Lecture 7: Classification Models and Cross Validation

INFO 1998: Introduction to Machine Learning

CDS Education

We explore, learn, and educate big minds.

Agenda

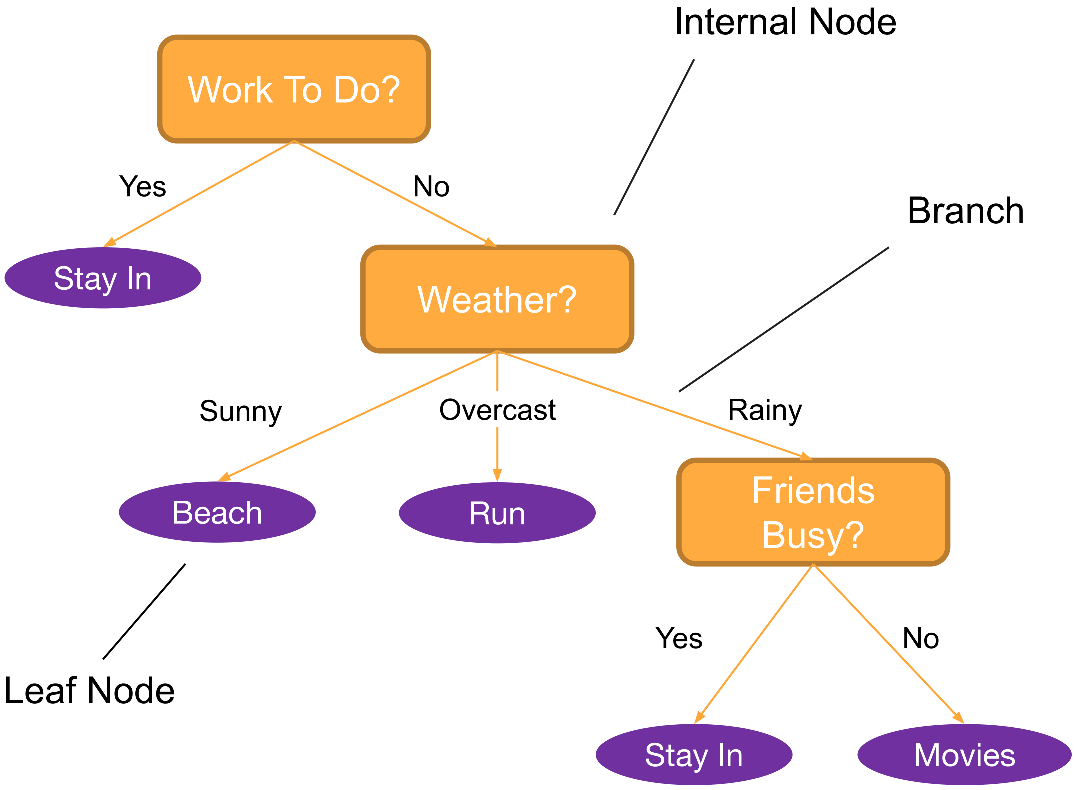
1. Decision Trees
2. Logistic Regression and Its Applications
3. Cross Validation

Decision Trees

Source

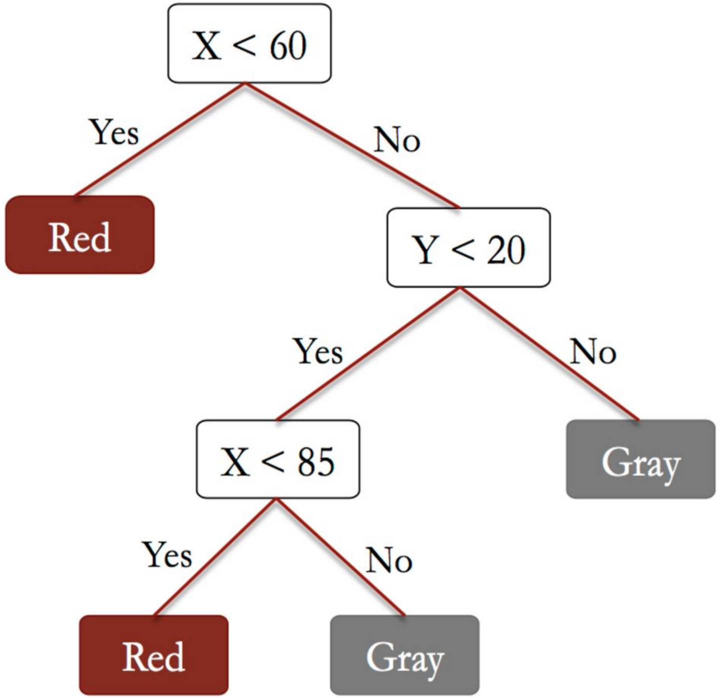
How Should I Spend My Weekends

A **decision tree** is a supervised machine learning model used to predict a target by learning decision rules from features. As the name suggests, we can think of this model as **breaking down** our data by **making a decision** based on **asking a series of questions.**



CART (Classification and Regression Trees)

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

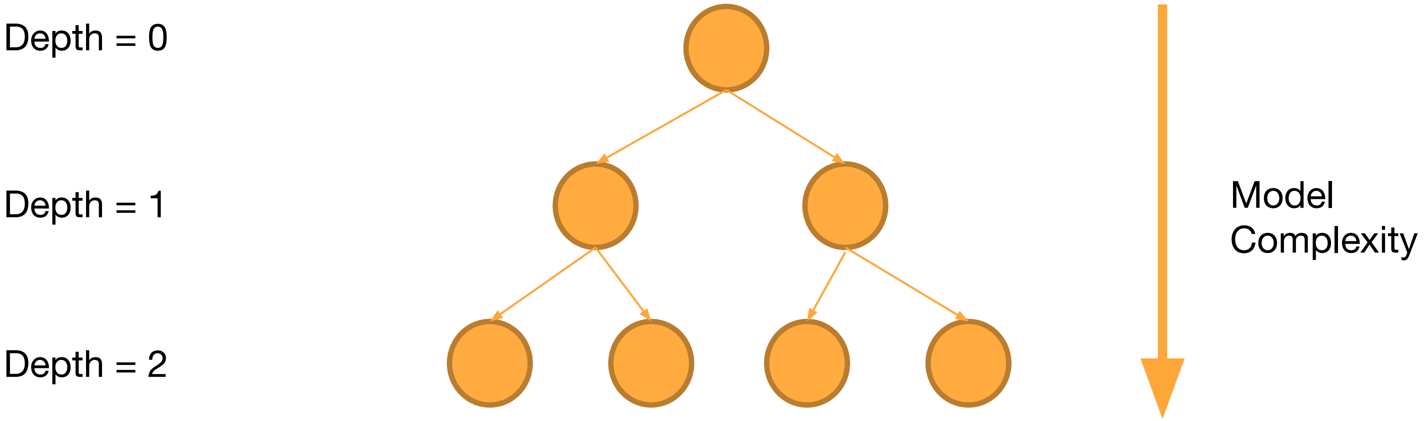


Pros and Cons of Using Decision Trees

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 capture non-linear relationships	

How to Reduce Overfitting

1. Limit the max depth of the tree

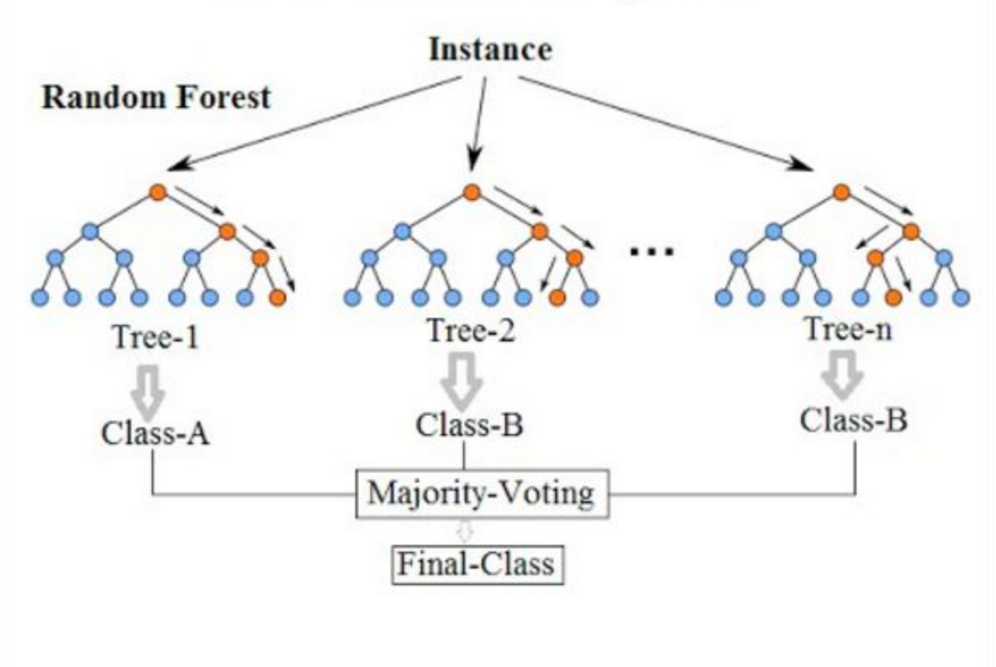


When training a decision tree, we have to specify the maximum depth a constructed tree can have

Source

How to Reduce Overfitting

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:

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

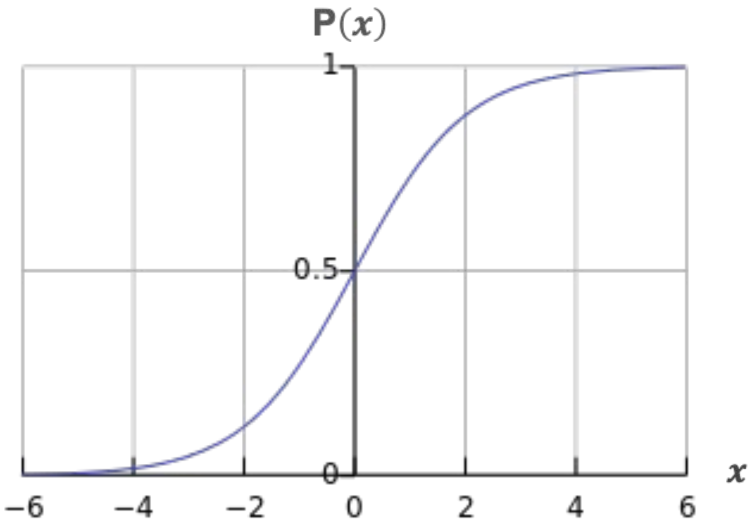
- Fits a linear relationship between the variables
- Transforms the linear relationship of probability that the outcome is 1 by using the **sigmoid function**

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

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



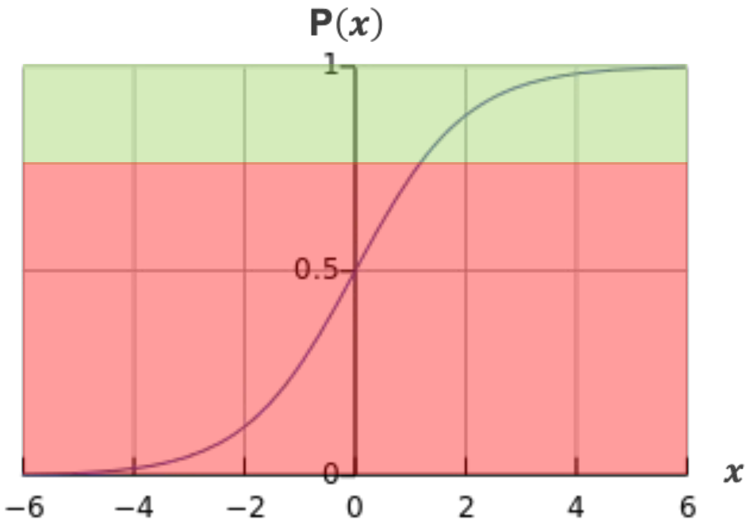
The Logistic Function "**squeezes**" numbers to be between 0 and 1

→ 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	
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

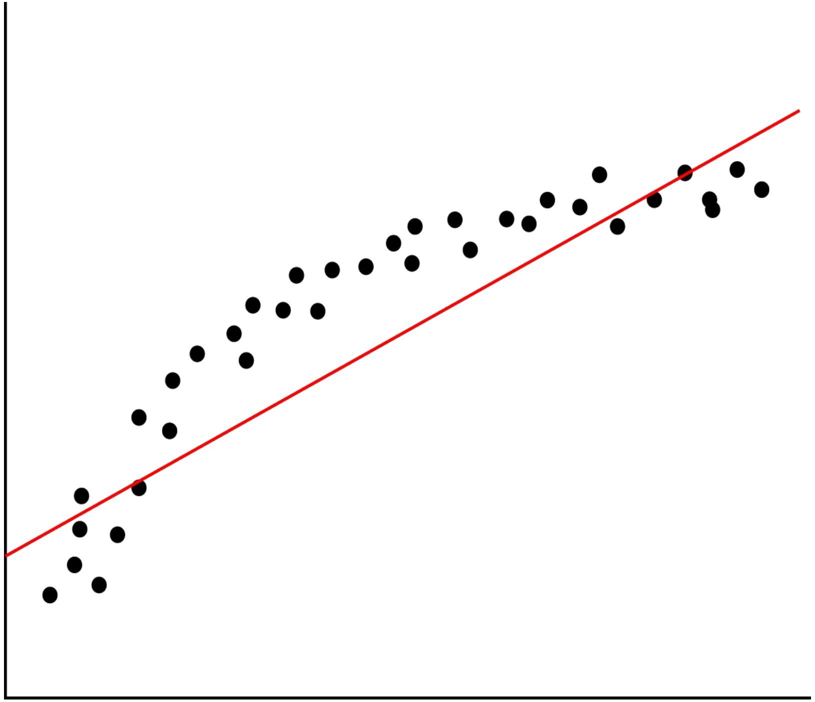
Demo

Cross Validation

Underfitting

Underfitting means we have high bias and low variance.

- Lack of relevant variables/factor
- Imposing limiting assumptions
  - Linearity
  - Assumptions on distribution
  - Wrong values for parameters

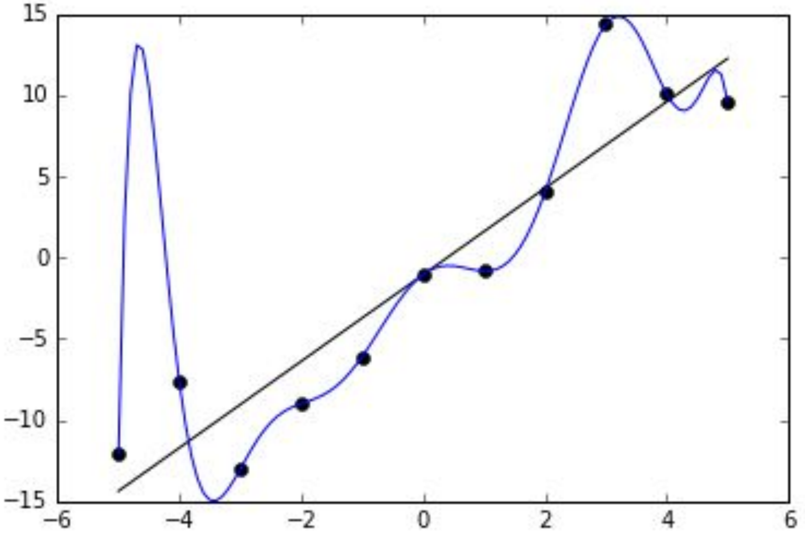


Source

Overfitting

Overfitting means we have low bias and high variance.

- Model fits too well to specific cases
- Model is over-sensitive to sample-specific noise
- Model introduces too many variables/complexities than needed



Source

K-fold Cross Validation

Often used in practice with *k*=5 or *k*=10.

Create equally sized *k* partitions, or **folds**, of training data

For each fold:

- Treat the *k-1* other folds as training data.
- Test on the chosen fold.

The average of these errors is the validation error

K-fold Cross Validation

Dataset

Suppose K = 5,

5-Fold CV

K-fold Cross Validation

Fold 1 | Fold 2 | Fold 3 | Fold 4 | Fold 5

K-fold Cross Validation

Test Sample	Training Sample	Training Sample	Training Sample	Training Sample

Calculate MSE = mse1

K-fold Cross Validation

Training Sample	Test Sample	Training Sample	Training Sample	Training Sample

Calculate MSE = mse2

K-fold Cross Validation

Training Sample	Training Sample	Test Sample	Training Sample	Training Sample

Calculate MSE = mse3

K-fold Cross Validation

And so on

K-fold Cross Validation

Fold 1	Fold 2	Fold 3	Fold 4	Fold 5

MSE = Avg(mse1…5)

K-fold Cross Validation

Matters less how we divide up

Selection bias not present

Leave-1-Out Cross Validation

For each sample:

- Treat all other data as training data.
- Test on that one sample

The average of these errors is the validation error

Pro: Better on small datasets

Pro: More realistic (trained on most of the data)

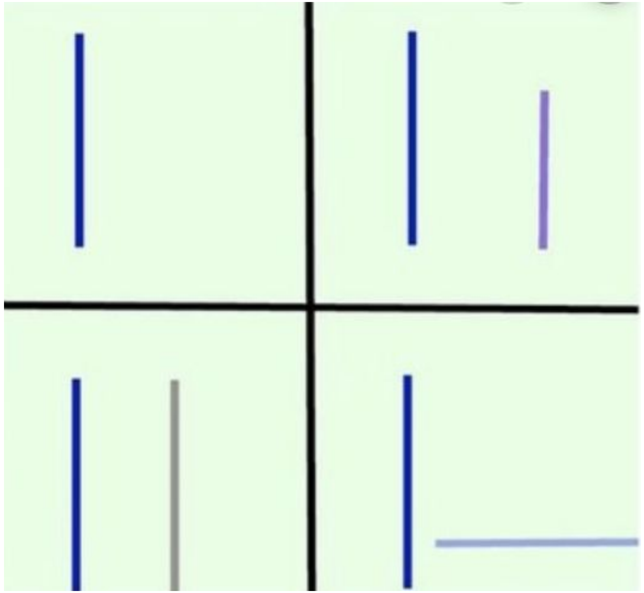
Con: Takes longer to run

Demo

Review

Loss, Cost, and Score Functions

- **Loss Function**
  - Penalty for missing a single data point
- **Cost Function**
  - Indicates how bad the whole model is
  - Applies loss function to each point, then combines that into a single number
    - ex: average of (loss from each point)
- **Score Function**
  - A more interpretable version of the cost function



Cost -> Accuracy Score

- sklearn's score function is:

  1 - ([Cost of model] / [Cost of baseline])

- **1** is very, very good
- **0** means you were as bad as the baseline
- **<0** means either your baseline predictions were very good, or you really, really messed up

Balancing Bias and Variance

$$\text{Error} = (\text{Bias})^2 + (\text{Variance}) + (\varepsilon)$$

Bias = expected loss of accuracy

Variance = inconsistency of model

ε = irreducible error

Linear Regression

$$y = B_0 + B_1 x_1 + \ldots + B_p x_p + \varepsilon$$

- x is an input; $x_1, x_2, \ldots, x_p$ are the features of x
- y is an output (usually a single value)
- B's are "weights"
  - A linear regression equation is defined by its B's
  - This linear regression equation is the "program" produced by ML
- Given a set of x's and y's, the program finds a set of B's that (almost) satisfies the equation above for all x's and y's
  - **Minimizes bias, but not variance**
  - Then, you can plug in the feature values of a new x and to predict its y

Coming Up

- **Assignment 7**: Due 4:30pm on Nov. 10
- **Next Lecture**: Linear Classifiers and Model Validation
- **Final Project Check in:** Due 4:30pm on Nov. 10

CDS Education

We explore, learn, and educate big minds.