DSC 140A

Probabilistic Modeling & Machine Learning

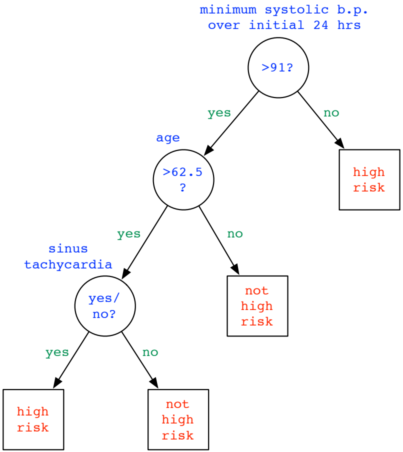
Lecture 17 | Part 1

Decision Trees

The Problem

- UCSD Medical Center (1970s): identify patients at risk of dying within 30 days after heart attack.

A Decision Tree

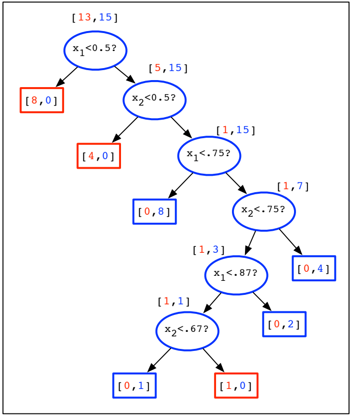


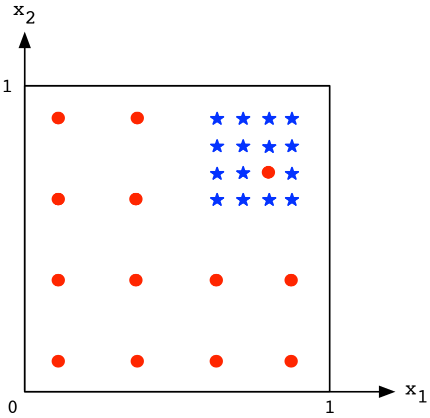
Decision Trees

- A **decision tree** is a rooted tree.
- Internal nodes ask yes/no questions.
  - **Categorical**: Is patient a male?
  - **Numerical**: Is patient's age > 62.5 years?
- Leaf nodes are decisions (class labels).
- Path from root is a sequence of "and"s:
  - Is patient over 62.5 **and** male **and** BP > 100?
    Then high risk.

Prediction

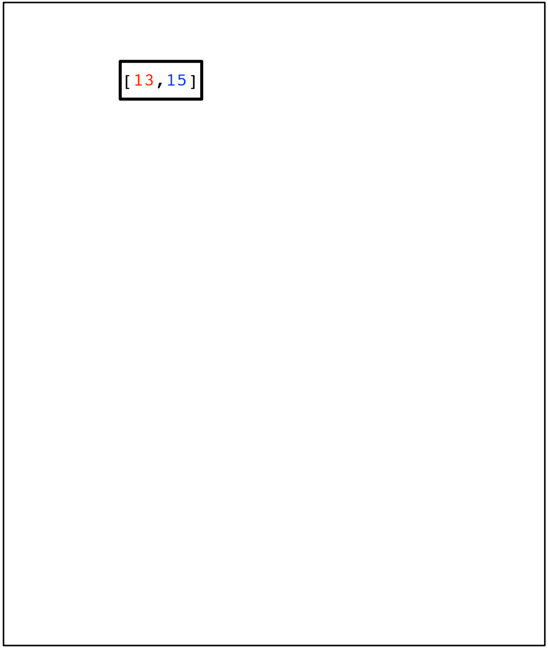
- To make prediction, traverse tree.
  - Example: (0.75, 0.6)

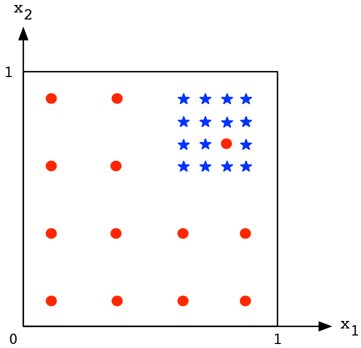


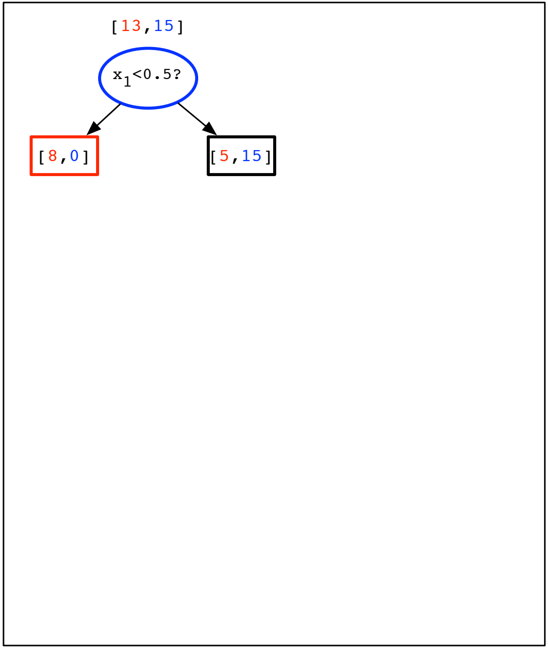


Learning Decision Trees

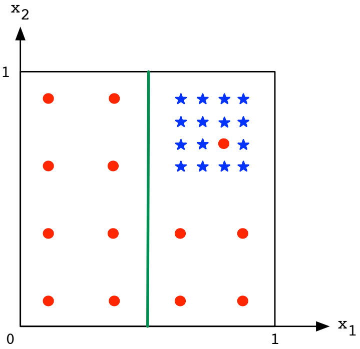
- How do we **learn** a tree from data?
  - Find right sequence of questions so that each training point is correctly classified.


[13,15]

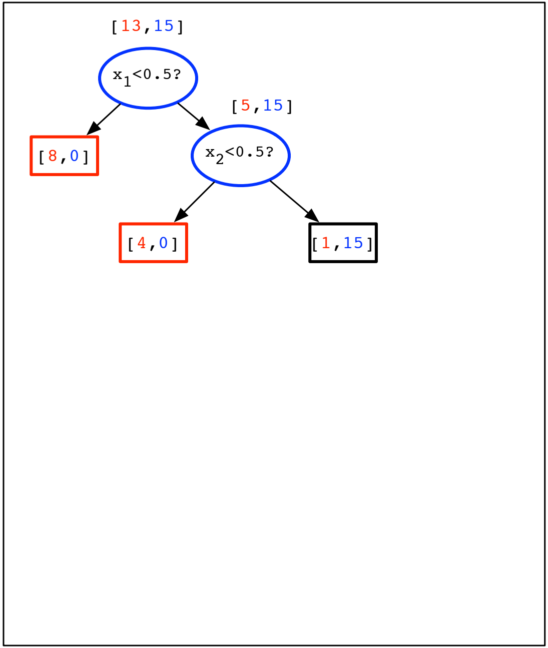

x_2
1
0
1
x_1

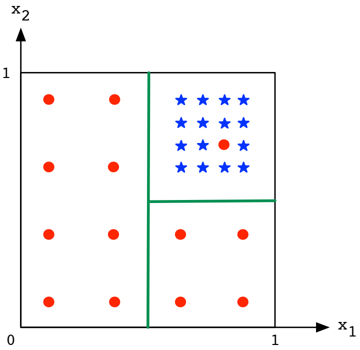


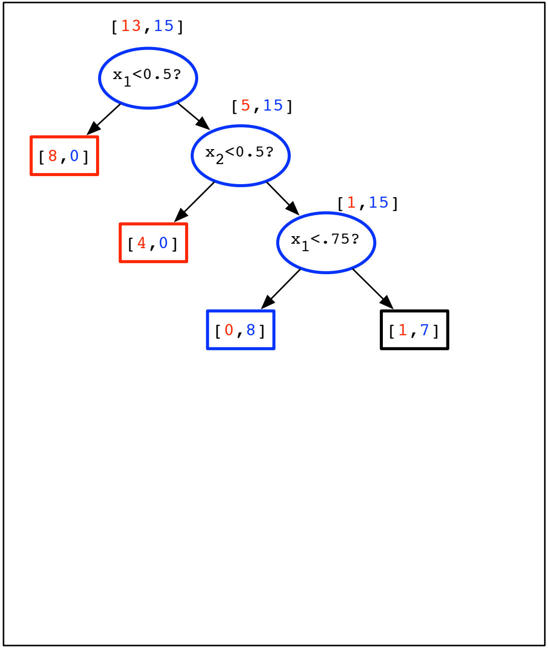
[13,15]
x1<0.5?
[8,0]
[5,15]

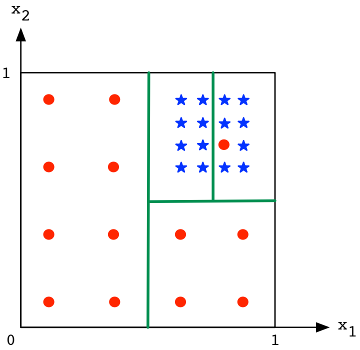


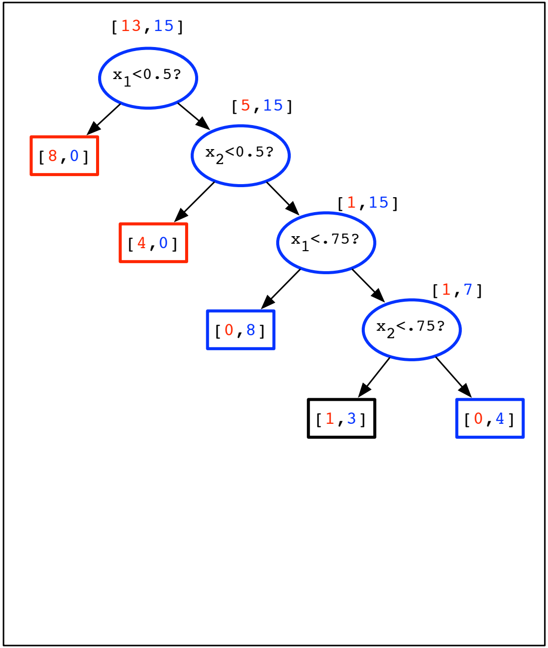
x2
1
0
1
x1

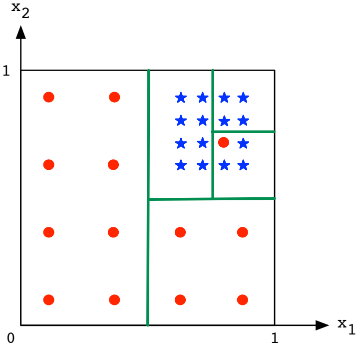

[13,15]
x1<0.5?
[5,15]
[8,0]
x2<0.5?
[4,0]
[1,15]

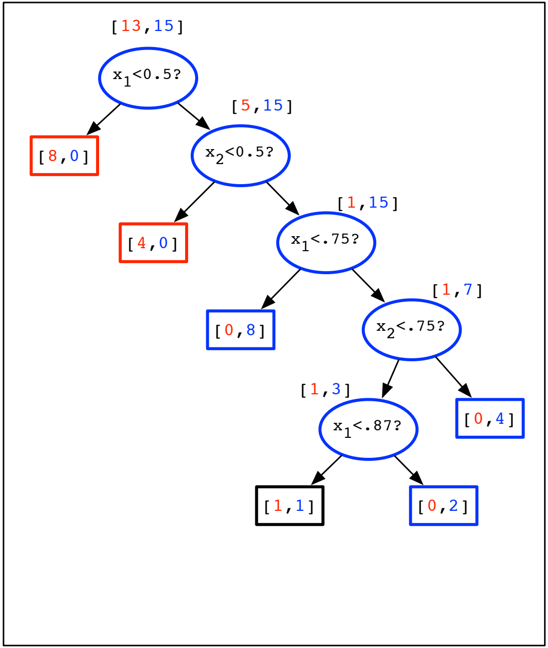

x2
1
0
1
x1

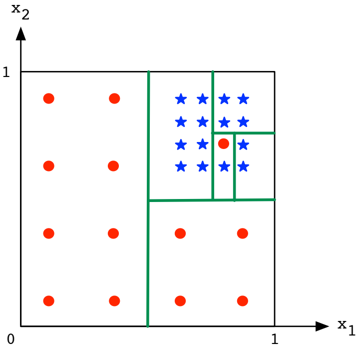

[13,15]
x1<0.5?
[8,0]
[5,15]
x2<0.5?
[4,0]
[1,15]
x1<.75?
[0,8]
[1,7]

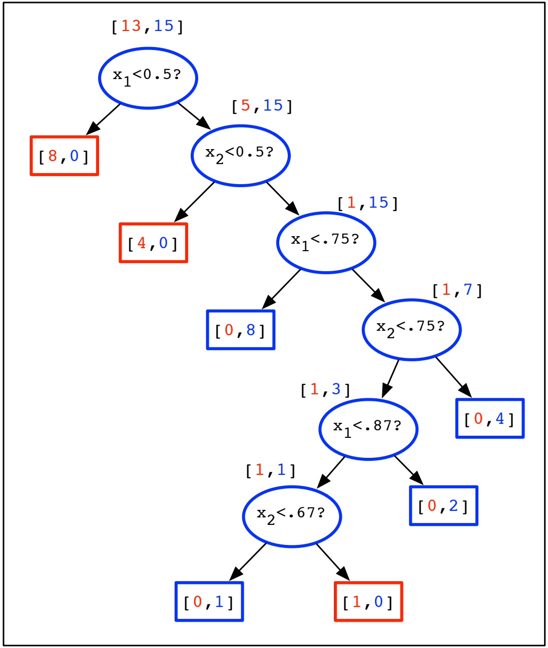

x2
1
0
1
x1

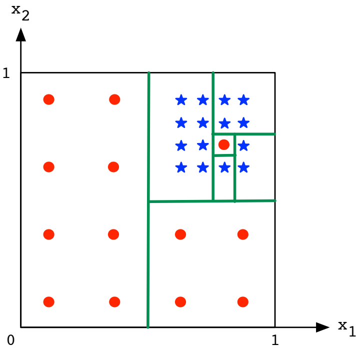

[13,15]
x1<0.5?
[8,0]
[5,15]
x2<0.5?
[4,0]
[1,15]
x1<.75?
[0,8]
[1,7]
x2<.75?
[1,3]
[0,4]


x2
1
0
1
x1


[13,15]
x1<0.5?
[8,0]
[5,15]
x2<0.5?
[4,0]
[1,15]
x1<.75?
[0,8]
[1,7]
x2<.75?
[1,3]
x1<.87?
[0,4]
[1,1]
[0,2]


x2
1
0
1
x1


[13,15]
x1<0.5?
[8,0]
[5,15]
x2<0.5?
[4,0]
[1,15]
x1<.75?
[0,8]
[1,7]
x2<.75?
[1,3]
x1<.87?
[0,4]
[1,1]
x2<.67?
[0,2]
[0,1]
[1,0]


x2
1
0
1
x1

Learning Decision Trees

- Start with single node containing all data points

- Repeat greedy procedure:
  - Look at all possible questions (splits)
  - Pick the one that most reduces **uncertainty**.

- Stop when each leaf node is **pure**.

Aside: Generating Possible Questions

- **Categorical**: One question per value seen.

- E.g., county of residence.
  - Patient is from San Diego County?
  - Patient is from Riverside County?
  - Patient is from Orange County?

Aside: Generating Possible Questions

- **Numerical**: one question between each pair of consecutive values.
- E.g., ages in data = $\{42, 43, 55, 57, 61, 75\}$
  - Patient is < 42.5?
  - Patient is < 49?
  - ...
  - Patient is < 68?

Measuring Uncertainty

- A good question splits the data by class.

[15, 12]

↓

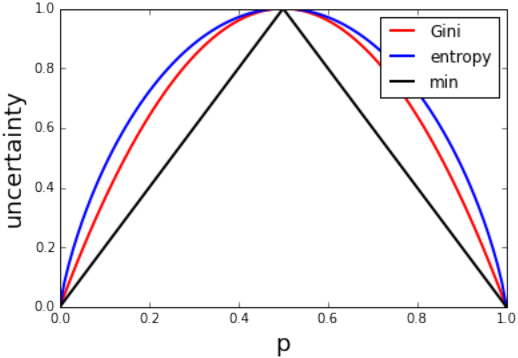
"Is patient > 62?"

yes → [13, 3]

no → [2, 9]

Measuring Uncertainty

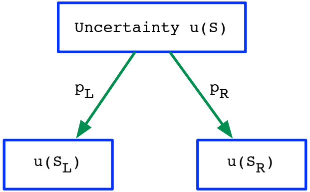
- Suppose our node contains proportions:
  - p from class +
  - $(1 - p)$ from class -
- Common **uncertainty scores**:
  - **Misclassification rate**: $\min\{p, 1 - p\}$
  - **Gini index**: $2p(1 - p)$
  - **Entropy**: $p \log \frac{1}{p} + (1 - p) \log \frac{1}{1-p}$



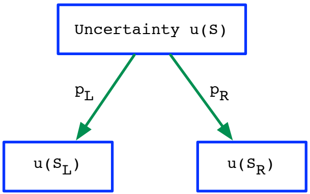
Gini
entropy
min
uncertainty
p
0.0
0.2
0.4
0.6
0.8
1.0

Benefit of a Question

- Let $u(S)$ be the uncertainty score for a set of labeled points, S.
  - Consider a particular question (split):



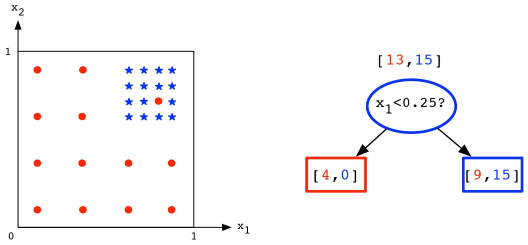
Benefit of a Question



- Resulting uncertainty:

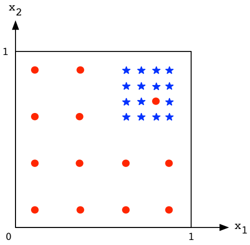
$$p_L\, u(S_L) + p_R\, u(S_R)$$

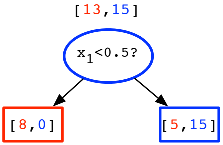
Example



- Initial Gini uncertainty: $2 \times \frac{13}{28} \times \frac{15}{28}$.
- $p_L\, u(S_L) + p_R\, u(S_R) = \frac{4}{28} \cdot 0 + \frac{24}{28} \cdot 2 \cdot \frac{9}{24} \cdot \frac{15}{24} = \frac{45}{112}$

Example





- Initial Gini uncertainty: $2 \times \frac{13}{28} \times \frac{15}{28}$.
- $p_L\, u(S_L) + p_R\, u(S_R) = \frac{8}{28} \cdot 0 + \frac{20}{28} \cdot 2 \cdot \frac{5}{20} \cdot \frac{15}{20} = \frac{30}{112}$

Example

- Because the second split (is $x_1 < 0.5$?) has lower uncertainty, it is “better” than the first.

- To pick the best question, we need to consider all possible splits, choose the one that minimizes uncertainty.
  - $x_1 < 0.25$?
  - $x_1 < 0.5$?
  - $\vdots$
  - $x_2 < 0.8$?
  - $x_2 < 0.9$?

Summary

To learn a decision tree:

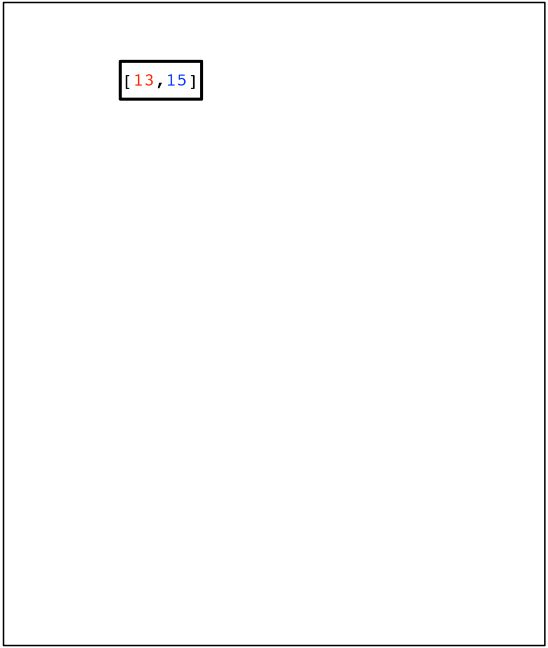
- Pick a measure of uncertainty (Gini, Entropy, etc.)
- Recursively ask question minimizing uncertainty.

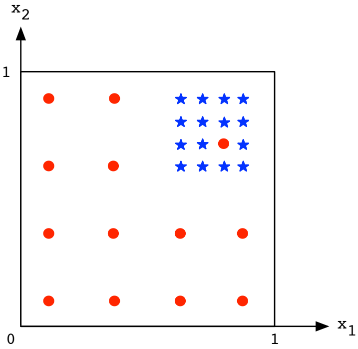
DSC 140A

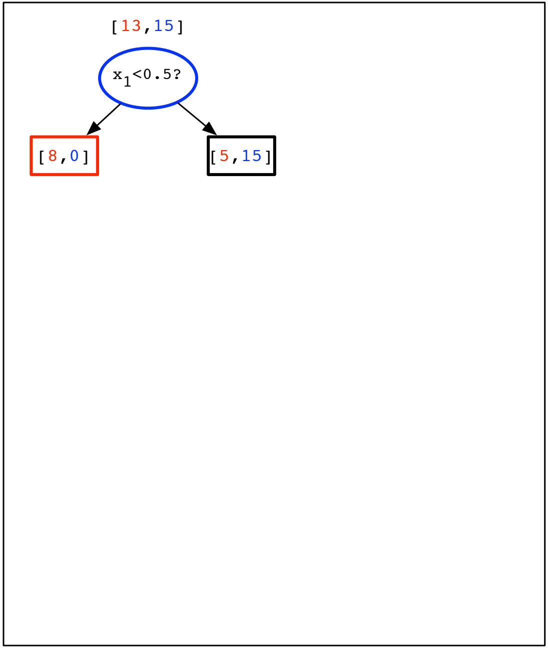
Probabilistic Modeling & Machine Learning

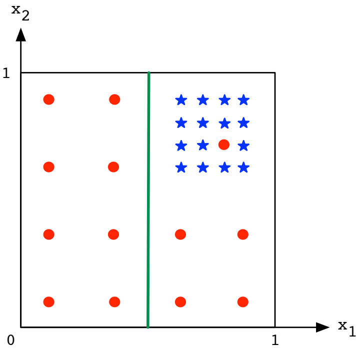
Lecture 17 | Part 2

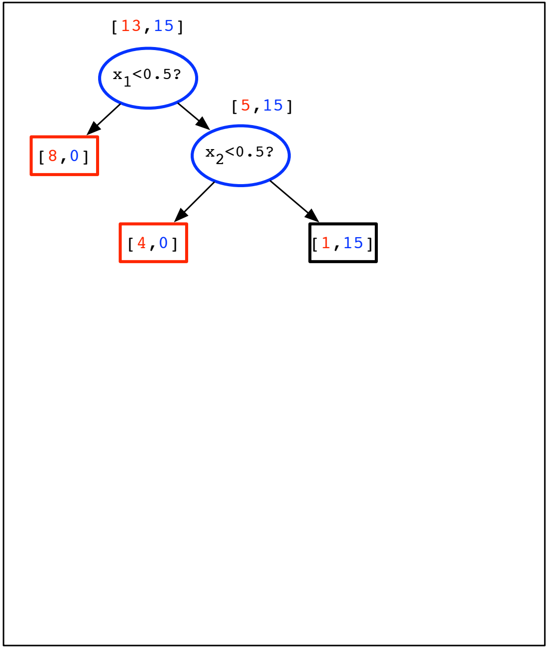
Overfitting


[13,15]

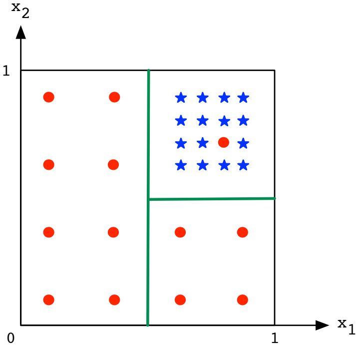

x2
1
0
1
x1


[13,15]
x1<0.5?
[8,0]
[5,15]

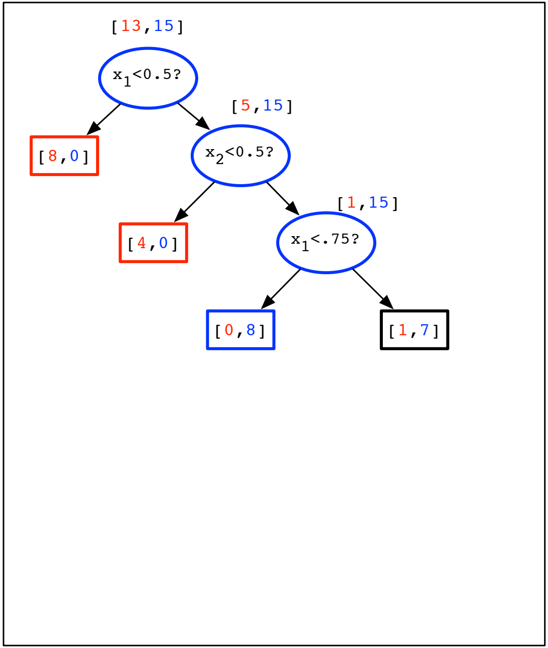

x2
1
0
1
x1



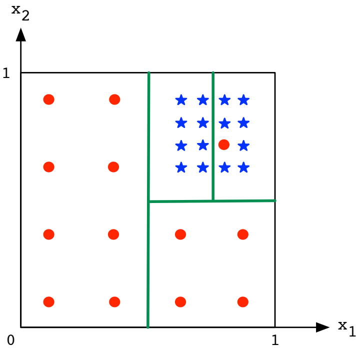
[13,15]
x1<0.5?
[5,15]
[8,0]
x2<0.5?
[4,0]
[1,15]



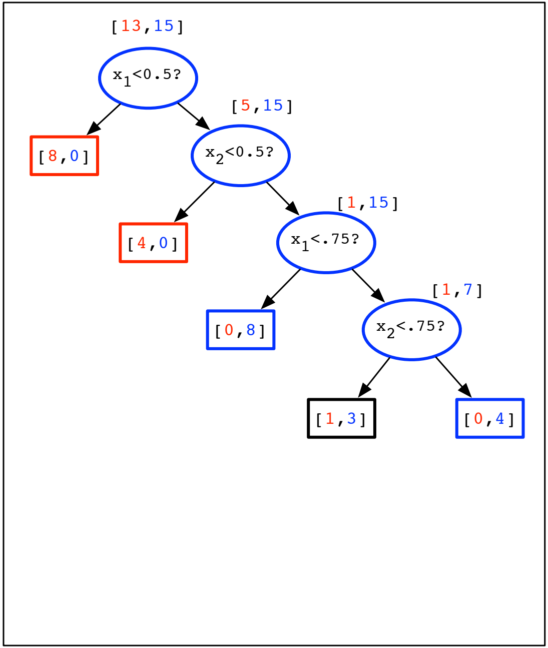
x2
1
0
1
x1

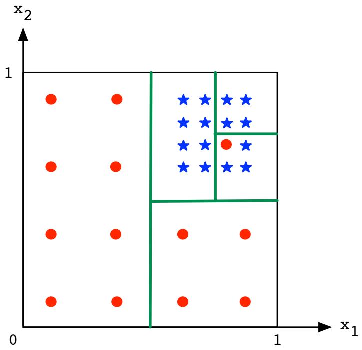


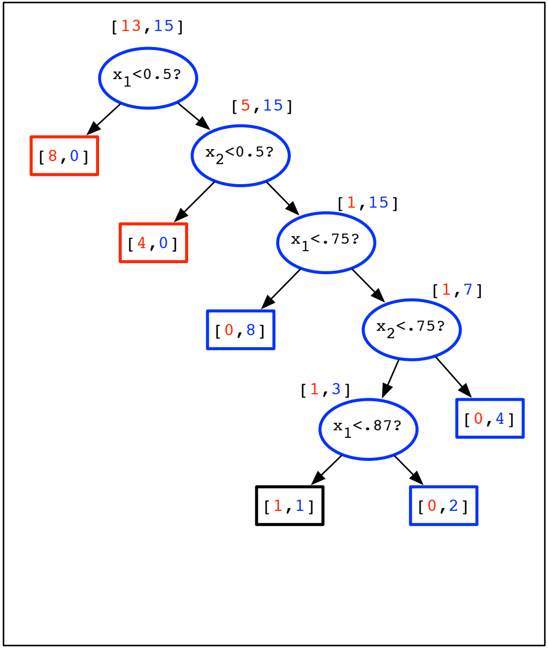
[13,15]
x1<0.5?
[8,0]
[5,15]
x2<0.5?
[4,0]
[1,15]
x1<.75?
[0,8]
[1,7]

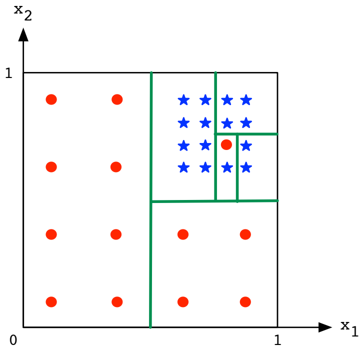


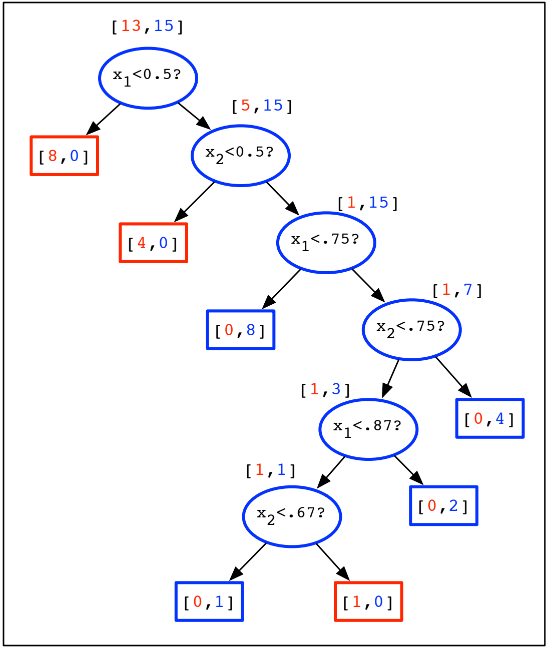
x2
1
0
1
x1

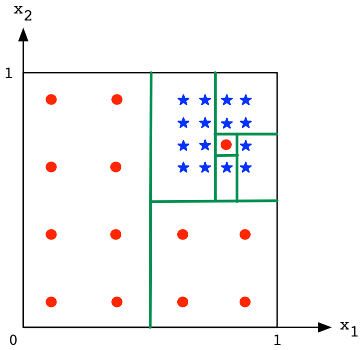

[13,15]
x1<0.5?
[8,0]
[5,15]
x2<0.5?
[4,0]
[1,15]
x1<.75?
[0,8]
[1,7]
x2<.75?
[1,3]
[0,4]


x2
1
0
1
x1


[13,15]
x1<0.5?
[8,0]
[5,15]
x2<0.5?
[4,0]
[1,15]
x1<.75?
[0,8]
[1,7]
x2<.75?
[1,3]
x1<.87?
[0,4]
[1,1]
[0,2]

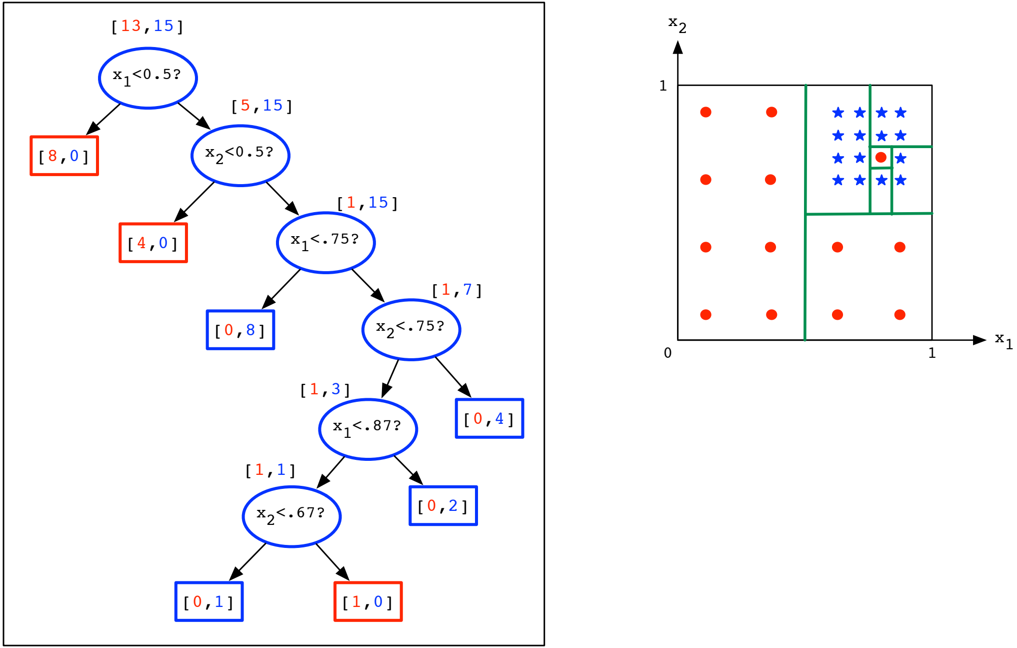

x2
1
0
1
x1


[13,15]
x1<0.5?
[8,0]
[5,15]
x2<0.5?
[4,0]
[1,15]
x1<.75?
[0,8]
[1,7]
x2<.75?
[1,3]
x1<.87?
[0,4]
[1,1]
x2<.67?
[0,2]
[0,1]
[1,0]

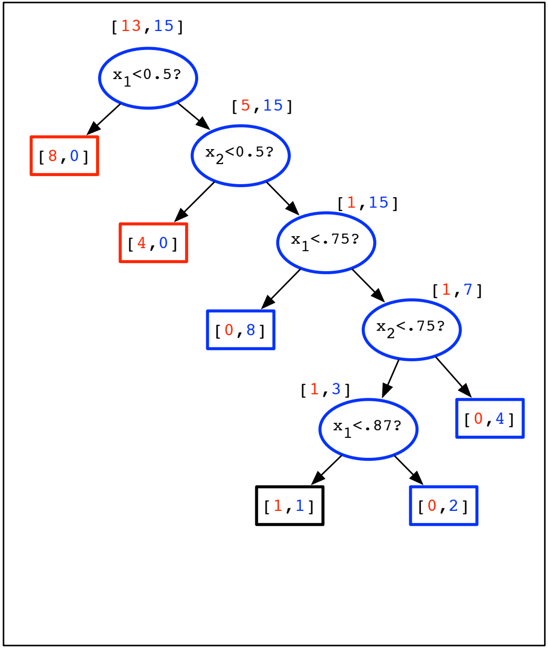

x2
1
0
1
x1

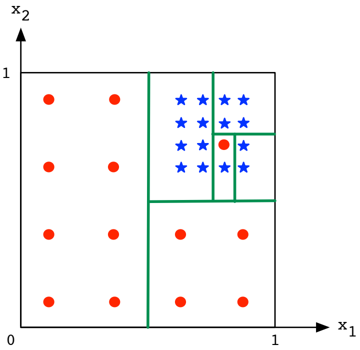
Overfitting

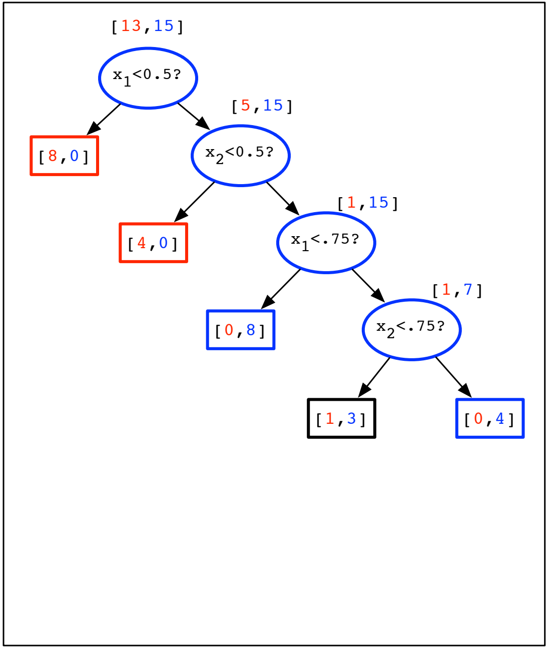
- The training error is **zero**.
  - We might be **overfitting**.
- (One) **solution**: rewind a few steps.

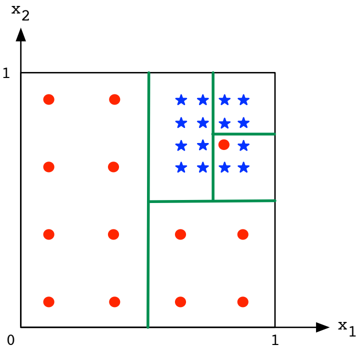


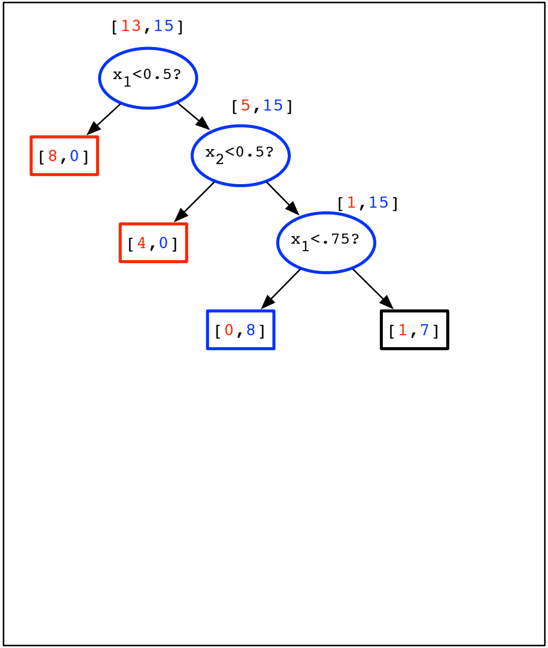
[13,15]
x1<0.5?
[8,0]
[5,15]
x2<0.5?
[4,0]
[1,15]
x1<.75?
[0,8]
[1,7]
x2<.75?
[1,3]
x1<.87?
[0,4]
[1,1]
x2<.67?
[0,2]
[0,1]
[1,0]
x2
1
0
1
x1

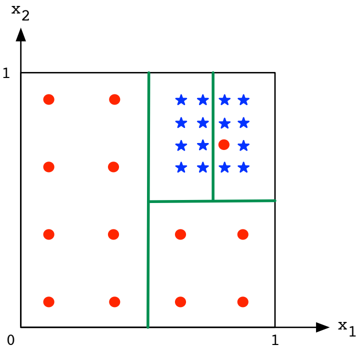

[13,15]
x1<0.5?
[8,0]
[5,15]
x2<0.5?
[4,0]
[1,15]
x1<.75?
[0,8]
[1,7]
x2<.75?
[1,3]
x1<.87?
[0,4]
[1,1]
[0,2]

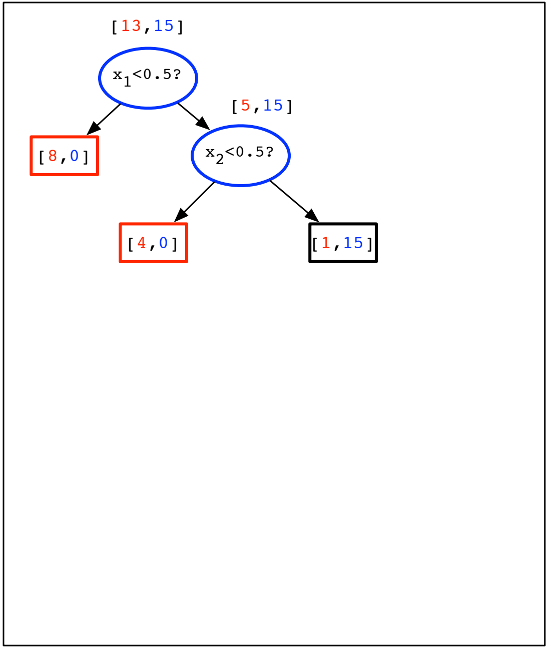

x2
1
0
1
x1

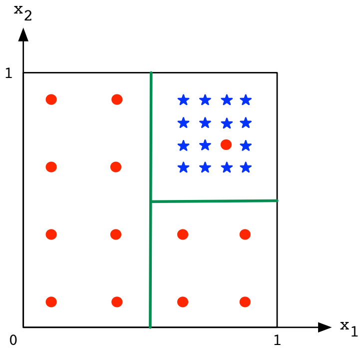

[13,15]
x1<0.5?
[8,0]
[5,15]
x2<0.5?
[4,0]
[1,15]
x1<.75?
[0,8]
[1,7]
x2<.75?
[1,3]
[0,4]


x2
1
0
1
x1

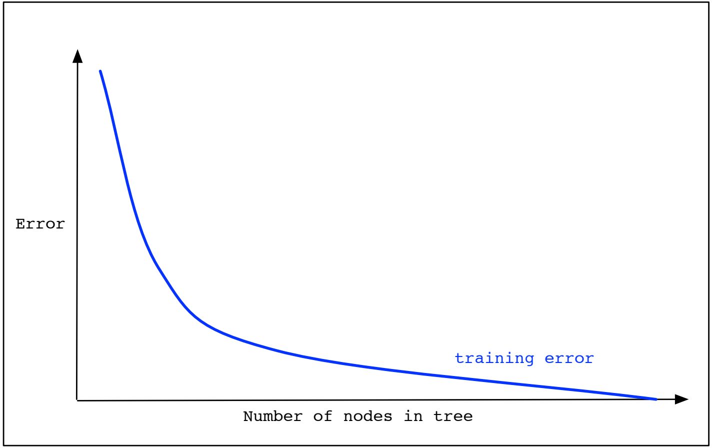

[13,15]
x1<0.5?
[8,0]
[5,15]
x2<0.5?
[4,0]
[1,15]
x1<.75?
[0,8]
[1,7]


x2
1
0
1
x1

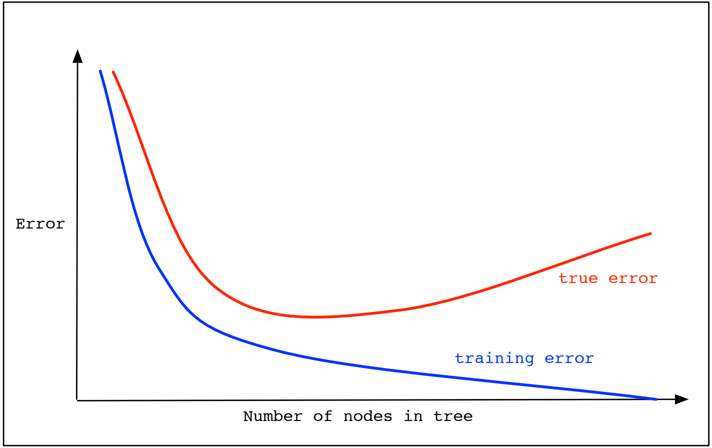

[13,15]
x1<0.5?
[8,0]
[5,15]
x2<0.5?
[4,0]
[1,15]


x2
1
0
1
x1

Overfitting



Overfitting



Two Strategies

- **Pruning**: simplify already-constructed tree.
- **Early-stopping**: stop early.

Pruning

- Given a full decision tree.
- Starting with predecessors of leaf nodes, replace node by most common class.
- If the change reduces validation error, keep it. Otherwise reverse it.

Early-Stopping

- Stop recursion when:
  - node is “pure enough” (uncertainty is low).
  - tree is too deep.

Decision Tree Properties

Very expressive:

- Can accommodate any type of data
  - numerical, Boolean, etc.
- Can accommodate any number of classes
- Can perfectly fit any data set
  - If data has no duplicates from different classes.
  - **Danger**! Overfitting!