Introduction to Machine Learning

Dor Bank

Lecture #8B
Decision Trees

On last time

- Constrained optimization
- SVM
- Kernel methods

Today – Decision Trees

- Architecture
- Learning regression decision trees
- Learning classification decision trees
- Pruning

Interpretable Models

- Machine learning problems make predictions
- It is often important to explain this predictions for:
  - Scientific understanding
  - Legal reasons (e.g., avoiding discrimination)
  - “Debugging” models

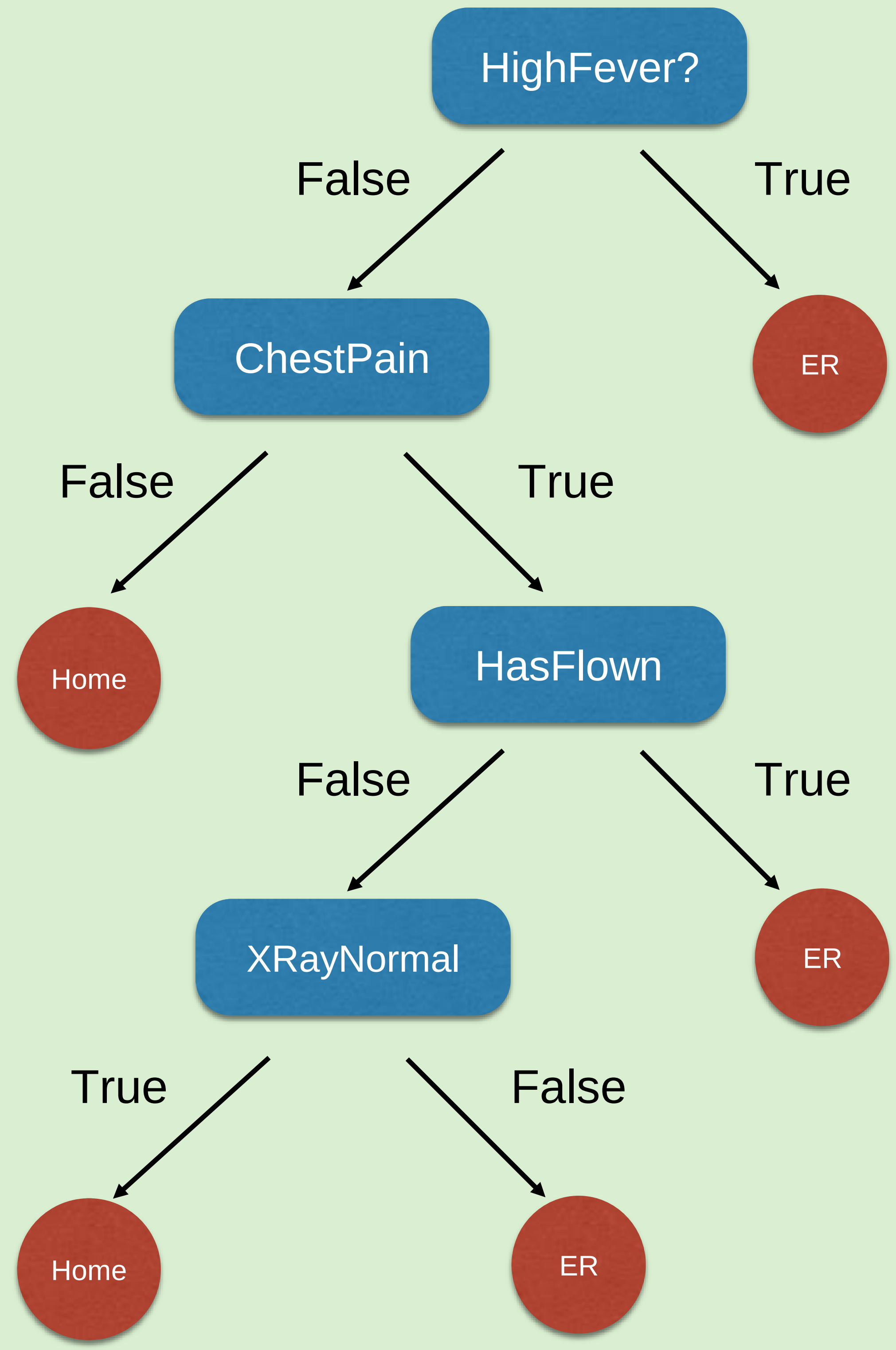
- Kernel SVM and deep learning are not very interpretable

Visiting a Doctor

- Should a doctor determine to send you to the ER?
- To decide, the doctor will ask a **sequence** of questions:
  - Do you have fever?
  - Did you have chest pains?
  - Did you fly recently?
  - …

Decision Trees

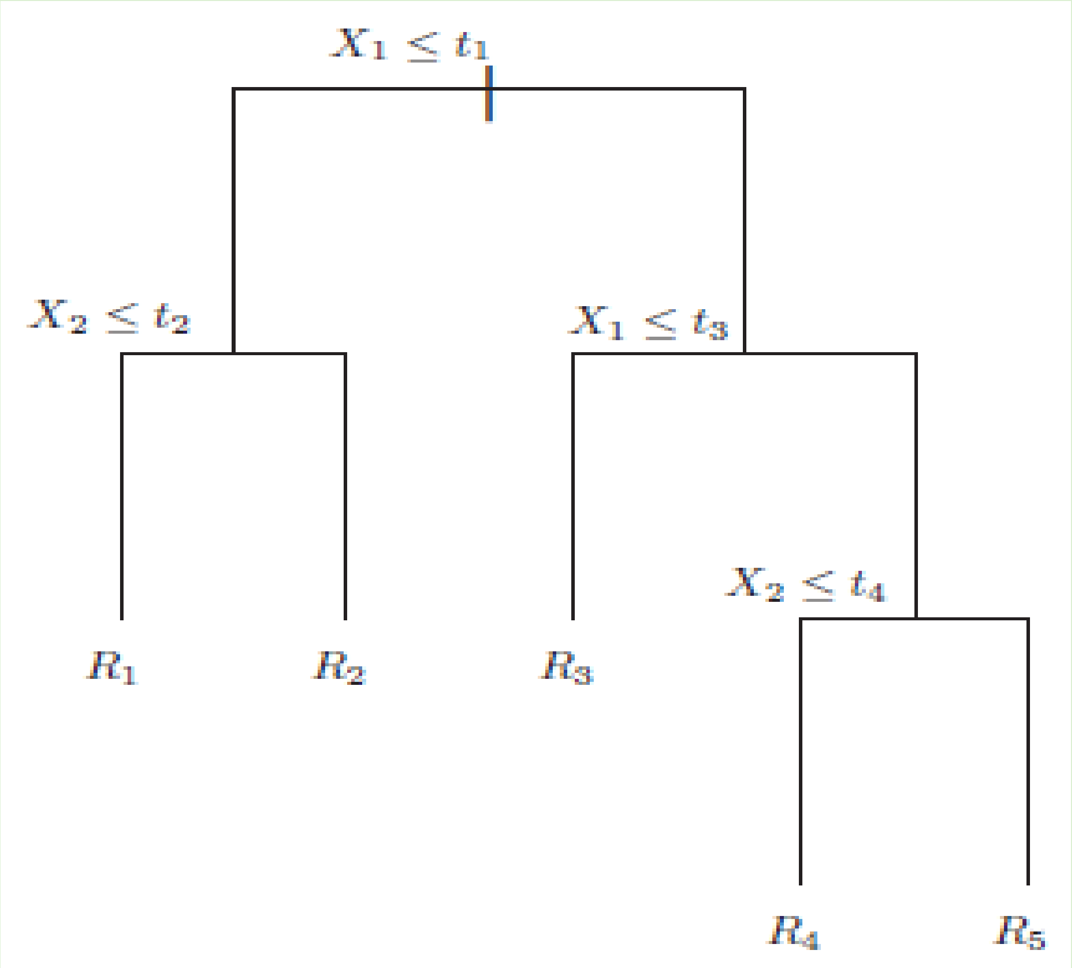
- Classifiers based on sequence of questions
- Easy to understand
- Often perform well (especially random forests which we'll discuss later)

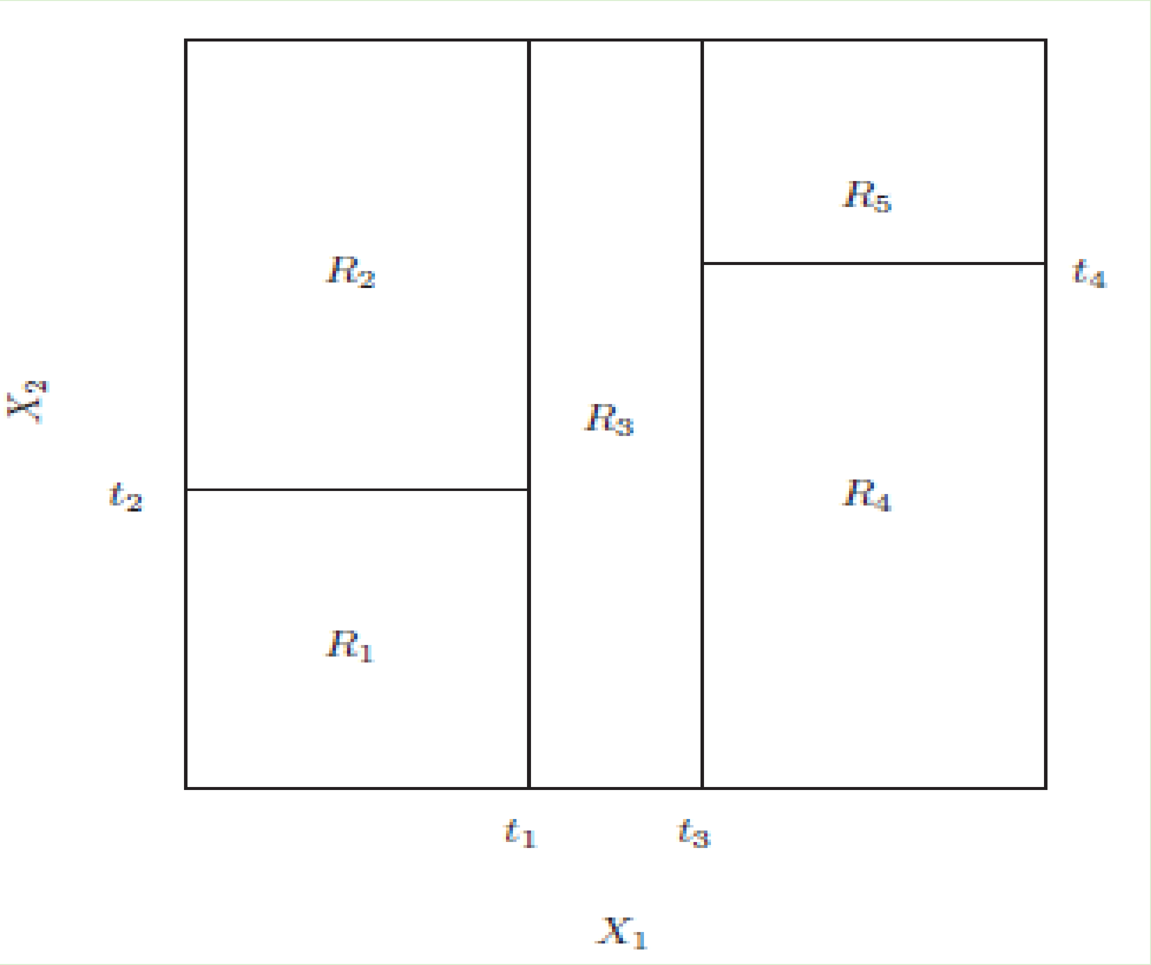


Decision Trees

- A decision tree partitions the features space to disjoint regions $R_1, \ldots, R_J$, as represented by the terminal nodes:

$$\forall_{i \neq j}\, R_i \cap R_j = \phi, \qquad \bigcup_i R_i = entire\ feature\ space$$





Introduction to Statistical Learning – G. James, D. Witten, T. Hastie, R. Tibshirani

Decision Trees

- Algorithm outline
  - Split the data by some criterion on one of the features
  - Split recursively on the right side
  - Split recursively on the left side

Decision Trees

- 3 decisions for building a DT model
  - Splitting criterion: which condition to put at each partition?
  - Which prediction to output at the leaves?
  - When to stop partitioning?

Today – Decision Trees

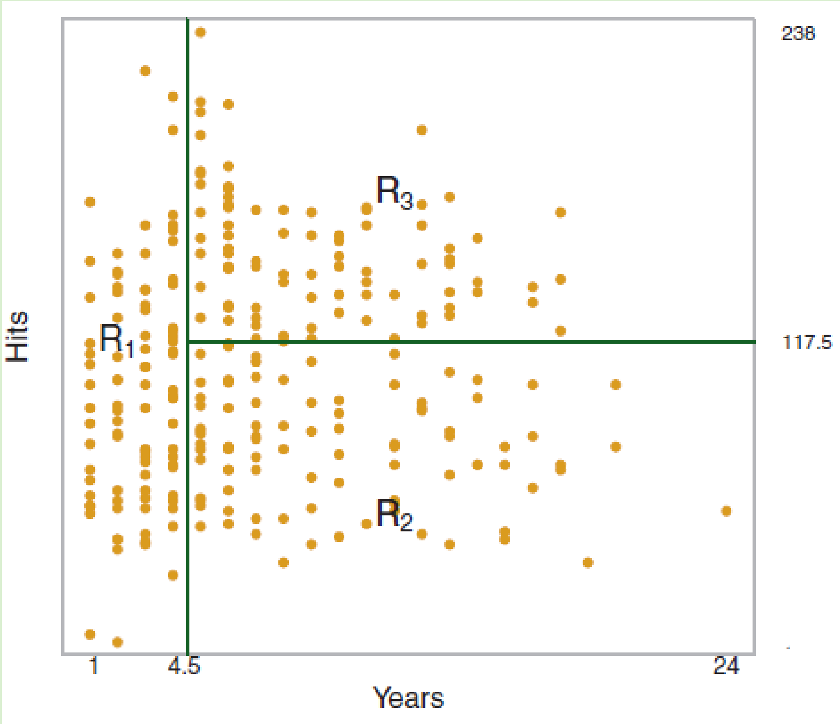
- Architecture
- Learning regression decision trees
- Learning classification decision trees
- Pruning

Decision Trees - regression

- 3 decisions for building a DT model
  - Splitting criterion: which condition to put at each partition?
  - Which prediction to output at the leaves?
    - The mean of the training samples in that leaf
  - When to stop partitioning?

Decision Trees - regression

- Example:
  - Predicting baseball (log) salaries, hitters dataset
  - Years feature – number of years playing
  - Hits – number of hits made last year



$$R_1 = \{X|\ Years < 4.5\}$$
$$R_2 = \{X|\ Years \geq 4.5, Hits < 117.5\}$$
$$R_3 = \{X|\ Years \geq 4.5, Hits \geq 117.5\}$$

Introduction to Statistical Learning – G. James, D. Witten, T. Hastie, R. Tibshirani

Decision Trees - tradeoff

- Minimizing E_{in}?
  - Minimizing the MSE is easy!
  - Just build a giant tree until each leaf contains a single sample – zero error!
- Bias-Variance
  - Large trees have high model complexity (can model almost everything)
  - Large trees suffer from large variance – a change in even one sample can lead to a totally different tree (model)

Decision Trees - tradeoff

- So lets constrain the DT to have J regions:

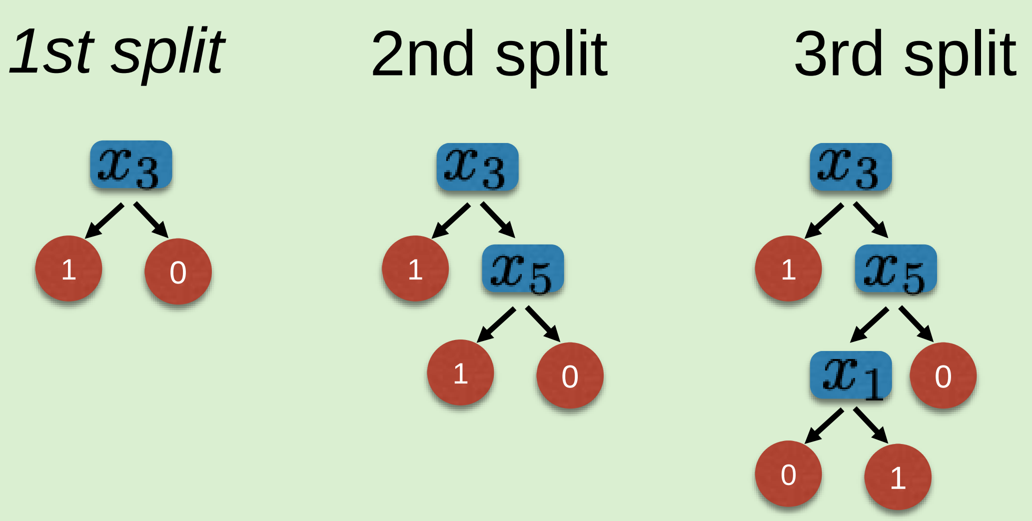
$$\underset{\{R_1,\ldots,R_J\}}{\operatorname{argmin}}\left(E_{in}\right) = \underset{\{R_1,\ldots,R_J\}}{\operatorname{argmin}} \sum_{j=1}^{J} \sum_{i \in R_j} \left(y_i - \hat{y}_{R_j}\right)^2$$

- Bad news: finding those J regions is *NP HARD*
- Good news: greedy algorithms work reasonably well

Greedy DT Training

- **Approach**: add one split at a time.

- For example:



- Once we decide on a split, we never change it!

Decision Trees - regression

- 3 decisions for building a DT model
  - Splitting criterion: which condition to put at each partition?
    - For the feature j and value s we divide the (sub) tree to:
      $$R_1(j,s) = \{X|X_j < s\} \quad \text{and} \quad R_2(j,s) = \{X|X_j \geq s\}$$
    - Find the feature j and value s that minimize
      $$\sum_{i:x_i \in R_1(j,s)} \left(y_i - \hat{y}_{R_1}\right)^2 + \sum_{i:x_i \in R_2(j,s)} \left(y_i - \hat{y}_{R_2}\right)^2$$

  Can be found efficiently!

  - Which prediction to output at the leaves?
    - The mean of the training samples in that leaf
  - When to stop partitioning?

Decision Trees - regression

- 3 decisions for building a DT model
  - Splitting criterion: which condition to put at each partition?
  - Which prediction to output at the leaves?
  - When to stop partitioning?
    - At (maximal) tree depth D
    - At (most) J regions
    - At l samples in region (or l% of the initial data size)
    - When the $MSE_{before\ split} - MSE_{after\ split} < threshold$
    - Regularization: minimizing $\sum_{j=1}^{J} \sum_{i \in R_j} \left(y_i - \hat{y}_{R_j}\right)^2 + \alpha \cdot J$

Cross validation

Prunes the tree (lasso)

Decision Trees - regression

- Hitters data set: overfitting with 10 regions

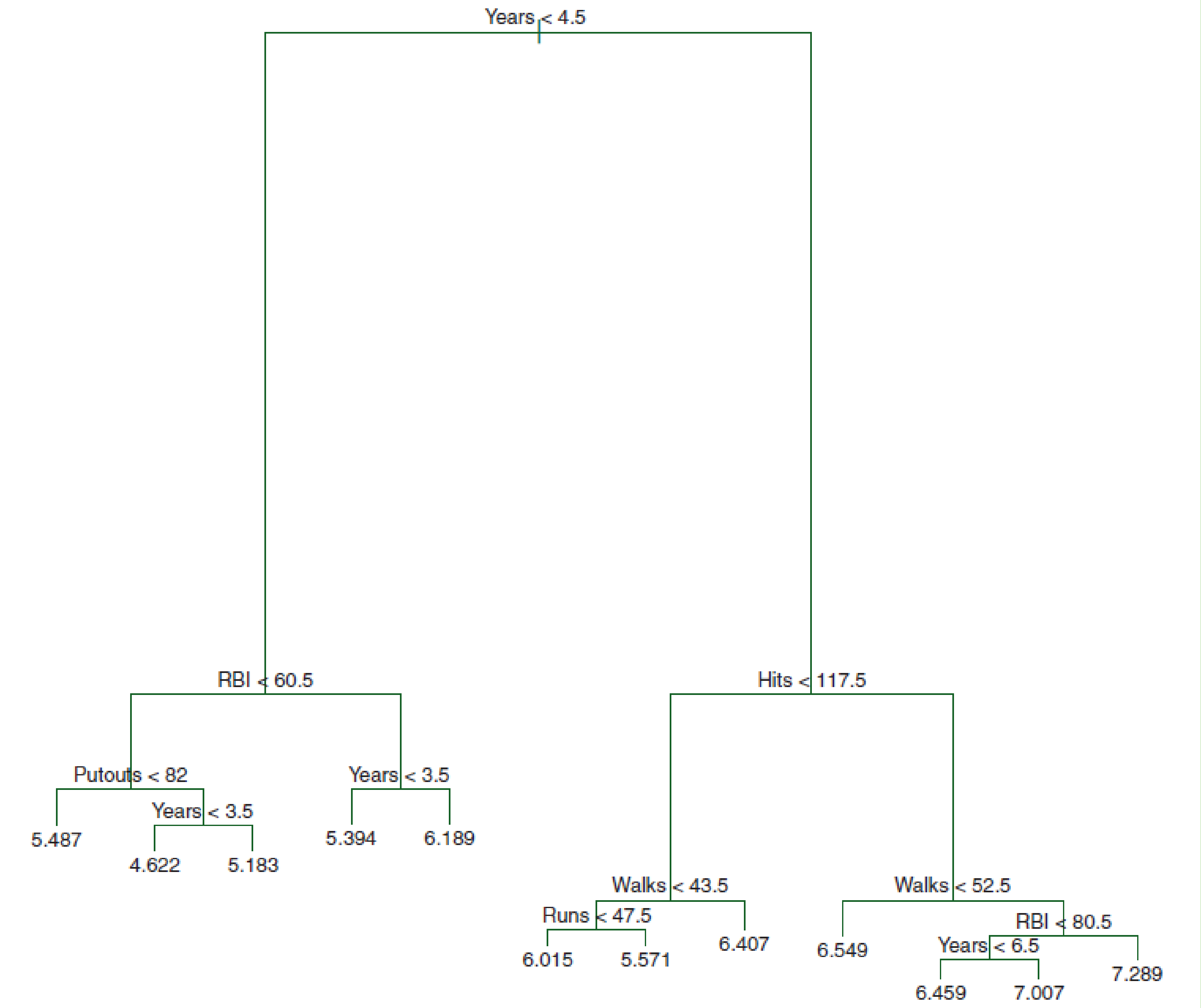


FIGURE 8.4. *Regression tree analysis for the* `Hitters` *data. The unpruned tree that results from top-down greedy splitting on the training data is shown.*

Decision Trees - regression

- Hitters data set: cross validating the tree size

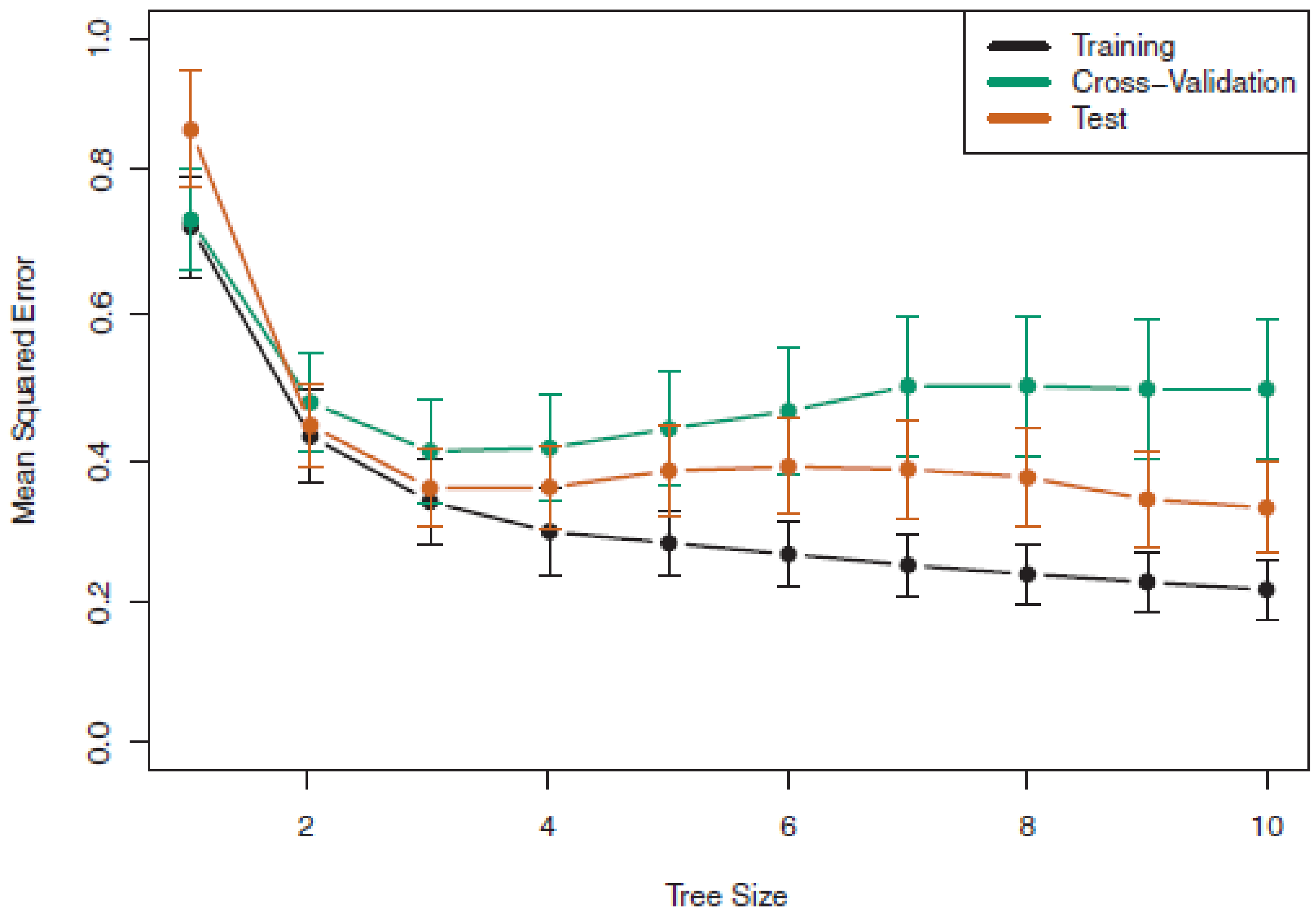


FIGURE 8.5. *Regression tree analysis for the* `Hitters` *data. The training, cross-validation, and test MSE are shown as a function of the number of terminal nodes in the pruned tree. Standard error bands are displayed. The minimum cross-validation error occurs at a tree size of three.*

Decision Trees – regression mean absolute error (MAE)

- Splitting criterion: which condition to put at each partition?
  - For the feature j and value s we divide the (sub) tree to:
    $$R_1(j,s) = \{\mathrm{X} | \mathrm{X}_j < s\} \quad \text{and} \quad R_2(j,s) = \{\mathrm{X} | \mathrm{X}_j \geq s\}$$
  - Find the feature j and value s that minimize
    $$\sum_{i:x_i \in R_1(j,s)} \left| y_i - \hat{y}_{R_1} \right| + \sum_{i:x_i \in R_2(j,s)} \left| y_i - \hat{y}_{R_2} \right|$$
- Which prediction to output at the leaves?
  - The median of the training samples in that leaf

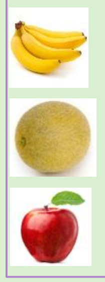
Can be computed as efficient as MSE. Advantages?

Today – Decision Trees

- Architecture
- Learning regression decision trees
- Learning classification decision trees
- Pruning

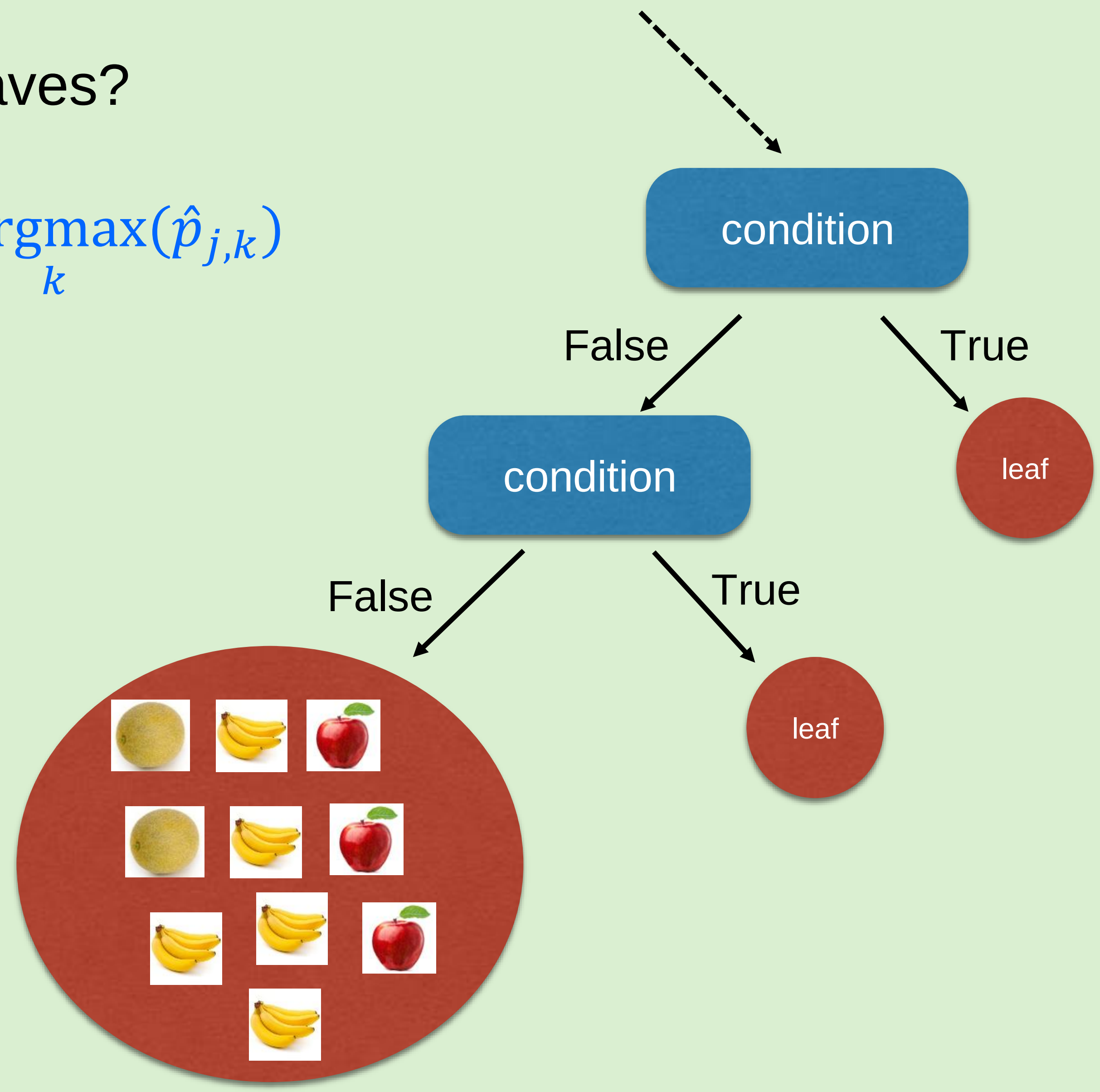
Decision Trees - classification

- 3 decisions for building a DT model
  - Splitting criterion: which condition to put at each partition?
  - Which prediction to output at the leaves?
    - The most common class k: $\underset{k}{\operatorname{argmax}}(\hat{p}_{j,k})$
  - When to stop partitioning?



$$\hat{p}_{j,0} = \frac{5}{10} = 0.5$$
$$\hat{p}_{j,1} = \frac{2}{10} = 0.2$$
$$\hat{p}_{j,2} = \frac{3}{10} = 0.3$$

$\cdots$



Decision Trees - classification

- 3 decisions for building a DT model
  - Splitting criterion: which condition to put at each partition?
    - For the feature j and value s we divide the (sub) tree to:
      $$R_1(j,s) = \{X | X_j < s\} \quad \text{and} \quad R_2(j,s) = \{X | X_j \geq s\}$$
    - Find the j and s that minimize the misclassification error
      $$\frac{N_{left}}{N}\left[1 - \max_k(\hat{p}_{1,k})\right] + \frac{N_{right}}{N}\left[1 - \max_k(\hat{p}_{2,k})\right]$$
  - Which prediction to output at the leaves?
    - The most common class k: $\underset{k}{\operatorname{argmax}}(\hat{p}_{j,k})$
  - When to stop partitioning?

Are we sure this is optimal?

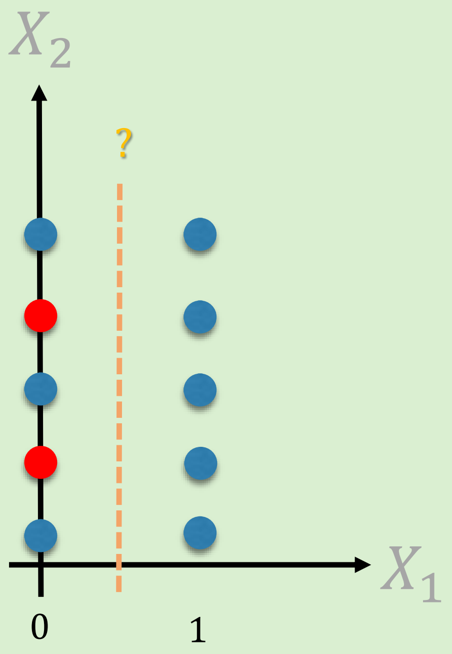
Limitation of misclassification – binary example

- Consider the following dataset s where:

$$P_s[Y=1]=0.8, \qquad P_s[Y=1|X_1=0]=0.6$$
$$P_s[X_1=1]=0.5, \qquad P_s[Y=1|X_1=1]=1$$

- Looks like splitting at $X_1 \leq \frac{1}{2}$ would be wonderful!

- Would we get it?



Limitation of misclassification – binary example

- Consider the following dataset s where:

$$P_s[Y=1]=0.8, \qquad P_s[Y=1|X_1=0]=0.6$$
$$P_s[X_1=1]=0.5, \qquad P_s[Y=1|X_1=1]=1$$

- $Pre-split-error$

$$1-\max_k(\hat{p}_k)=1-P_s[Y=1]=0.2$$

Need a different measure

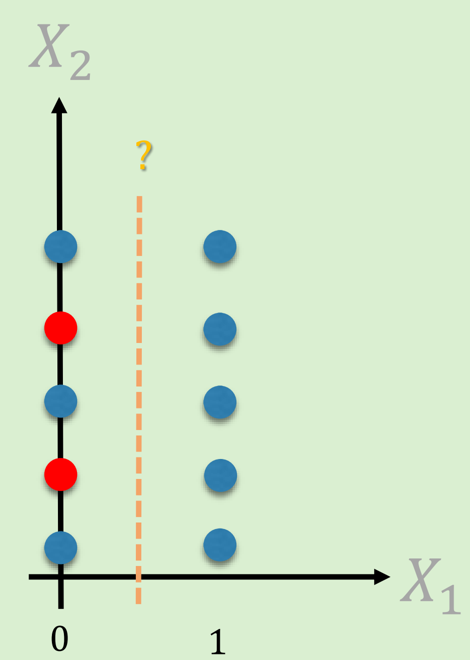
- $Post-split-error$

$$\frac{N_{left}}{N}\left(1-\max_k(\hat{p}_{left,k})\right)+\frac{N_{right}}{N}\left(1-\max_k(\hat{p}_{right,k})\right)=$$
$$\frac{5}{10}\cdot 0.4+\frac{5}{10}\cdot 0=0.2$$

- $Accuracy\ gain$

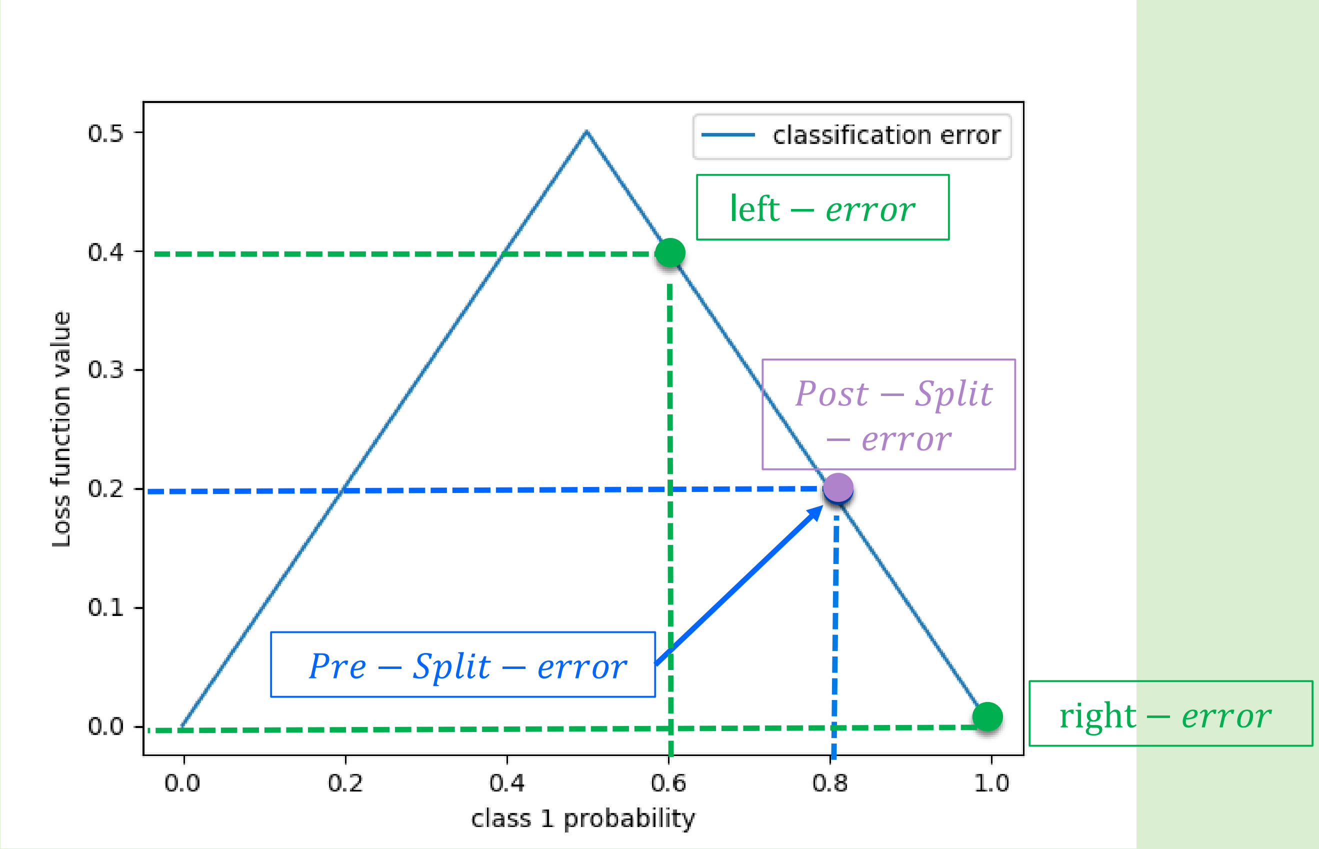
$$Pre-split-error-Post-split-error=$$
$$0.2-0.2=0$$

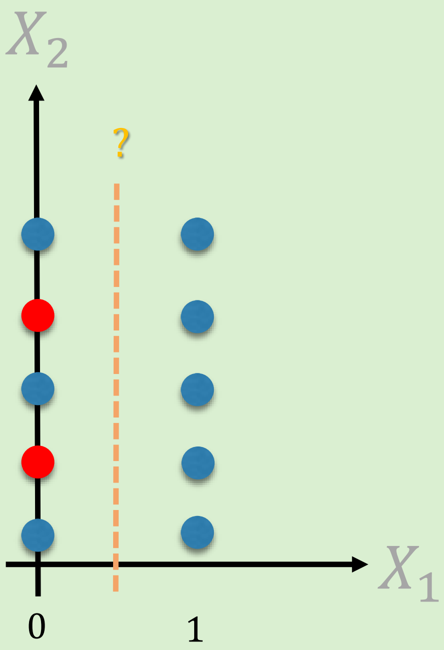
- Misclassification indicates we do not gain from this split



Limitation of misclassification – binary example

- Graphically – consider binary classification
- $P[Y=1] = 1 - P[Y=0] = \alpha$
- misclassification error loss function is a function of α
- $L_{misclassification\_error}(\alpha) = 1 - \max(\alpha, 1-\alpha)$





Note: the post split error is in the exact middle because half of the sample went to each side

Decision Trees - classification

- 3 decisions for building a DT model
  - Splitting criterion: which condition to put at each partition?
    - For the feature j and value s we divide the (sub) tree to:
      $$R_1(j,s) = \{X|X_j < s\} \quad \text{and} \quad R_2(j,s) = \{X|X_j \geq s\}$$
    - Find the j and s that minimize the ~~misclassification error~~
      $$\cancel{\frac{N_{left}}{N}\left[1 - \max_k(\hat{p}_{1,k})\right] + \frac{N_{right}}{N}\left[1 - \max_k(\hat{p}_{2,k})\right]}$$

We will now go over two objective alternatives to misclassification error:

- Entropy function
- Gini index

The entropy function

- Consider a random variable X with K values
- Denote its distribution function by $P[X = i] = p_i$
- Define the ***entropy*** of $\mathbf{p}$:

$$H[X] = \sum_{i=1}^{K} p_i \log \frac{1}{p_i} = -\sum_{i=1}^{K} p_i \log p_i$$

- Some properties:
  - Non negative
  - Is zero if and only if p is **deterministic**.
  - Maximized by uniform $\mathbf{p}$. Max value: $\log K$

Assume log base 2

The entropy function

- Entropy measures the uncertainty about the value of X
- Higher entropy -> More uncertainty
- Least uncertainty for deterministic (entropy=0)
- Most uncertainty for uniform (entropy=$\log K$)
- The fundamental measure in Shannon's information theory (used for compression, channel coding etc.)

The entropy function – binary example

- Consider the following dataset s where:

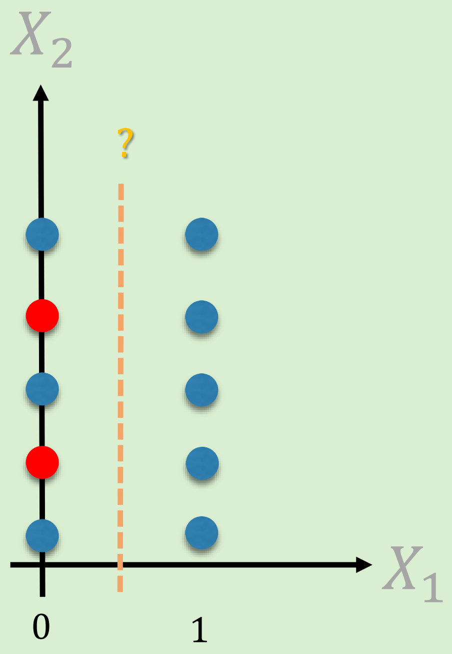
$$P_s[Y=1]=0.8, \qquad P_s[Y=1|X_1=0]=0.6$$
$$P_s[X_1=1]=0.5, \qquad P_s[Y=1|X_1=1]=1$$

- $Pre-split-entropy$

$$-\sum_k p_k \log p_k = -(p_1 \log p_1 + (1-p_1)\log(1-p_1)) =$$
$$-(0.8 \log 0.8 + 0.2 \log 0.2) = 0.72$$

- $Post-split-entropy$

$$\frac{N_{left}}{N}\left(-\sum_k p_{left,k} \log p_{left,k}\right) + \frac{N_{right}}{N}\left(-\sum_k p_{right,k} \log p_{right,k}\right) =$$
$$\frac{5}{10} \cdot 0.97 + \frac{5}{10} \cdot 0 = 0.485$$



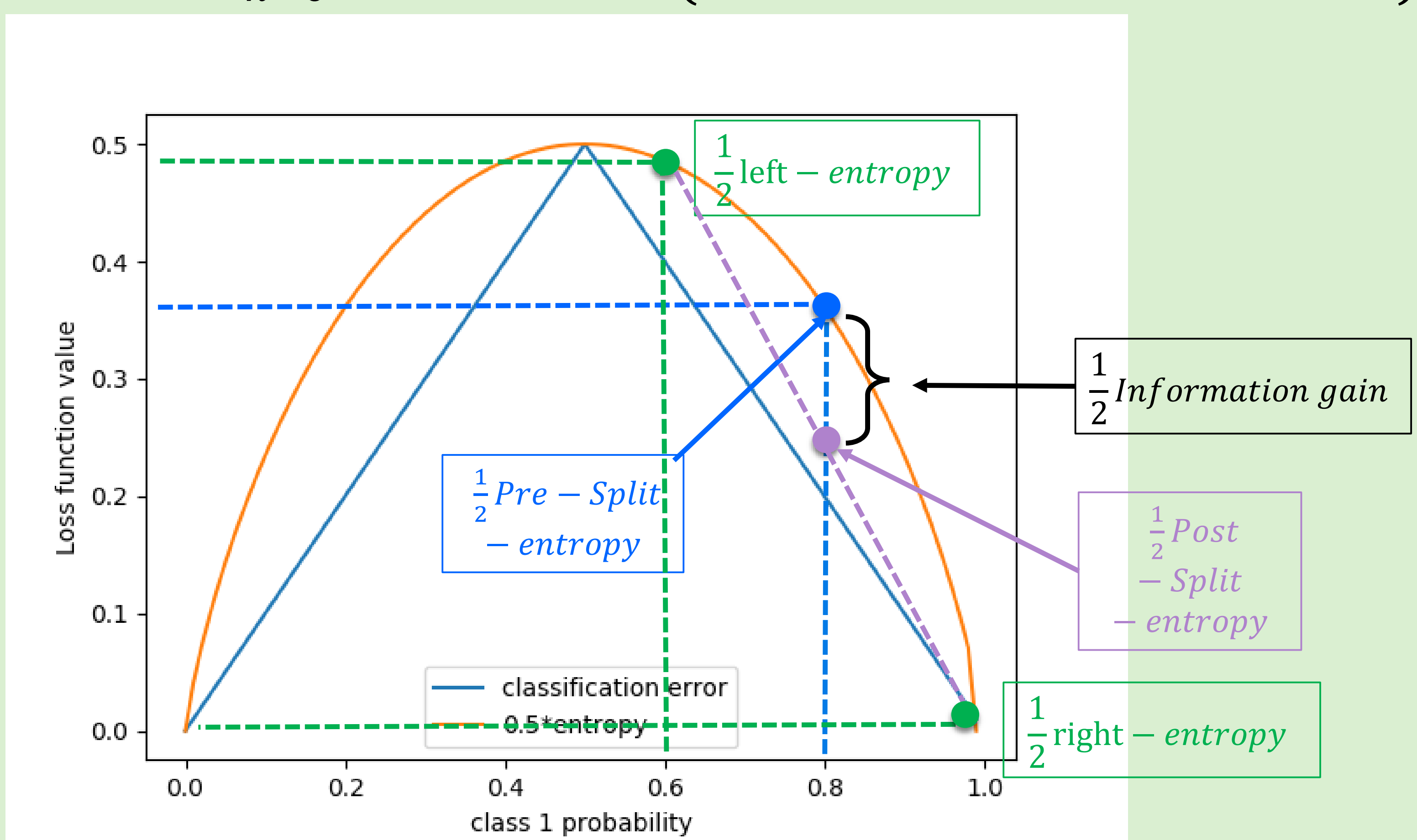
- $Information\ gain$:

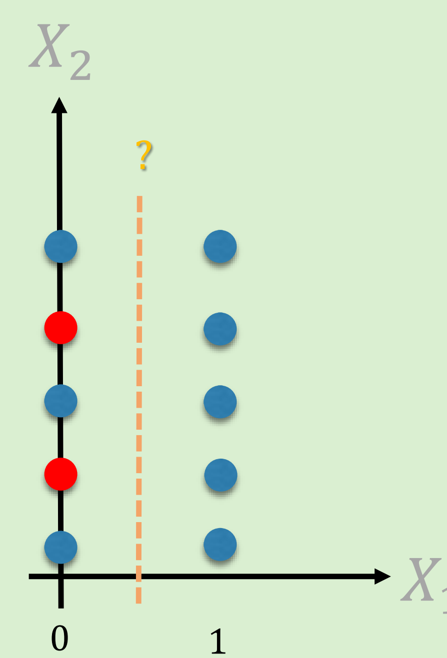
$$Pre-split-error - Post-split-error =$$
$$0.72 - 0.485 = 0.235$$

- Entropy loss indicates we **DO** gain from this split

The entropy function – binary example

- Graphically – consider binary classification
- $P[Y = 1] = 1 - P[Y = 0] = \alpha$
- The entropy loss function is a function of α
- $L_{entropy}(\alpha) = -\sum_{k=0}^{1} p_k \log p_k = -\big((1-\alpha)\log(1-\alpha) + \alpha \log \alpha\big)$





The Gini index

- A different measurement commonly used, with similar behavior as Entropy

- The ***Gini Index*** is defined by

$$G = \sum_{k=1}^{K} p_{j,k}\,(1 - p_{j,k})$$

- Measures the total variance across K classes, and refers to the nodes purity

- Smallest where all probabilities are close to 1 or 0

The Gini index– binary example

- Consider the following dataset s where:

$$P_s[Y=1]=0.8, \qquad P_s[Y=1|X_1=0]=0.6$$
$$P_s[X_1=1]=0.5, \qquad P_s[Y=1|X_1=1]=1$$

- $Pre-split-Gini\ index$

$$\sum_k p_k(1-p_k) = (p_1(1-p_1)+(1-p_1)p_1) = 2p_1(1-p_1) =$$
$$2\cdot 0.8\cdot(1-0.8) = 0.32$$

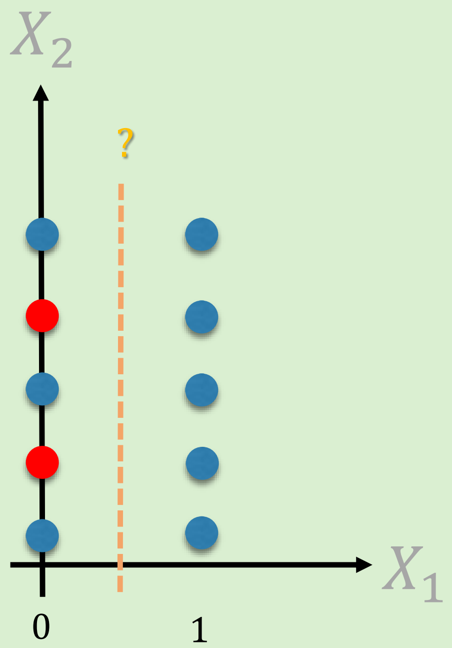
- $Post-split-Gini\ index$

$$\frac{N_{left}}{N}\left(\sum_k p_{left,k}(1-p_{left,k})\right)+\frac{N_{right}}{N}\left(\sum_k p_{right,k}(1-p_{right,k})\right)=$$
$$\frac{5}{10}\cdot 2\cdot 0.6\cdot 0.4+\frac{5}{10}\cdot 2\cdot 1\cdot 0 = 0.24$$

- $Information\ gain$

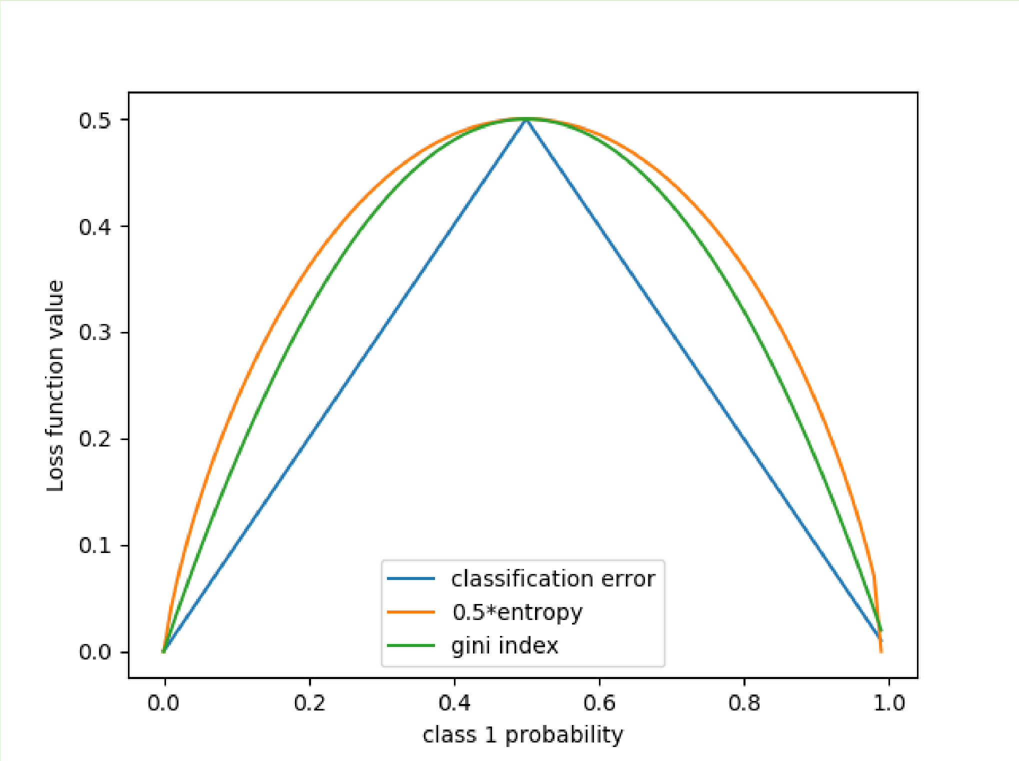
$$Pre-split-error - Post-split-error =$$
$$0.32-0.24=0.08$$

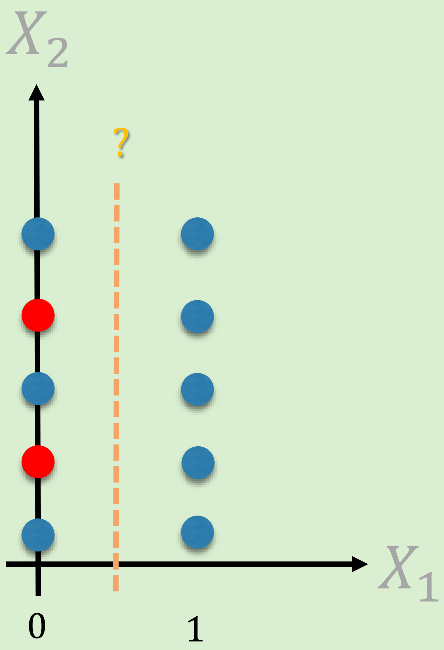
- Gini index indicates we **DO** gain from this split



The Gini index – binary example

- Graphically – consider binary classification
- $P[Y=1] = 1 - P[Y=0] = \alpha$
- The Gini index is a function of α
- $L_{Gini}(\alpha) = \sum_{k=0}^{1} p_{j,k}\,(1 - p_{j,k}) = (1-\alpha)\alpha + \alpha(1-\alpha) = 2\alpha(1-\alpha)$
- We have a similar concave shape which would result with the calculated gain





Measurement examples

p1	p2	Entropy	Gini	misclassification
1	0	0	0	0
0.7	0.3	0.88	0.42	0.3
0.5	0.5	1	0.5	0.5

p1	p2	p3	Entropy	Gini	misclassification
0.6	0.2	0.2	1.37	0.56	0.4
0.5	0.49	0.01	1.01	0.51	0.5

Decision Trees - classification

- 3 decisions for building a DT model
  - Splitting criterion: which condition to put at each partition?
    - For the feature j and value s we divide the (sub) tree to:
      $$R_1(j,s) = \left\{\mathrm{X} | \mathrm{X}_j < s\right\} \quad \text{and} \quad R_2(j,s) = \left\{\mathrm{X} | \mathrm{X}_j \geq s\right\}$$
    - Find the j and s that maximize the gain (minimize the impurity)
      $$Gain(N) = i(N) - \left[\frac{N_{right}}{N} i\left(N_{right}\right) + \frac{N_{right}}{N} i\left(N_{right}\right)\right]$$
  - Which prediction to output at the leaves?
    - The most common class k: $\underset{k}{\operatorname{argmax}}(\hat{p}_{j,k})$
  - When to stop partitioning?

Decision Trees - classification

- 3 decisions for building a DT model
  - Splitting criterion: which condition to put at each partition?
    - For the feature j and value s we divide the (sub) tree to:
      $R_1(j,s) = \{X|X_j < s\}$ and $R_2(j,s) = \{X|X_j \geq s\}$
    - Find the j and s that maximize the gain (minimize the impurity)
      $$\Delta i(N) = i(N) - \left[\frac{N_{left}}{N} i(N_{left}) + \frac{N_{right}}{N} i(N_{right})\right]$$
  - Which prediction to output at the leaves?
    - The most common class k: $\underset{k}{\operatorname{argmax}}(\hat{p}_{j,k})$
  - When to stop partitioning?
    - Same as regression

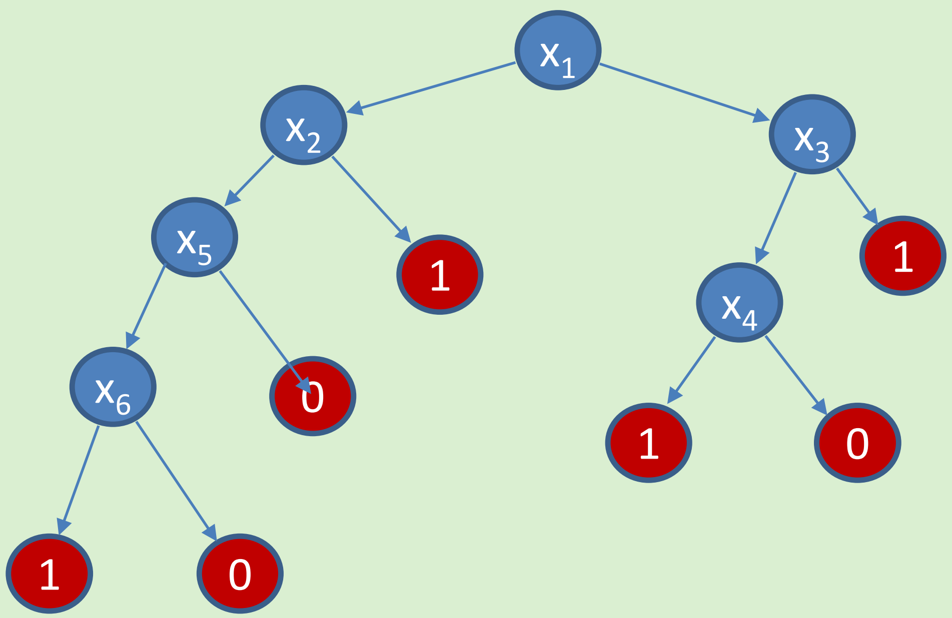
Maybe, instead of stopping, we can prune?

Today – Decision Trees

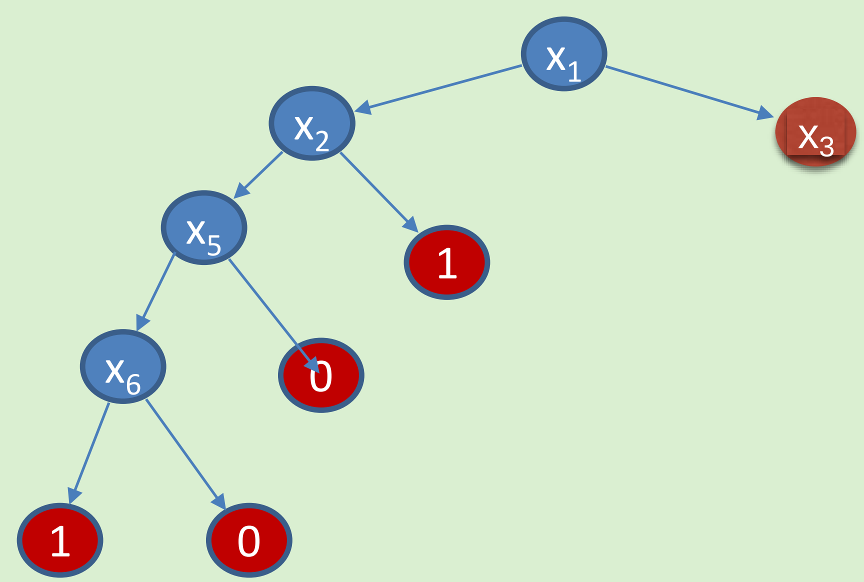
- Architecture
- Learning regression decision trees
- Learning classification decision trees
- Pruning

Pruning Decisions

- The pruning process takes a large tree, and considers cutting sub-trees



Pruned Tree



- We should prune if pruned tree has better true error than original true

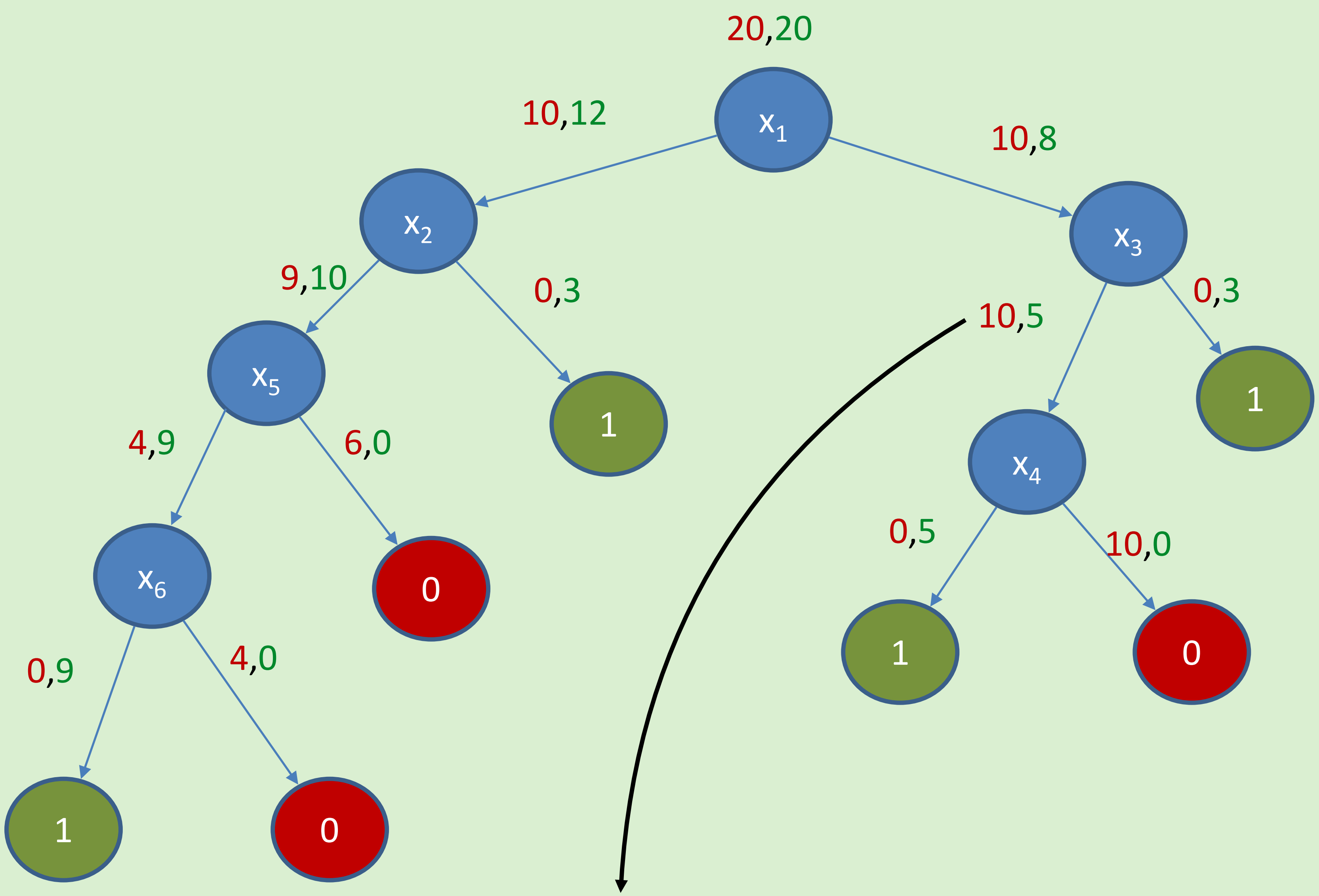
Estimating True Error

- How can we estimate the true error of a DT:
  - Using generalization bounds
  - Using holdout set. Train on S_1, evaluate on S_2

Example - Training

Result of training on S_1 of size 40

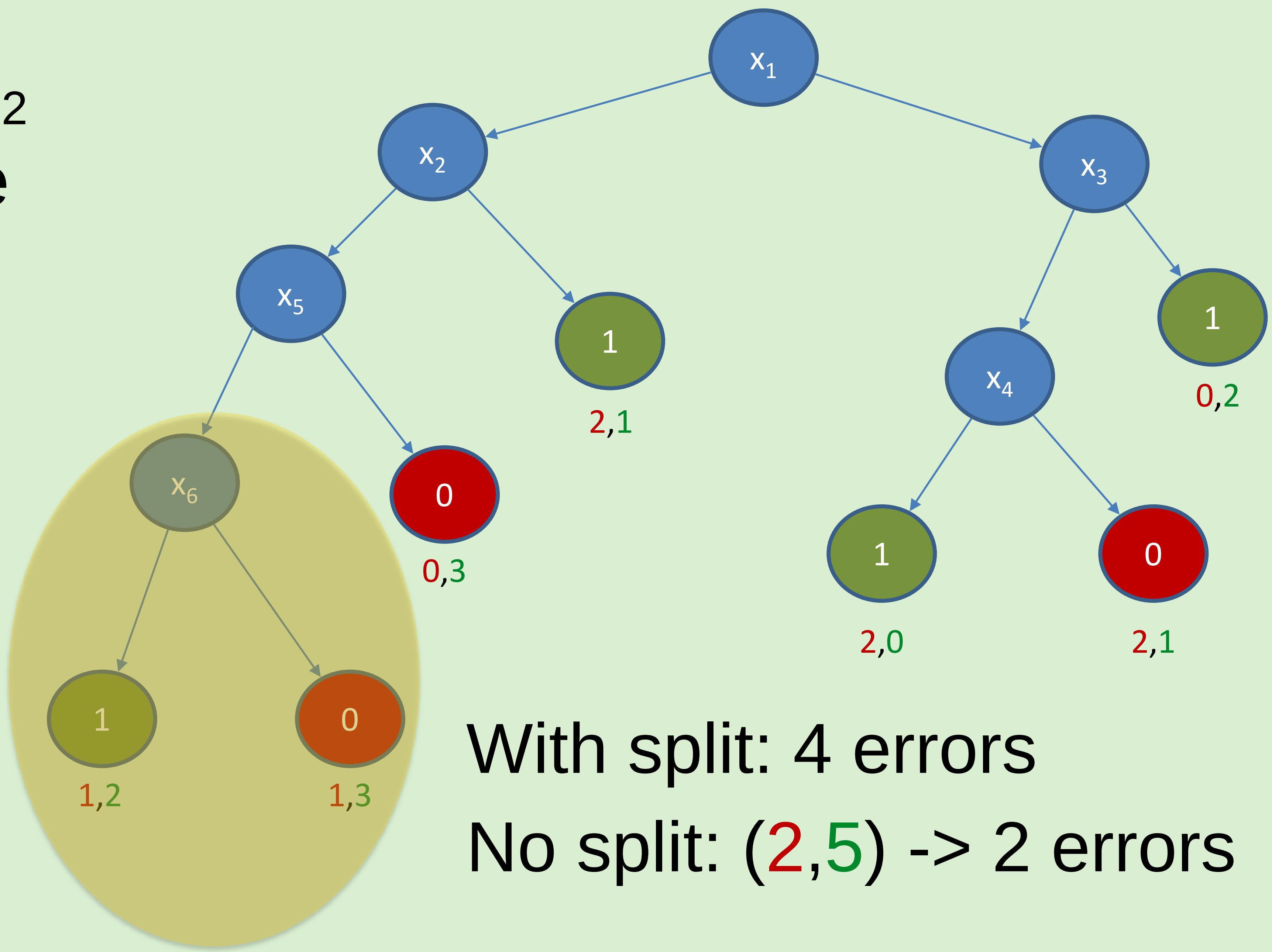
- 20 of label 0
- 20 of label 1



The data at this point has 5 Y=1 and 10 Y=0

Example Holdout

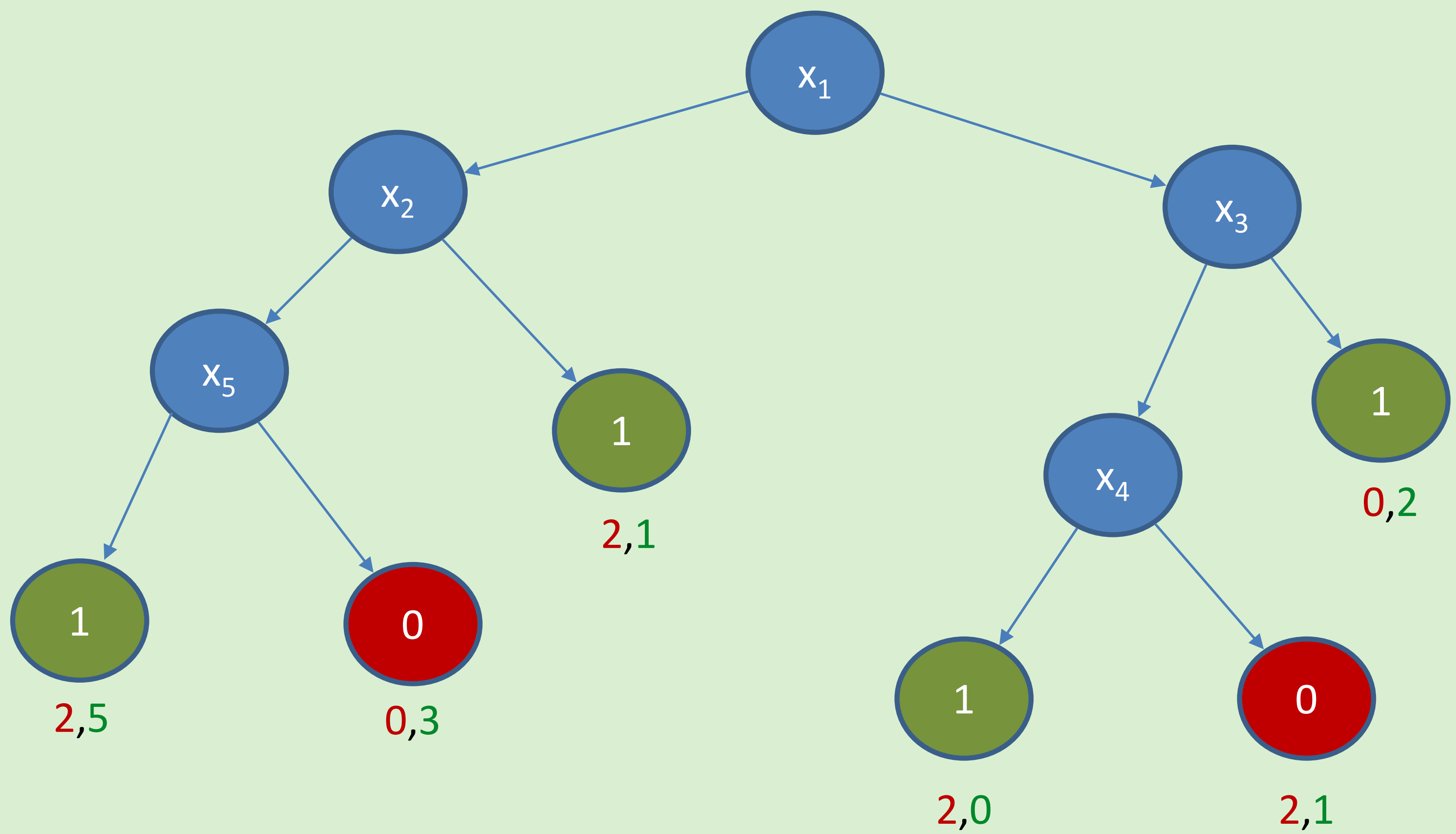
Evaluate S_2 on this tree



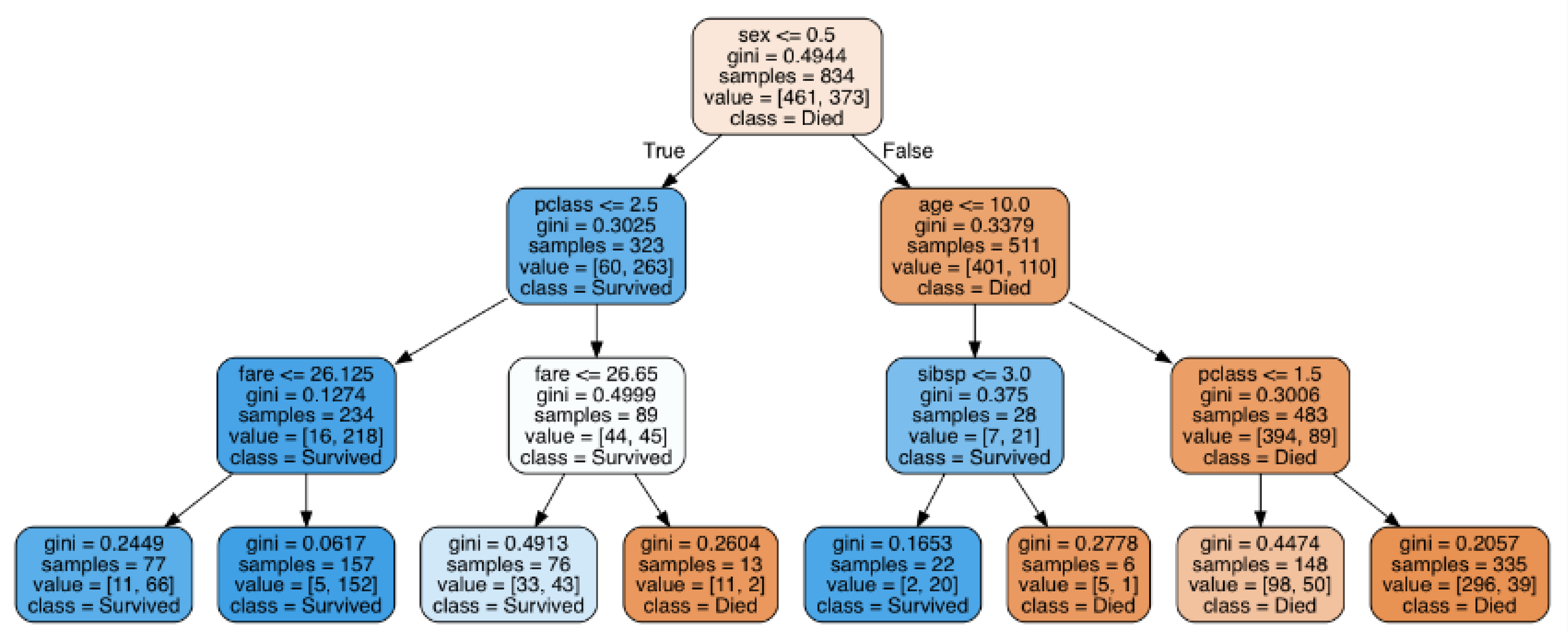
With split: 4 errors

No split: (2,5) -> 2 errors

After Pruning



DT plot with graphiz



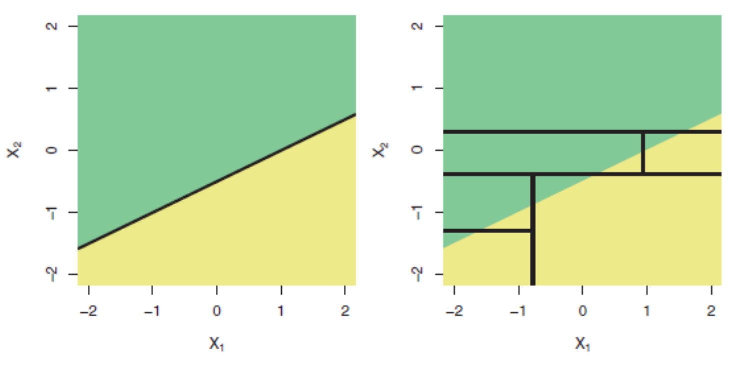
Titanic Decision Tree learned by Scikit

https://redislabs.com/blog/introduction-redis-ml-part-five/

Decision Trees - Summary

- Pros
  - Simple
  - Intuitive
  - “natural” Nan dealing
  - No need for dummy variables

- Cons
  - Bad model
  - High variance – small change in data might change the entire tree
  - Can’t deal with simple model such as linear



Introduction to Statistical Learning – G. James, D. Witten, T. Hastie, R. Tibshirani

Next week

- Though a single decision tree is not very useful, we shall use it as a building block for state-of-the-art models!

http://blog.yhat.com/posts/random-forests-in-python.html

Questions?