Lecture 8: Supervised Learning Pt. 2

More Models, Bootstrapping, and Bagging

INFO 1998: Introduction to Machine Learning

CDS Education

Agenda

1. Decision Trees
2. Logistic Regression
3. Validation Techniques
    - Bootstrapping & Bagging

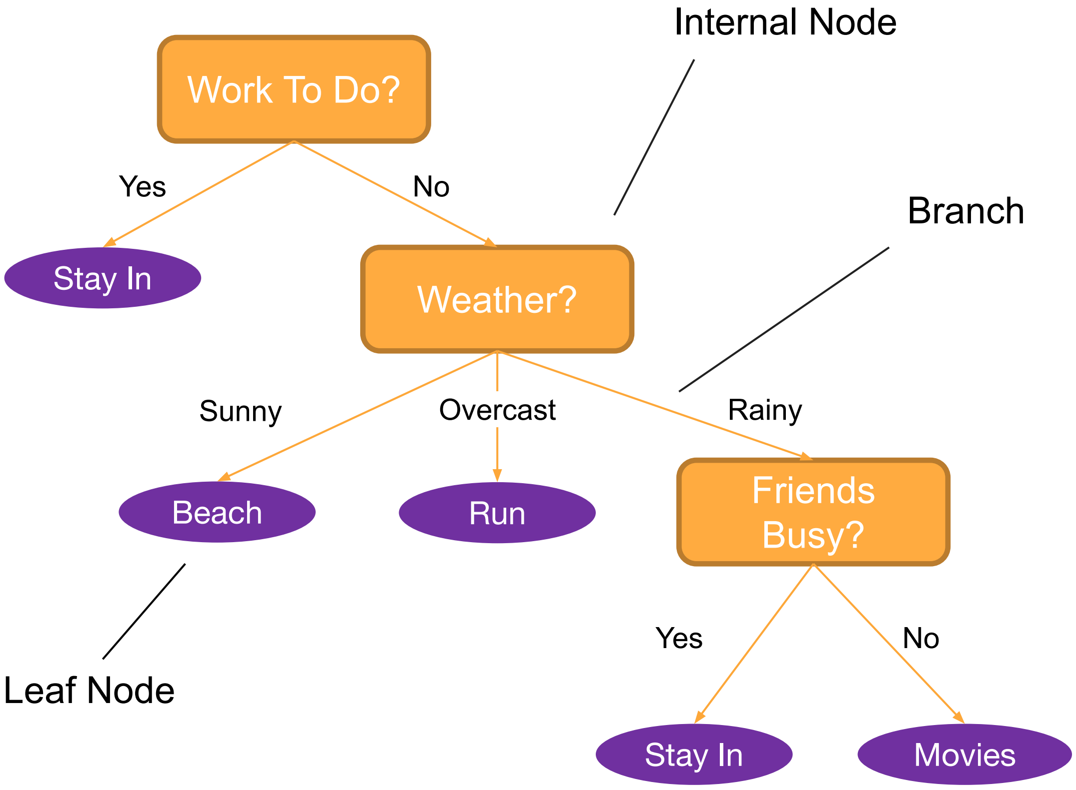
Decision Trees

Source

How Should I Spend My Weekends

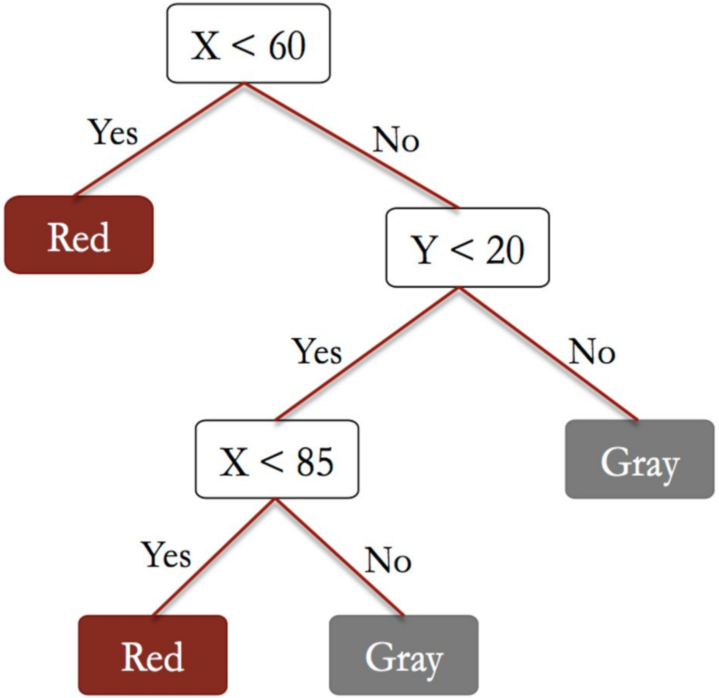
Decision Tree

- *supervised* machine learning model
- **breaking down** our data by **making a decision** based on **asking a series of questions** based on features



CART (Classification and Regression Trees)

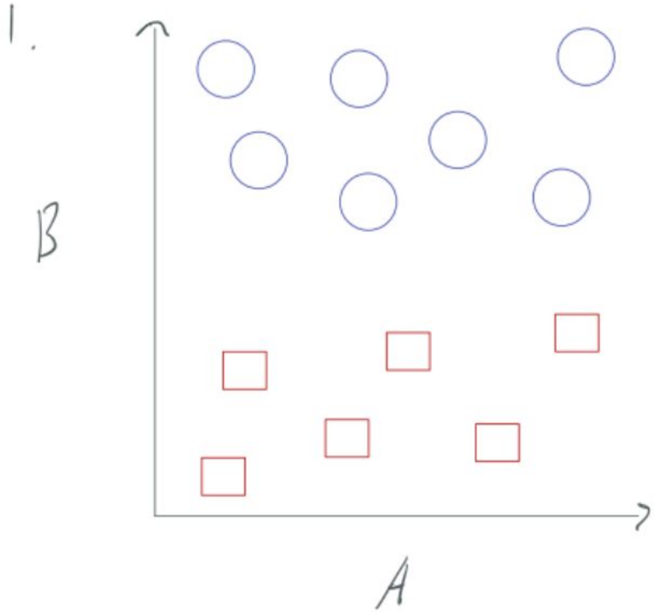
- Used for Classification and Regression
- At each node, split on variables
- Each split minimizes error/impurity function
- Very interpretable
- <u>Models a non-linear relationship!</u>

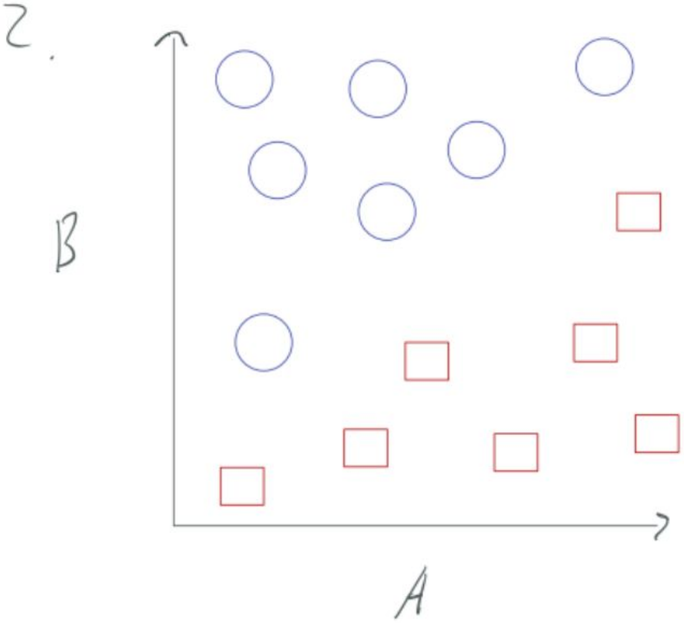


Pros and Cons of Using Decision Trees

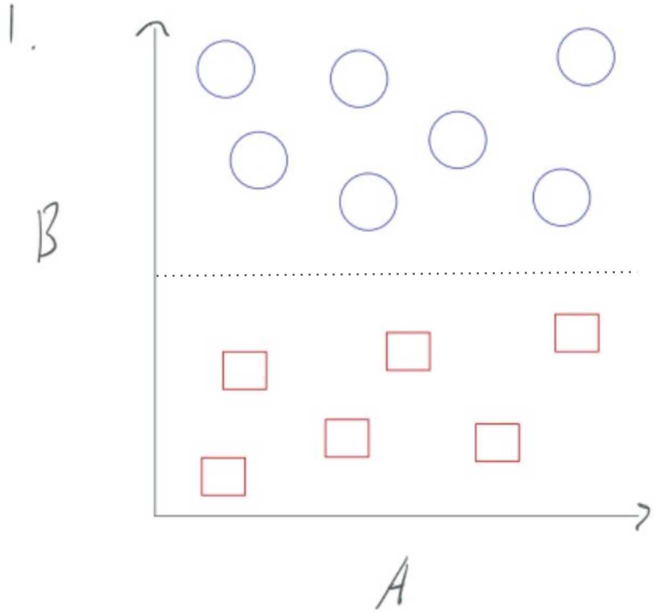
Pros	Cons
Easy to interpret	Overfitting ☹️
Requires little data preparation (robust to missing data)	Requires parameter tuning (max depth)
Can use a lot of features	Can only make horizontal/vertical splits (solvable with feat. eng. / ensembling)
Can capture non-linear relationships	

What would these decision boundaries look like?

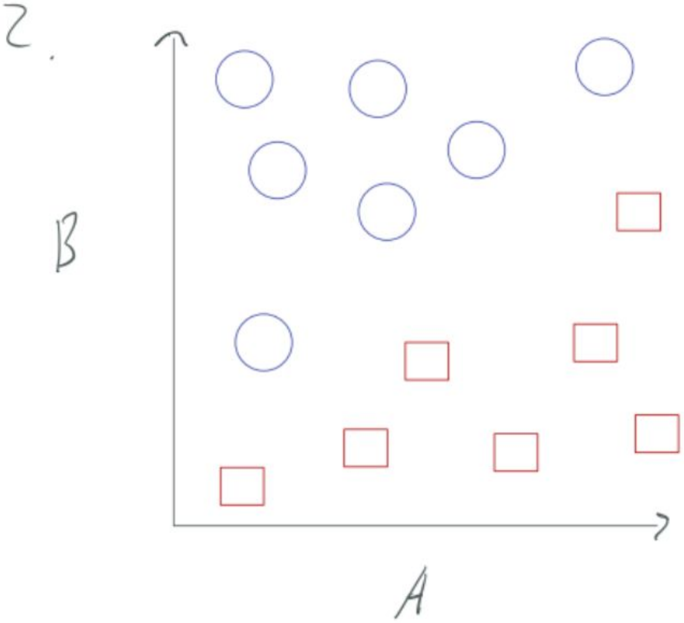




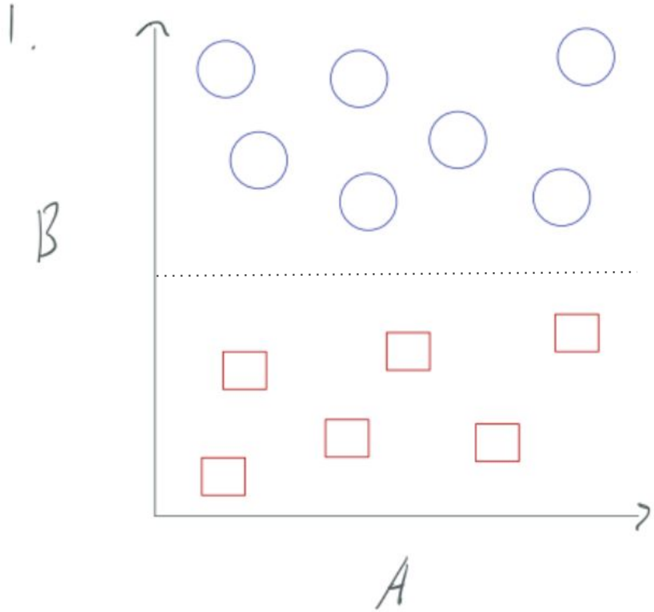
What would these decision boundaries look like?



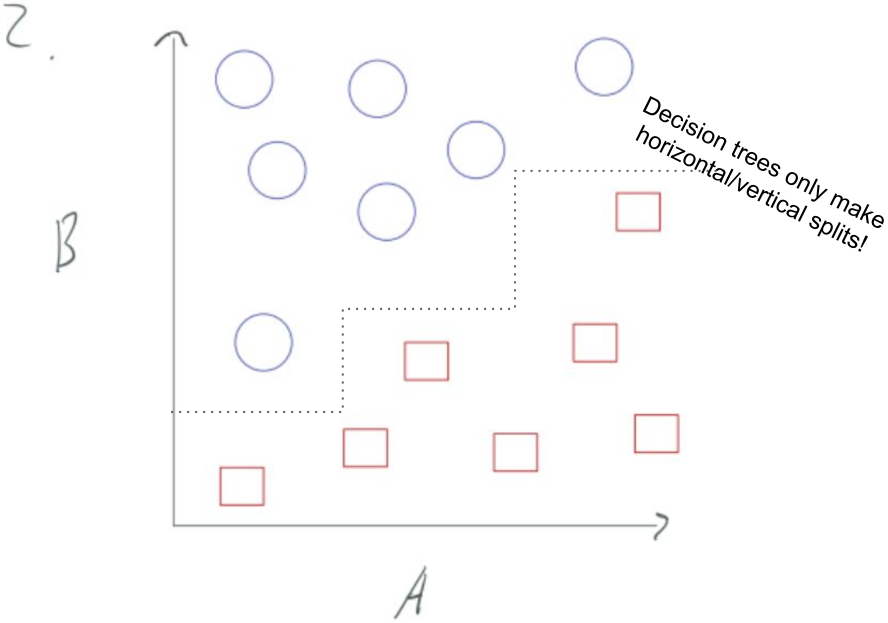
"If B less than this value, it's a red square. Otherwise, it's a blue circle."



What would these decision boundaries look like?

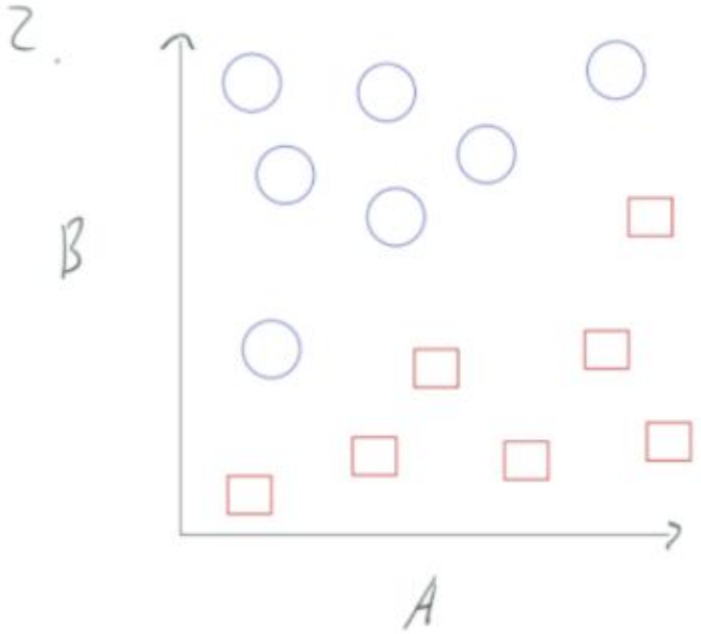


“If B less than this value, it’s a red square. Otherwise, it’s a blue circle.”

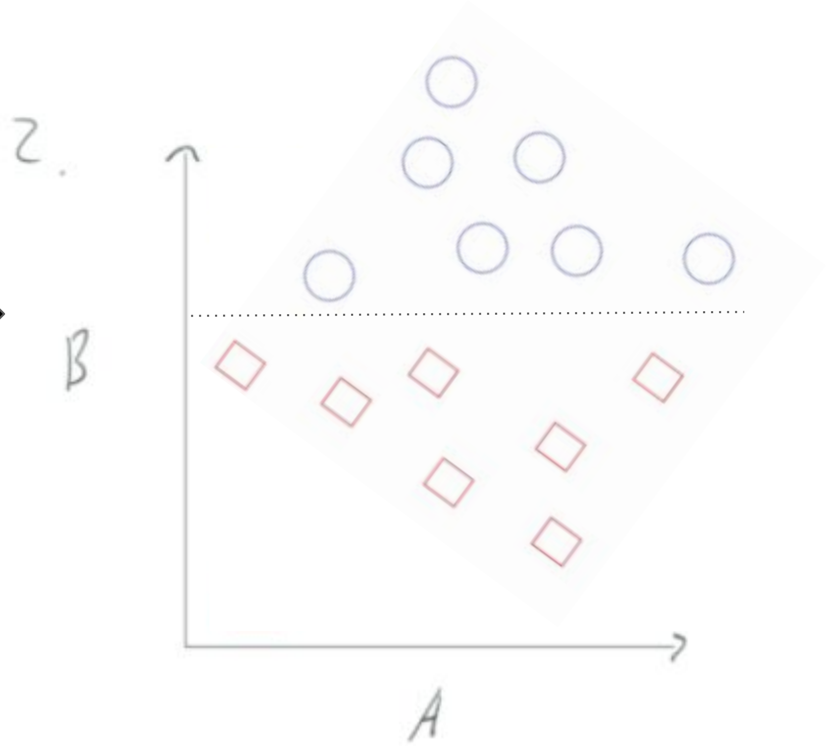


Uhhh…

But...

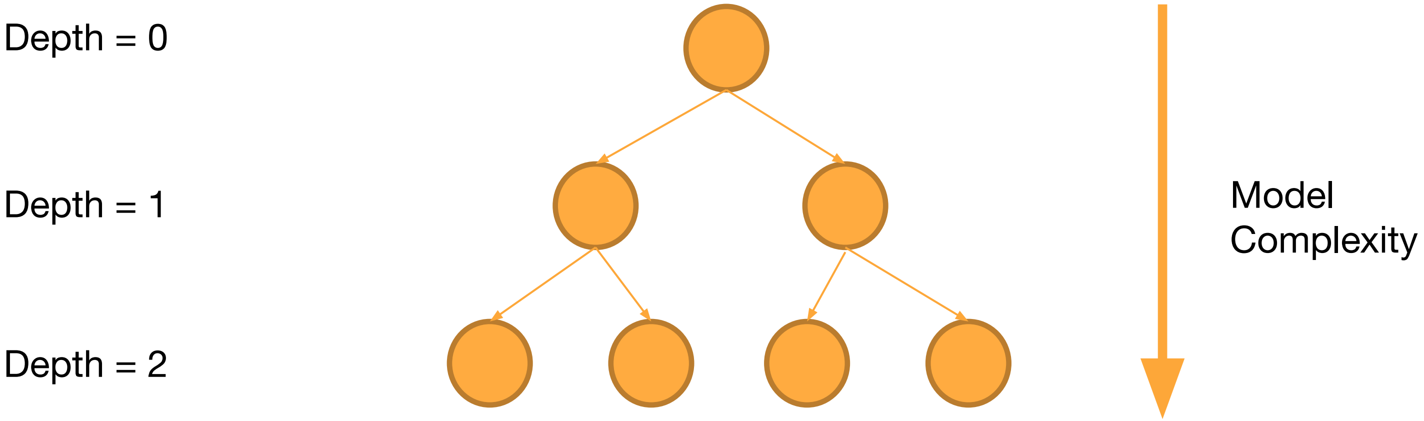


Rotate!



How to Reduce Overfitting

1. Limit the max depth of the tree

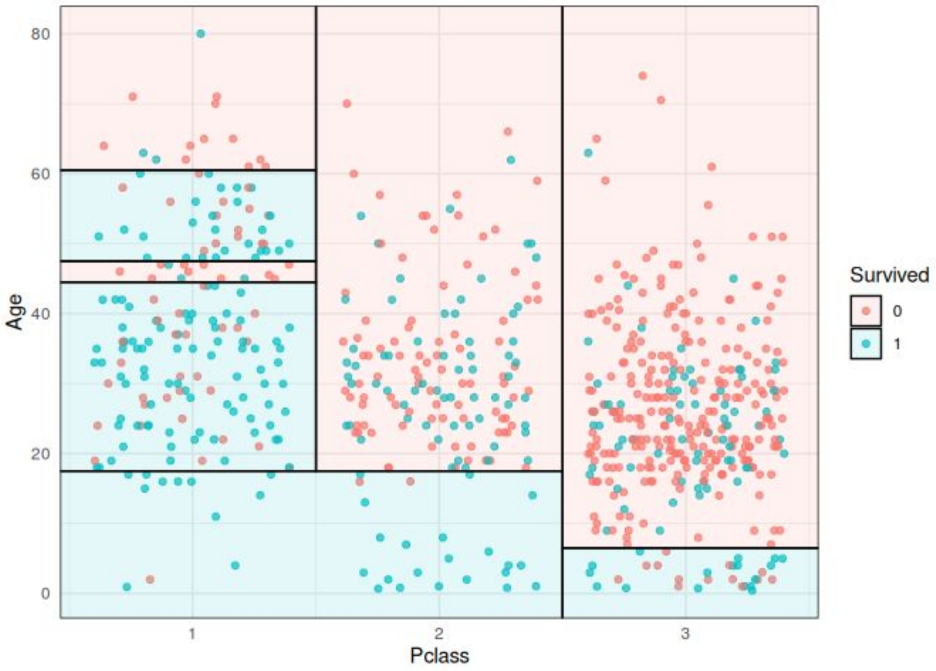


When training a decision tree, we have to specify the maximum depth a constructed tree can have

Source

How to Reduce Overfitting

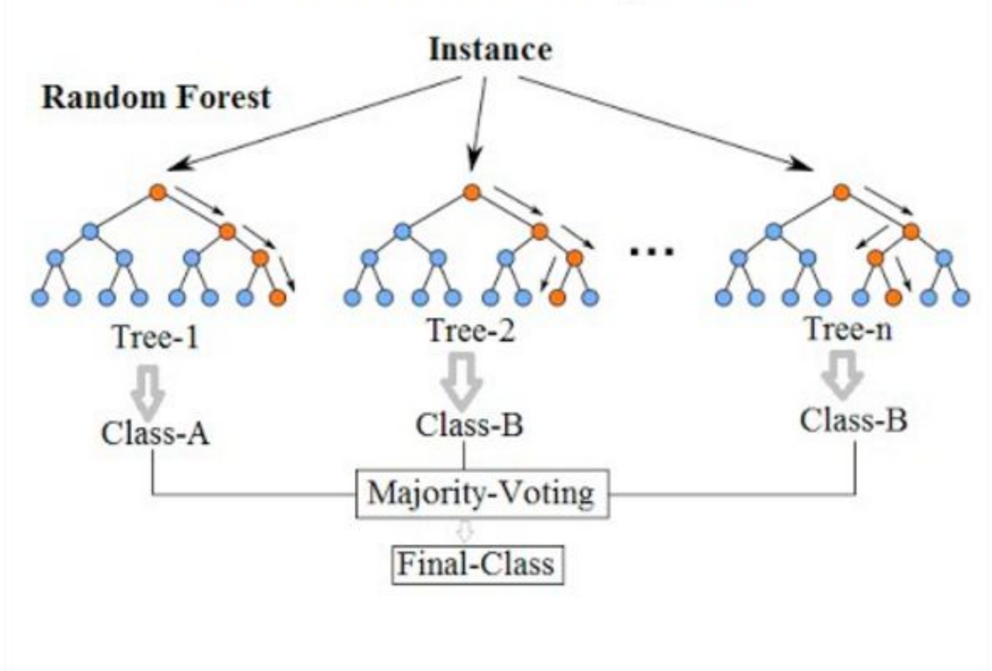
- There are no “curves” for each decision tree boundary line
- Limiting the depth of the tree limits the number of lines you are splitting on



Source

How to Reduce Overfitting

2) Train multiple decision trees and determine final output based on output of each decision tree



This is called a
Random Forest Classifier

Demo

Logistic Regression

Logistic Regression

- Used for Binary Classification:

$$Y = \begin{cases} 1 \\ 0 \end{cases}$$

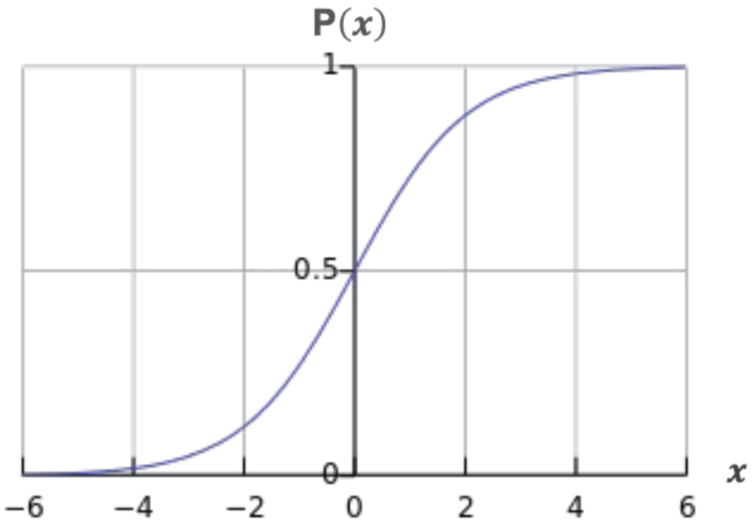
- Fits a linear relationship between the variables
- Transforms the linear relationship of probability that the outcome is 1 by using the **sigmoid function**

Formula:

$$P(x) = \frac{1}{1 + e^{-(\beta_0 + \beta_1 x_1 + \ldots + \beta_k x_k)}} \longrightarrow \ln\left(\frac{P}{1-P}\right) = \beta_0 + \beta_1 x_1 + \ldots + \beta_k x_k$$

Logistic Function

$$P(x) = \frac{1}{1+e^{-x}}$$



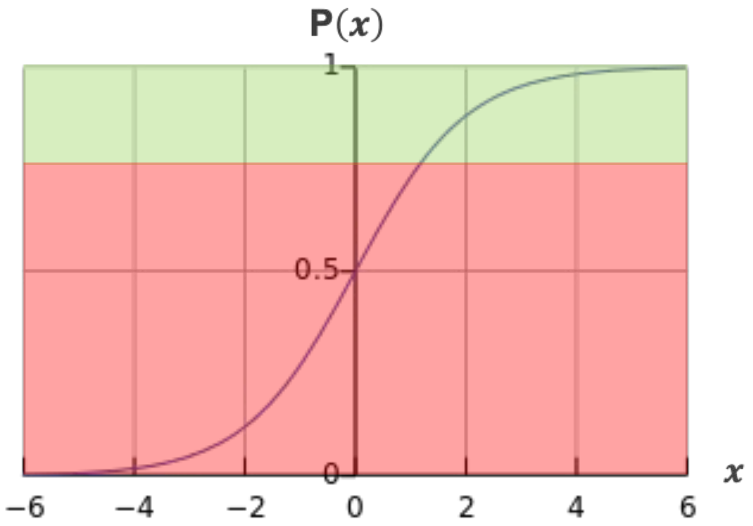
The Logistic Function "**squeezes"** numbers to be between 0 and 1

Allows us to interpret our prediction as a "**probability**" that something is true

Threshold

At what point point do we differentiate between our classifications?

- f(x) below threshold: predict 0
- f(x) above threshold: predict 1



Pros and Cons of Using Logistic Regression

Pros	Cons
Easy to interpret (probability)	Only Capable of Binary Classification
Computationally efficient to compute	No closed form solution (requires use of optimization algorithms)
Does not require parameter tuning	

Logistic Regression is a simple model, therefore, oftentimes it is used as a good “baseline” to compare more complex models to

Validation Techniques

Review: Regression vs. Classification

Regression

- Predict Continuous Data
- “On average, **how wrong** are we?”

Classification

- Predict Discrete or Categorical data
- “**How many** points do we get wrong?”

Numbers ≠ Continuous

Leave-P-out

Let **D** be our whole dataset

Choose a **P**

For every combination of **P** points in **D**:

Use a train/test split with those **P** points as test, the rest as train

Leave-P-out: different from K-fold!

Let's say **D** has a size of 4. There are four data points: *a*, *b*, *c*, and *d*.

K-fold:

- K = 2.
- Each fold has a size of 2: {*a*,*b*} and {*c*,*d*}
- So, we only have 2 possible test sets:

  {*a*,*b*} and {*c*,*d*}

Leave-P-out:

- P = 2.
- We have 6 possible test sets:

  {*a*,*b*}, {*a*,*c*}, {*a*,*d*}, {*b*,*c*}, {*b*,*d*}, and {*c*,*d*}

Leave-P-out

Pros:

- Dependable (not random)
- Representative — checks all combinations

Cons:

- Slow!
    - Runtime <u>increases</u> with larger datasets
    - Runtime <u>explodes</u> with larger P

Monte Carlo Cross Validation

- Getting accuracy **1** time doesn’t tell us much
- Getting accuracy **2** times tells us a bit
- Getting accuracy **3** times tells us a bit more
- ...
- Getting accuracy **N** times might be good enough!

Take the average of those **N** times

Monte Carlo CV

- Need to use **new**, **random** train/test split each time
  - If you use the same train/test split each time, you're not getting any new information!
- Pros:
  - easy to implement
  - easy to make faster/slower by changing number of iterations
- Cons:
  - random -> train/test splits not guaranteed to be representative of dataset (might overlap, or miss some data)
  - harder to calculate how many iterations you need

The Bootstrap

What if we don't have enough data?

- Use **bootstrap datasets** to approximate the test error
- **Sample with replacement** from the original training dataset (with n samples) to generate **bootstrap datasets** of size n
  - Some data points may appear more than once in the generated data
  - Some data points may not appear
- Estimate of test error = average error among bootstrap datasets

Demo

Why do we still use Bootstrap?

- Bootstrap allows us to use a computer to mimic the process of obtaining new data sets.
- Can be used to quantify the uncertainty associated with a given estimator or statistical learning method.
- Provides an estimate of the standard error of a coefficient, or a confidence interval for that coefficient.
  - i.e. the variability of the model!

Bagging (Bootstrap Aggregating)

What if we don't have enough data?

- Bagging is a common technique that builds on Bootstrapping

- Main Idea: Do Bootstrapping a bunch and make a classifier for each bootstrap, then choose majority prediction.

- Many weak learners aggregated typically outperform a single learner over the entire set, and overfits less.
    - Principle behind Random Forests ("forest" of decision trees)

Coming Up

- **Assignment 7:** Due tonight at 11:59pm
- **Assignment 8**: Due November 6th, 2024
- **Final Project:** Due November 20th, 2024
- **Next Lecture**: Applications of *Un*supervised Learning

CDS Education