Introduction

Ryan Miller

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 accuracy 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 accuracy with experience)
- How is machine learning similar/different from computer programming?

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 accuracy with experience)
- How is machine learning similar/different from computer programming?
  - Computer programs rely on written rules, machine learning develops rules by finding patterns in examples

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 accuracy with experience)
- How is machine learning similar/different from computer programming?
  - Computer programs rely on written rules, machine learning develops rules by finding patterns in examples
- How is machine learning similar/different from statistics?

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 accuracy with experience)
- How is machine learning similar/different from computer programming?
  - Computer programs rely on written rules, machine learning develops rules by finding patterns in examples
- How is machine learning similar/different from statistics?
  - Statistics focuses on inference about a population using a sample, machine learning seeks generalizable predictive patterns

What is Machine Learning?

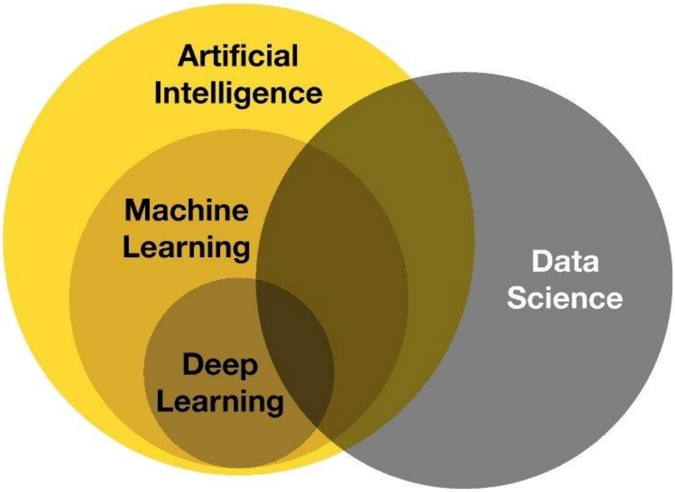
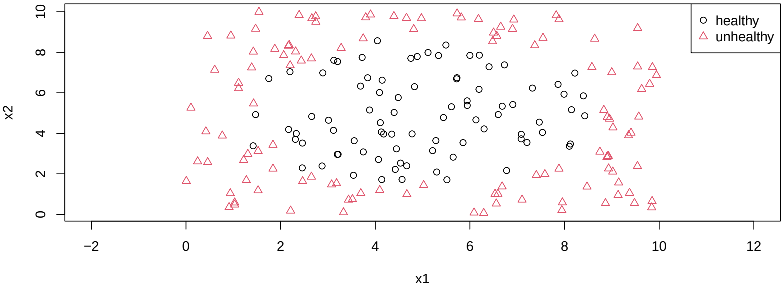


Image credit: BBN Times

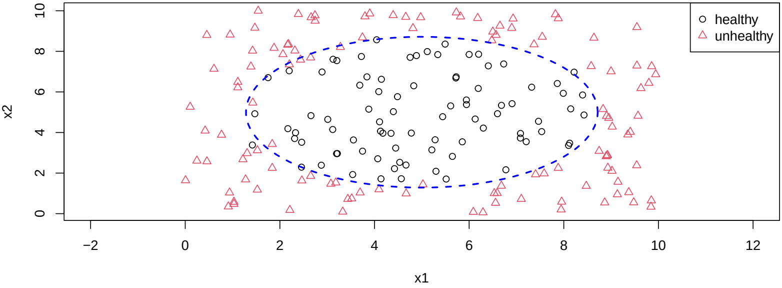
Example

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?



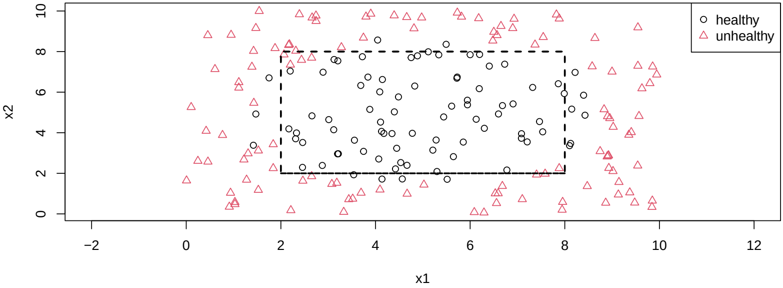
Example (cont.)

Yes! In this example, the true relationship between predictors and the outcome is given by the blue ellipse



Learning?

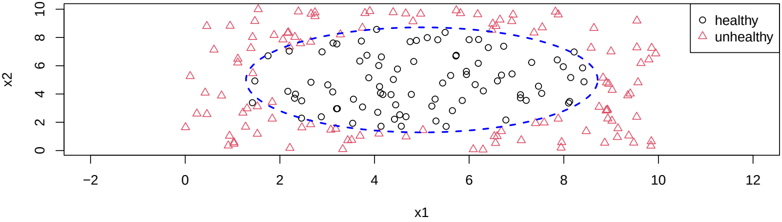
As a human, you might observe that the healthy data-points tend to fall between 2 and 8 in x_1 and x_2, so you might propose the following *classification model*:



This simple model correctly classifies 178 of 200 data-points.

Reducible vs. Irreducible Error

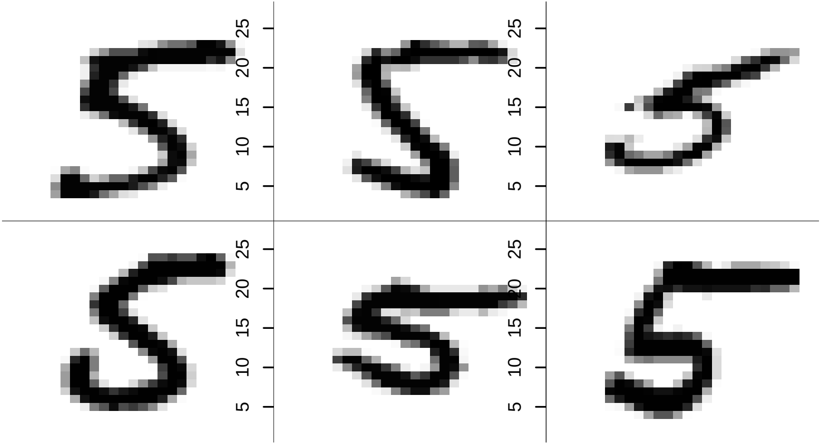
Let's revisit the true relationship between x_1, x_2, and y. Notice that some "healthy" data-points are outside the ellipse, and some "unhealthy" ones are inside it:



- The misclassification of these examples is known as the **irreducible error** (sometimes called "Bayes error")
  - Even the *best possible model* still cannot perfectly predict every outcome

Irreducible Error in Real Life

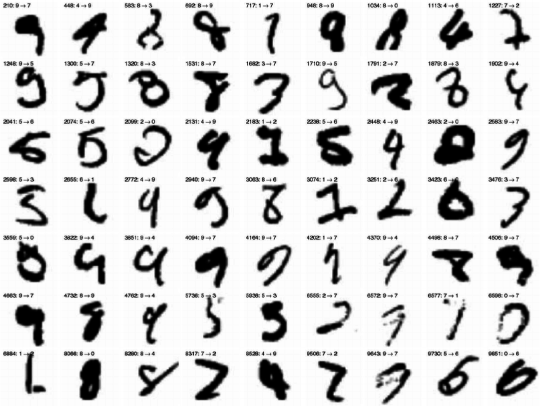
Is a digit a "5" or something else?



How might the concept of irreducible error manifest in this application?

Irreducible Error in Real Life

We could know the exact “rules” used to make a “5”, but it’s possible we encounter examples of “5” that look more like a “6”.



Even state of the art classifiers (which approach the irreducible error rate) misclassify $\sim 0.5\%$ on handwritten digits (source)

Irreducible Error in Real Life

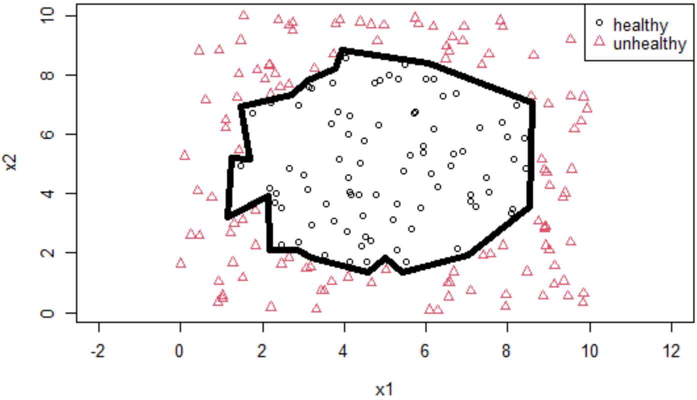
Irreducible error will always exist, the important question is "how much?" Consider the following scenarios:

- Classifying examples of "5" handwritten by doctors
- Classifying examples of "5" created by a laser printer

While the precise amount of irreducible error in a machine learning problem is generally unknown, we often have a sense of what it might be.

Reducible Error

The primary goal of machine learning is to *learn rules* that minimize *reducible error*. Consider the following classifier:



Has this classifier reduced the error rate to zero?

Training vs. Testing Splits

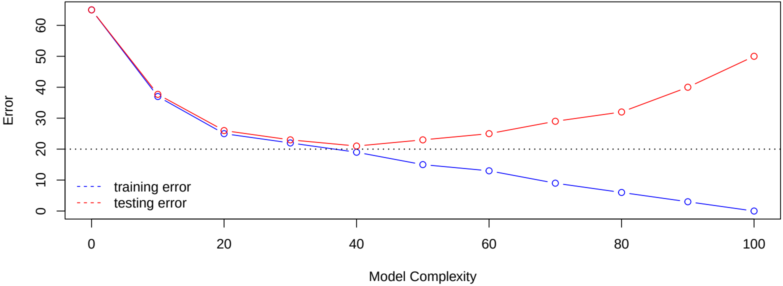
- We generally aren't interested in the error rate for the *observed examples*
  - Instead, we'd like to minimize reducible error on *new examples* that our model *hasn't yet seen*

Training vs. Testing Splits

- We generally aren't interested in the error rate for the *observed examples*
  - Instead, we'd like to minimize reducible error on *new examples* 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
  - The testing data is used to evaluate how well these rules perform on data that hasn't been seen by the learner

Training, Testing, and Error

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



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

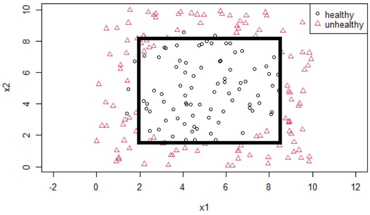
Bias vs. Variance

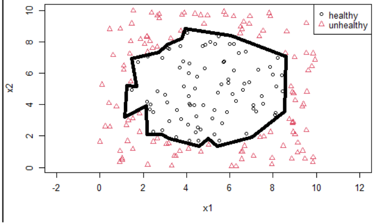
Reducible error can arise in one of two ways: **bias** or **variance**

- **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*

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 categorical outcomes next week
- 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.

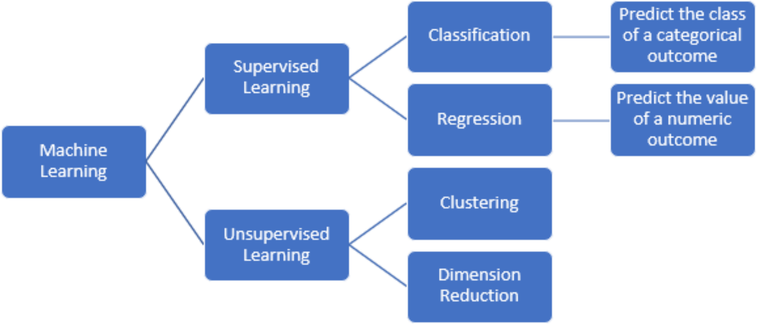
Classification vs. Regression

- Machine learning applications involving a *numeric outcome* are called **regression tasks**
  - Applications involving a *categorical outcome* are called **classification tasks**
- We define error differently for each type of task
  - The bias-variance trade-off and irreducible error still apply to both scenarios

Machine Learning without an Outcome?

- For most of this semester, we'll focus on machine learning tasks involving a pre-selected or derived outcome
  - These are known as **supervised learning** tasks
- Other learning tasks, such as clustering or dimension reduction, can be achieved without designating an outcome
  - These are known as **unsupervised learning** tasks

Overview


Machine Learning
Supervised Learning
Unsupervised Learning
Classification
Regression
Clustering
Dimension Reduction
Predict the class of a categorical outcome
Predict the value of a numeric outcome

Things to know for Thursday's quiz

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