NYU Center for Data Science

DS-GA 3001.007
Introduction to Machine Learning

Lecture 14
Combining Models I – Trees and Random Forests

Ensemble methods take a set of models trained in series or in parallel to combine predictions

NYU Center for Data Science

DS-GA 3001.007
Introduction to Machine Learning

Lecture 14
Combining Models I – Trees and Random Forests

Ensemble methods take a set of models trained in series or in parallel to combine predictions

NYU Center for Data Science

DS-GA 3001.007
Introduction to Machine Learning

Different from approach in kernels that focused on extending models through feature selection

Lecture 14
Combining Models I – Trees and Random Forests

Announcements

- Project
  - Feedback on Project Milestone is available
  - Submit Project Report by Sunday December 15 at 11:59PM
- Homework
  - Submit by Thursday December 12 at 11:59PM
- Final Exam
  - The exam is scheduled for Wednesday December 18 between 12-1:50PM

Announcements

- Project
  - Feedback on Project Milestone is available
  - Submit Project Report by Sunday December 15 at 11:59PM
- Homework
  - Submit by Thursday December 12 at 11:59PM
- Final Exam
  - The exam is scheduled for Wednesday December 18 between 12-1:50PM

Responsible for poster along with notebook. See project guidelines for more information.

Announcements

- Project
  - Feedback on Project Milestone is available
  - Submit Project Report by Sunday December 15 at 11:59PM
- Homework
  - Submit by Thursday December 12 at 11:59PM
- Final Exam
  - The exam is scheduled for Wednesday December 18 between 12-1:50PM

We will review next week. See Week 9 – Week 14 agendas for more practice questions.

Review

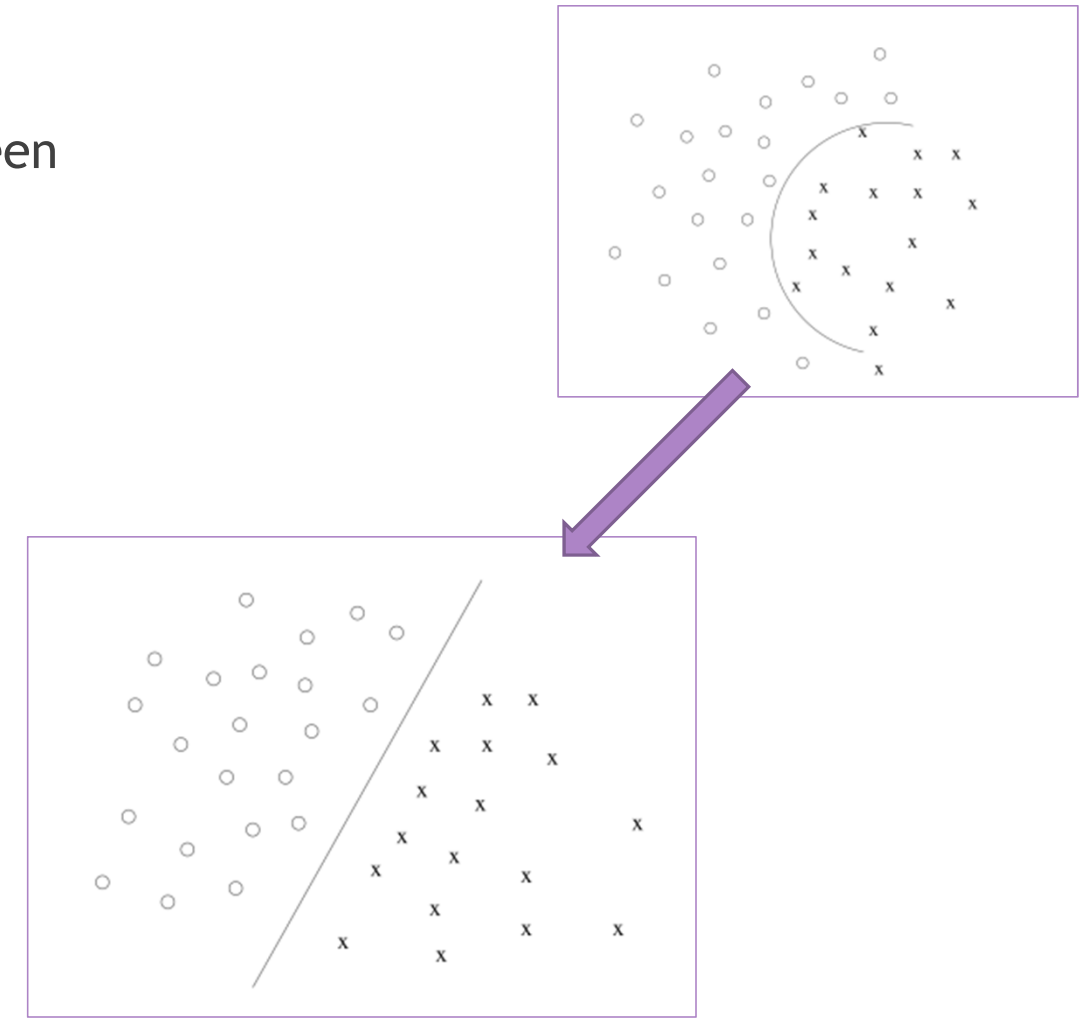
- Extending Linear Models
  - Feature Engineering
    - Polynomial Transformation for Numerical Data
    - One-Hot Encoding for Categorical Data
  - Feature Selection
    - Forward Feature Selection
    - Backward Features Selection
    - Regularization

Use 0 or 1 like False and True to indicate categories.

Helps reduce…
- Estimation Error
- Approximation Error
- Optimization Error

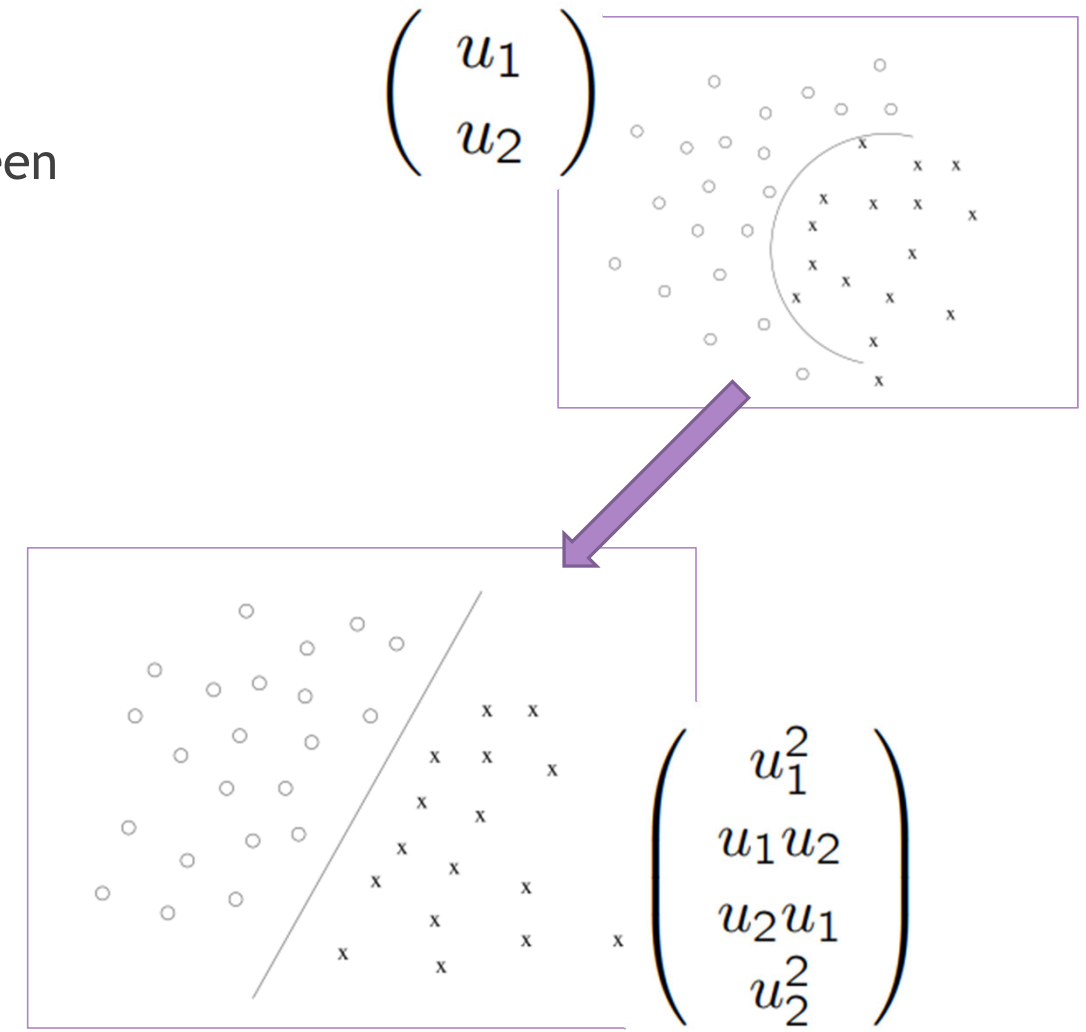
Review

- Kernels – Relationships between Features
  - Examples
    - Linear Kernel
    - Polynomial Kernel
    - Gaussian Kernel



Review

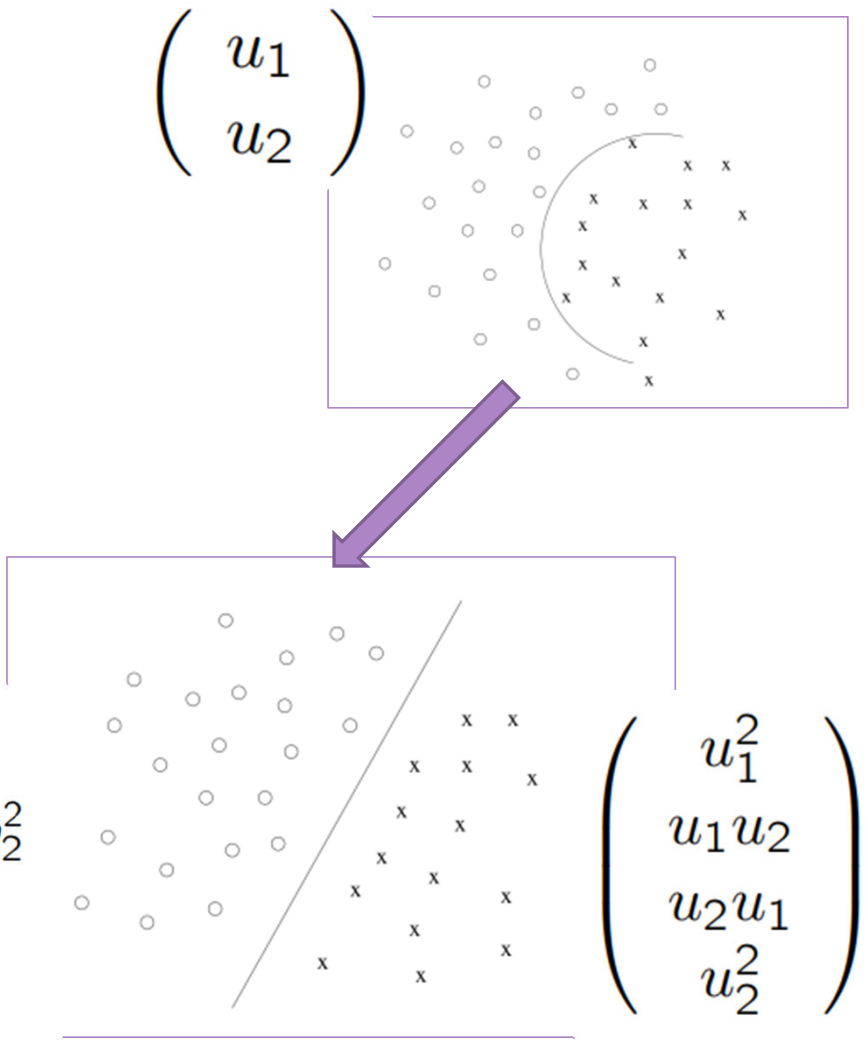
- Kernels – Relationships between Features
  - Examples
    - Linear Kernel
    - Polynomial Kernel
    - Gaussian Kernel



Review

- Kernels – Relationships between Features
  - Examples
    - Linear Kernel
    - Polynomial Kernel
    - Gaussian Kernel

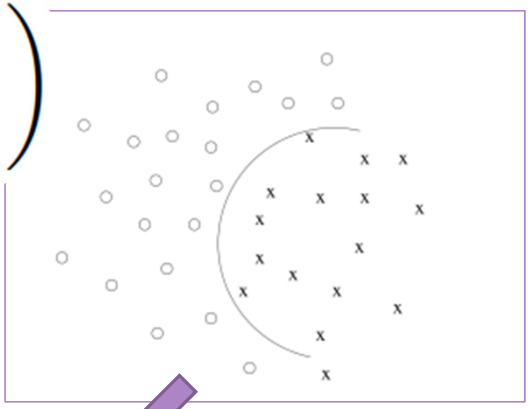
$$\begin{pmatrix} u_1^2 \\ u_1u_2 \\ u_2u_1 \\ u_2^2 \end{pmatrix} \cdot \begin{pmatrix} v_1^2 \\ v_1v_2 \\ v_2v_1 \\ v_2^2 \end{pmatrix} = u_1^2v_1^2 + 2u_1v_1u_2v_2 + u_2^2v_2^2 = (u_1v_1 + u_2v_2)^2 = (u.v)^2$$



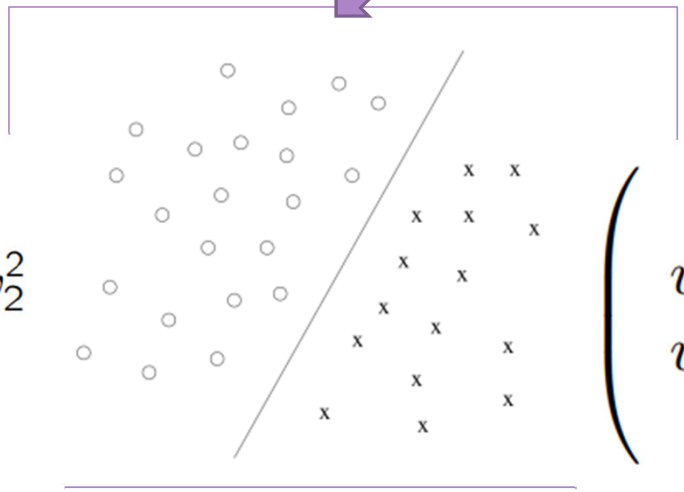
Review

- Kernels – Relationships between Features
  - Examples
    - Linear Kernel
    - Polynomial Kernel
    - Gaussian Kernel

$$\begin{pmatrix} u_1 \\ u_2 \end{pmatrix}$$



Kernel Trick



$$\begin{pmatrix} u_1^2 \\ u_1 u_2 \\ u_2 u_1 \\ u_2^2 \end{pmatrix} \cdot \begin{pmatrix} v_1^2 \\ v_1 v_2 \\ v_2 v_1 \\ v_2^2 \end{pmatrix} = u_1^2 v_1^2 + 2u_1 v_1 u_2 v_2 + u_2^2 v_2^2 = (u_1 v_1 + u_2 v_2)^2 = (u.v)^2$$

$$\begin{pmatrix} u_1^2 \\ u_1 u_2 \\ u_2 u_1 \\ u_2^2 \end{pmatrix}$$

Review

- Kernels: Relationships between Features
  - Determining Kernels
    - Sum, Product, Scale by Positive Number

Question

If we have a function

$k: X \times X \longrightarrow R$, then can we find

features $f: X \longrightarrow R^d$ such that

$k(x,x) = f(x) \cdot f(x)$?

Review

- Kernels: Relationships between Features
  - Determining Kernels
    - Sum, Product, Scale by Positive Number

Question

If we have a function

k: X x X $\longrightarrow$ R, then can we find

features f: X $\longrightarrow$ R^d such that

k(x,x) = f(x) . f(x) ?

Answer

Definition: $k : \mathcal{X}^2 \to \mathbb{R}$ is a **kernel** if it is

1. Symmetric: $k(u, v) = k(v, u)$, and
2. Positive semidefinite: every Gram matrix $K_{ij} = k(x_i, x_j)$ is positive semidefinite.

Review

- Kernels: Relationships between Features
  - Determining Kernels
    - Sum, Product, Scale by Positive Number
    - Properties of Gram matrix

Gram Matrix

$$K = (\langle x_i, x_j \rangle)_{i,j} = \begin{pmatrix} \langle x_1, x_1 \rangle & \cdots & \langle x_1, x_n \rangle \\ \vdots & \ddots & \cdots \\ \langle x_n, x_1 \rangle & \cdots & \langle x_n, x_n \rangle \end{pmatrix}$$

Answer

Definition: $k : \mathcal{X}^2 \to \mathbb{R}$ is a **kernel** if it is

1. Symmetric: $k(u, v) = k(v, u)$, and
2. Positive semidefinite: every Gram matrix $K_{ij} = k(x_i, x_j)$ is positive semidefinite.

Review

- Kernels – Relationships between Features
  - Models Admitting Kernels
    - Ridge Regression
    - Support Vector Machine
    - Representer Theorem
      - Projecting a vector onto a subspace reduces size
      - Linear combinations of the points in the training set minimize the empirical risk function

Nondecreasing function of the dot product

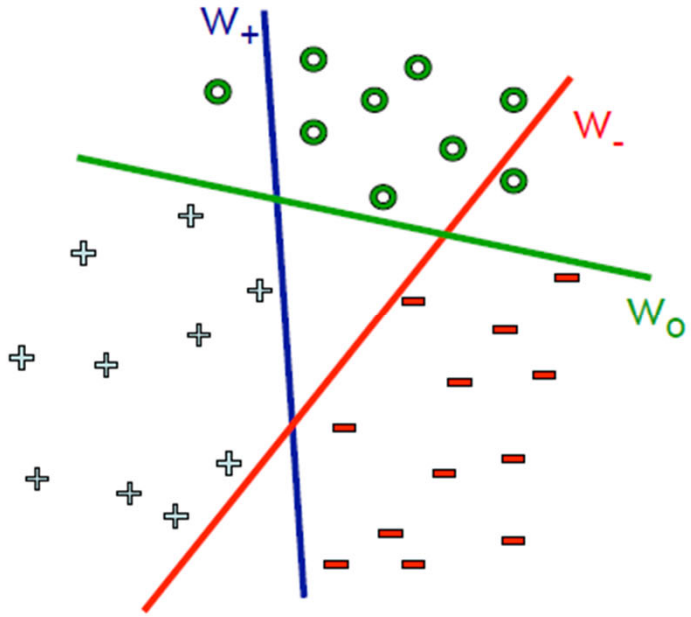
$$J(w) = \|Xw - y\|_1 + \lambda\|w\|_2^2.$$

Function of the dot product between points in training set and weights

Review

- Multiple Categories
  - One-vs-All

Default for scikit-learn. Used in Lab 11 and Homework 5



Learn 3 classifiers:

- \- vs {o,+}, weights w_-
- \+ vs {o,-}, weights w_+
- o vs {+,-}, weights w_o

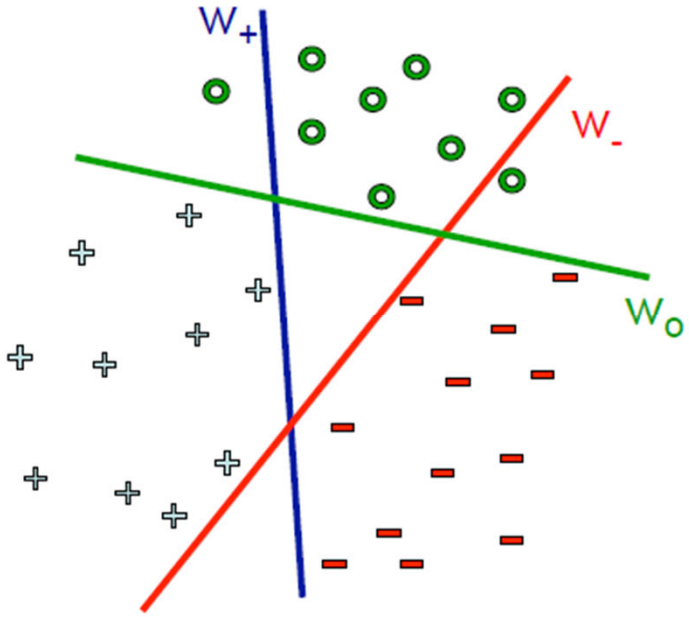
Predict label using:

$$\hat{y} \leftarrow \arg\max_k \; w_k \cdot x + b_k$$

Review

- Multiple Categories
  - One-vs-All
  - One-vs-One

Sometimes called One-vs-Rest and All-vs-All.



Learn 3 Classifiers

- - vs + weights w_{+-}
- - vs o weights w_{o-}
- + vs o weights w_{o+}

Predict Label using

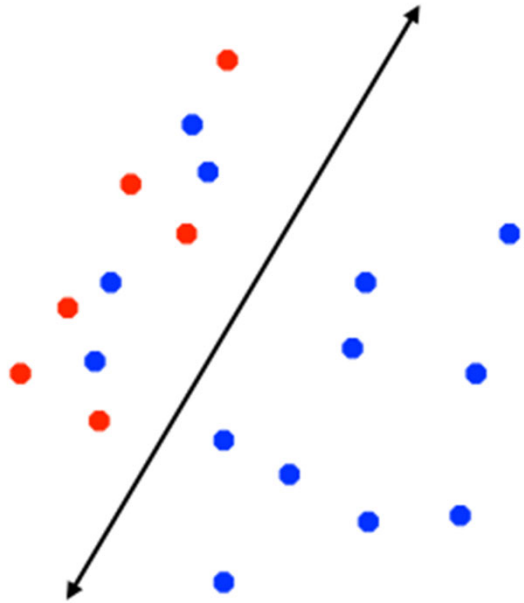
$\text{argmax}_i \ \Sigma_j w_{ij} x + b_{ij}$

Note that $w_{ij} x + b_{ij} = -w_{ji} x - b_{ji}$

Review

- Multiple Categories
  - One-vs-All
  - Ove-vs-One
- Issues
  - Large datasets
  - Imbalanced Data

$$N = N_+ + N_-$$

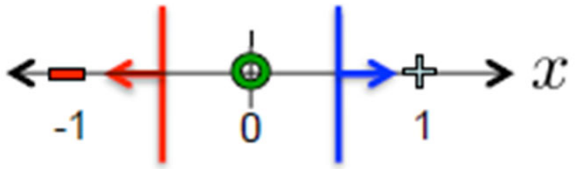


$$\min_{w,b} \quad ||w||_2^2 + \frac{CN}{2N_+} \sum_{j:y_j=+1} \xi_j + \frac{CN}{2N_-} \sum_{j:y_j=-1} \xi_j$$

Class-specific weighting of the slack variables

Review

- Multiple Categories
  - One-vs-All
  - Ove-vs-One
- Issues
  - Large datasets
  - Imbalanced Data
  - Different Scales
  - Not Separable



$$w_- = -1 \qquad w_+ = 1$$
$$b_- = -.5 \qquad b_+ = -.5$$
$$w_o = 0$$
$$b_o = .001$$

$$\text{minimize}_{\mathbf{w},b} \quad \sum_y \mathbf{w}^{(y)}.\mathbf{w}^{(y)} + C \sum_j \xi_j$$
$$\mathbf{w}^{(y_j)}.\mathbf{x}_j + b^{(y_j)} \geq \mathbf{w}^{(y')}.\mathbf{x}_j + b^{(y')} + 1 - \xi_j, \quad \forall y' \neq y_j, \quad \forall j$$
$$\xi_j \geq 0, \quad \forall j$$

To predict, we use:

$$\hat{y} \leftarrow \arg\max_k \; w_k \cdot x + b_k$$

Agenda

- Lesson
  - Trees
    - Dividing Data by Categorical and Numerical Features
    - Determining Partitions of the Data
  - Random Forests
    - Preventing Overfitting
    - Using Bootstrap with Random Forests
- Demo
  - Handling Missing Data
  - Working with Categorical Variables

Objectives

- **How can a tree be used for classification and regression?**
  - **How should we assess partitions of the data by features?**
- How can we combine predictions from trees to increase precision?
  - Would the same approach be useful for other models?
- **Readings:**
  - Hastie, Tibshirani, Friedman 9.2
  - James, Witten, Hastie, Tibshirani 8.1
  - Murphy 16.2
  - Geron 6

Agenda

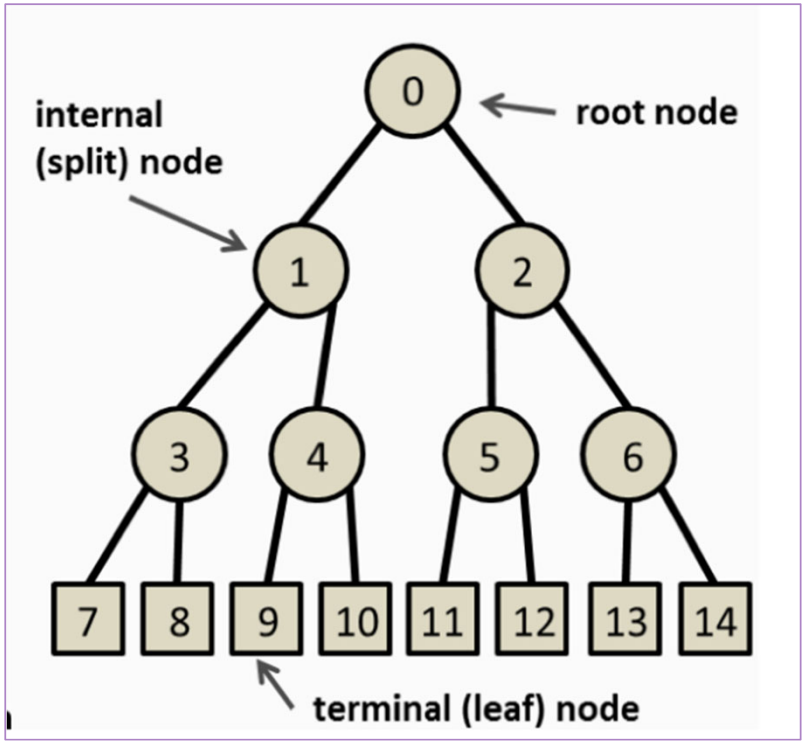
- Lesson
  - Trees
  - Dividing Data by Categorical and Numerical Features
    - Determining Partitions of the Data
  - Random Forests
    - Preventing Overfitting
    - Using Bootstrap with Random Forests
- Demo
  - Handling Missing Data
  - Working with Categorical Variables

Objectives

- **How can a tree be used for classification and regression?**
  - **How should we assess partitions of the data by features?**
- How can we combine predictions from trees to increase precision?
  - Would the same approach be useful for other models?
- **Readings:**
  - Hastie, Tibshirani, Friedman 9.2
  - James, Witten, Hastie, Tibshirani 8.1
  - Murphy 16.2
  - Geron 6

Trees

- Graphs are diagrams containing
  - Dots called Nodes
  - Lines called Edges
- Tree: Graph without loops
  - Root
  - Leaf Node
  - Internal Node
  - Height
  - Branch

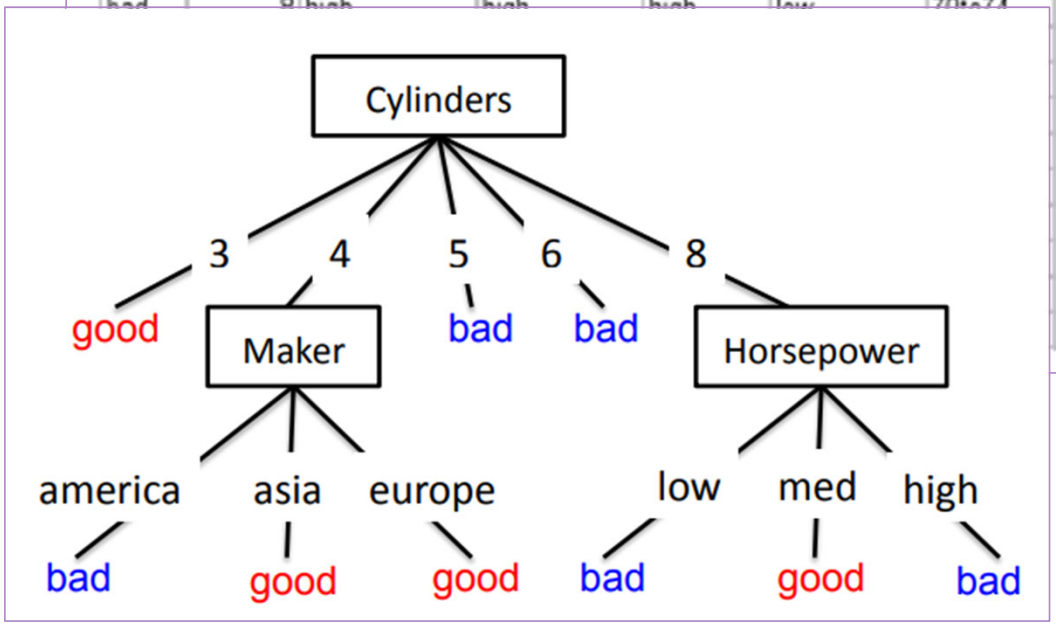


Trees

Categorical Data

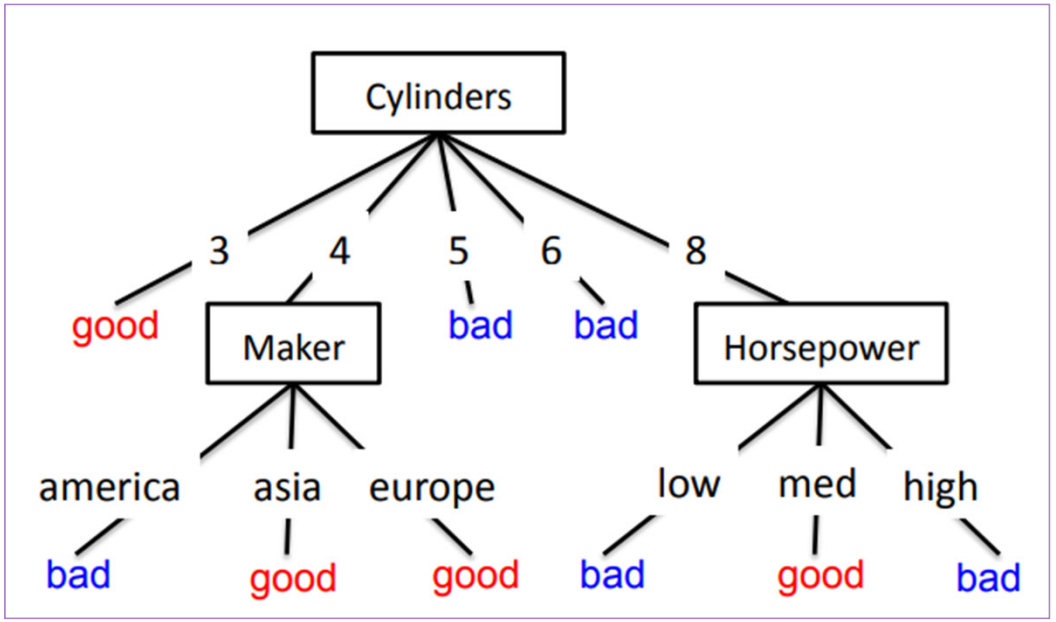
- Classification
  - Each internal node has an associated feature
    - Partition data by values of feature
  - Each leaf assigns label for classification
    - With input x traverse the tree from root to leaf
    - Output the label y

mpg	cylinders	displacement	horsepower	weight	acceleration	modelyear	maker
good	4	low	low	low	high	75to78	asia
bad	6	medium	medium	medium	medium	70to74	america
bad	4	medium	medium	medium	low	75to78	europe
bad	8	high	high	high	low	70to74	america
bad	6	medium	medium	medium	medium	70to74	america
bad	4	low	medium	low	medium	70to74	asia
bad	4	low	medium	low	low	70to74	asia
bad	8	high	high	high	low	75to78	america
:	:	:	:	:	:	:	:
:	:	:	:	:	:	:	:
:	:	:	:	:	:	:	:
bad	8	high	high	high	low	70to74	america
							america
							america
							america
							america
							america
							america
							america
							america
							europe
							europe



Trees

- How to use Tree model?
  - With larger and larger trees could represent any relationship between input and output
  - Learning the smallest tree is not computationally reasonable
  - Instead use recursion to build up the tree from the best features

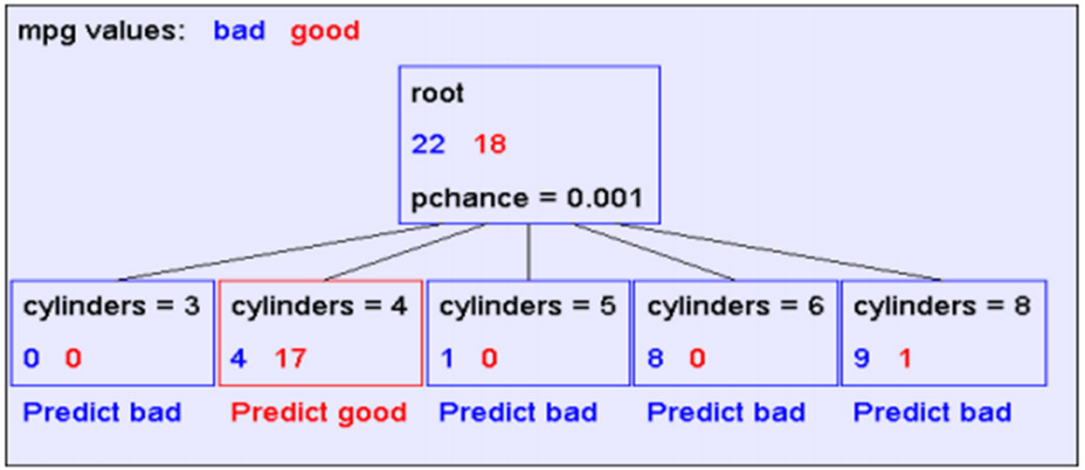


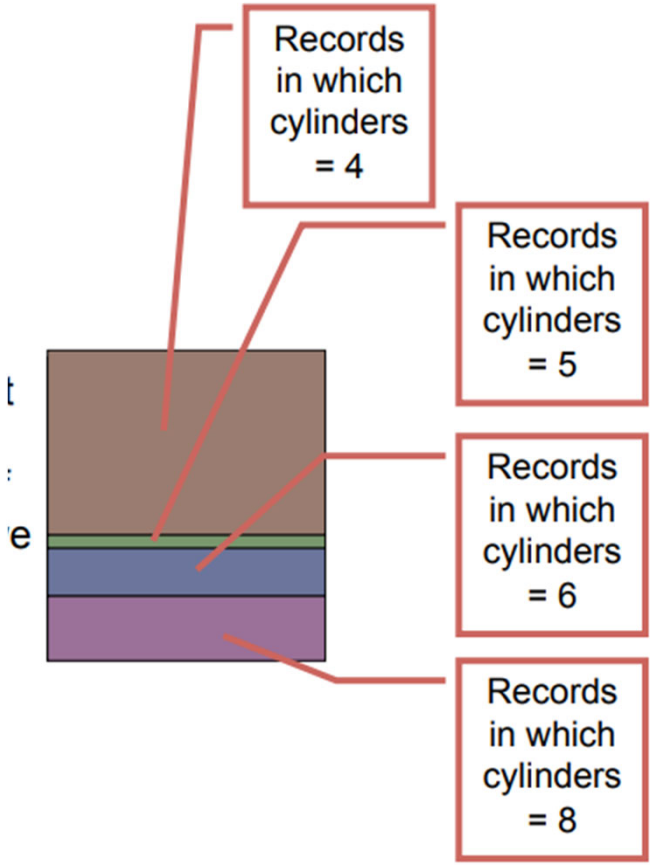
Similar to Forward Feature Selection

Trees

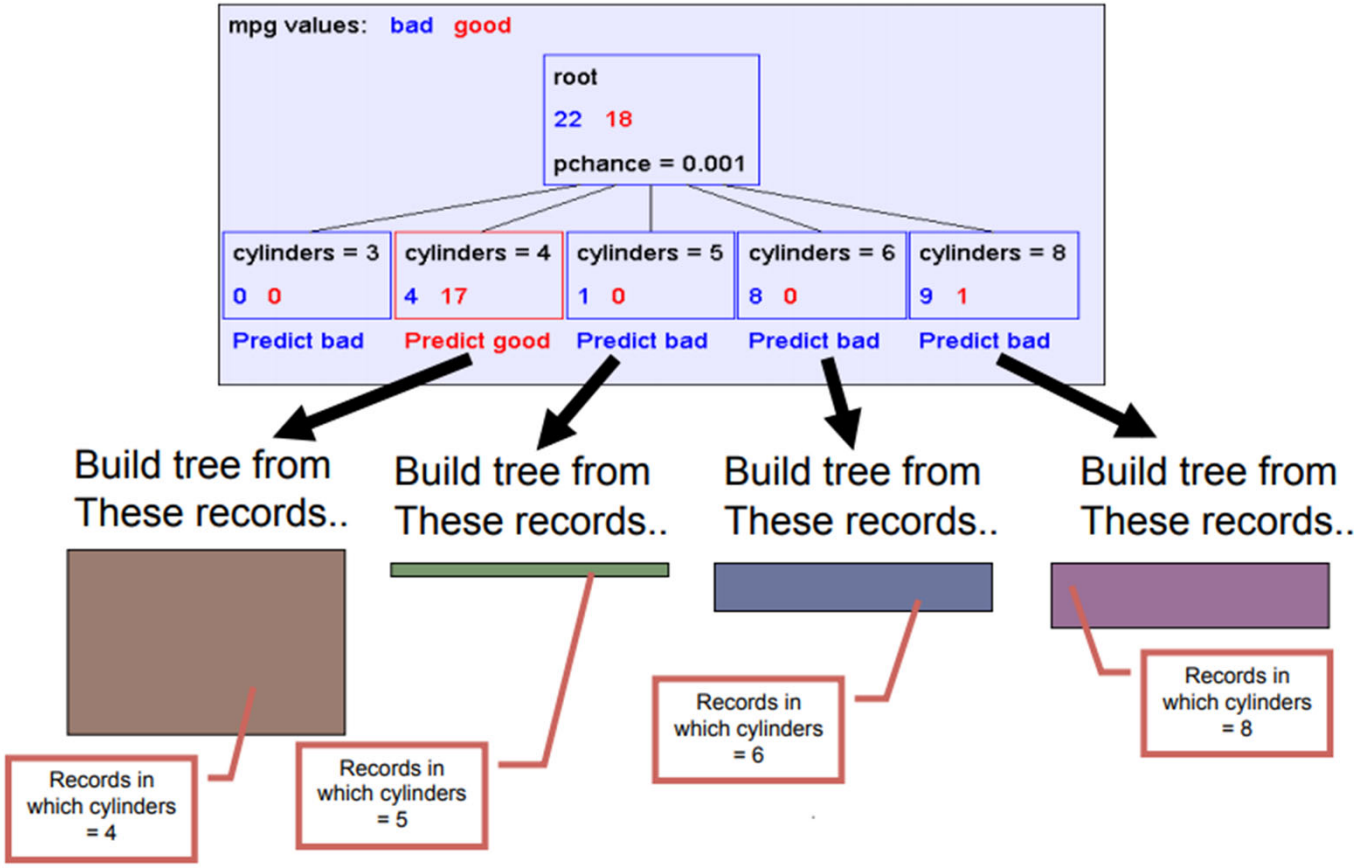
- Procedure

1. Select a feature at parent node
2. Split the data on the values of the feature
3. Repeat Step 2, 3 on children

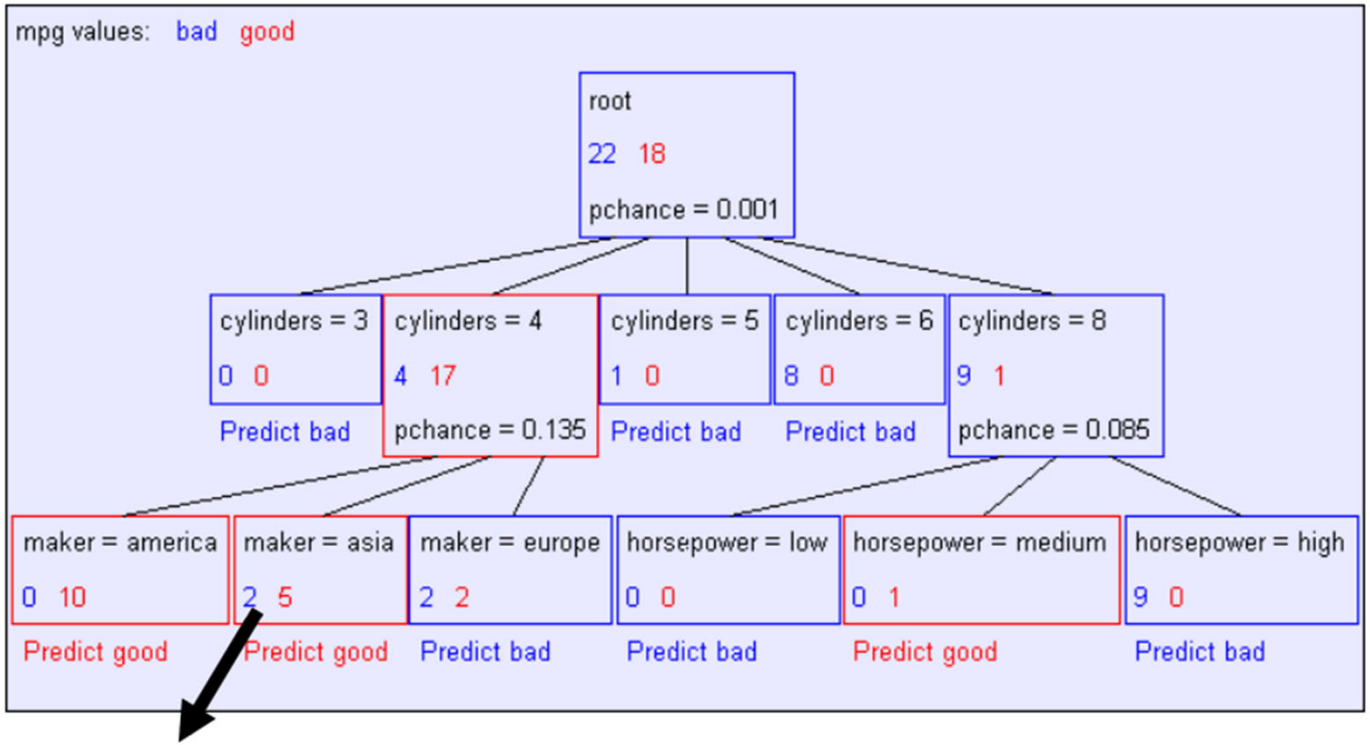




Trees


mpg values: bad good
root
22 18
pchance = 0.001
cylinders = 3
0 0
Predict bad
cylinders = 4
4 17
Predict good
cylinders = 5
1 0
Predict bad
cylinders = 6
8 0
Predict bad
cylinders = 8
9 1
Predict bad
Build tree from These records..
Build tree from These records..
Build tree from These records..
Build tree from These records..
Records in which cylinders = 4
Records in which cylinders = 5
Records in which cylinders = 6
Records in which cylinders = 8

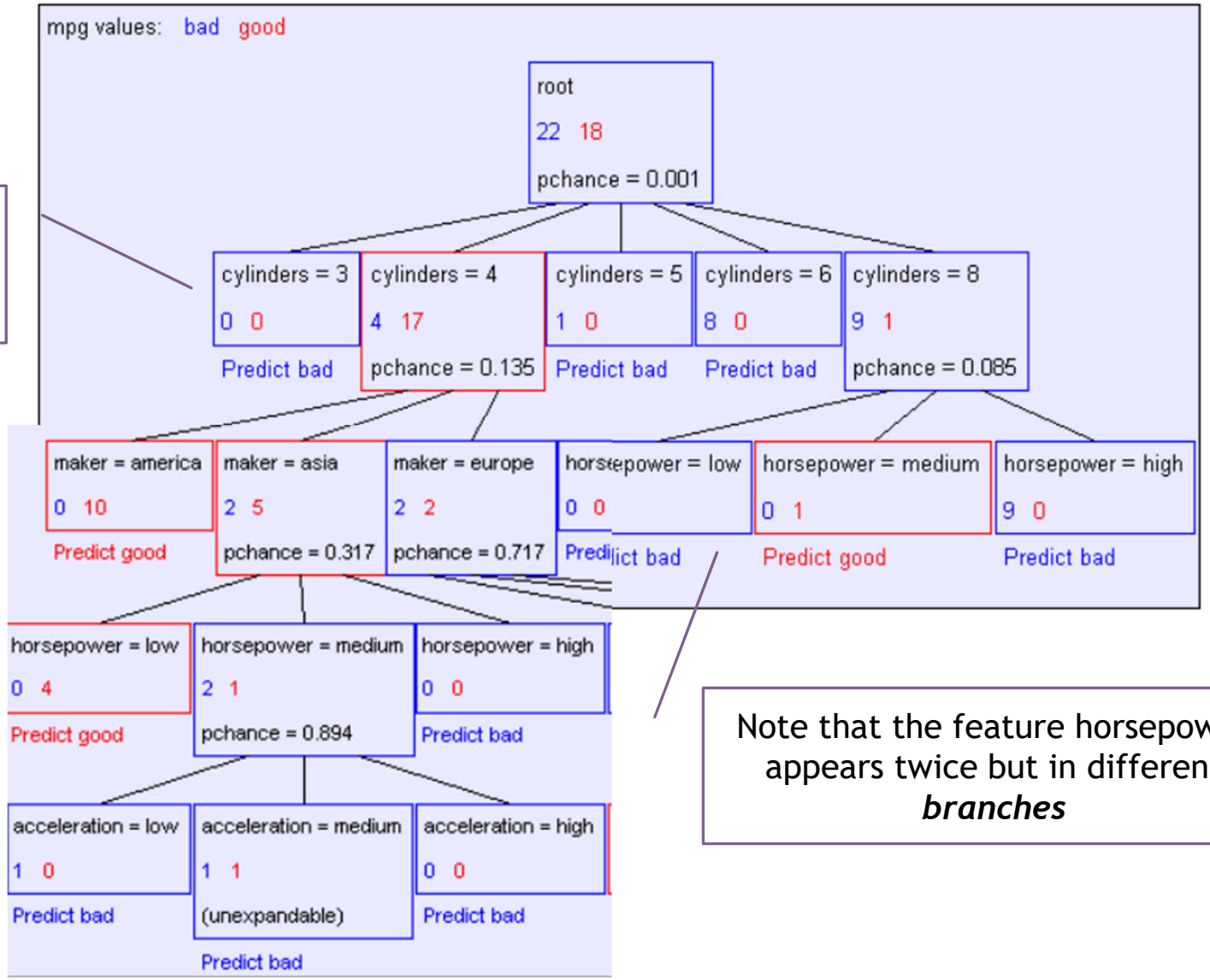
Trees



Recursively build a tree from the seven records in which there are four cylinders and the maker was based in Asia

Trees

After splitting with a feature, we do not use it in the subsequent splits



Note that the feature horsepower appears twice but in different ***branches***

Trees

Numerical Data

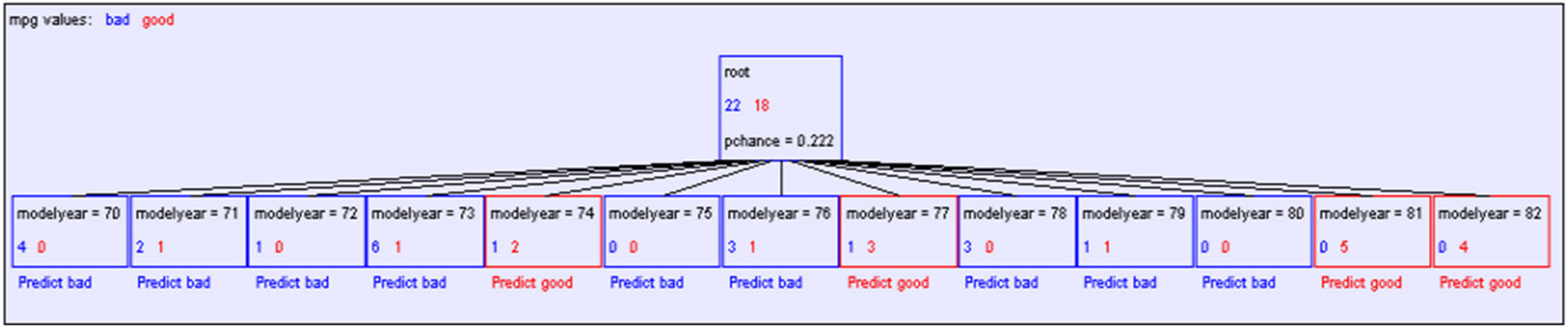
- Classification
  - Each internal node has an associated feature
    - Partition data by values of feature
  - Each leaf assigns label for classification
    - With input x traverse the tree from root to leaf
    - Output the label y

mpg	cylinders	displacemen	horsepower	weight	acceleration	modelyear	maker
good	4	97	75	2265	18.2	77	asia
bad	6	199	90	2648	15	70	america
bad	4	121	110	2600	12.8	77	europe
bad	8	350	175	4100	13	73	america
bad	6	198	95	3102	16.5	74	america
bad	4	108	94	2379	16.5	73	asia
bad	4	113	95	2228	14	71	asia
bad	8	302	139	3570	12.8	78	america
:	:	:	:	:	:	:	:
:	:	:	:	:	:	:	:
:	:	:	:	:	:	:	:
good	4	120	79	2625	18.6	82	america
bad	8	455	225	4425	10	70	america
good	4	107	86	2464	15.5	76	europe
bad	5	131	103	2830	15.9	78	europe

Trees

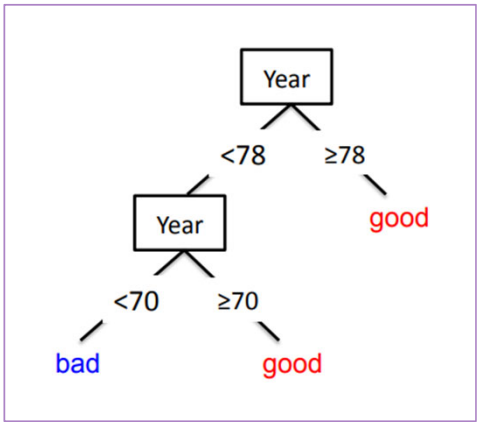
Numerical Data

mpg	cylinders	displacemen	horsepower	weight	acceleration	modelyear	maker
good	4	97	75	2265	18.2	77	asia
bad	6	199	90	2648	15	70	america
bad	4	121	110	2600	12.8	77	europe
bad	8	350	175	4100	13	73	america
bad	6	198	95	3102	16.5	74	america
bad	4	108	94	2379	16.5	73	asia
bad	4	113	95	2228	14	71	asia
bad	8	302	139	3570	12.8	78	america
:	:	:	:	:	:	:	:
:	:	:	:	:	:	:	:
:	:	:	:	:	:	:	:
good	4	120	79	2625	18.6	82	america
bad	8	455	225	4425	10	70	america
good	4	107	86	2464	15.5	76	europe
bad	5	131	103	2830	15.9	78	europe

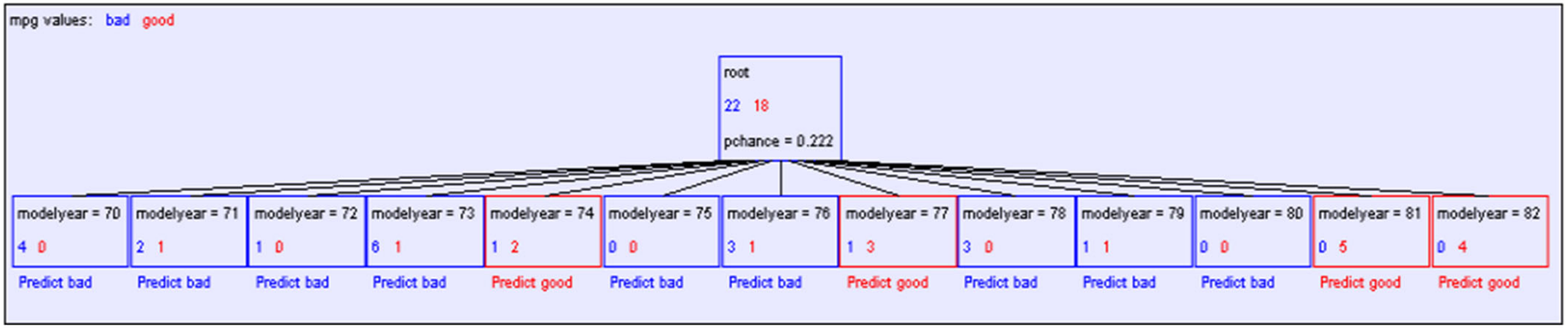


Trees

Numerical Data



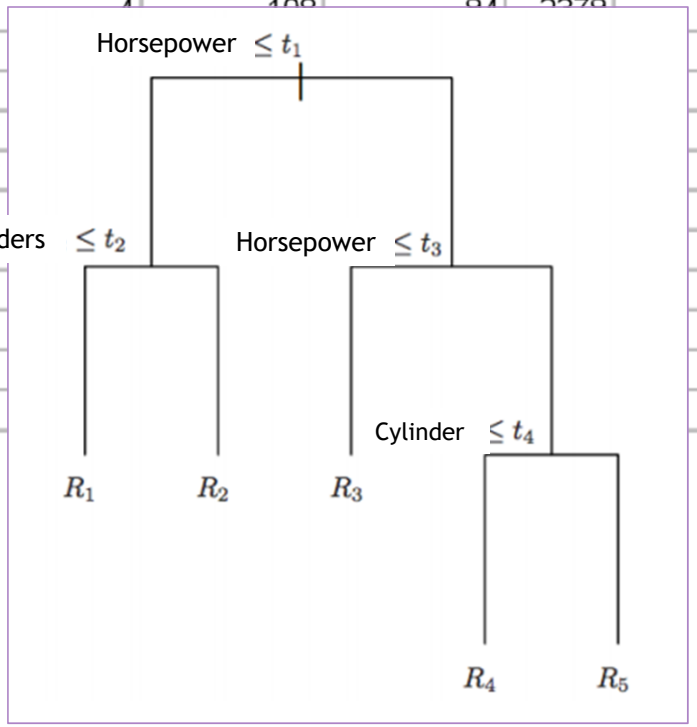
mpg	cylinders	displacemen	horsepower	weight	acceleration	modelyear	maker
good	4	97	75	2265	18.2	77	asia
bad	6	199	90	2648	15	70	america
bad	4	121	110	2600	12.8	77	europe
bad	8	350	175	4100	13	73	america
bad	6	198	95	3102	16.5	74	america
bad	4	108	94	2379	16.5	73	asia
bad	4	113	95	2228	14	71	asia
bad	8	302	139	3570	12.8	78	america
:	:	:	:	:	:	:	:
:	:	:	:	:	:	:	:
:	:	:	:	:	:	:	:
good	4	120	79	2625	18.6	82	america
bad	8	455	225	4425	10	70	america
good	4	107	86	2464	15.5	76	europe
bad	5	131	103	2830	15.9	78	europe



Trees

- Classification
  - Each internal node has an associated feature
    - Partition data by values of feature
  - Each leaf assigns label for classification
    - With input x traverse the tree from root to leaf
    - Output the label y

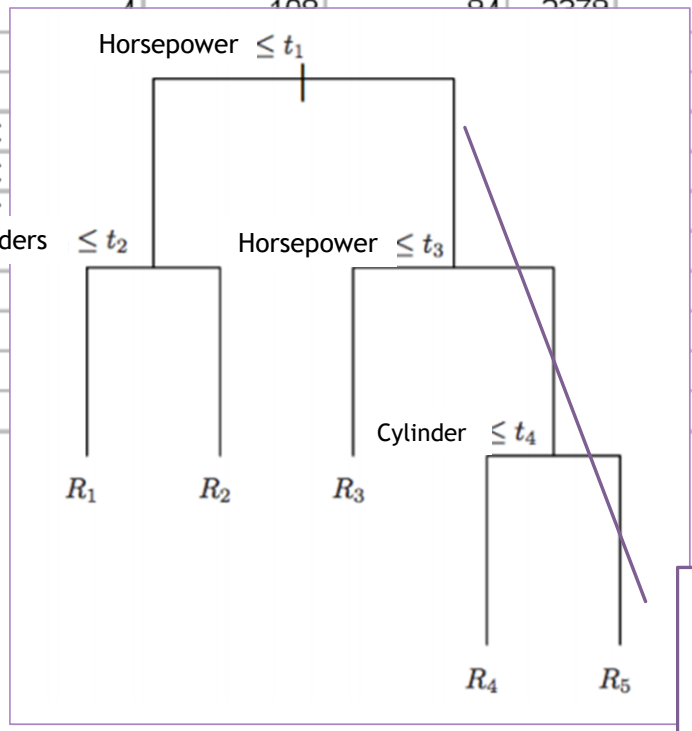
mpg	cylinders	displacemen	horsepower	weight	acceleration	modelyear	maker
good	4	97	75	2265	18.2	77	asia
bad	6	199	90	2648	15	70	america
bad	4	121	110	2600	12.8	77	europe
bad	8	350	175	4100	13	73	america
bad	6	198	95	3102	16.5	74	america
bad					16.5	73	asia
bad					14	71	asia
bad					12.8	78	america
:	:					:	:
:	:					:	:
:	:					:	:
gc					18.6	82	america
bad					10	70	america
good					15.5	76	europe
bad					15.9	78	europe



Trees

- Classification
  - Each internal node has an associated feature
    - Partition data by values of feature
  - Each leaf assigns label for classification
    - With input x traverse the tree from root to leaf
    - Output the label y

mpg	cylinders	displacemen	horsepower	weight	acceleration	modelyear	maker
good	4	97	75	2265	18.2	77	asia
bad	6	199	90	2648	15	70	america
bad	4	121	110	2600	12.8	77	europe
bad	8	350	175	4100	13	73	america
bad	6	198	95	3102	16.5	74	america
bad					16.5	73	asia
bad					14	71	asia
bad					12.8	78	america
:	:					:	:
:	:					:	:
:	:					:	:
gc					18.6	82	america
bad					10	70	america
good					15.5	76	europe
bad					15.9	78	europe

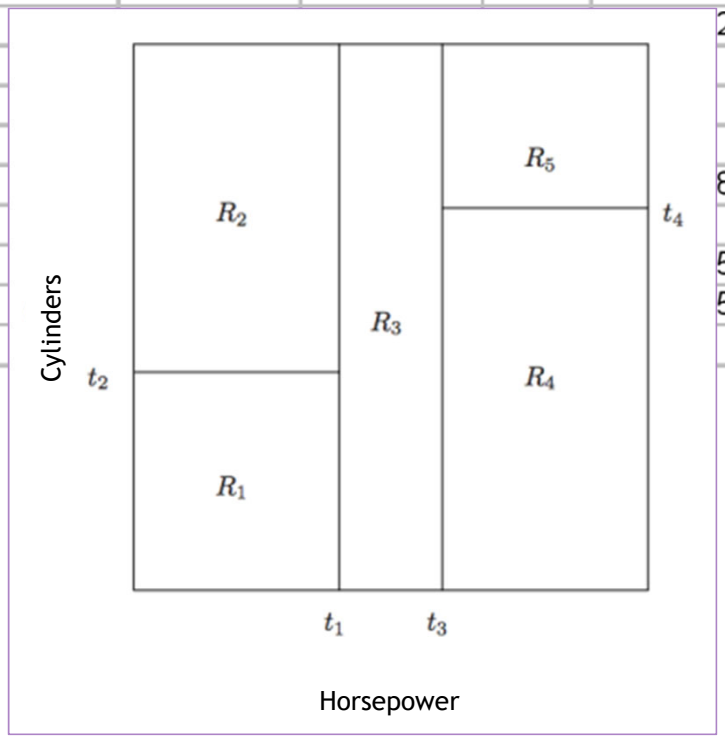


Note multiple splits along same branch

Trees

- Classification
  - Each internal node has an associated feature
    - Partition data by values of feature
  - Each leaf assigns label for classification
    - With input x traverse the tree from root to leaf
    - Output the label y

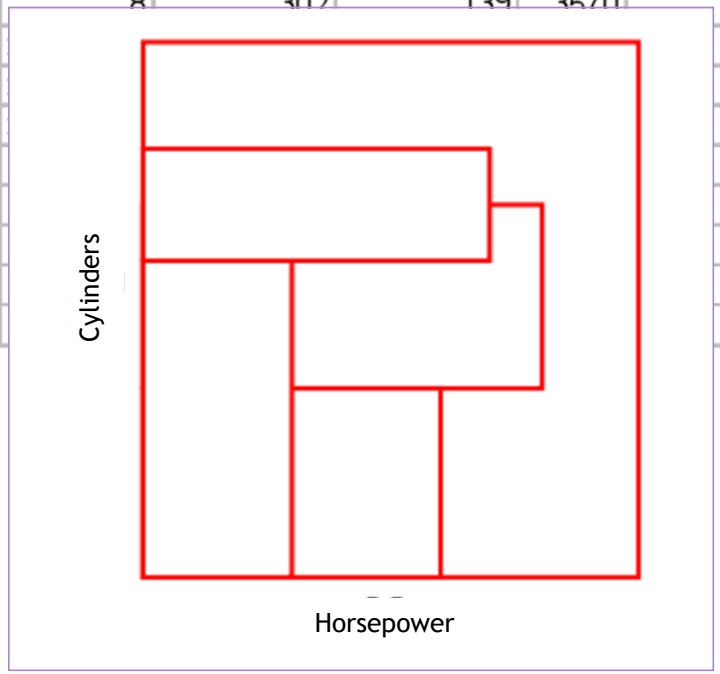
mpg	cylinders	displacemen	horsepower	weight	acceleration	modelyear	maker
good	4	97	75	2265	18.2	77	asia
bad	6	199	90	2648	15	70	america
bad	4	121	110	2600	12.8	77	europe
bad	8	350	175	4100	13	73	america
bad	6	198	95	3102	16.5	74	america
bad	4	108	94	2379	16.5	73	asia
bad	4	113	95	2228	14	71	asia
bad					2.8	78	america
:	:					:	:
:	:					:	:
:	:					:	:
good					8.6	82	america
bad					10	70	america
good					5.5	76	europe
bad					5.9	78	europe



Trees

- Classification
  - Each internal node has an associated feature
    - Partition data by values of feature
  - Each leaf assigns label for classification
    - With input x traverse the tree from root to leaf
    - Output the label y

mpg	cylinders	displacemen	horsepower	weight	acceleration	modelyear	maker
good	4	97	75	2265	18.2	77	asia
bad	6	199	90	2648	15	70	america
bad	4	121	110	2600	12.8	77	europe
bad	8	350	175	4100	13	73	america
bad	6	198	95	3102	16.5	74	america
bad	4	108	94	2379	16.5	73	asia
bad	4	113	95	2228	14	71	asia
bad	8	302	139	3570	12.8	78	america
:	:	:	:	:	:	:	:
:	:	:	:	:	:	:	:
:	:	:	:	:	:	:	:
good	4	97	75	2265	18.6	82	america
bad	6	199	90	2648	10	70	america
good	4	121	110	2600	15.5	76	europe
bad	8	350	175	4100	15.9	78	europe



Trees

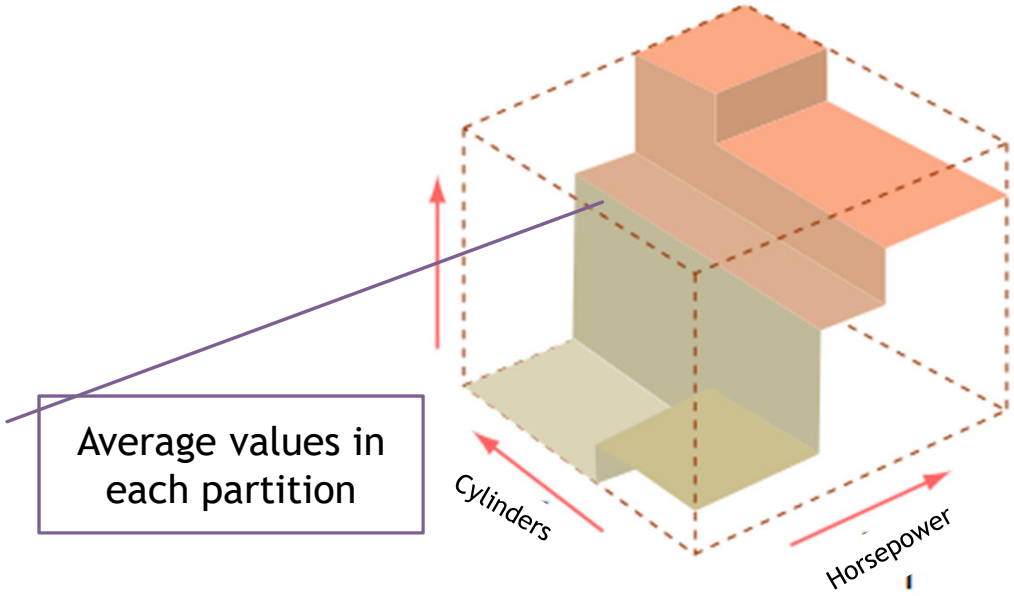
- Issues
  - Should we use a threshold for categorical data?
    - Ordinal Data
    - Group Categories Together

mpg	cylinders	displacement	horsepower	weight	acceleration	modelyear	maker
good	4	low	low	low	high	75to78	asia
bad	6	medium	medium	medium	medium	70to74	america
bad	4	medium	medium	medium	low	75to78	europe
bad	8	high	high	high	low	70to74	america
bad	6	medium	medium	medium	medium	70to74	america
bad	4	low	medium	low	medium	70to74	asia
bad	4	low	medium	low	low	70to74	asia
bad	8	high	high	high	low	75to78	america
:	:	:	:	:	:	:	:
:	:	:	:	:	:	:	:
:	:	:	:	:	:	:	:
bad	8	high	high	high	low	70to74	america
good	8	high	medium	high	high	79to83	america
bad	8	high	high	high	low	75to78	america
good	4	low	low	low	low	79to83	america
bad	6	medium	medium	medium	high	75to78	america
good	4	medium	low	low	low	79to83	america
good	4	low	low	medium	high	79to83	america
bad	8	high	high	high	low	70to74	america
good	4	low	medium	low	medium	75to78	europe
bad	5	medium	medium	medium	medium	75to78	europe

Could replace category with fraction of bad. After transformation, we can treat as numerical data.

Trees

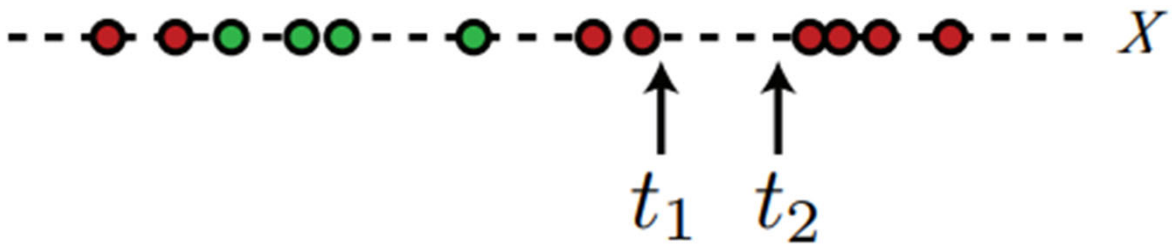
- Issues
  - How could we adapt the approach for classification to regression?



- Regression
  - Each internal node has an associated feature
    - Partition data by value of feature
  - Each leaf assigns number for regression
    - For input x traverse the tree from root to leaf
    - Output aggregate of values in leaf

Trees

- Issues
  - **How could we choose between different thresholds**
  - How should we choose between different thresholds?

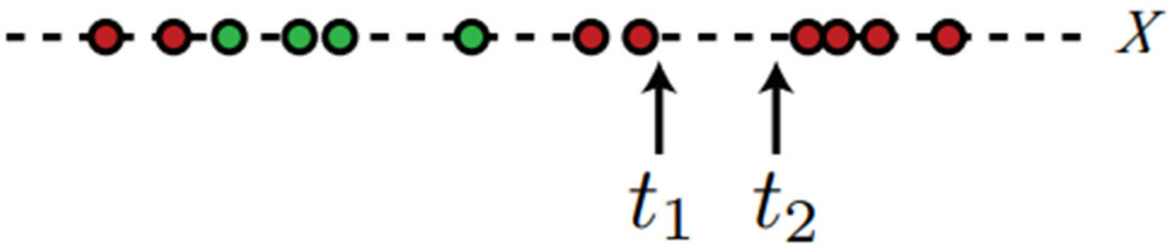


Only finitely many values of the threshold t are important

Trees

- Issues
  - **How could we choose between different thresholds**
  - How should we choose between different thresholds?

Sort the data into order $x_1,\ldots,x_m$.
Split at $x_i + (x_{i+1}-x_i)/2$

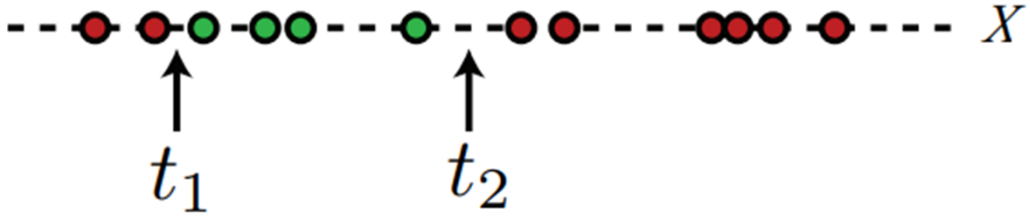


Only finitely many values of the threshold t are important

Trees

- Issues
  - **How could we choose between different thresholds**
  - How should we choose between different thresholds?

Sort the data into order $x_1,\ldots,x_m$.
Split at $x_i + (x_{i+1}-x_i)/2$



Only finitely many values of the threshold t are important

Only splits between different classes matter

Agenda

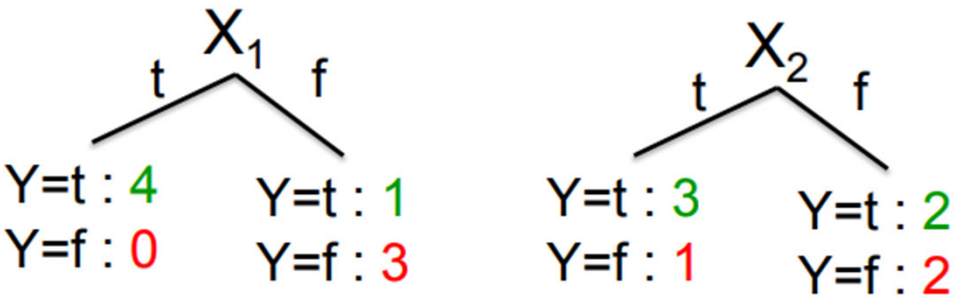
- Lesson
  - Trees
    - Dividing Data by Categorical and Numerical Features
    - Determining Partitions of the Data
  - Random Forests
    - Preventing Overfitting
    - Using Bootstrap with Random Forests
- Demo
  - Handling Missing Data
  - Working with Categorical Variables

Objectives

- **How can a tree be used for classification and regression?**
  - **How should we assess partitions of the data by features?**
- How can we combine predictions from trees to increase precision?
  - Would the same approach be useful for other models?
- **Readings:**
  - Hastie, Tibshirani, Friedman 9.2
  - James, Witten, Hastie, Tibshirani 8.1
  - Murphy 16.2
  - Geron 6

Trees

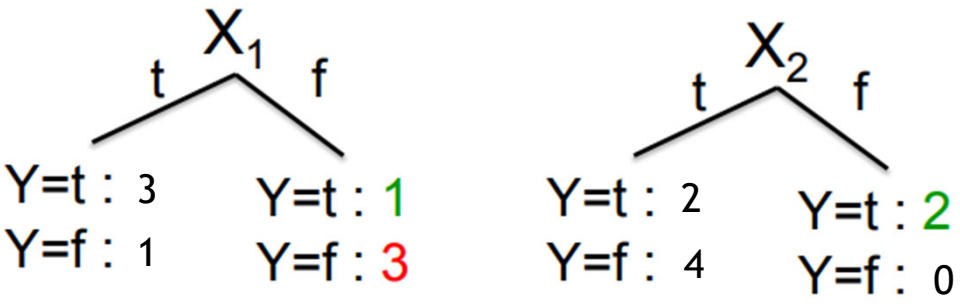
- Determining Features for Split
  - Use counts at leaves to evaluate different features
  - Want the data in the partitions to be the same



X_1	X_2	Y
T	T	T
T	F	T
T	T	T
T	F	T
F	T	T
F	F	F
F	T	F
F	F	F

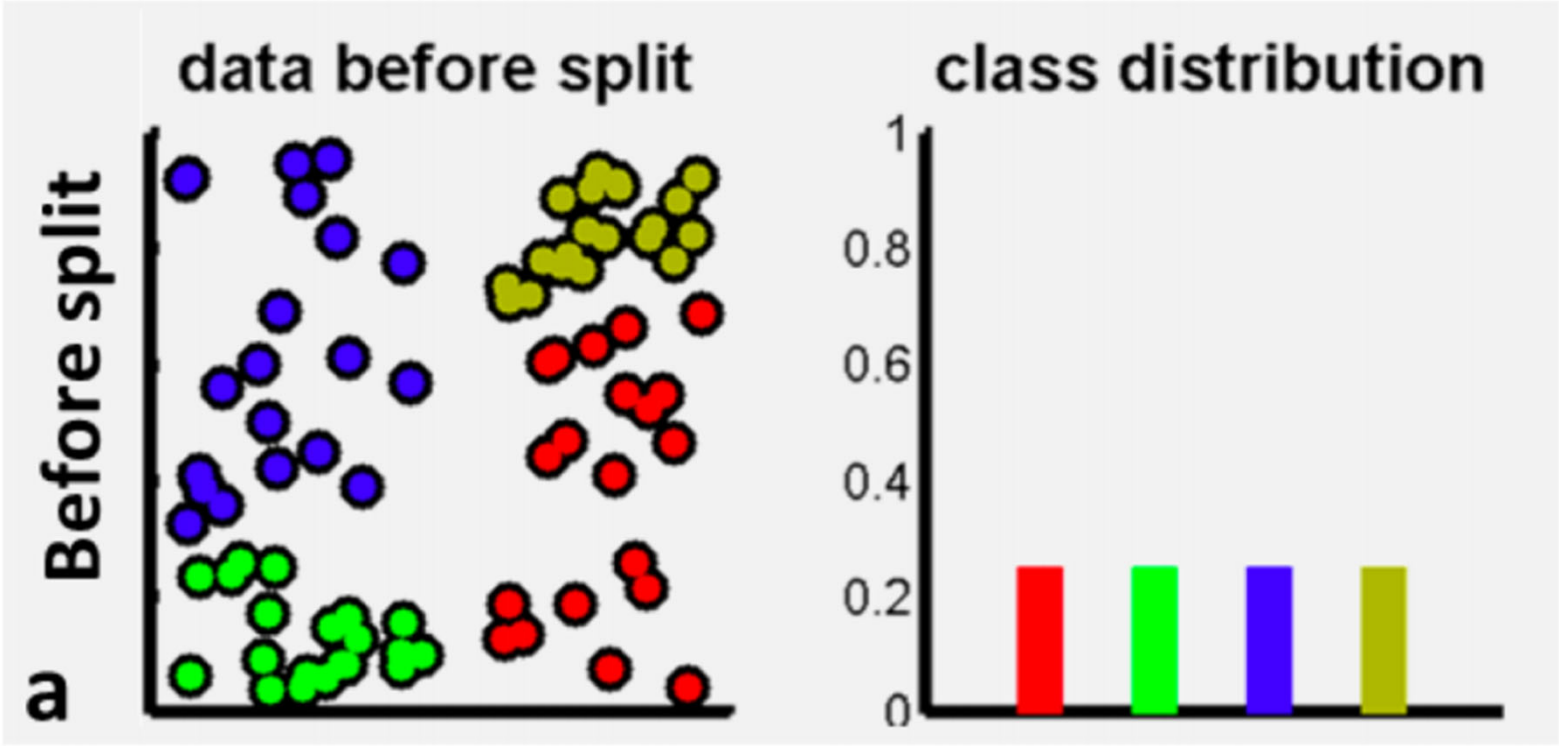
Trees

- Determining Features for Split
  - Use counts at leaves to evaluate different features
  - Want the data in the partitions to be the same

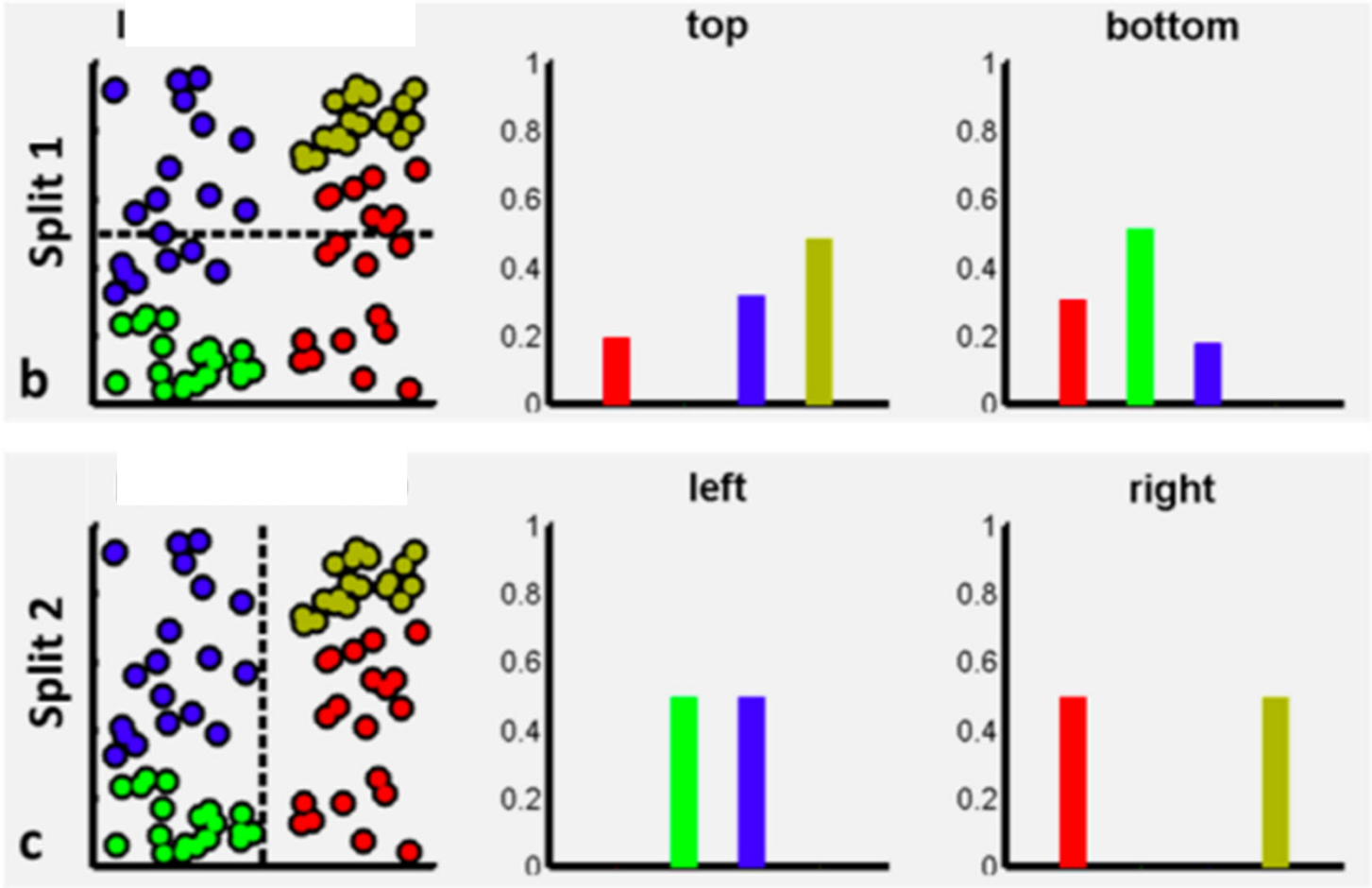


X_1	X_2	Y
T	T	T
T	F	T
T	T	T
T	F	T
F	T	T
F	F	F
F	T	F
F	F	F

Trees

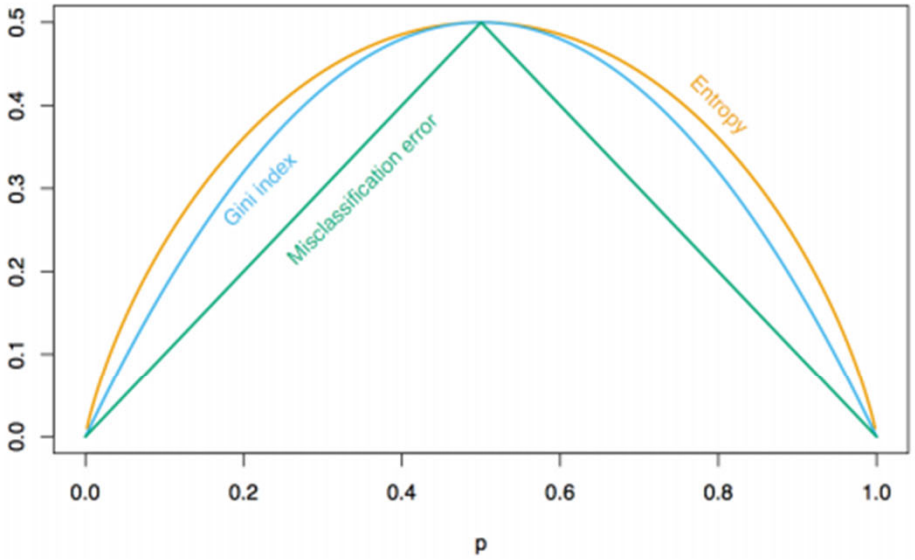

data before split
class distribution
Before split
a
1
0.8
0.6
0.4
0.2
0

Trees



Trees

- Consider binary classification
- Let p be the relative frequency of class 1.
- Here are three node impurity measures as a function of p



Misclassification

$$1 - \hat{p}_{mk(m)}.$$

Gini Index

$$\sum_{k=1}^{K} \hat{p}_{mk}(1 - \hat{p}_{mk})$$

Entropy

$$-\sum_{k=1}^{K} \hat{p}_{mk} \log \hat{p}_{mk}.$$

Trees

- Information

- $I(p) \geq 0, I(1) = 0$; the information of any event is non-negative, no information from events with prob 1
- $I(p_1 \cdot p_2) = I(p_1) + I(p_2)$; the information from two independent events should be the sum of their informations
- $I(p)$ is continuous, slight changes in probability correspond to slight changes in information

Trees

- Information

Together these lead to:

$$I(p^2) = 2I(p) \text{ or generally } I(p^n) = nI(p),$$

this means that

$$I(p) = I\left((p^{1/m})^m\right) = mI\left(p^{1/m}\right) \text{ so } \frac{1}{m}I(p) = I\left(p^{1/m}\right)$$

and more generally,

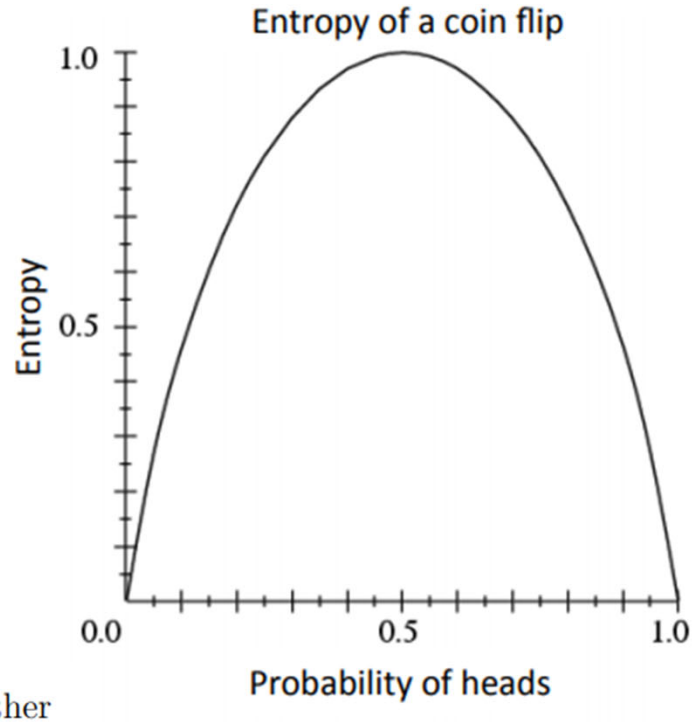
$$I\left(p^{n/m}\right) = \frac{n}{m}I(p).$$

Only functions with this property are logarithm…

Trees

- Information

$$I(p) = -\log_b(p)$$



Flipping a fair coin gives $-\log_2(1/2) = 1$ bit of information if it comes up either heads or tails.

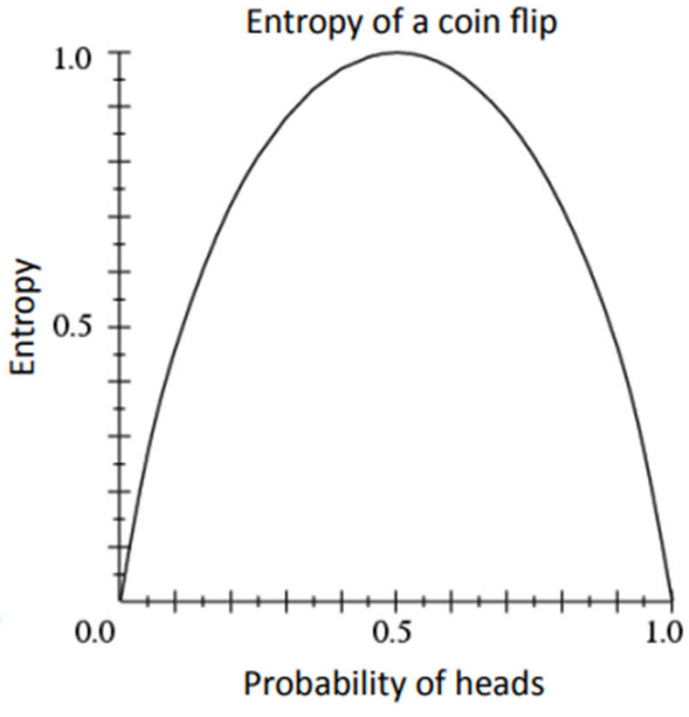
A biased coin landing on heads with $p = .99$ gives $-\log_2(.99) = .0145$ bits of information.

A biased coin landing on heads with $p = .01$ gives $-\log_2(.01) = 6.643$ bits of information.

Trees

- Entropy

$$H(Y) = -\sum_{i=1}^{k} P(Y = y_i) \log_2 P(Y = y_i)$$



So if there are only 2 events (binary), with probabilities $\mathbf{p} = [p, 1-p]$,

$$H(\mathbf{p}) = -p \log_2(p) - (1-p) \log_2(1-p).$$

If the probabilities were $[1/2, 1/2]$,

$$H(\mathbf{p}) = -2\frac{1}{2} \log_2 \frac{1}{2} = 1$$

Or if the probabilities were $[0.99, 0.01]$,

$$H(\mathbf{p}) = 0.08 \text{ bits.}$$

Trees

- Determining Features for Split
  - Split should make use more certain about classification
  - High entropy means fractions are the same so classification is less predictable
  - Low entropy means fractions are like 0 and 1 so classification more predictable

X_1	X_2	Y
T	T	T
T	F	T
T	T	T
T	F	T
F	T	T
F	F	F

$P(Y=t) = 5/6$

$P(Y=f) = 1/6$

$H(Y) = - 5/6 \log_2 5/6 - 1/6 \log_2 1/6$

$= 0.65$

Trees

- Determining Features for Split
  - Split should make use more certain about classification
  - High entropy means fractions are the same so classification is less predictable
  - Low entropy means fractions are like 0 and 1 so classification more predictable

X_1	X_2	Y
T	T	T
T	F	T
T	T	T
T	F	T
F	T	T
F	F	F

$P(Y=t) = 5/6$

$P(Y=f) = 1/6$

$H(Y) = - 5/6 \log_2 5/6 - 1/6 \log_2 1/6$

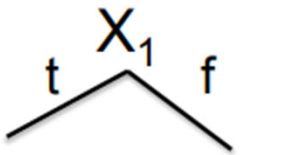
$= 0.65$

Trees

- Conditional Entropy

$$H(Y \mid X) = -\sum_{j=1}^{v} P(X = x_j) \sum_{i=1}^{k} P(Y = y_i \mid X = x_j) \log_2 P(Y = y_i \mid X = x_j)$$

X_1	X_2	Y
T	T	T
T	F	T
T	T	T
T	F	T
F	T	T
F	F	F



$P(X_1=t) = 4/6$
$P(X_1=f) = 2/6$

Y=t : 4
Y=f : 0

Y=t : 1
Y=f : 1

$H(Y|X_1) = -\ 4/6\ (1 \log_2 1 + 0 \log_2 0)$
$-\ 2/6\ (1/2 \log_2 1/2 + 1/2 \log_2 1/2)$
$= 2/6$

Trees

- Information Gain

$$IG(X) = H(Y) - H(Y \mid X)$$

X_1	X_2	Y
T	T	T
T	F	T
T	T	T
T	F	T
F	T	T
F	F	F

$H(Y) = - 5/6 \log_2 5/6 - 1/6 \log_2 1/6$

$= 0.65$

$H(Y|X_1) = - 4/6 (1 \log_2 1 + 0 \log_2 0)$

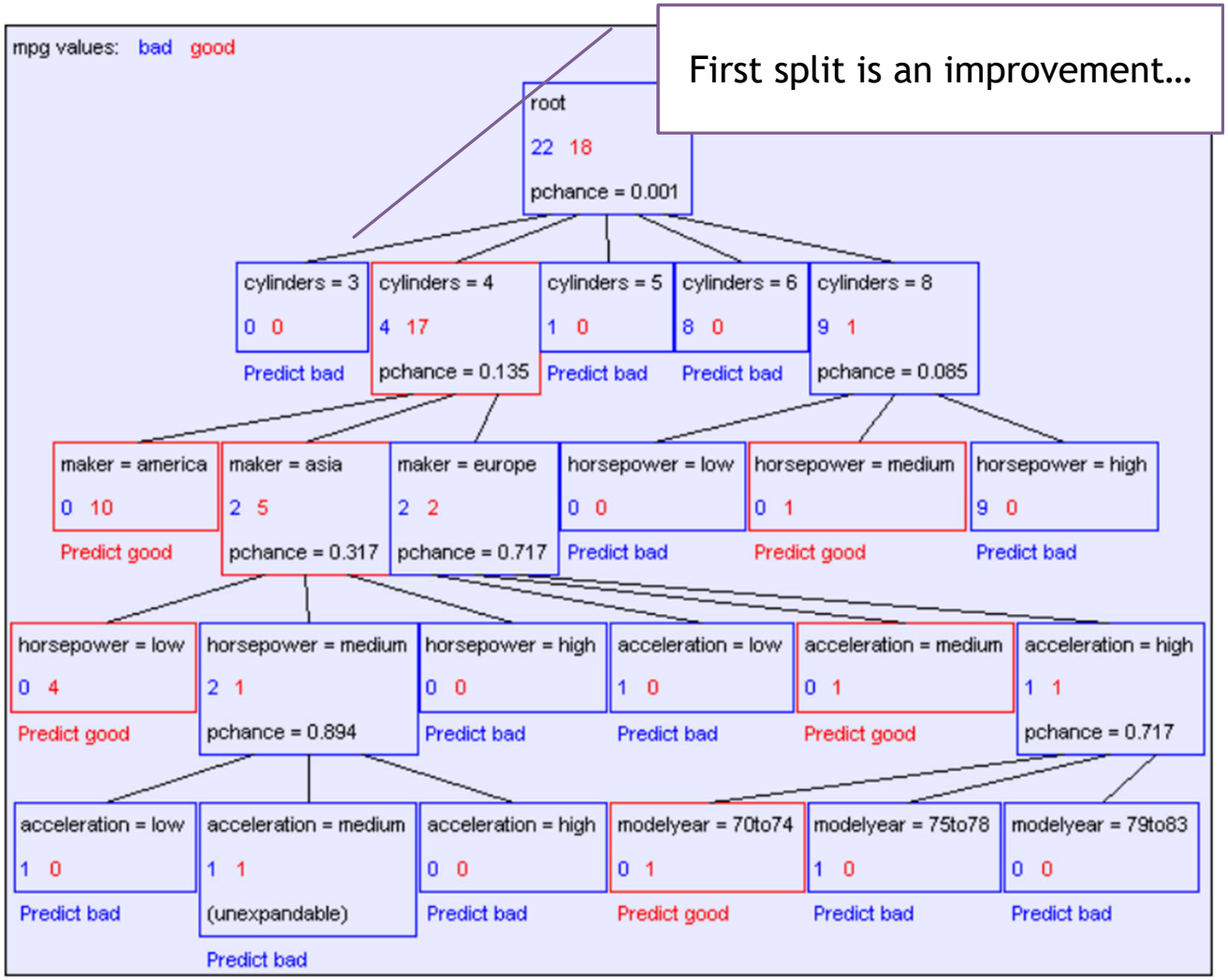
$- 2/6 (1/2 \log_2 1/2 + 1/2 \log_2 1/2)$

$= 2/6$

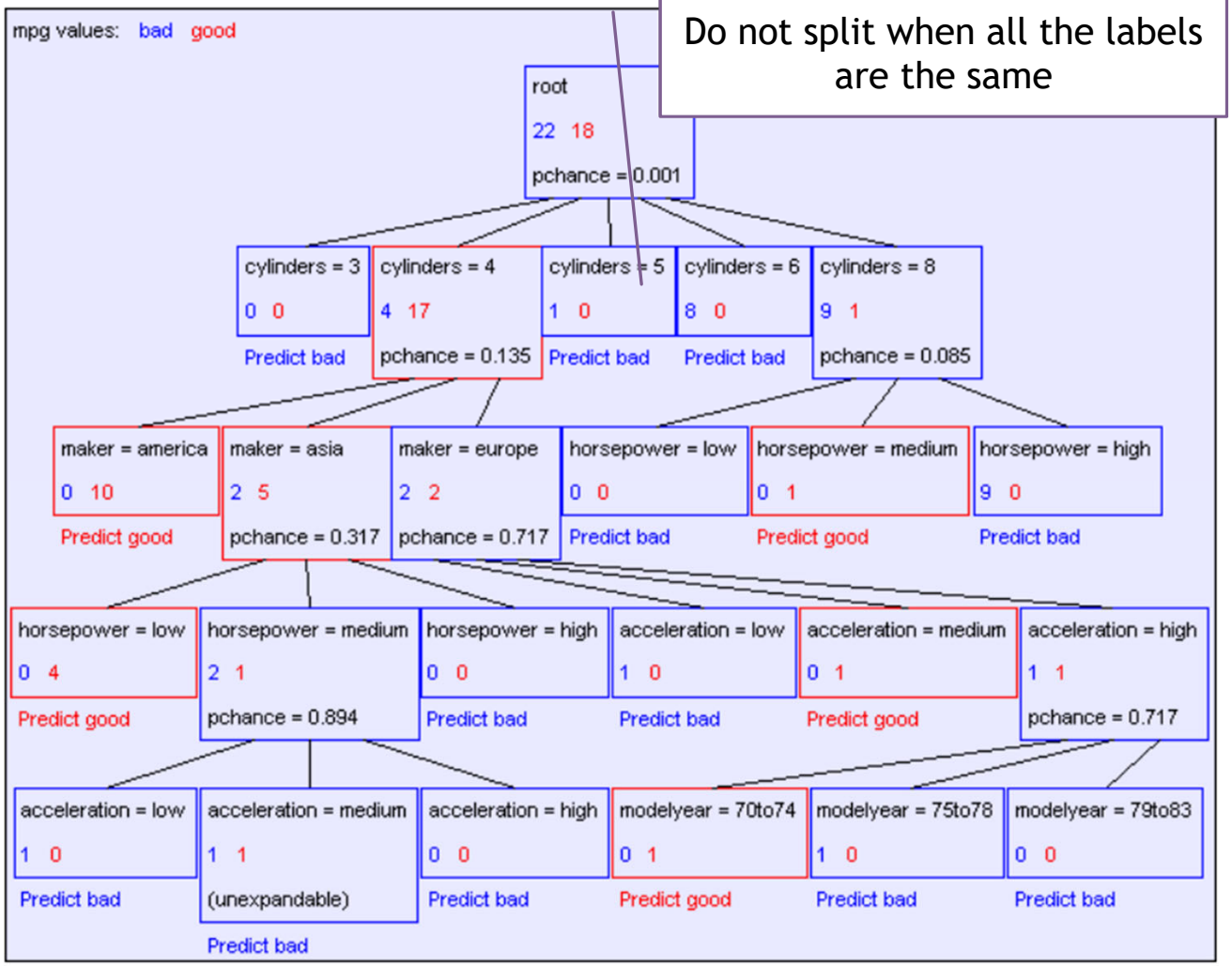
$IG(X_1) = H(Y) - H(Y|X_1)$

$= 0.65 - 0.33$

Trees

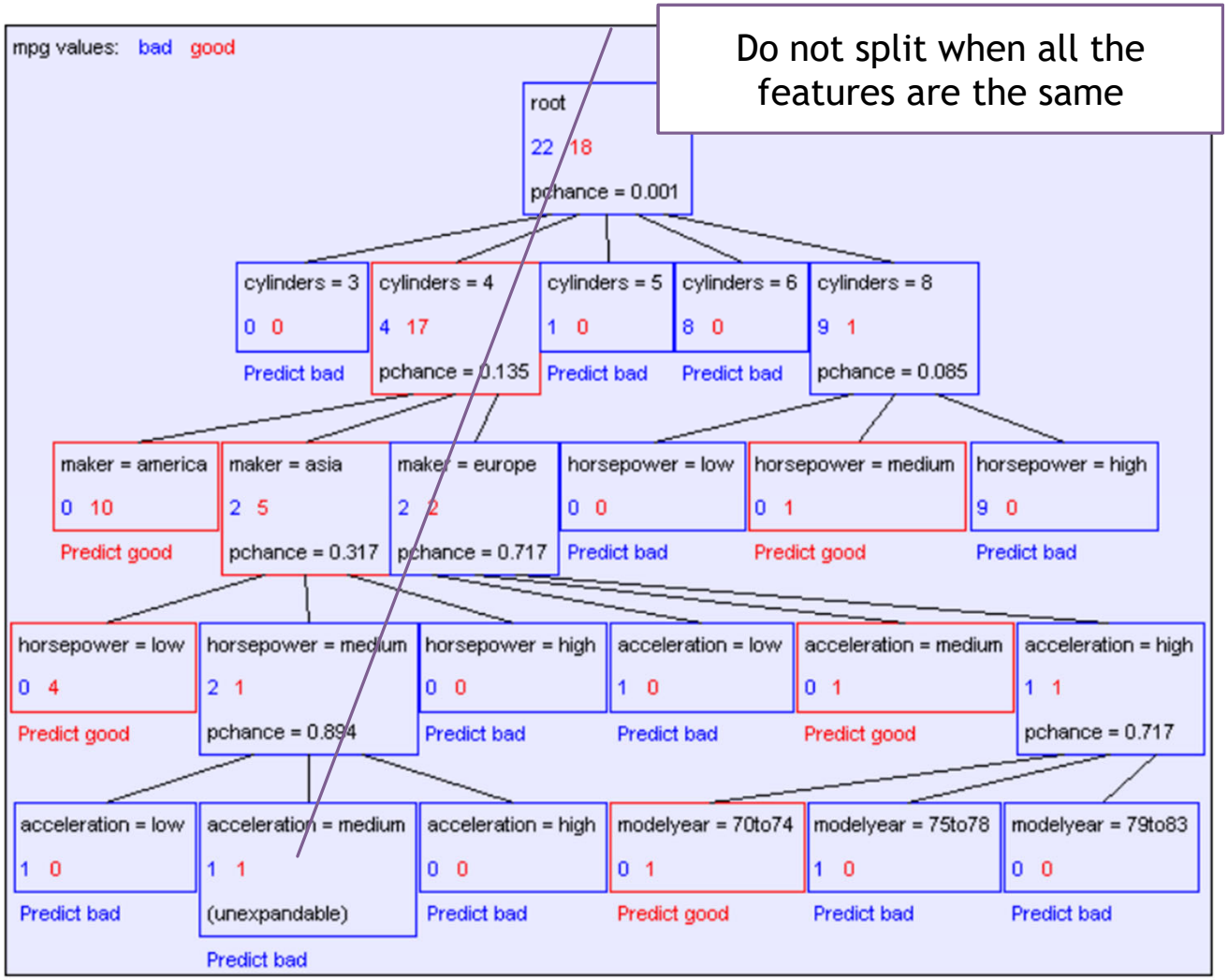

mpg values: bad good
First split is an improvement…
root
22 18
pchance = 0.001
cylinders = 3
0 0
Predict bad
cylinders = 4
4 17
pchance = 0.135
cylinders = 5
1 0
Predict bad
cylinders = 6
8 0
Predict bad
cylinders = 8
9 1
pchance = 0.085
maker = america
0 10
Predict good
maker = asia
2 5
pchance = 0.317
maker = europe
2 2
pchance = 0.717
horsepower = low
0 0
Predict bad
horsepower = medium
0 1
Predict good
horsepower = high
9 0
Predict bad
horsepower = low
0 4
Predict good
horsepower = medium
2 1
pchance = 0.894
horsepower = high
0 0
Predict bad
acceleration = low
1 0
Predict bad
acceleration = medium
0 1
Predict good
acceleration = high
1 1
pchance = 0.717
acceleration = low
1 0
Predict bad
acceleration = medium
1 1
(unexpandable)
Predict bad
acceleration = high
0 0
Predict bad
modelyear = 70to74
0 1
Predict good
modelyear = 75to78
1 0
Predict bad
modelyear = 79to83
0 0
Predict bad

Trees



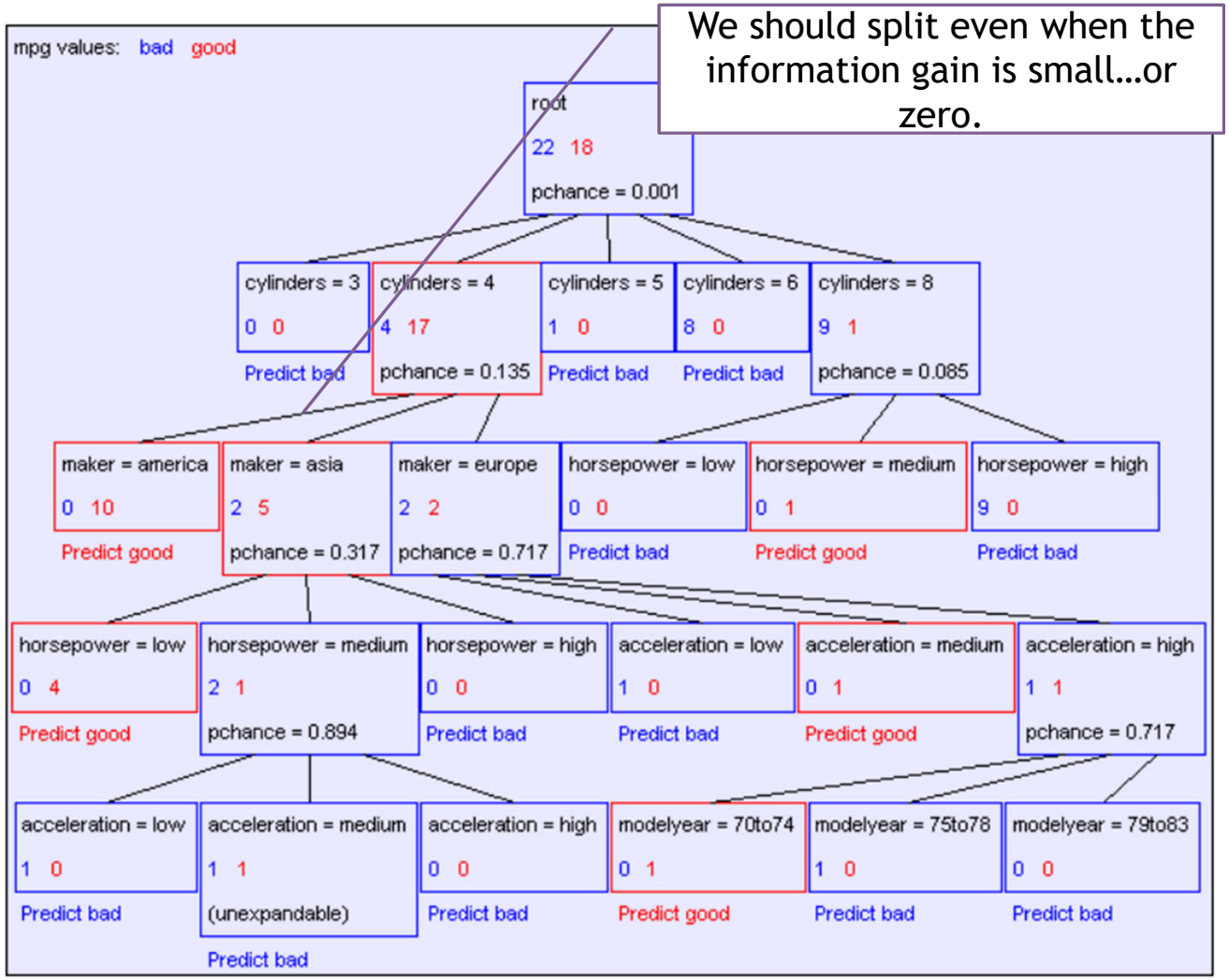
Do not split when all the labels are the same

Trees



Do not split when all the features are the same

Trees

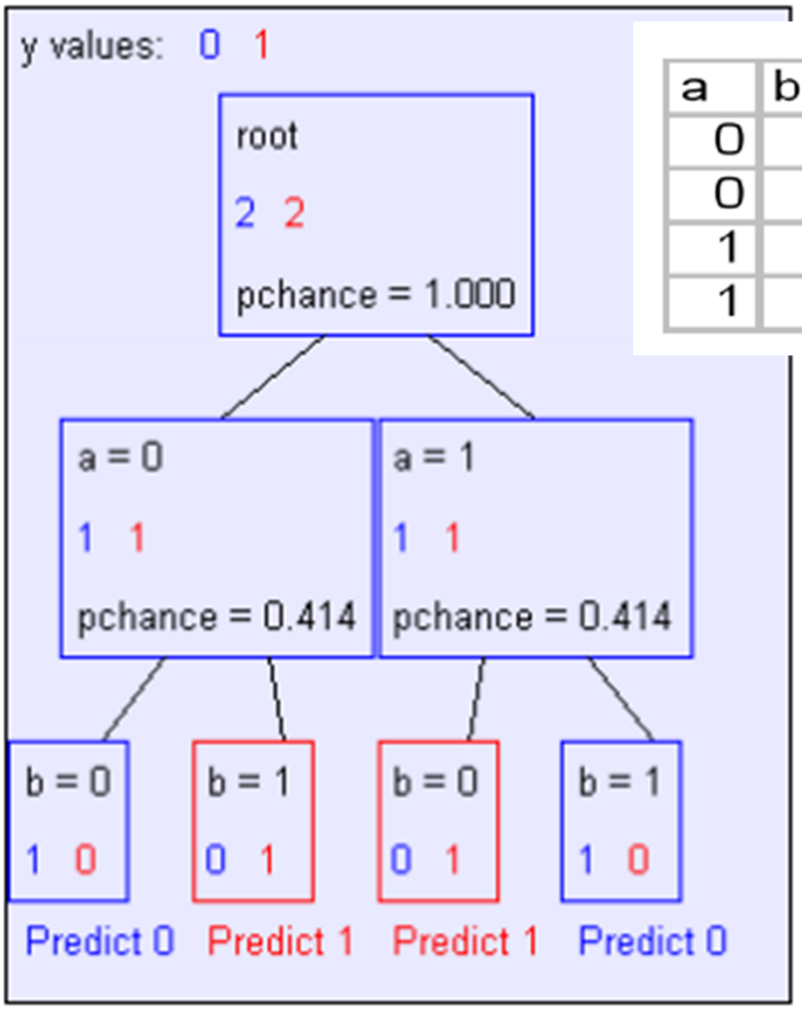

We should split even when the information gain is small…or zero.
mpg values: bad good
root
22 18
pchance = 0.001
cylinders = 3
0 0
Predict bad
cylinders = 4
4 17
pchance = 0.135
cylinders = 5
1 0
Predict bad
cylinders = 6
8 0
Predict bad
cylinders = 8
9 1
pchance = 0.085
maker = america
0 10
Predict good
maker = asia
2 5
pchance = 0.317
maker = europe
2 2
pchance = 0.717
horsepower = low
0 0
Predict bad
horsepower = medium
0 1
Predict good
horsepower = high
9 0
Predict bad
horsepower = low
0 4
Predict good
horsepower = medium
2 1
pchance = 0.894
horsepower = high
0 0
Predict bad
acceleration = low
1 0
Predict bad
acceleration = medium
0 1
Predict good
acceleration = high
1 1
pchance = 0.717
acceleration = low
1 0
Predict bad
acceleration = medium
1 1
(unexpandable)
Predict bad
acceleration = high
0 0
Predict bad
modelyear = 70to74
0 1
Predict good
modelyear = 75to78
1 0
Predict bad
modelyear = 79to83
0 0
Predict bad

Trees

Information gains using the training set (4 records)

y values: 0 1

Input	Value	Distribution	Info Gain
a	0		0
	1		
b	0		0
	1		



a	b	y
0	0	0
0	1	1
1	0	1
1	1	0

Agenda

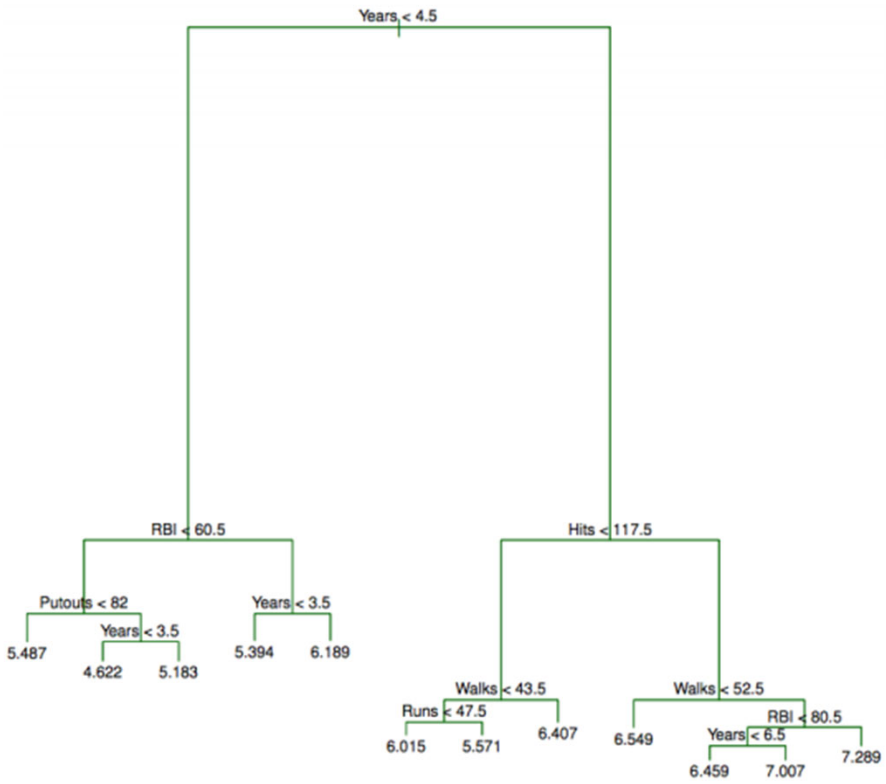
- Lesson
  - Trees
    - Dividing Data by Categorical and Numerical Features
    - Determining Partitions of the Data
  - Random Forests
  - Preventing Overfitting
    - Using Bootstrap with Random Forests
- Demo
  - Handling Missing Data
  - Working with Categorical Variables

Objectives

- **How can a tree be used for classification and regression?**
  - **How should we assess partitions of the data by features?**
- How can we combine predictions from trees to increase precision?
  - Would the same approach be useful for other models?
- **Readings:**
  - Hastie, Tibshirani, Friedman 9.2
  - James, Witten, Hastie, Tibshirani 8.1
  - Murphy 16.2
  - Geron 6

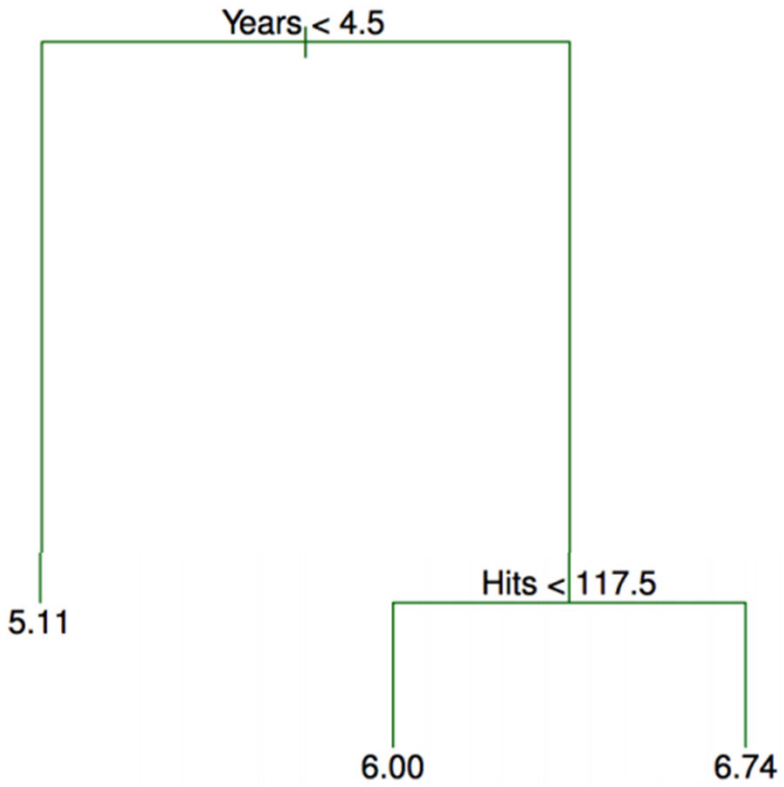
Trees

- When to stop splitting the tree into branches?
  - All the labels are the same
  - All the features are the same
  - ~~Information gain is small~~
- How to prevent against overfitting?
  - Minimum number of data for each leaf
  - Maximum depth of the tree
    - Early Stopping
  - Pruning
    - Backward Feature Selection



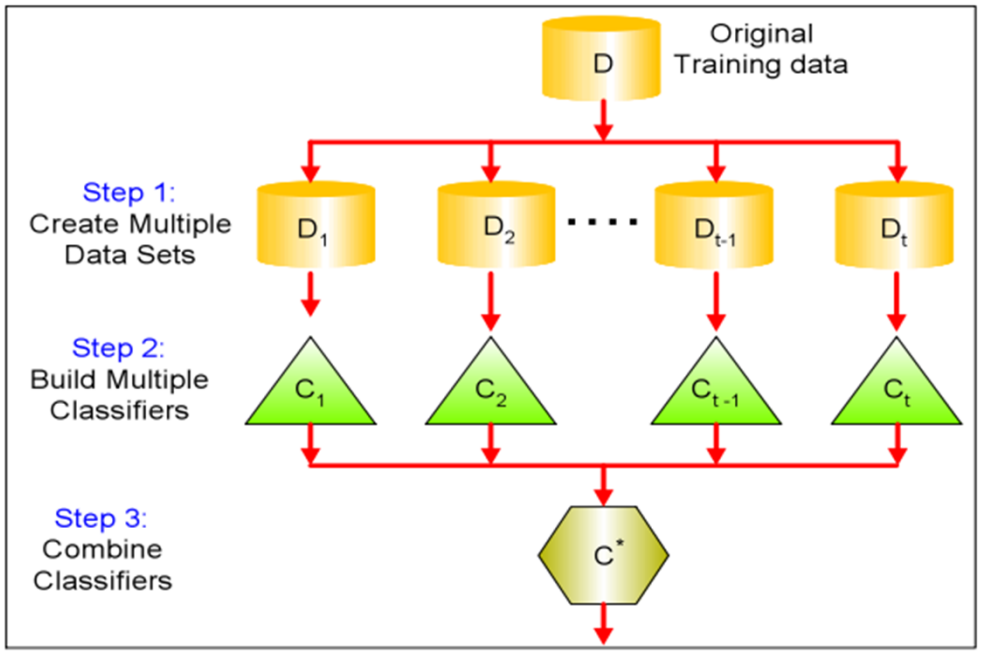
Trees

- When to stop splitting the tree into branches?
  - All the labels are the same
  - All the features are the same
  - ~~Information gain is small~~
- How to prevent against overfitting?
  - Minimum number of data for each leaf
  - Maximum depth of the tree
    - Early Stopping
  - Pruning
    - Backward Feature Selection



Trees

- Combining Trees
  - Averaging reduces Variance
  - However we usually have single training set
    - Bootstrap Resampling
  - Bagging
    - Random Forest to mimic independent samples



Summary

- Why use Trees?
  - Intuitive and Interpretable
    - Mimics steps to make decisions
  - Have low approximation error
    - Overfitting an Issue
  - Works with both regression and classification data
    - Prediction function for regression is discontinuous
  - Fits complex, nonlinear boundaries without feature engineering
    - Scale of features is not relevant. No distance in calculation

Larger trees are less interpretable than smaller trees

Summary

- Why use Trees?
  - Intuitive and Interpretable
    - Mimics steps to make decisions
  - Have low approximation error
    - Overfitting an Issue
  - Works with both regression and classification data
    - Prediction function for regression is discontinuous

Can reduce overfitting by restricting the size of the tree…both the width and the height….

- Fits complex, nonlinear boundaries without feature engineering
  - Scale of features is not relevant. No distance in calculation

Summary

- Why use Trees?
  - Intuitive and Interpretable
    - Mimics steps to make decisions
  - Have low approximation error
    - Overfitting an Issue
  - Works with both regression and classification data
    - Prediction function for regression is discontinuous

…imagine splitting by the row number in a dataset

- Fits complex, nonlinear boundaries without feature engineering
  - Scale of features is not relevant. No distance in calculation

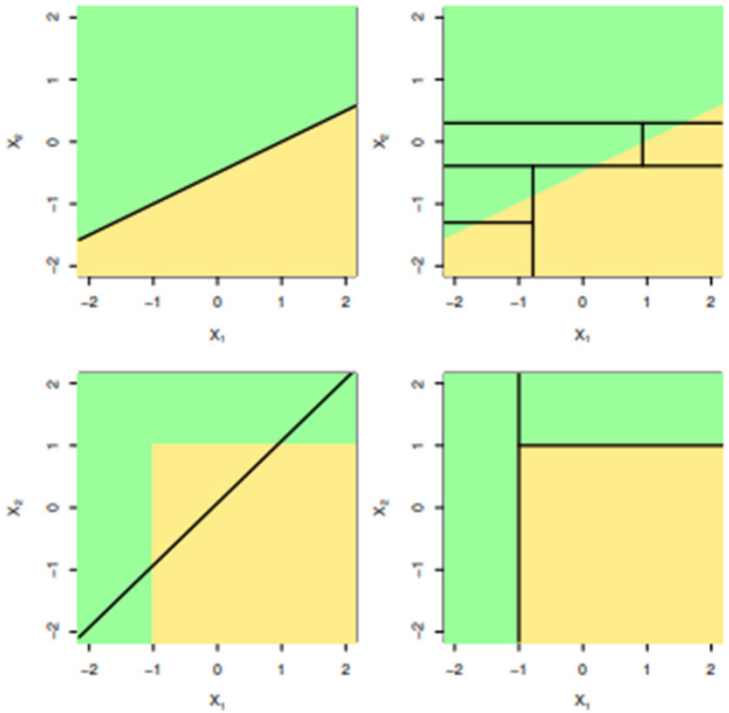
Summary

- Why use Trees?
  - Intuitive and Interpretable
    - Mimics steps to make decisions
  - Have low approximation error
    - Overfitting an Issue
  - Works with both regression and classification data
    - Prediction function for regression is discontinuous
- Fits complex, nonlinear boundaries without feature engineering
  - Scale of features is not relevant. No distance in calculation

Compare to Demo 12 where approximation of target function by step functions

Summary

- Why use Trees?
  - Intuitive and Interpretable
    - Mimics steps to make decisions
  - Have low approximation error
    - Overfitting an Issue
  - Works with both regression and classification data
    - Prediction function for regression is discontinuous



- Fits complex, nonlinear boundaries without feature engineering
  - Scale of features is not relevant. No distance in calculation

Summary

- Why use Trees?
  - Intuitive and Interpretable
    - Mimics steps to make decisions
  - Have low approximation error
    - Overfitting an Issue
  - Works with both regression and classification data
    - Prediction function for regression is discontinuous
- Fits complex, nonlinear boundaries without feature engineering
  - Scale of features is not relevant. No distance in calculation

However tree can be used for feature engineering…