Introduction to Machine Learning

Ryan Miller

Grinnell College
Statistics

What is Machine Learning?

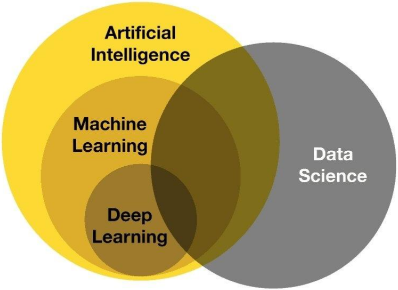
- How would you define “machine learning”?

What is Machine Learning?

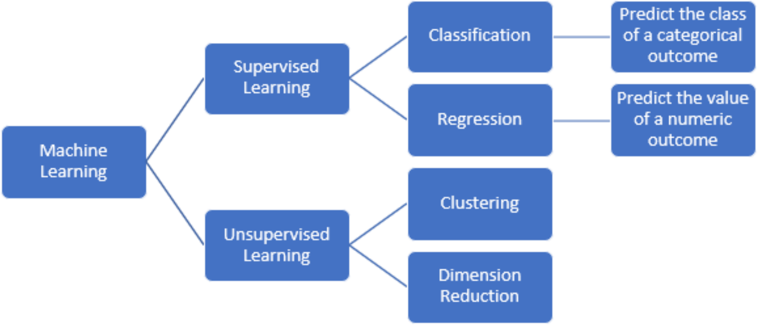
- How would you define “machine learning”?
  - IBM: a branch of artificial intelligence (AI) focusing on the use of data and algorithms to imitate the way humans learn (gradually improving with experience)

What is Machine Learning?

- How would you define "machine learning"?
  - IBM: a branch of artificial intelligence (AI) focusing on the use of data and algorithms to imitate the way humans learn (gradually improving with experience)



An Overview of Machine Learning



Getting Started

We'll begin by working with **tabular data** that is organized in a matrix, **X**, consisting of n samples (rows) with p features (columns)

$$\mathbf{X} = \begin{pmatrix} x_{1,1} & x_{1,2} & \cdots & x_{1,p} \\ x_{2,1} & x_{2,2} & \cdots & x_{2,p} \\ \vdots & \vdots & \ddots & \vdots \\ x_{n,1} & x_{n,2} & \cdots & x_{n,p} \end{pmatrix}$$

- Bold capital letters denote a matrix, bold lower-case letters denote a vector
- Samples are indexed by i (ie: $\mathbf{x}_i$ denotes all data for the i^{th} sample), features are indexed by j (ie: $\mathbf{x}_j$ denotes data for all samples for the j^{th} variable)

Getting Started (cont.)

In supervised learning we focus on a target variable, Y. We'll adopt the general framework for regression tasks:

$$y_i = f(\mathbf{x}_i) + \text{Noise}$$

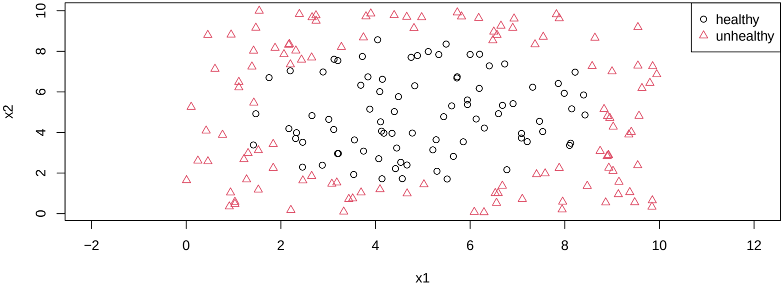
And a similar framework for classification tasks:

$$Pr(Y = y_i) = f(\mathbf{x_i}) + \text{Noise}$$

Here $f()$ is some unknown function of the data, and our goal is to approximate the behavior of $f()$ with an algorithm

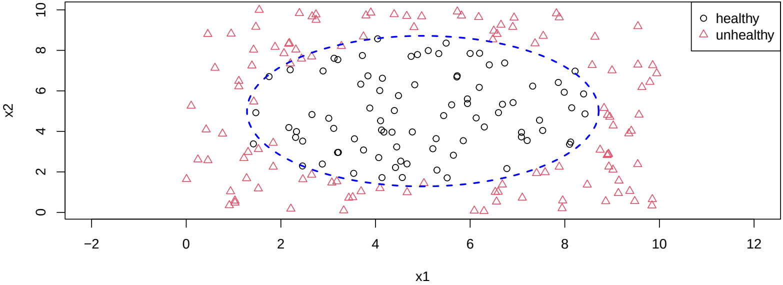
A Simple Example

Consider two predictors, X_1 and X_2, and an outcome y of "healthy" or "unhealthy". Can these predictors be used to accurately *classify* an observation?



Example (cont.)

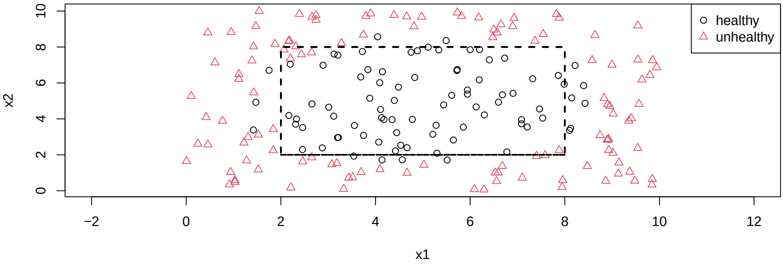
Yes! In this example, the true $f()$ is given by the blue ellipse:



Grinnell College
Statistics

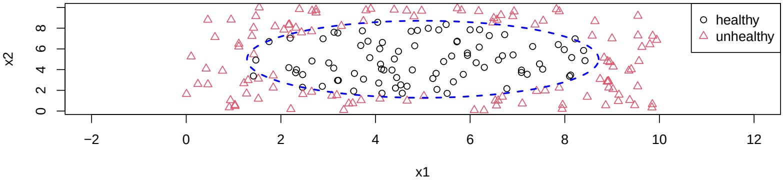
Example (cont.)

As a human, you might observe that the healthy data-points tend to fall between 2 and 8 in x_1 and x_2, so you might propose the *classification model* shown below. This model correctly classifies 178 of 200 data-points (89% accuracy).



Reducible vs. Irreducible Error

Now let's revisit the true relationship between x_1, x_2, and y. Notice that some "healthy" data-points are outside the ellipse, and some "unhealthy" ones are inside it:



These misclassifications contribute to **irreducible error**. Even if we perfectly recover $f()$ we will not achieve 100% accuracy.

Irreducible Error in Real Life

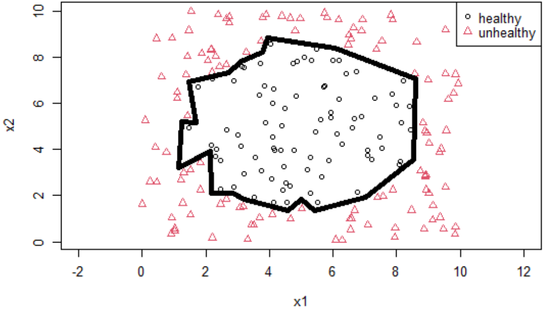
- Last Spring I worked with a group of students to use machine learning to automatically determine the destination of incoming Grinnell ITS help tickets
  - It has historically been the job of an ITS employee to route new tickets to the appropriate department/personnel
  - We used data from past tickets and where they were sent to build our models
- Why might we not be able to predict the destination of a ticket with 100% accuracy?

Irreducible Error in Real Life

- Last Spring I worked with a group of students to use machine learning to automatically determine the destination of incoming Grinnell ITS help tickets
  - It has historically been the job of an ITS employee to route new tickets to the appropriate department/personnel
  - We used data from past tickets and where they were sent to build our models
- Why might we not be able to predict the destination of a ticket with 100% accuracy?
  - The human employees may not have been routing tickets perfectly
  - That is, they might rely upon some rules, but they likely also introduce noise into in their decision making (subjective feelings, etc.)

Reducible Error

Considering the presence of irreducible error, we can frame the goal of machine learning as minimizing *reducible error*.



Has this classifier (black line) reduced the error rate to zero?

Training vs. Testing Splits

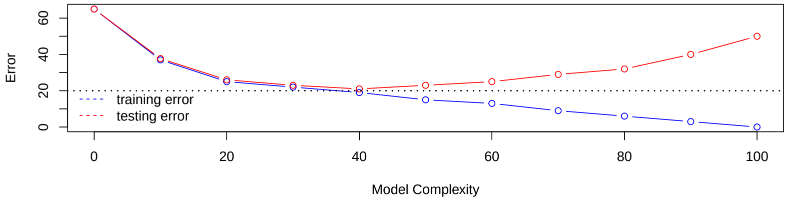
- We generally aren't interested in the error rate for the *observed data*
  - Instead, we'd like to minimize reducible error on *new data* that our model *hasn't yet seen*

Training vs. Testing Splits

- We generally aren't interested in the error rate for the *observed data*
  - Instead, we'd like to minimize reducible error on *new data* that our model *hasn't yet seen*
- Standard procedure is to divide the available data into **training** and **testing** sets
  - The training set is used to learn a collection of rules (ie: estimate $f()$)
  - The testing set is used to evaluate how well these rules perform on new, unseen data

Training, Testing, and Error

Consider an example with an irreducible error of "20 units":



- Training error can always be reduced by increasing the model complexity (ie: learning more rules)
- Testing error will never drop below the irreducible error rate (probabilistically speaking, it might for a given test set)

Bias vs. Variance

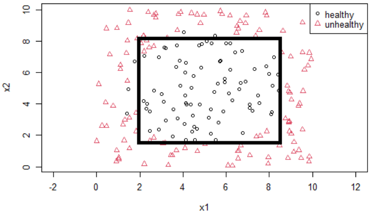
Reducible error can be attributed to one of two sources: **bias** or **variance**

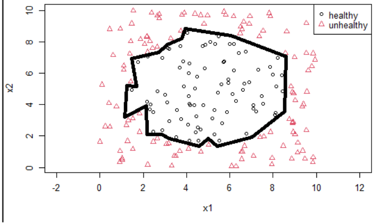
- **Bias** is when a learner lacks the structural flexibility to detect aspects of the true relationship between the predictors and the outcome
- **Variance** is when a learner is overly sensitive to chance artifacts present in the data (ie: the manifestations of irreducible error)

Poor performance due to high bias is called *underfitting*, while poor performance due to high variance is called *overfitting*

Bias vs. Variance

How would you compare the bias and variance of the following learners (a rectangle vs. an n-dimensional polygon)?





Defining Error

- So far we've focused on classifying a binary categorical outcome, a scenario where *classification accuracy* provides a natural framework for understanding a method's error
  - We'll talk about more sophisticated ways to evaluate error for classification tasks later on
- What if our goal is to predict a numeric outcome?

Defining Error

For a numeric outcome, it's most natural to measure error by summarizing the distances between predicted and observed outcomes:

- **Root Mean Squared Error**: $RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}$
- **Mean Absolute Error**: $MAE = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i|$

In each definition, y_i is the observed outcome for the i^{th} example (data-point) and $\hat{y}_i$ is the predicted outcome for that example.

Things to Know for the First Quiz

1. Differences between classification and regression
2. Definitions and examples of reducible vs. irreducible error
3. The bias-variance trade-off
4. The reason for creating a training and testing split