Machine Learning

CS 539

Worcester Polytechnic Institute
Department of Computer Science
Instructor: Prof. Kyumin Lee

Project Team

- Will Buchta, Sarah Semy, Deena Selitsk, and Humza Qureshi

Upcoming Schedule

- Project Teams
  - I will randomly assign team members and announce it next week.
- HW1 is out.
- Quiz1 will be taken on June 4 – it will be available only during the day

Previous Class…

Which attribute is the best classifier?
➔ Information Gain
➔ Entropy

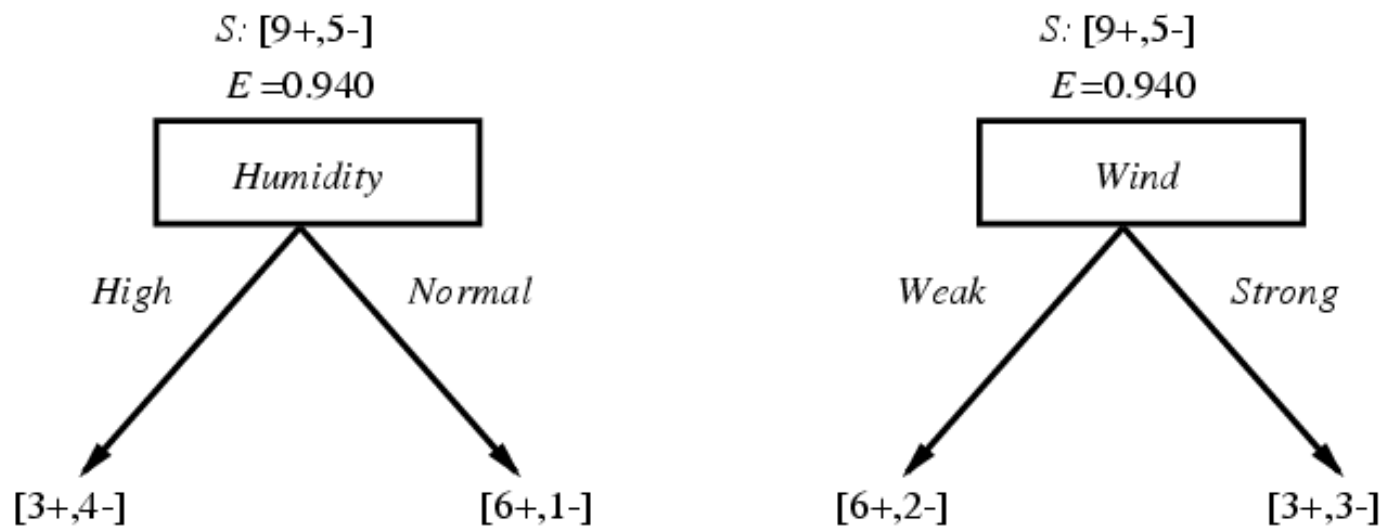
Decision Trees

Training Example

Day	Outlook	Temperature	Humidity	Wind	PlayTenr
Dl	Sunny	Hot	High	Weak	No
D2	Sunny	Hot	High	Strong	No
D3	Overcast	Hot	High	Weak	Yes
D4	Rain	Mild	High	Weak	Yes
D5	Rain	Cool	Normal	Weak	Yes
D6	Rain	Cool	Normal	Strong	No
D7	Overcast	Cool	Normal	Strong	Yes
D8	Sunny	Mild	High	Weak	No
D9	Sunny	Cool	Normal	Weak	Yes
DlO	Rain	Mild	Normal	Weak	Yes
Dll	Sunny	Mild	Normal	Strong	Yes
D12	Overcast	Mild	High	Strong	Yes
D13	Overcast	Hot	Normal	Weak	Yes
D14	Rain	Mild	High	Strong	No

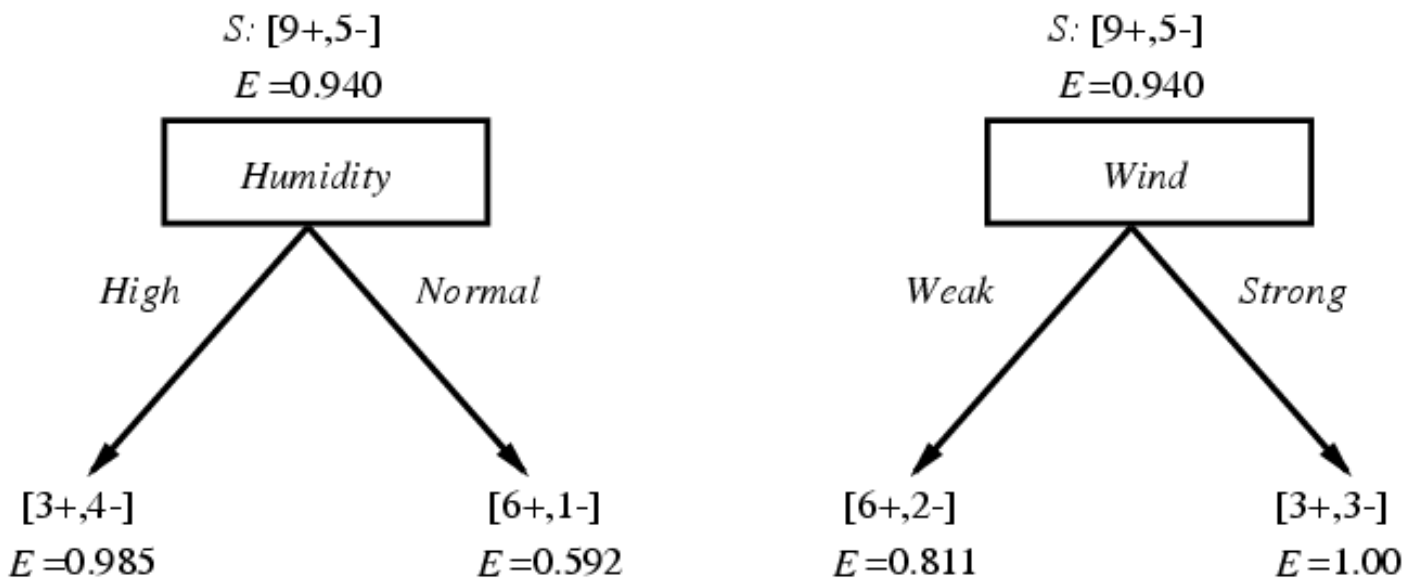
Selecting the Next Attribute

Which attribute is the best classifier?



Selecting the Next Attribute

Which attribute is the best classifier?



Gain (S, Humidity)

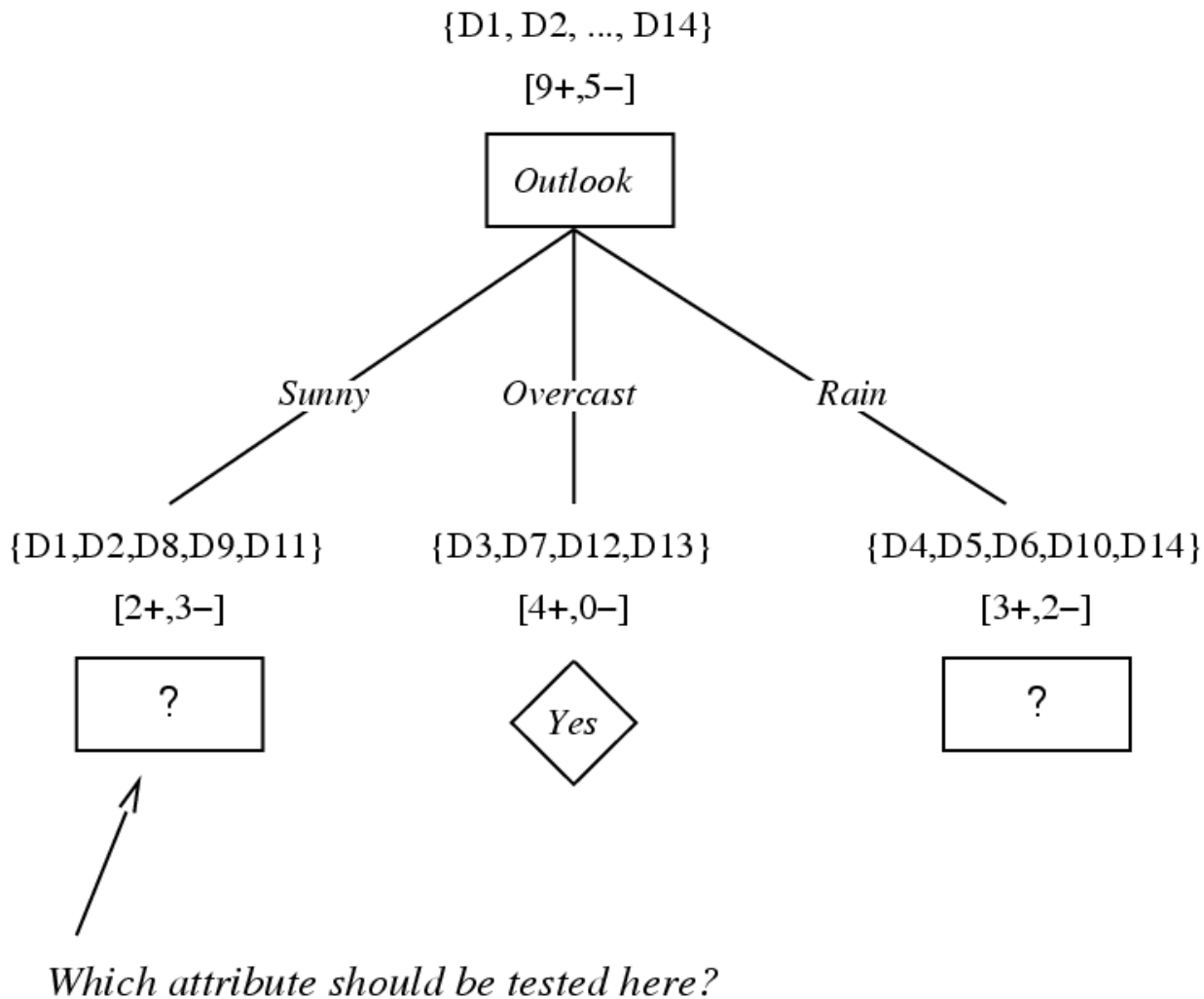
= .940 - (7/14).985 - (7/14).592
= .151

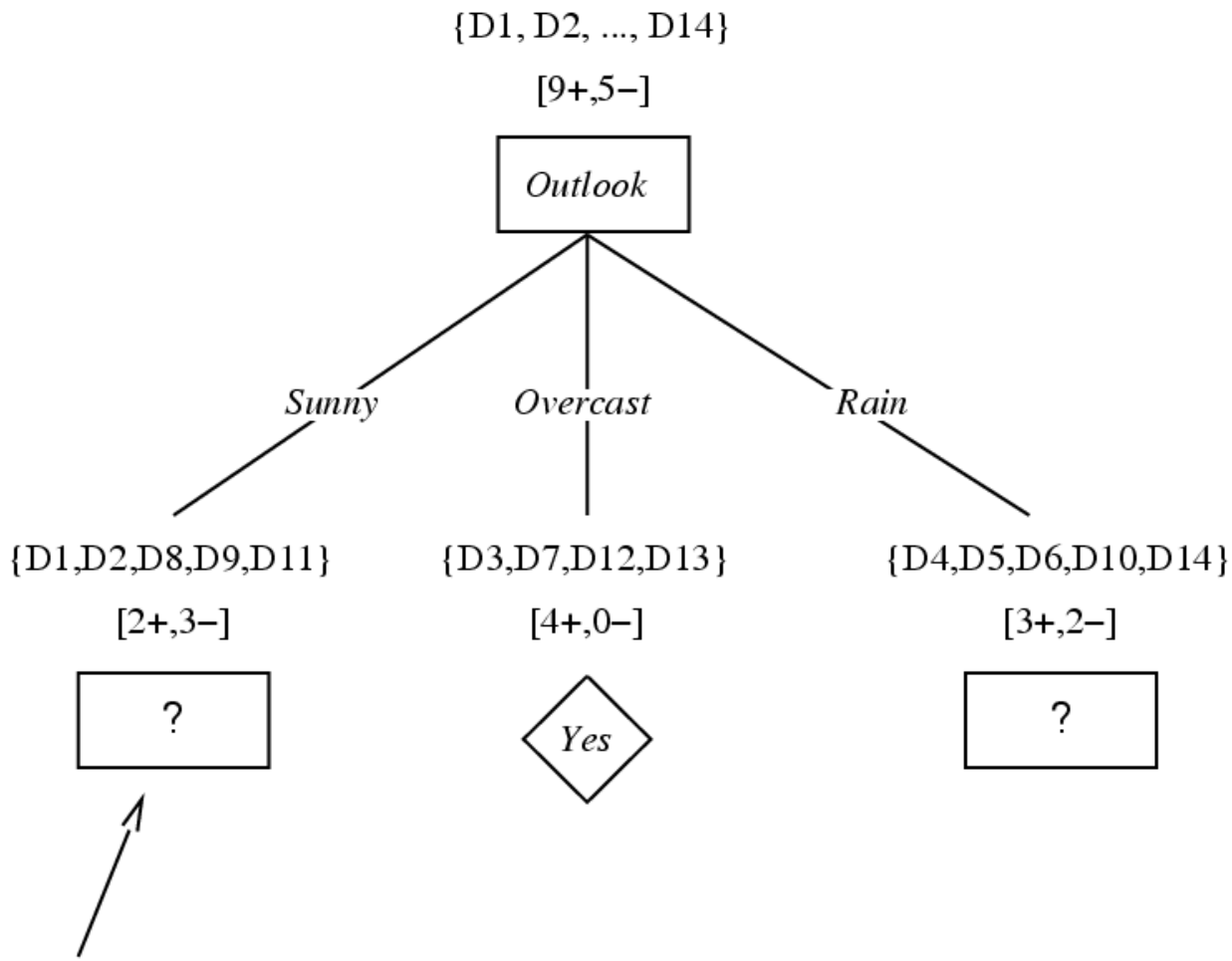
Gain (S, Wind)

= .940 - (8/14).811 - (6/14)1.0
= .048

First step: which attribute to test at the root?

- Which attribute should be tested at the root?
  - $Gain(S, Outlook) = 0.246$
  - $Gain(S, Humidity) = 0.151$
  - $Gain(S, Wind) = 0.084$
  - $Gain(S, Temperature) = 0.029$
- *Outlook* provides the best prediction for the target


{D1, D2, ..., D14}
[9+,5−]
Outlook
Sunny
Overcast
Rain
{D1,D2,D8,D9,D11}
[2+,3−]
{D3,D7,D12,D13}
[4+,0−]
{D4,D5,D6,D10,D14}
[3+,2−]
?
Yes
?
Which attribute should be tested here?



Which attribute should be tested here?

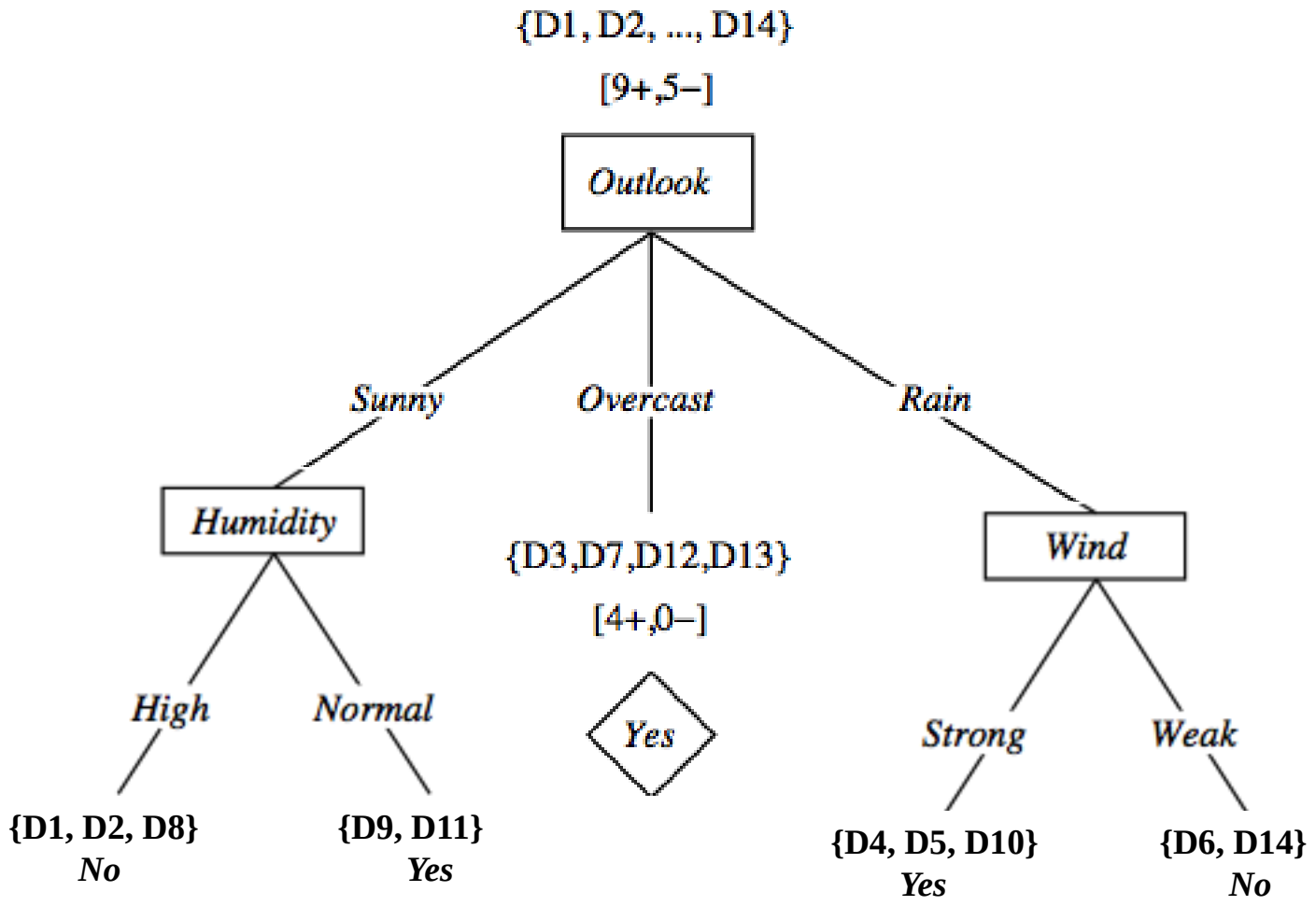
S_{sunny} = {D1,D2,D8,D9,D11}

Gain (S_{sunny} , *Humidity*) = .970 − (3/5) 0.0 − (2/5) 0.0 = .970

Gain (S_{sunny} , *Temperature*) = .970 − (2/5) 0.0 − (2/5) 1.0 − (1/5) 0.0 = .570

Gain (S_{sunny} , *Wind*) = .970 − (2/5) 1.0 − (3/5) .918 = .019

Second and third steps



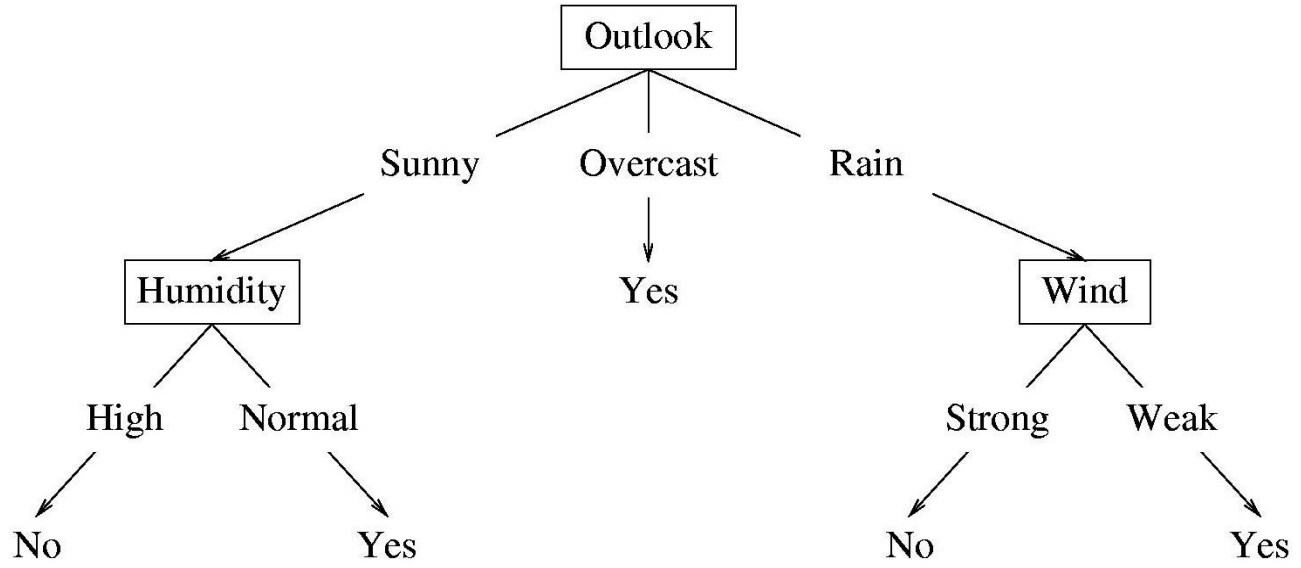
ID3: algorithm

ID3(*X, T, Attrs*) *X*: training examples:
T: target attribute (e.g. *PlayTennis*),
Attrs: other attributes, initially all attributes

Create *Root* node
If all X's are +, *return Root* with class +
If all X's are –, *return Root* with class –
If Attrs is empty *return Root* with class most common value of *T* in *X*
else
  A ← best attribute; decision attribute for Root ← *A*
  For each possible value v_i of *A:*
   - add a new branch below *Root,* for test $A = v_i$
   - X_i ← subset of *X* with $A = v_i$
   - *If* X_i is empty *then* add a new leaf with class the most common value of *T* in *X*
     else add the subtree generated by ID3(X_i, *T, Attrs* – {*A*})
return Root

Refer to https://en.wikipedia.org/wiki/ID3_algorithm

Summary of Decision Trees (so far)



- Decision tree induction -> choose the best attribute
  - Choose split via information gain
  - Build tree greedily, recursing on children of split
  - Stop when we achieve homogeny
    - i.e., when all instances in a child have the same class

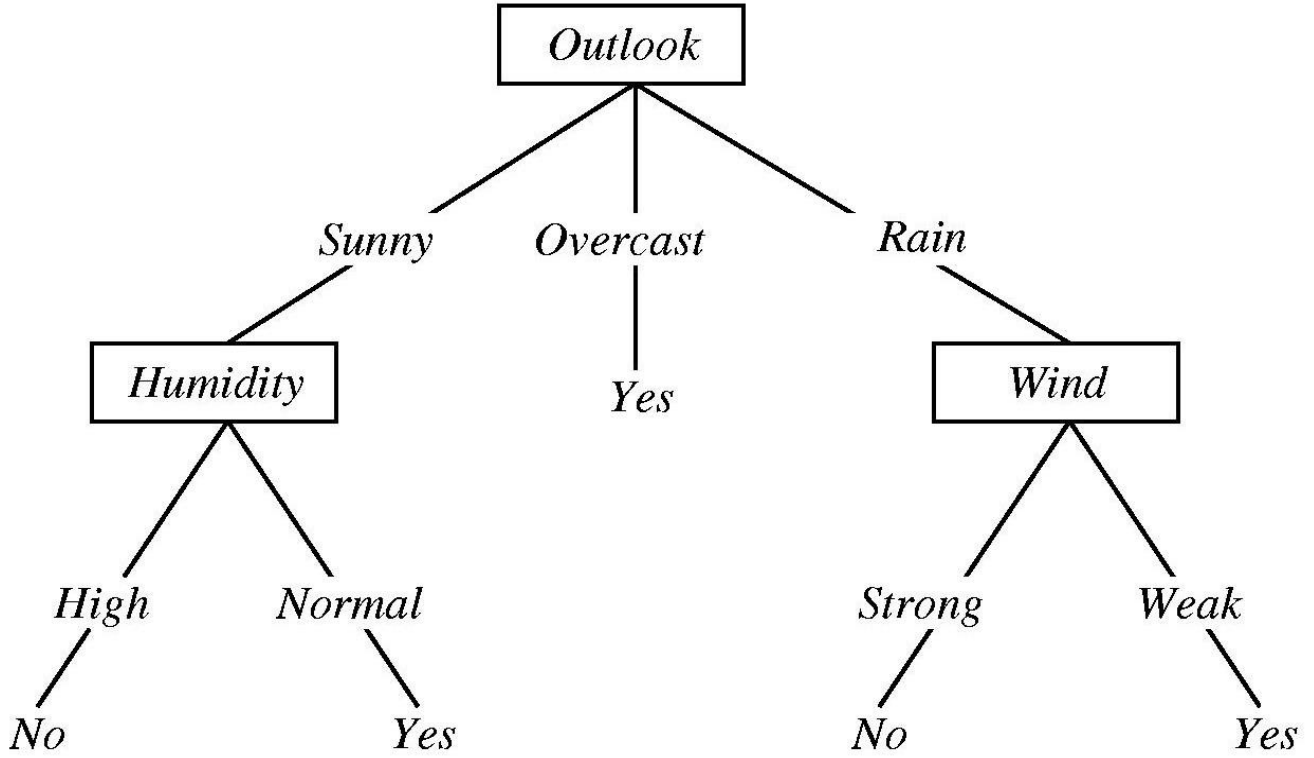
Overfitting

- Irrelevant attributes can result in *overfitting* the training example data
  - If hypothesis space has many dimensions (large number of attributes), we may find **meaningless regularity** in the data that is irrelevant to the true, important, distinguishing features

- If we have too little training data, even a reasonable hypothesis space will ‘overfit’

Overfitting in Decision Trees

Consider adding a noisy training example to the following tree:



What would be the effect of adding:

<outlook=sunny, temperature=hot, humidity=normal, wind=strong, playTennis=No> ?

=> New noisy example causes splitting of second leaf node.

Overfitting

Consider error of hypothesis h over

- training data: $error_{train}(h)$
- entire distribution $\mathcal{D}$ of data: $error_{\mathcal{D}}(h)$

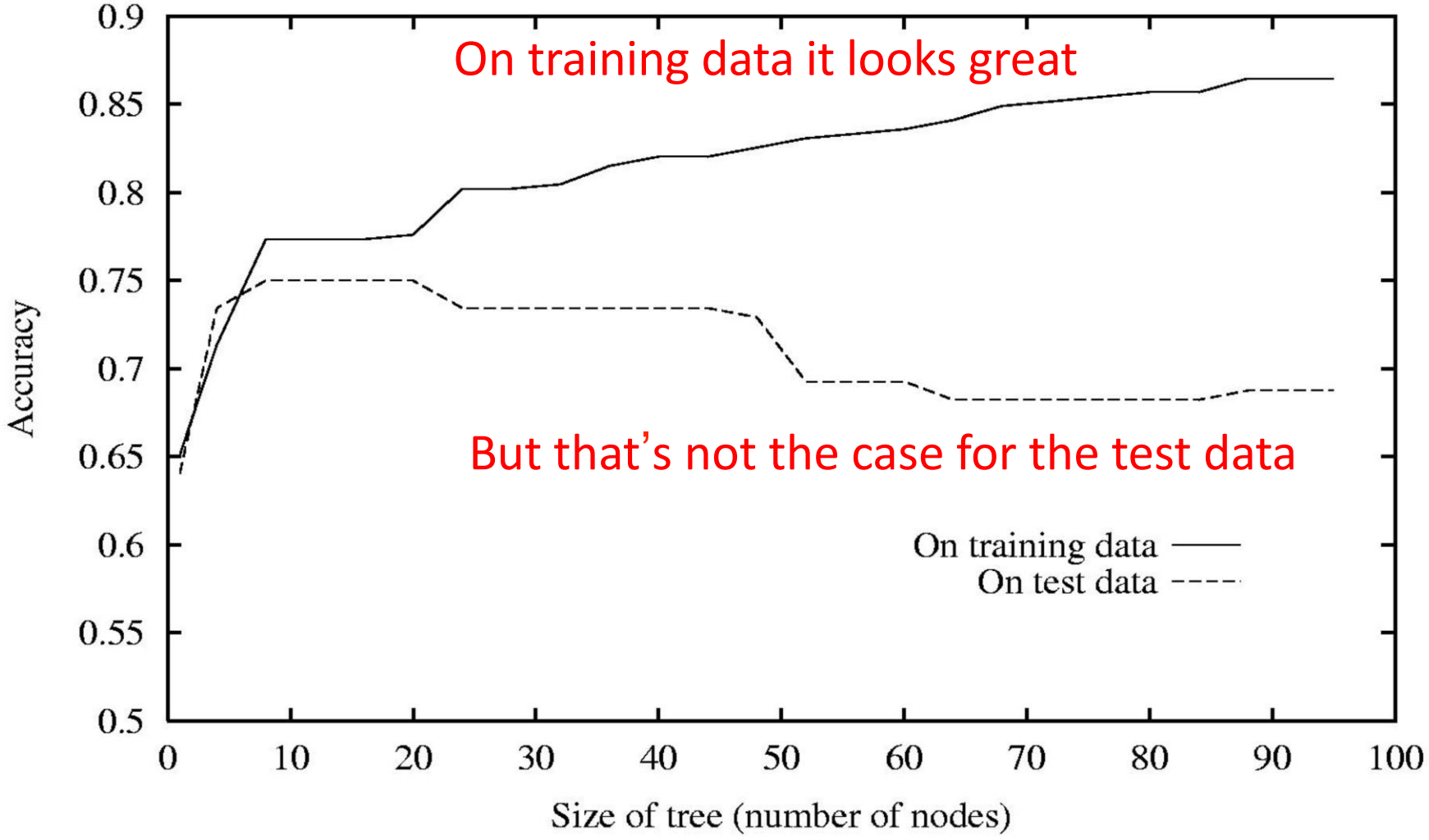
Hypothesis $h \in H$ **overfits** training data if there is an alternative hypothesis $h' \in H$ such that

$$error_{train}(h) < error_{train}(h')$$

and

$$error_{\mathcal{D}}(h) > error_{\mathcal{D}}(h')$$

Overfitting in Decision Tree Learning



Avoiding Overfitting

- How can we avoid overfitting?
  - Stop growing when data split is not statistically significant (e.g., chi-squared test)
  - Acquire more training data
  - Allow the tree to *overfit* the data (i.e., grow full tree), and then *post-prune* the tree

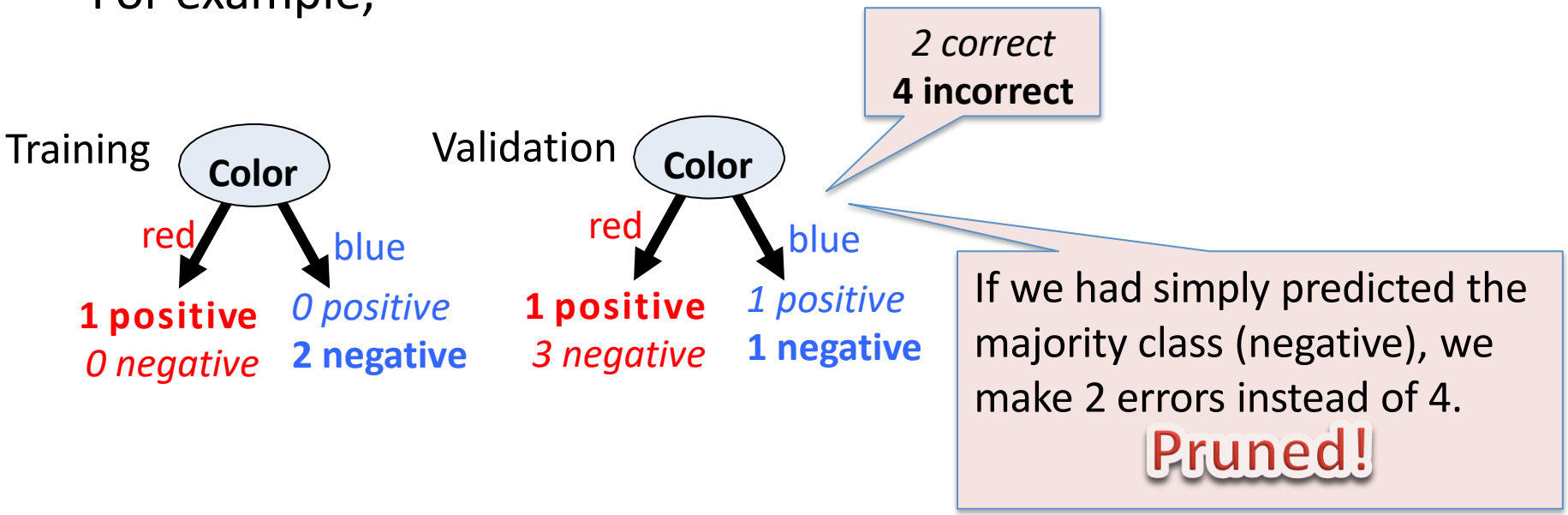
- Training and validation set
  - Split the training in two parts (training and validation) and use validation to assess the utility of post-pruning
    - Reduced error pruning

Reduced-Error Pruning

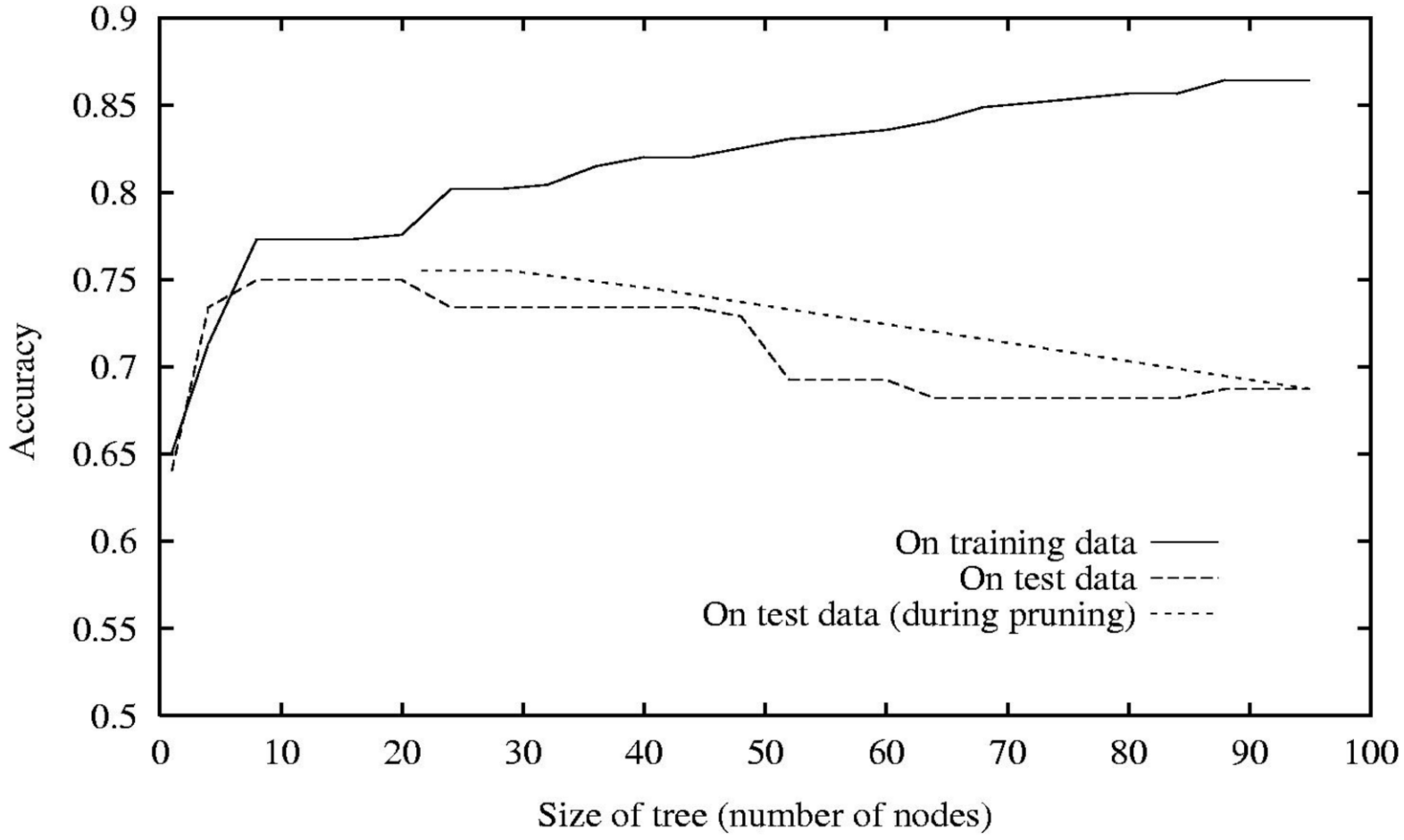
- Each node is a candidate for pruning
- *Pruning* consists in removing a subtree rooted in a node: the node becomes a leaf and is assigned the most common classification
- Nodes are removed only if the resulting tree performs no worse on the validation set.
- Nodes are pruned iteratively: at each iteration the node whose removal most increases accuracy on the validation set is pruned.
- Pruning stops when no pruning increases accuracy

Pruning Decision Trees

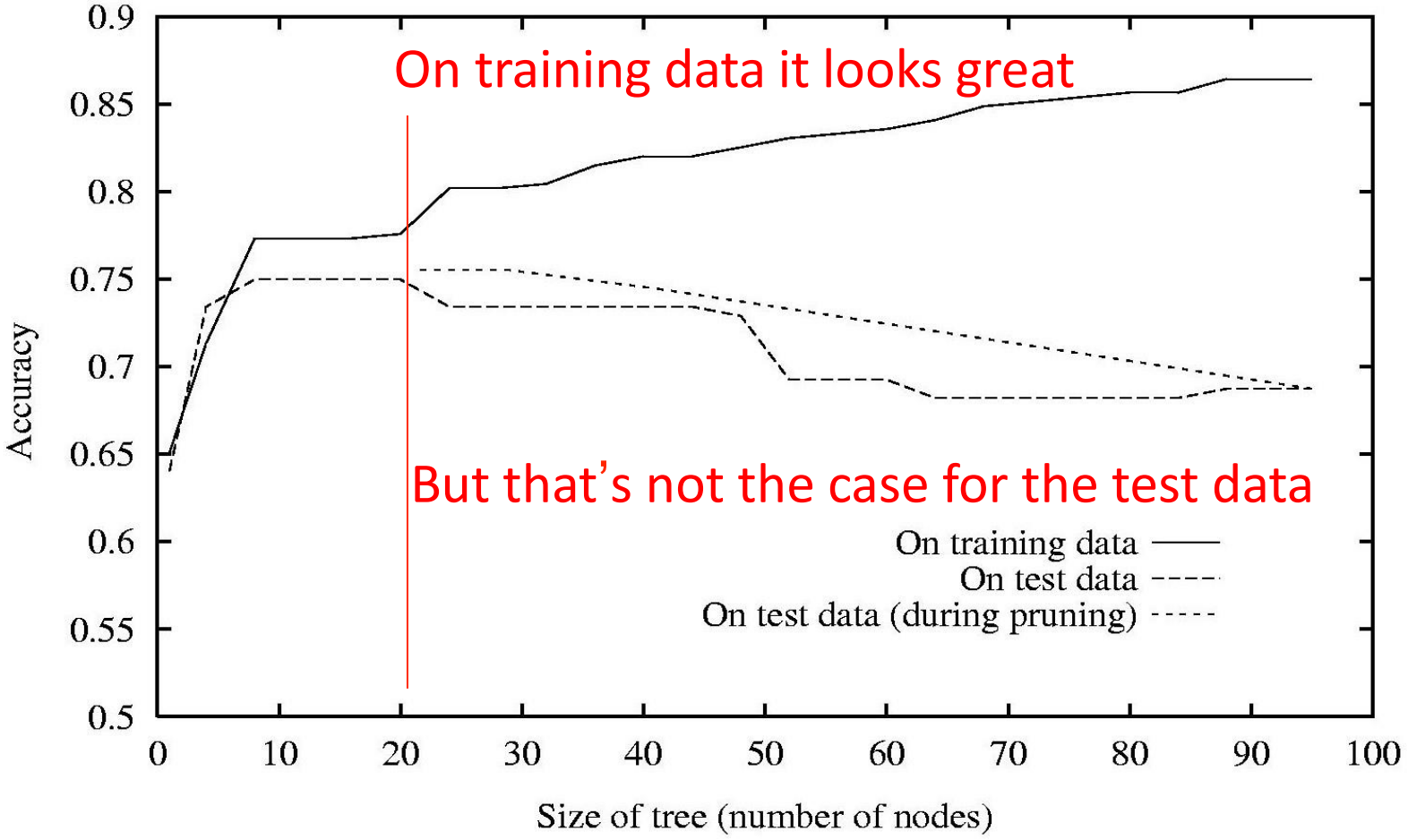
- Pruning of the decision tree is done by replacing a whole subtree by a leaf node.
- The replacement takes place if a decision rule establishes that the expected error rate in the subtree is greater than in the single leaf.
- For example,



Effect of Reduced-Error Pruning



Effect of Reduced-Error Pruning



The tree is pruned back to the red line where it gives more accurate results on the test data

Summary: Decision Tree Learning

- Representation: decision trees
- Bias: prefer small decision trees
- Search algorithm: greedy
- Heuristic function: information gain
- Overfitting / pruning

Evaluation

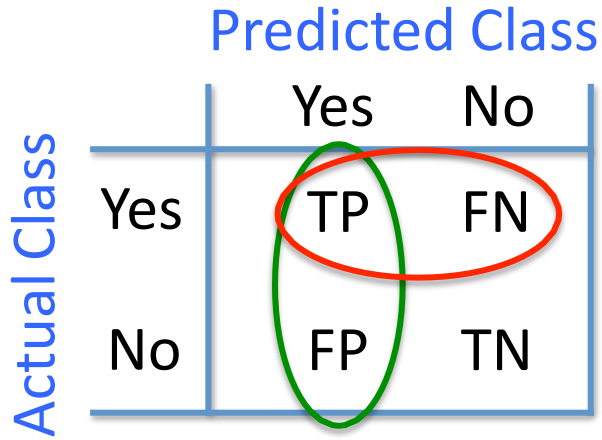
Classification Metrics

$$\text{accuracy} = \frac{\#\text{ correct predictions}}{\#\text{ test instances}}$$

$$\text{error} = 1 - \text{accuracy} = \frac{\#\text{ incorrect predictions}}{\#\text{ test instances}}$$

Confusion Matrix

- Given a dataset of *P* positive instances and *N* negative instances:



$$\text{accuracy} = \frac{TP + TN}{P + N}$$

- Imagine using classifier to identify positive cases (i.e., for information retrieval)

$$\text{precision} = \frac{TP}{TP + FP}$$

Probability that a randomly selected result is relevant

$$\text{recall} = \frac{TP}{TP + FN}$$

Probability that a randomly selected relevant document is retrieved

HW1

- https://canvas.wpi.edu/courses/58900/assignments/355140
- Due date is June 6th 11:59pm.