# CSDS 440: Machine Learning

Soumya Ray (he/him, sray@case.edu)

Olin 516

Office hours T, Th 11:15-11:45 or by appointment

# Today

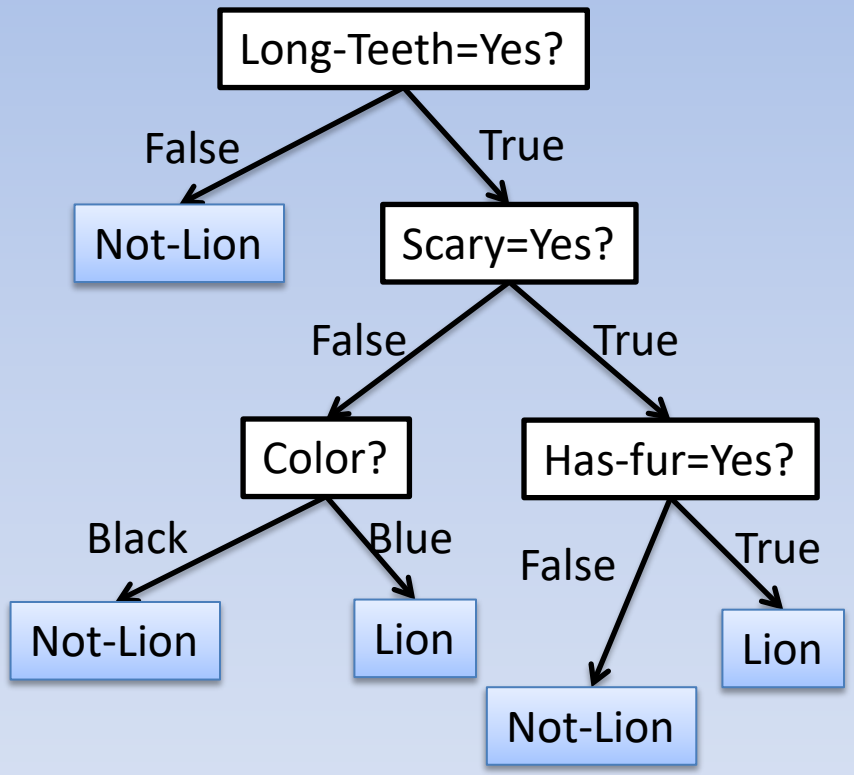
- Overfitting and overfitting control
- Evaluation Methodology and Metrics

# An Issue

- Given enough features, ID3 will usually be able to fit training examples exactly (i.e. every leaf is pure), because the tree can be grown as much as needed

- But real data is noisy

# Example




| | Has-fur? | Long-Teeth? | Scary? | Color? | *Lion?* |
|---|---|---|---|---|---|
| $Animal_1$ | Yes | No | No | Green | No |
| $Animal_2$ | No | Yes | No | Black | No |
| $Animal_3$ | Yes | Yes | Yes | Golden | Yes |
| $Animal_4$ | Yes | Yes | No | Blue | Yes |
| $Animal_5$ | Yes | Yes | Yes | Tawny | Yes |

# Overfitting

- If a learned concept $h$ has
  - Higher performance (lower error) on the training examples, BUT
  - Lower performance (higher error) on average across all examples

than some alternative concept $h'$ in the same hypothesis space, $h$ is said to have overfit to the training examples

# Overfitting

# Controlling Overfitting

- Introduce a restriction on the hypothesis space to prevent overly-complex hypotheses from being learned
  - Early Stopping
  - Post Pruning

# Early Stopping

- Standard algorithm stops growing the tree when $IG(X)=0$ for all $X$
- Early stopping stops growing the tree when $IG(X) \leq \varepsilon,$ for some chosen $\varepsilon$
- Sensitive to choice of $\varepsilon$
- Easy to implement, but does not work very well in practice

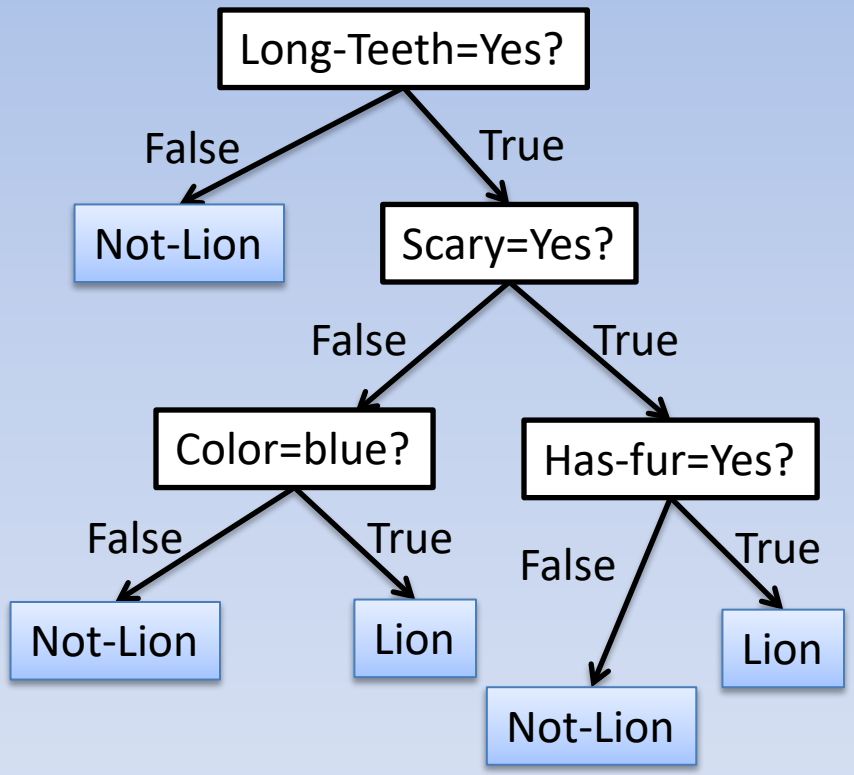
# Greedy post-pruning

- Hold aside some training examples at start (validation set)
- Grow tree as usual on remainder
- Then run a *greedy pruning* algorithm

# Greedy post-pruning

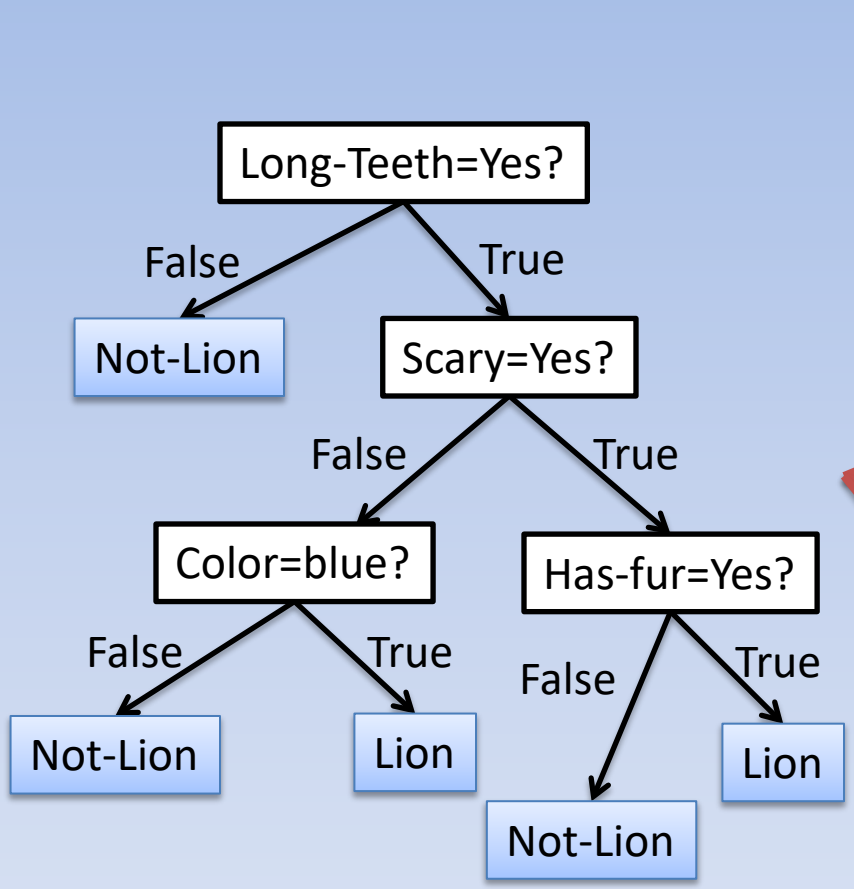
- For each internal node, construct a tree without that node
  - Convert node to leaf by predicting majority class
  - Delete subtree below node
- Evaluate this *pruned* tree on the validation set
- Find the single node that improves performance the most over the unpruned tree and remove it
- Repeat steps above until no node removal improves performance

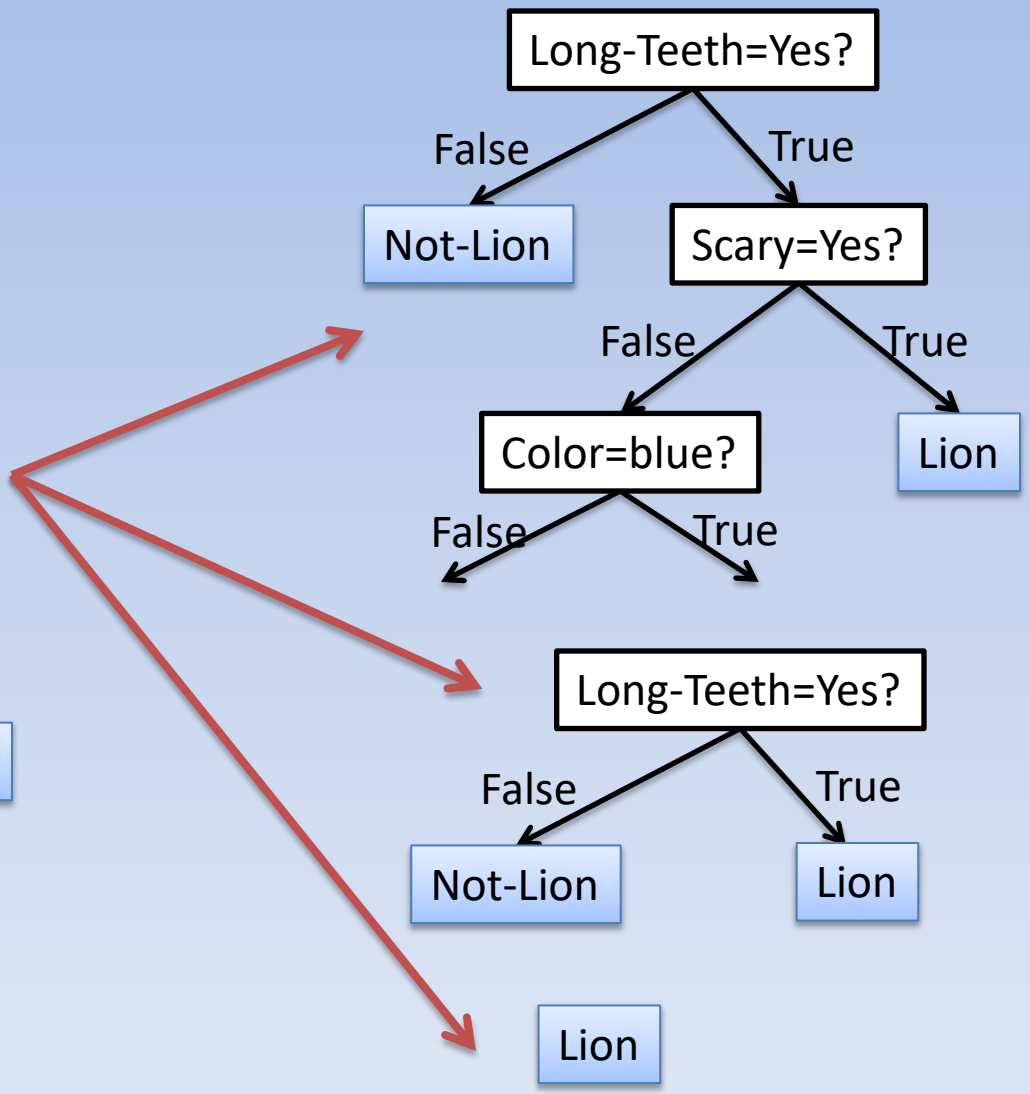
# Greedy Post Pruning



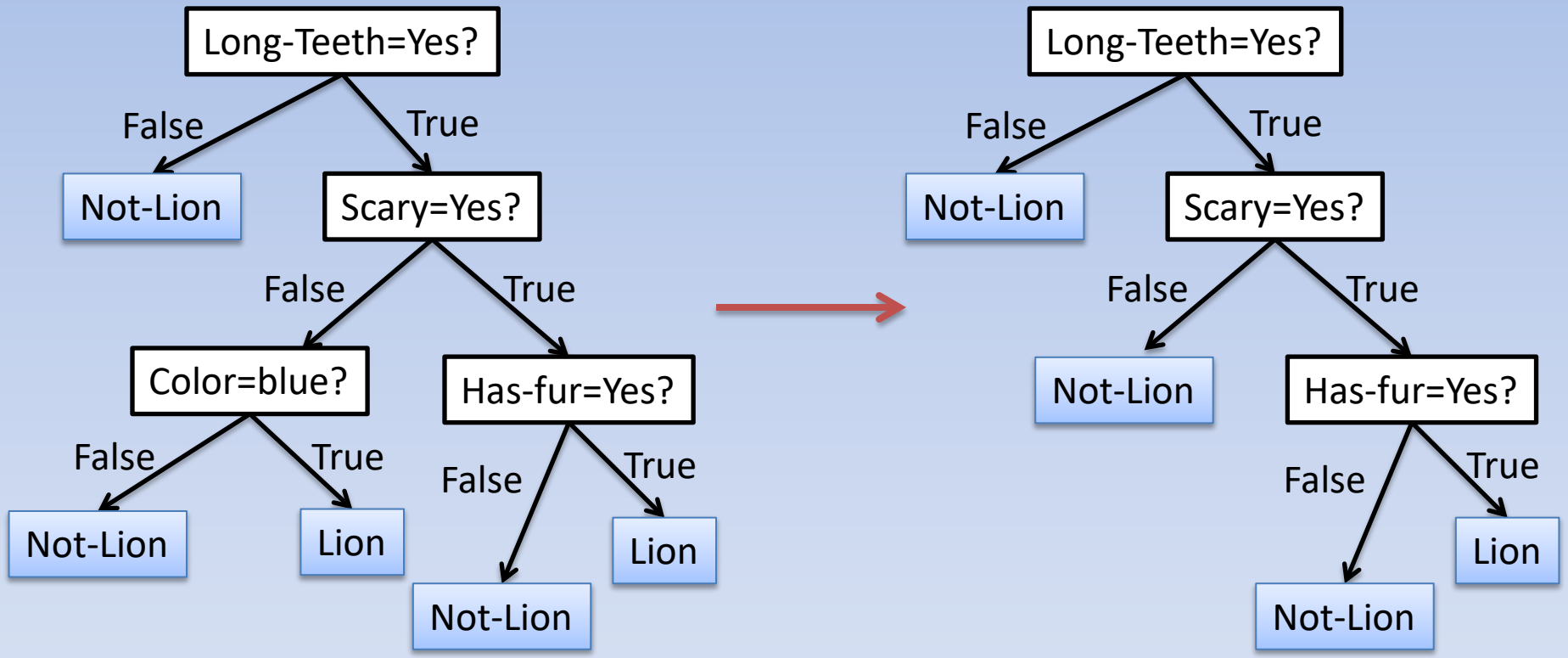

| | Has-fur? | Long-Teeth? | Scary? | Color? | *Lion?* |
|---|---|---|---|---|---|
| $Animal_1$ | Yes | No | No | Green | No |
| $Animal_2$ | No | Yes | No | Black | No |
| $Animal_3$ | Yes | Yes | Yes | Golden | Yes |
| $Animal_4$ | Yes | Yes | No | Blue | Yes |
| $Animal_5$ | Yes | Yes | Yes | Tawny | Yes |
| $Animal_6$ | No | Yes | No | Blue | No |

# Greedy Post Pruning

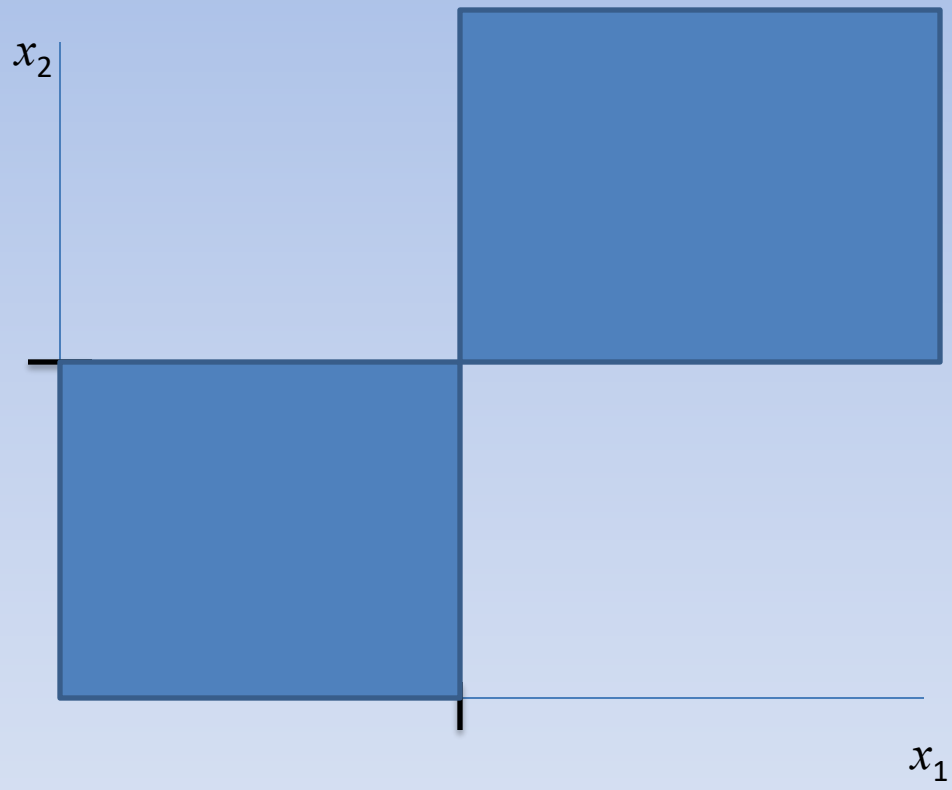


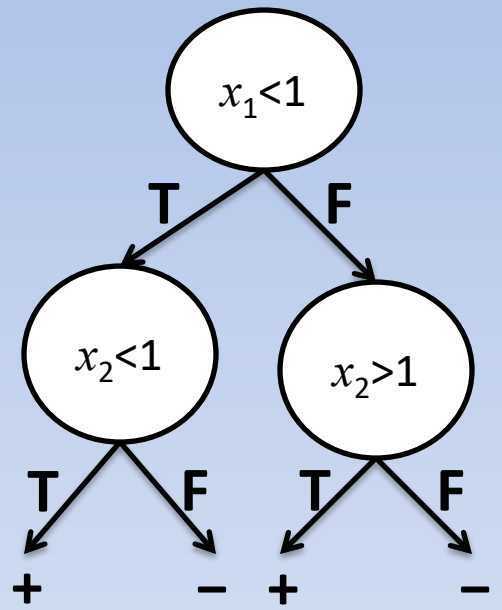



# Greedy Post Pruning



# Decision Tree Geometry (Continuous features)



# Extensions

- Idea can be extended to handle:
  - Multiclass classification
  - Regression
  - Functional tests in internal nodes (Function Trees)
  - More complex functions in leaves (Model Trees)
  - Density functions in leaves (PETs)

# Pros and Cons of Decision Trees

+ Does not require metric space representation
+ Produces human-comprehensible concepts
+ Can produce concepts with range of complexity
+ Easily extendable to various other scenarios
+ Easy to combine with other algorithms (general purpose partitioning)

− Attributes with lots of values (including continuous attributes)
− Attributes with complex interactions
− Partitioning strategy means easier to overfit as depth increases