinic, Department of Biostatistics; 2018. https://cran.r-project.org/web/packages/rpart/vignettes/longintro.pdf. Accessed July 3, 2019

163. Polley EC, van der Laan MJ. Super Learner In Prediction. 2010;(http://biostats.bepress.com/ucbbiostat/paper266/). (Accessed May 20, 2018)

164. Opitz D, Maclin R. Popular Ensemble Methods: An Empirical Study. *1*. 1999;11:169–198.

165. Markham K. In-depth introduction to machine learning in 15 hours of expert videos. https://www.r-bloggers.com/in-depth-introduction-to-machine-learning-in-15-hours-of-expert-videos/. Accessed July 3, 2019.

Table 1. Glossary of Machine Learning and Epidemiology Terminology

Machine Learning Term(s)	**Epidemiology Term(s)**	**Definition and Notes**	***Example***
Attribute, Feature, Predictor, or Field	Independent variable	Machine learning uses various terms to reference what epidemiologists would consider an "independent variable," including "attribute," "feature," "predictor," or "field".	*In a dataset with four independent variables (BMI, Age, Race, and SES) and a dependent variable (diabetes), BMI, age, race and SES are attributes.*
Domain	Range of possible variable values	The domain is the set of possible values of an attribute. It can be continuous or categorical/binary.	*If race is recorded in a dataset as 1=Caucasian, 2=African-American, and 3=Other, its domain is categorical and includes only the three referenced categories.*
Input and output	Independent (exposure) and	In machine learning, "input" refers to all of the predictors or independent variables that	*BMI, age, race, and SES are model input. In a binary classification algorithm, the*

	dependent (outcome) variables	enter your model, and "output" generally refers to the predicted value (whether a number, classification, etc.) of the dependent variable or outcome.	*model output is a prediction of whether a subject does (=1) or does not (=0) have diabetes.*
Classifier, Estimator	Model	"Classifiers" or "estimators" are used generally in the machine learning literature to refer to algorithms that perform a prediction or classification of interest. Their less common, although more technical, usage, specifically refers to fully parameterized models that are used to predict or classify.	*A decision tree is one type of machine learning classifier (general usage). The more specific usage of this term would refer only to a parameterized decision tree that has been fit in a dataset (e.g., that predicts diabetes outcomes from BMI, age, sex, and SES).*
Learner	Model fitting algorithm	A learner inputs a training set and outputs a classifier. Usually, but not always,	*In decision tree learning, the classification and regression tree algorithm,*

		learner refers to the fitting algorithm, while classifier the fitted model.	*developed by Breiman et al (27). in 1984, is one of multiple available learners for developing a decision tree classifier.*
Dimensionality	Number of covariates	Number of independent variables under consideration in a model.	*A dataset with four independent variables (BMI, Age, Race, and SES) and a dependent variable (diabetes), has four dimensions.*
Label	Value of dependent variables, outcomes	A variable's label is its value for each observation (e.g, 0 or 1). Although labels can technically describe any variable, common shorthand is that "labeled data" refers to data in which the dependent variable assumes a value for all observations.	*In a dataset in which you have collected information on diabetes status (outcome) for all subjects, this is "labeled" data. The label for diabetes is 0 or 1. Partially labeled data would have diabetes status missing for some subjects.*

Imbalanced Data	A dataset in which some cases or risk categories occur much less frequently than the others.	In imbalanced machine learning datasets, outcomes or another risk category of interest occur much less frequently, either due to the intrinsic nature of the problem (e.g., a rare disease in a database of medical records) or due to the sampling strategy (e.g., prevalence of cases in the study population is much lower than that in the target/source population). Heavily imbalanced data may pose challenges in some classification algorithms and require tuning parameters in order to correct for or otherwise address this imbalance. One method for addressing imbalanced datasets is to “balance” them	*Assume a hypothetical dataset of pediatric, normal-weight patients in which the prevalence of diabetes is 2%. This data is imbalanced because the outcome is very rare, which can lead to poor sensitivity of classification algorithms without parameter tuning or other corrective methods. This imbalance is due to the intrinsic nature of the population we are evaluating (i.e., healthy children) and not due to sampling strategy or other bias.*

		artificially, either by oversampling instances of the minority class or undersampling instances of the majority class.	
Loss function	Error measure	In machine learning, a loss function is generally considered a penalty for misclassification when assessing a model's predictive performance.	*A simple loss function may be the absolute value of (Predicted Value - True Value). If a model predicts a subject has diabetes (=1) and the subject does not (=0), the value of the loss function for this prediction is "1."*

BMI - Body Mass Index; SES - Socioeconomic Status

Table 2. Matrix of Joint Probabilities for BMI (x) and Diabetes (y) in a Dataset With Four Dichotomized Observations: (1, 0) (1, 0) (0, 1) (0, 0)

	Overweight BMI = 1	**Overweight BMI = 0**
Diabetes = 1	0	1/4
Diabetes = 0	1/2	1/4

BMI - Body Mass Index

Figure 1.

A) A single artificial neuron, also called a perceptron; B) A hypothetical feed-forward neural network examining the relationship between clinical and demographic predictors and a numerical outcome, fasting blood sugar level. Line (axon) thickness reflects input weight, and line type indicates direction of effect (solid = excitatory or positive; dashed = inhibitory or negative). Lack of a line (e.g., connecting 'sex' to neuron C) indicates no input. Connections between input and output layers are exclusively mediated through the hidden layer (more complex artificial neural networks can have multiple hidden layers). At hidden layer neuron A we observe that both body mass index (BMI) and age exert positive inputs, and they demonstrate interactive effects with each other and race (the latter's input is negative, as indicated by the dashed line). The weighted sum of these inputs results in activation of neuron A and positive output. In contrast, neuron B converts inputs from age, socio-economic status (SES), and race into negative output (inversely correlated with fasting blood sugar), while neuron C's inputs fail to surpass the activation function threshold, i.e., there is no effect on the outcome mediated through neuron C.

Abbreviations: BMI, body mass index; SES, social economic status.

Figure 2.

A hypothetical classification decision tree for predicting a binary outcome, type II diabetes. Body mass index (BMI) occupies the root node (most discriminatory variable in the dataset); age, consumption of sweetened beverages, and physical activity occupy daughter nodes; and predicted diabetes status (yes/no) is reflected in the terminal or "leaf" nodes. Terminal node predictions proceed based upon simple majority-rule (e.g., if 60% of patients in a terminal node are diabetes-positive, the entire terminal node will be classified as "Diabetes"). The cut-points for the continuous variables, BMI and age, are algorithm-derived. The presence of age at different cut-points in two different daughter nodes reflects likely interaction effects: the relationship between age and diabetes differs in patients with BMI ≤32 compared to patients with BMIs >32 who do not routinely consume sweetened beverages.

Abbreviations: BMI, body mass index.

Figure 3.

A) Hypothetical age and body mass index (BMI) distribution of diabetic and non-diabetic patients (black vs. grey) in two-dimensional space. B) After transformation, these dots/patients who are not linearly separable in two-dimensional space become linearly separable in three-dimensional space. A hyperplane in three-dimensional space is shown as a surface. Abbreviations: BMI, body mass index.

Figure 4.

A hypothetical Naive Bayes algorithm for predicting a binary outcome, type II diabetes, among the sub-population whose body mass index (BMI) is over 32, age is over 55, and who are female. Prior probability of the class (e.g., diabetes status) and a product of the likelihood functions, one for each patient characteristic, determine the class assignment. If the posterior probability of being diabetic in this population, P(D+|BMI>32 & Age>55 & Female), is larger than the posterior probability of *not* being diabetic in this population, P(D-|BMI >32 & Age >55 & Female), then this population would be classified as having diabetes. Prior probability of being diabetic, P(D+), approximates the overall diabetes prevalence. Due to the independence assumption, the likelihood of observing people with this set of attributes, BMI >32 & Age >55 & Female, among the diabetics (i.e., P(BMI >32 & Age >55 & Female | D+)) can be approximated by the product of observing each attribute among the diabetics (i.e., P(BMI >32 | D+) * P(Age >55 | D+) * P(Female | D+)). For example, P(BMI >32 | D+) represents, among the diabetics, the likelihood of observing people with BMI>32.

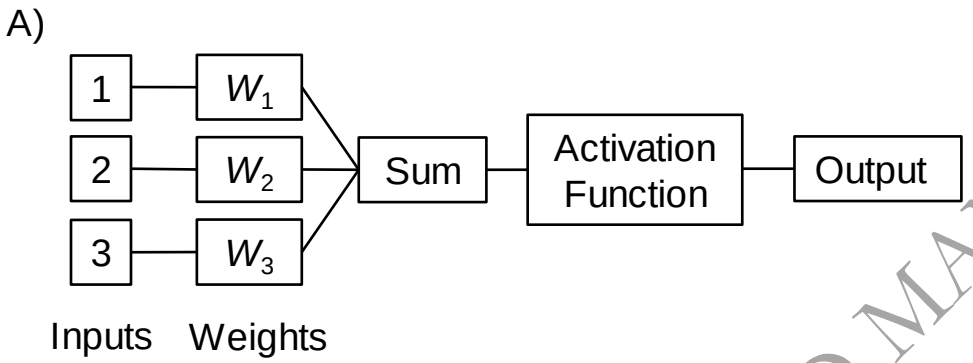
Abbreviations: BMI, body mass index.

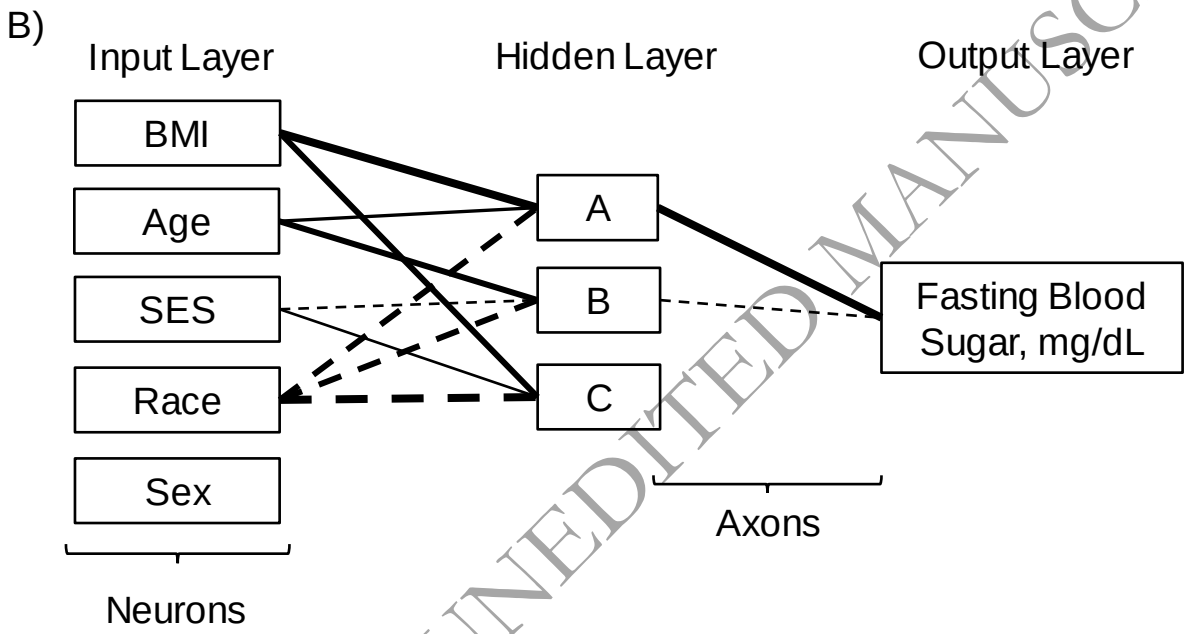
Figure 5.

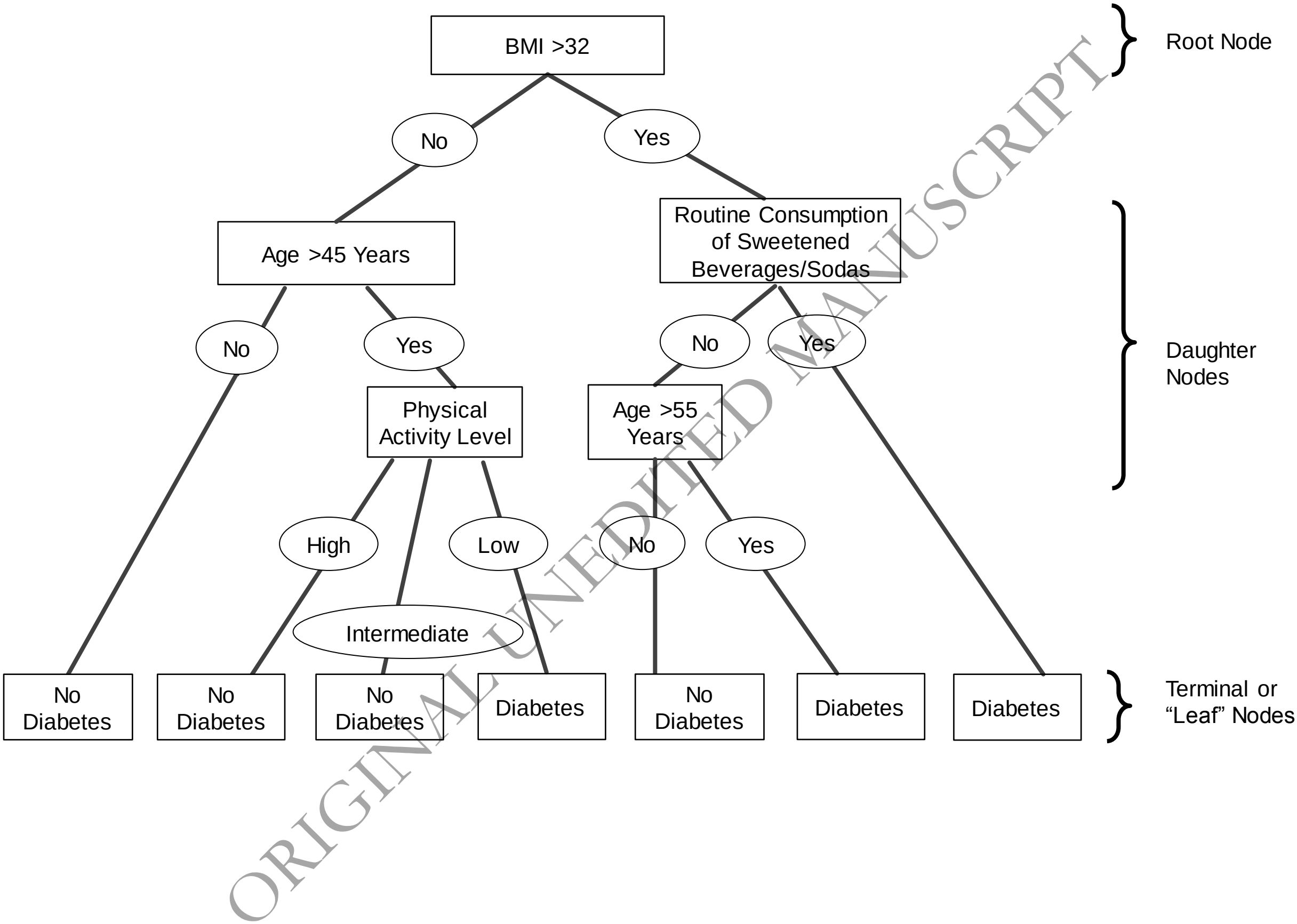
A) A hypothetical k-means algorithm for dichotomizing (k=2) patients based upon their age and body mass index (BMI). Each unclassified observation (hollow dots) is assigned to a diabetes classification (solid dots), with black and grey representing the predicted diabetes classifications at each step. Squares are centroids, with a single centroid per cluster. A) Unclassified data; B) randomly select k=2 centroids; C) assign each observation to its nearest centroid and predict its diabetes status (black dots are closer to the black square and grey dots are closer to the grey square); C) move the black centroid to the mean of all black dots, and similarly for the grey dots, as represented by centroid arrows; D) reclassify observations to the nearest, updated centroid; E) repeat step C, and F) final classifications, assuming clusters have stabilized. Abbreviations: BMI, body mass index.

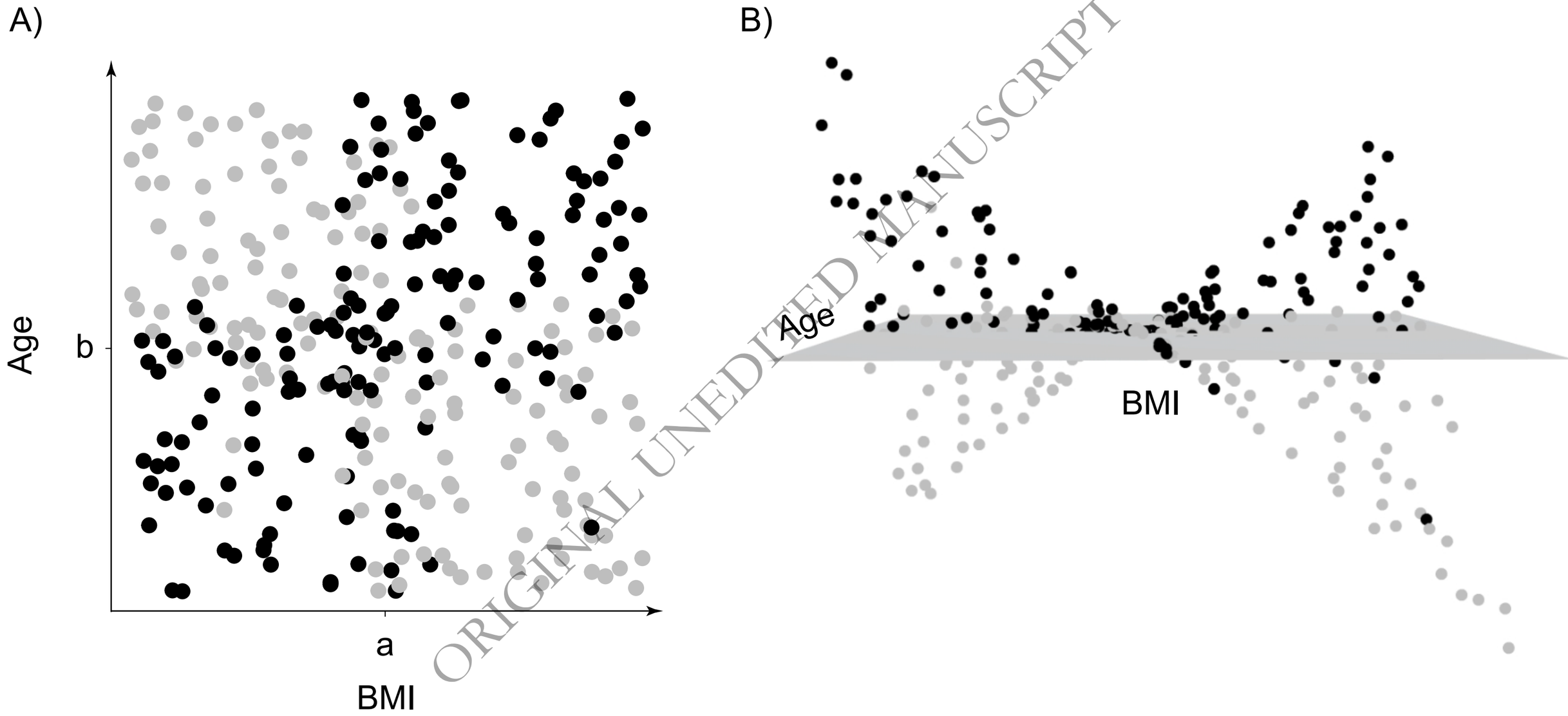
Figure 6.

One limitation of k-means algorithm, as illustrated with simulated data. When one cluster (upper right) is much larger than the other (lower left), k-means can produce counter-intuitive classifications (A) when the more intuitive classification is shown in the right panel (B).









Posterior Probability of Diabetes Status = Prior Probability × Likelihood

$$P(D+ \mid BMI > 32 \text{ and } Age > 55 \text{ and } Sex = F) = P(D+) \times \text{Product of} \begin{cases} P(BMI > 32 \mid D+) \\ P(Age > 55 \mid D+) \\ P(Sex = F \mid D+) \end{cases}$$

$$P(D- \mid BMI > 32 \text{ and } Age > 55 \text{ and } Sex = F) = P(D-) \times \text{Product of} \begin{cases} P(BMI > 32 \mid D-) \\ P(Age > 55 \mid D-) \\ P(Sex = F \mid D-) \end{cases}$$