Decision Trees and Random Forests

Ivan Corneillet

Data Scientist

GENERAL ASSEMBLY

Learning Objectives

After this lesson, you should be able to:

- Understand and build decision tree models for classification and regression
- Understand the differences between linear and non-linear models
- Understand and build random forest models for classification and regression
- Know how to extract the most important predictors in a random forest model

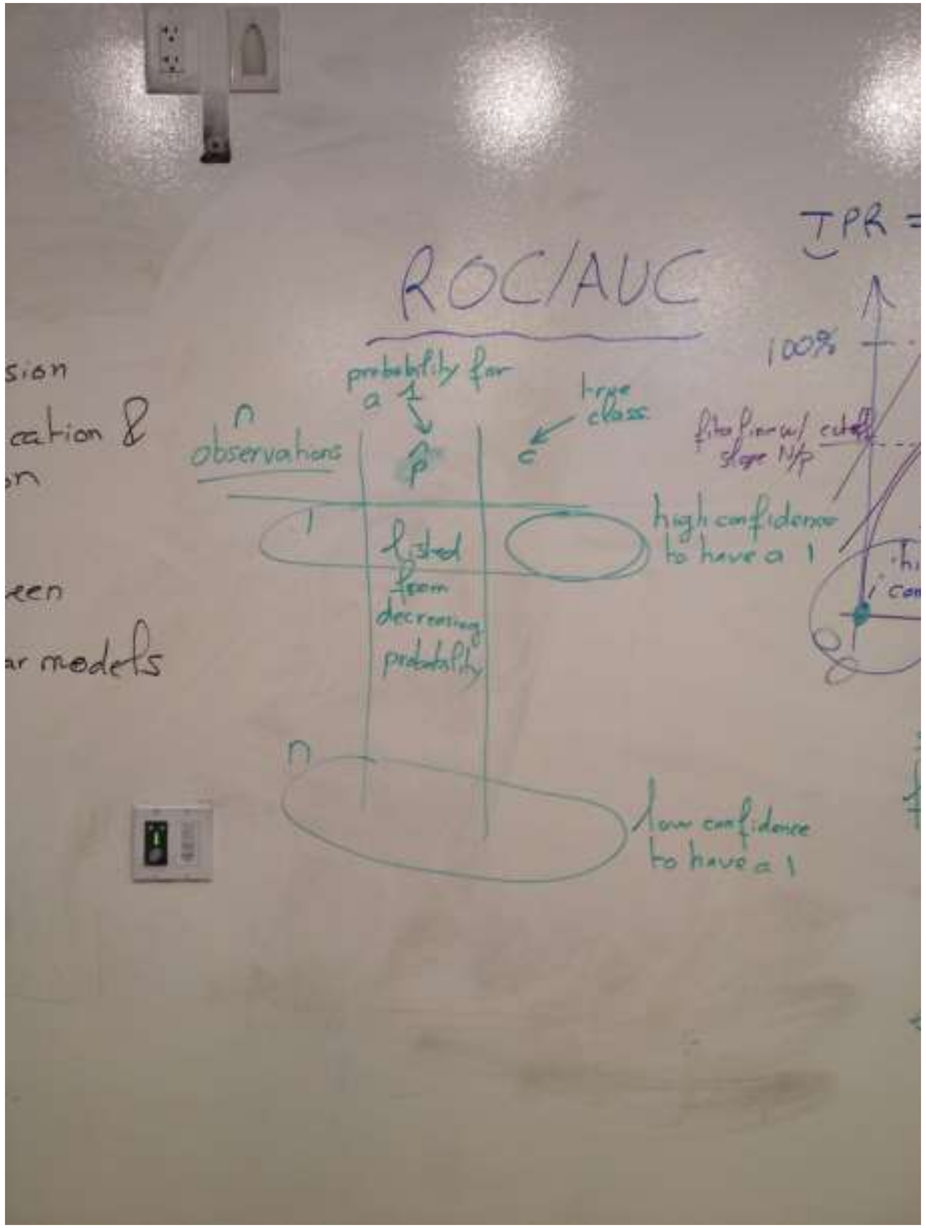
Announcements and Exit Tickets

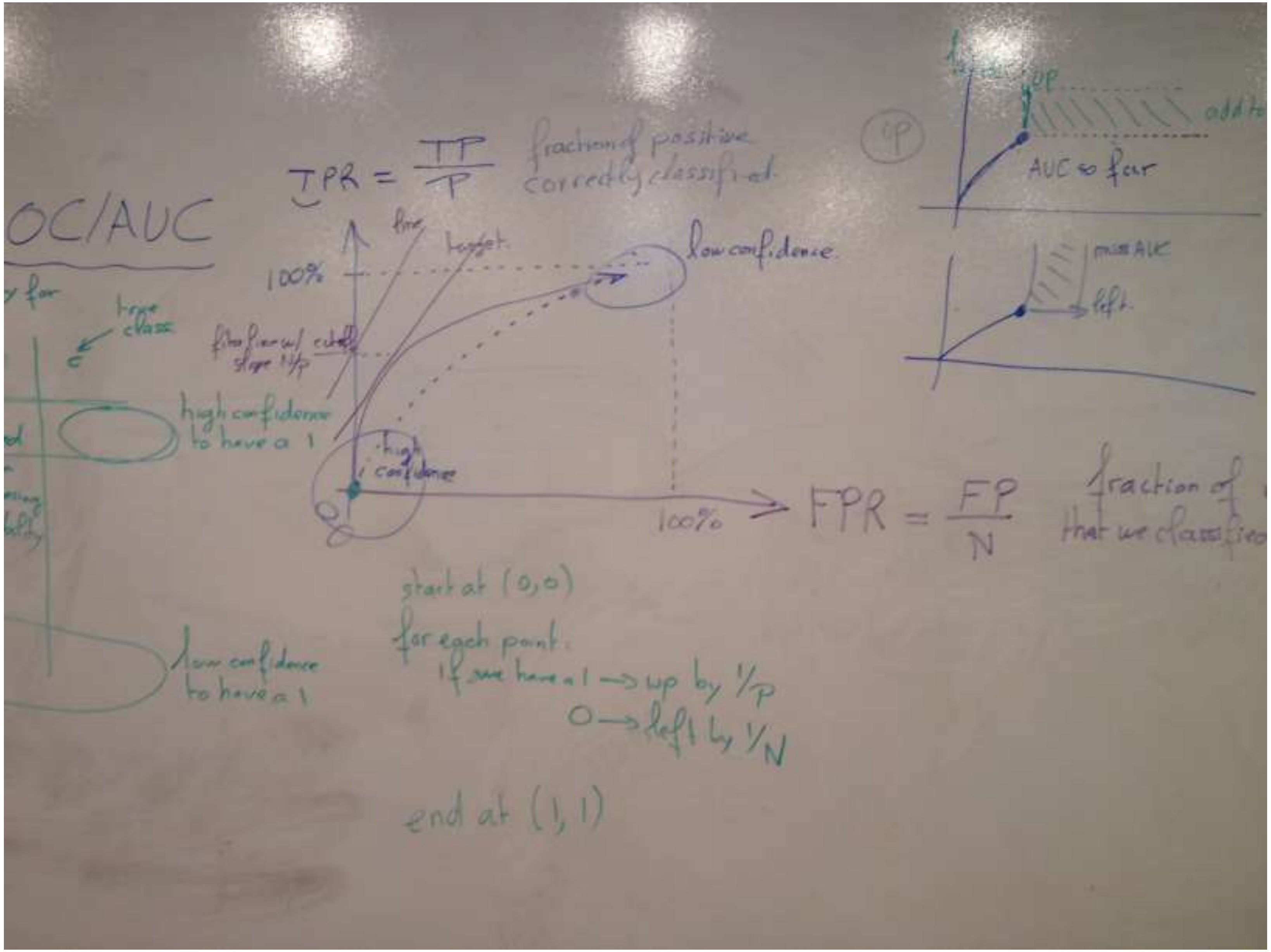
Q & A

Review

Review

ROC/AUC

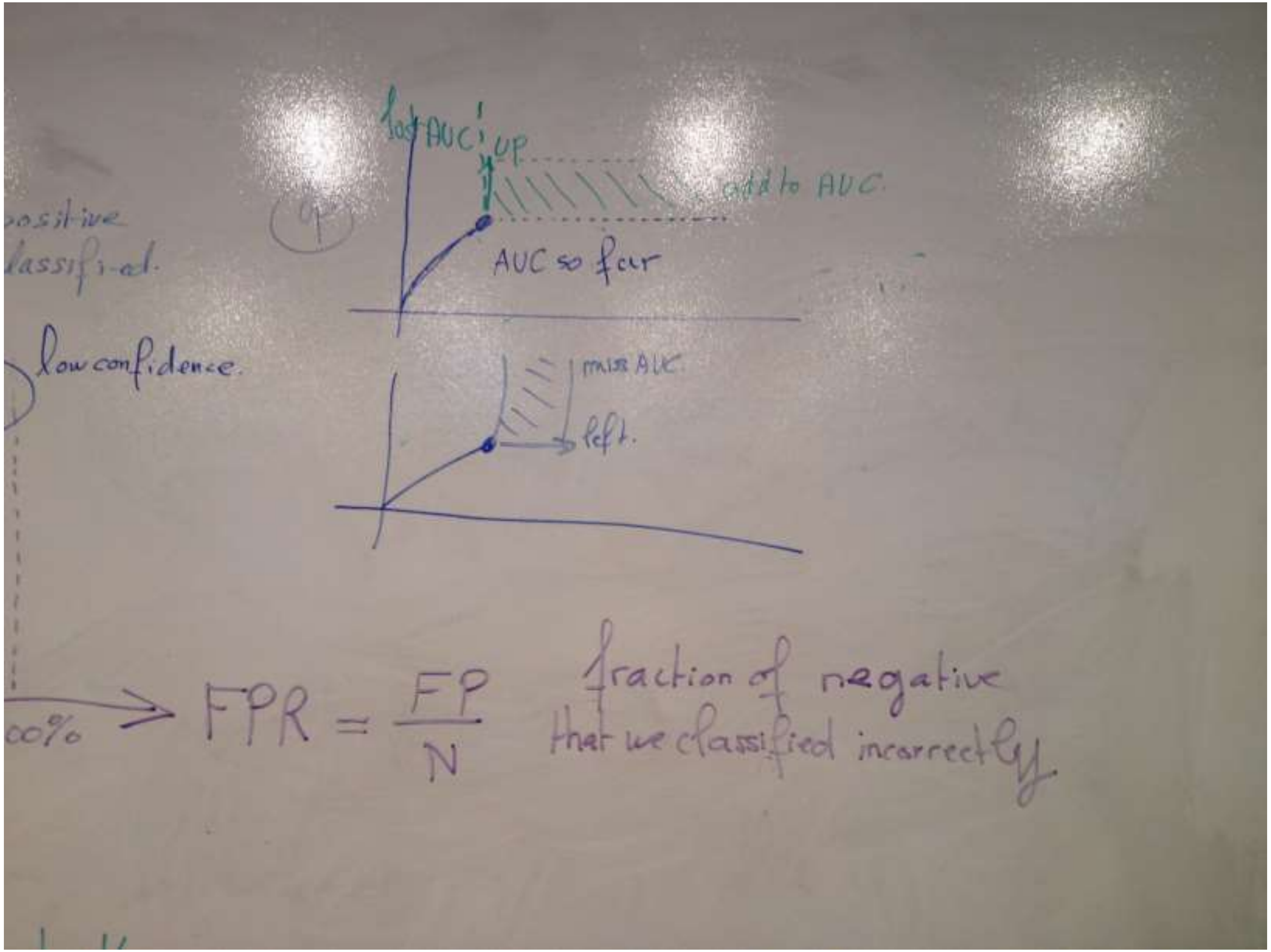

ROC/AUC
probability for a 1
true class
n
observations
$\hat{p}$
c
1
listed from decreasing probability
high confidence to have a 1
n
low confidence to have a 1
TPR =
100%
fit a line w/ cutoff slope N/P
sion
cation &
on
een
ar models



$TPR = \frac{TP}{P}$ fraction of positive correctly classified

$FPR = \frac{FP}{N}$ fraction of that we classified

```
start at (0,0)
for each point:
    if we have a 1 → up by 1/P
                 0 → left by 1/N
end at (1,1)
```



$FPR = \frac{FP}{N}$ fraction of negative that we classified incorrectly

Today

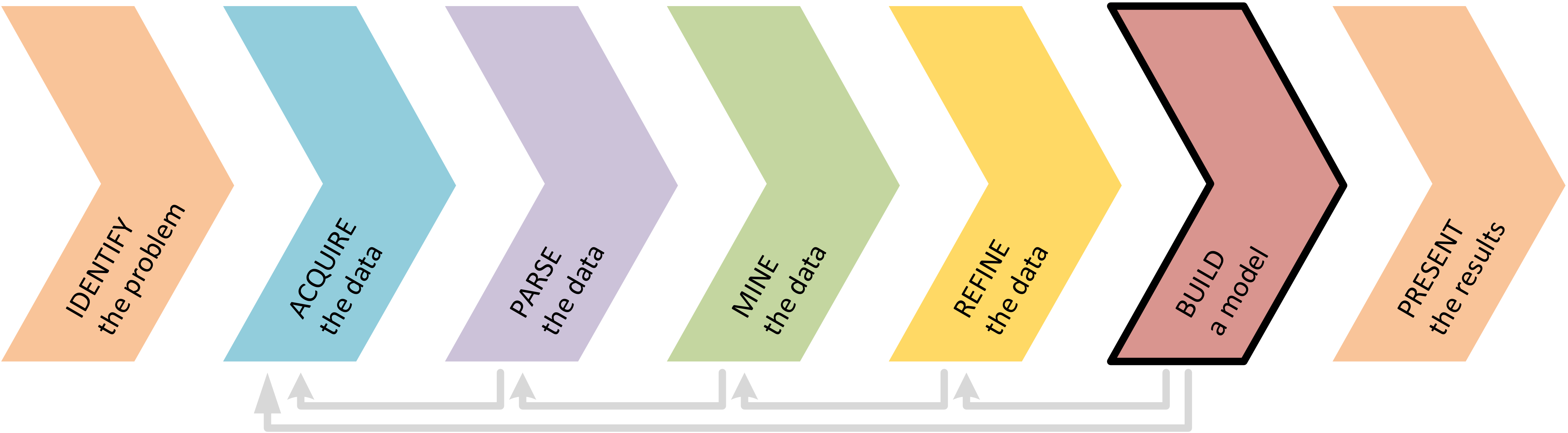
Here’s what’s happening today:

- Final Project 3 (due today)
- Announcements and Exit Tickets
- Review
- ❻ Build a Model | Decision Trees
    - The 2008 Democratic Primaries
    - What is a decision tree?
    - Structure
    - Predicting
    - Entropy
    - Training
    - Classification and Regression Decision Trees
    - Pros and Cons
    - Overfitting
- ❻ Build a Model | Random Forests
    - Pros and Cons
    - Training
    - Predicting
- Lab – Decision trees and random forests
- Review

Today, we are starting Unit 3 by introducing two new machine learning algorithms: first Decision Trees; then Random Forests (which will build upon decision trees)

Research Design and Data Analysis	Research Design	Data Visualization in *pandas*	Statistics	Exploratory Data Analysis in *pandas*
Foundations of Modeling	Linear Regression	Classification Models	Evaluating Model Fit	Presenting Insights from Data Models
Data Science in the Real World	Decision Trees and Random Forests	Time Series Data	Natural Language Processing	Databases

Therefore, today we are refocusing on BUILD a model step but with a focus on decision trees and random forests



Pre-Work

Pre-Work

Before this lesson, you should already be able to:

- Explain the concepts of cross-validation, logistic regression, and overfitting
- Know how to build and evaluate some classification models in *sklearn* using cross-validation and AUC

DS

Decision Trees

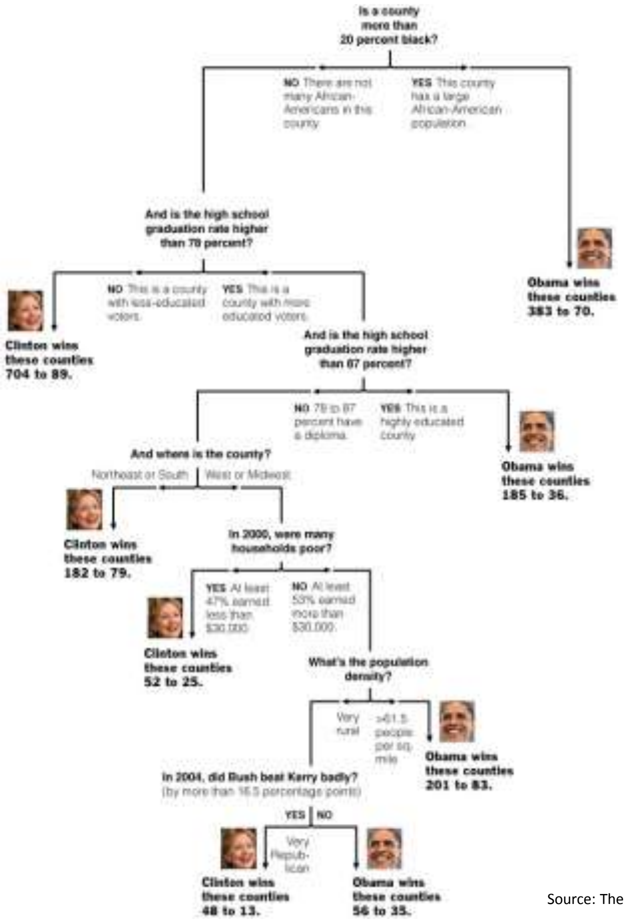
Decision Trees

Motivating Example and Activity | The 2008 Democratic Primaries

Motivating Example | The 2008 Democratic Primaries

EXAMPLE

- Decision Tree: The Obama-Clinton Divide
  - Published in The New York Time on April 16, 2008 while the Democratic Primaries were still running



Is a county more than 20 percent black?
NO There are not many African-Americans in this county
YES This county has a large African-American population
And is the high school graduation rate higher than 78 percent?
Obama wins these counties 383 to 70.
NO This is a county with less-educated voters.
YES This is a county with more educated voters.
Clinton wins these counties 704 to 89.
And is the high school graduation rate higher than 87 percent?
NO 78 to 87 percent have a diploma.
YES This is a highly educated county.
Obama wins these counties 185 to 36.
And where is the county?
Northeast or South
West or Midwest
Clinton wins these counties 182 to 79.
In 2000, were many households poor?
YES At least 47% earned less than $30,000
NO At least 53% earned more than $30,000
Clinton wins these counties 52 to 25.
What's the population density?
Very rural
>61.5 people per sq. mile
Obama wins these counties 201 to 83.
In 2004, did Bush beat Kerry badly? (by more than 16.5 percentage points)
YES
NO
Very Repub-lican
Clinton wins these counties 48 to 13.
Obama wins these counties 56 to 35.

Source: The New York Times

Activity | The 2008 Democratic Primaries

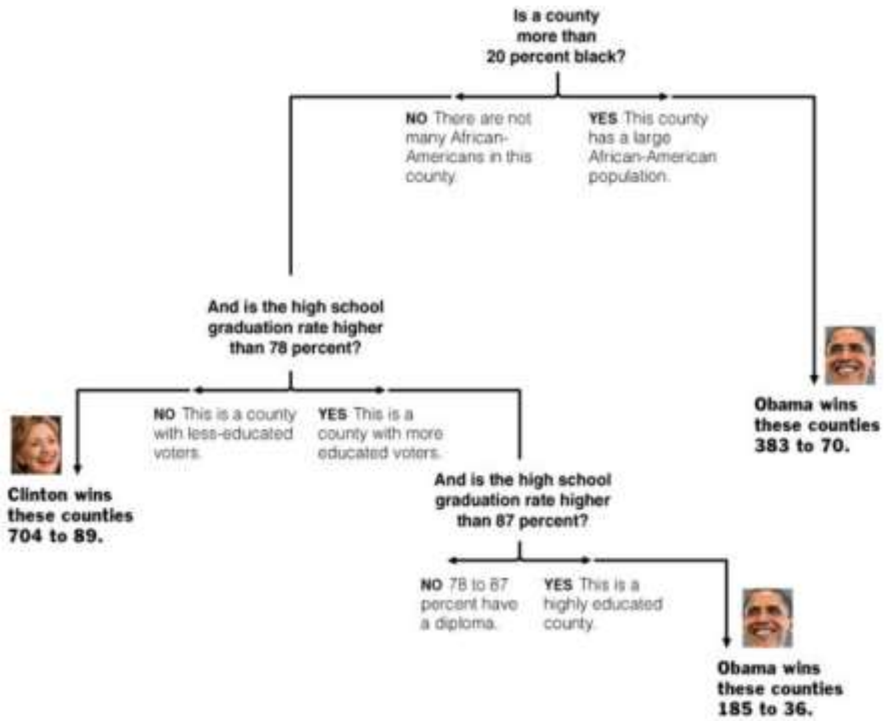
EXERCISE

DIRECTIONS (10 minutes)

1. In a couple of sentences, describe what counties senators Obama and Clinton have won so far in the nomination contest
2. Amanda Cox, a data scientist of The New York times created this excellent graphics. How do you think she proceeded to create it?
3. When finished, share your answers with your table

DELIVERABLE

Answers to the above questions



Is a county more than 20 percent black?

NO There are not many African-Americans in this county.

YES This county has a large African-American population.

And is the high school graduation rate higher than 78 percent?

NO This is a county with less-educated voters.

YES This is a county with more educated voters.

Clinton wins these counties 704 to 89.

Obama wins these counties 383 to 70.

And is the high school graduation rate higher than 87 percent?

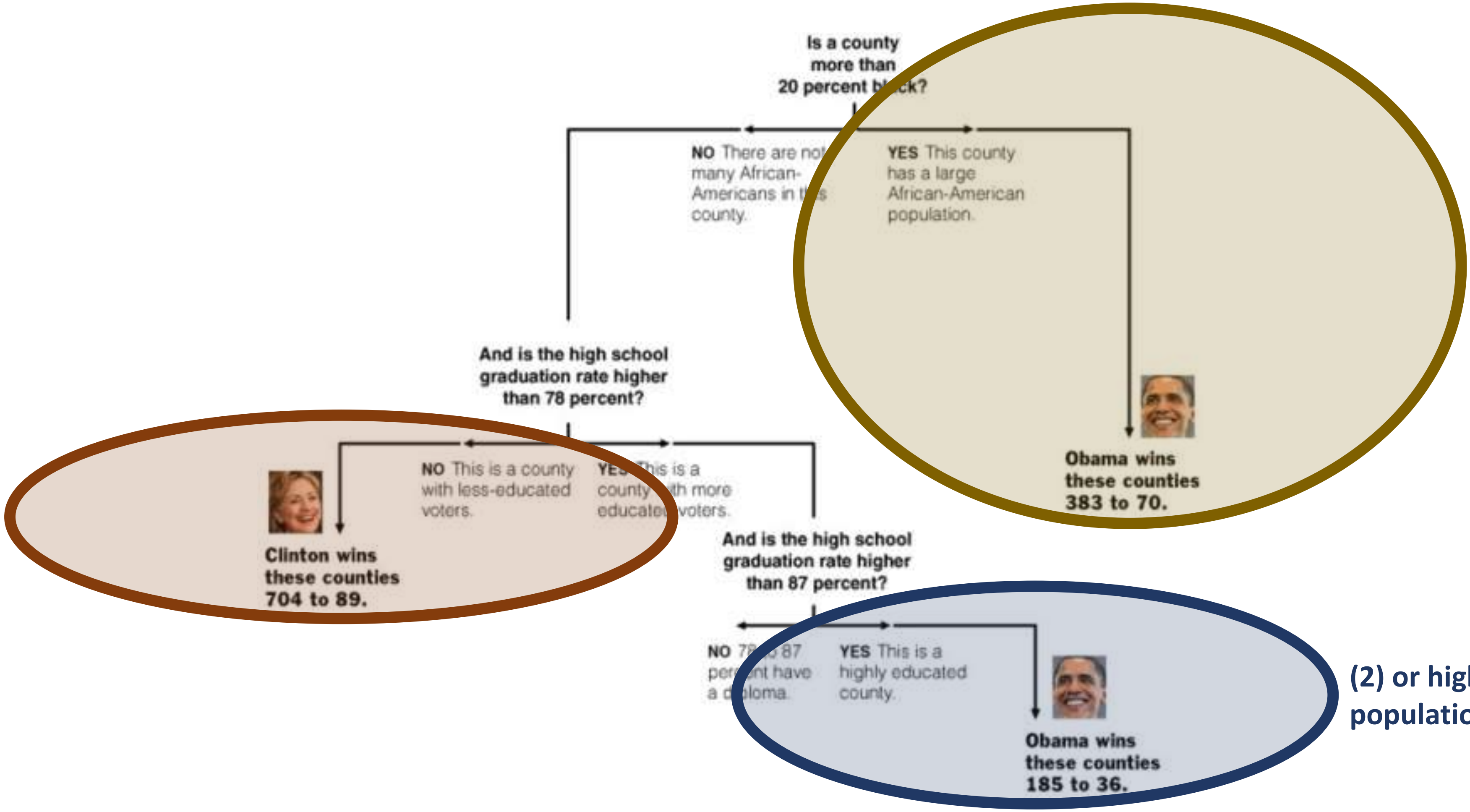
NO 78 to 87 percent have a diploma.

YES This is a highly educated county.

Obama wins these counties 185 to 36.

Activity | "In the nominating contests so far, …

(3) Senator Hillary Rodham Clinton has a commanding lead in less-educated counties dominated by whites"



Is a county more than 20 percent black?

NO There are not many African-Americans in this county.

YES This county has a large African-American population.

Obama wins these counties 383 to 70.

And is the high school graduation rate higher than 78 percent?

NO This is a county with less-educated voters.

YES This is a county with more educated voters.

Clinton wins these counties 704 to 89.

And is the high school graduation rate higher than 87 percent?

NO 78 to 87 percent have a diploma.

YES This is a highly educated county.

Obama wins these counties 185 to 36.

(1) Senator Barack Obama has won the vast majority of counties with large black populations

(2) or highly educated populations.

Activity | The 2008 Democratic Primaries (cont.)

- "In the nominating contests so far, Senator Barack Obama has won the vast majority of counties with large black or highly educated populations. Senator Hillary Rodham Clinton has a commanding lead in less-educated counties dominated by whites"

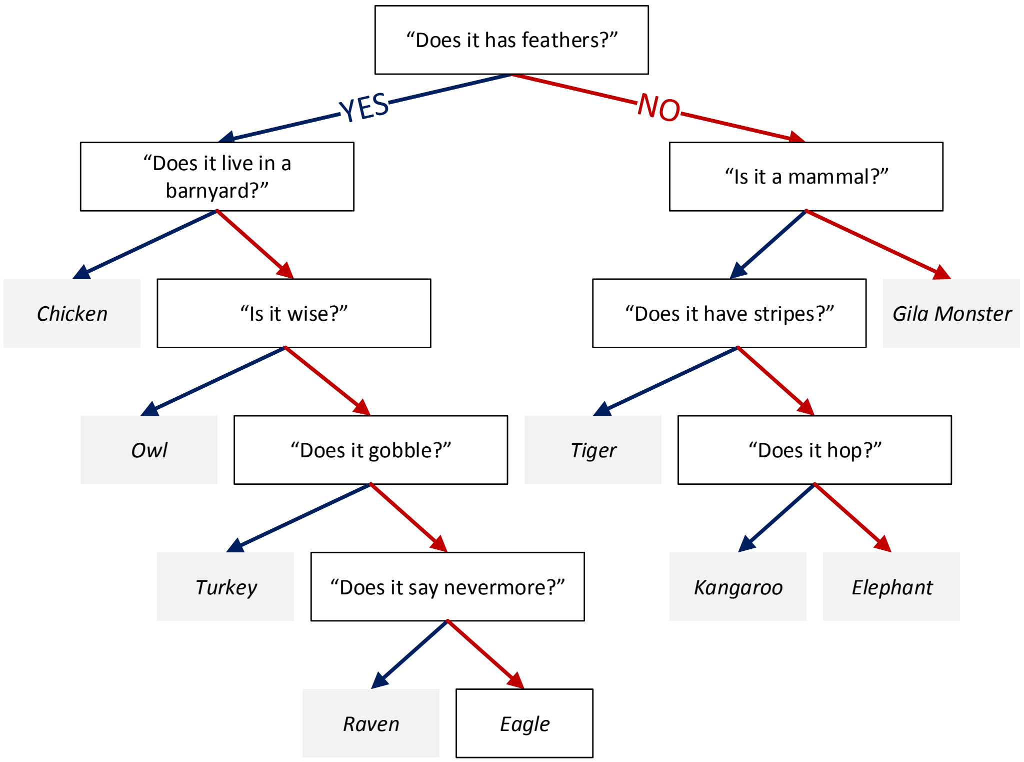
- Surely an over-simplification but it is easy to display, interpret, and explain

DS

Decision Trees

What is a decision tree?

Decision trees are like the game "20 questions." They make decision by answering a series of questions, most often binary questions. (yes or no) (cont.)

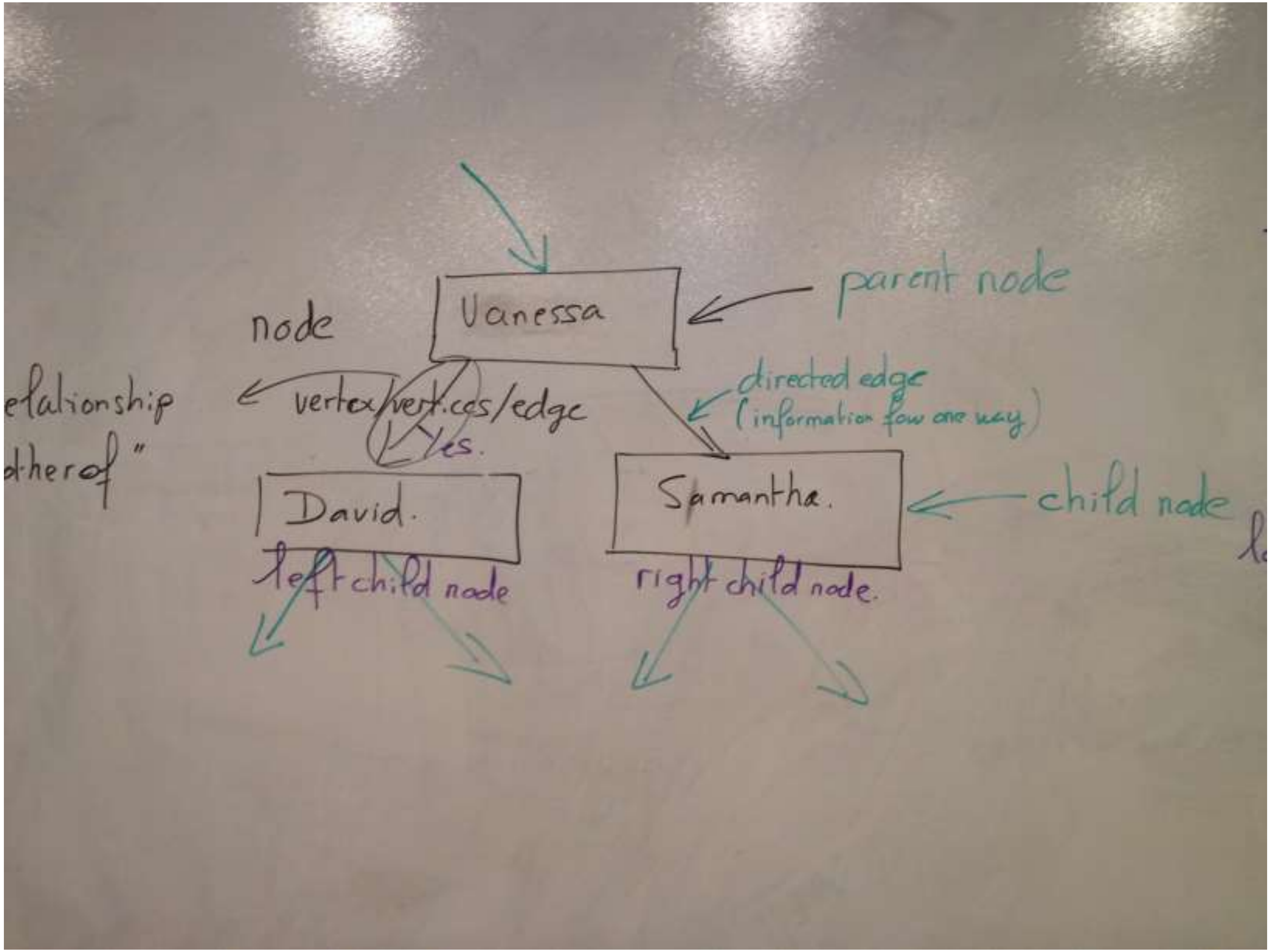


- We want the smallest set of questions to get to the right answer
- Each questions should reduce the search space as much as possible

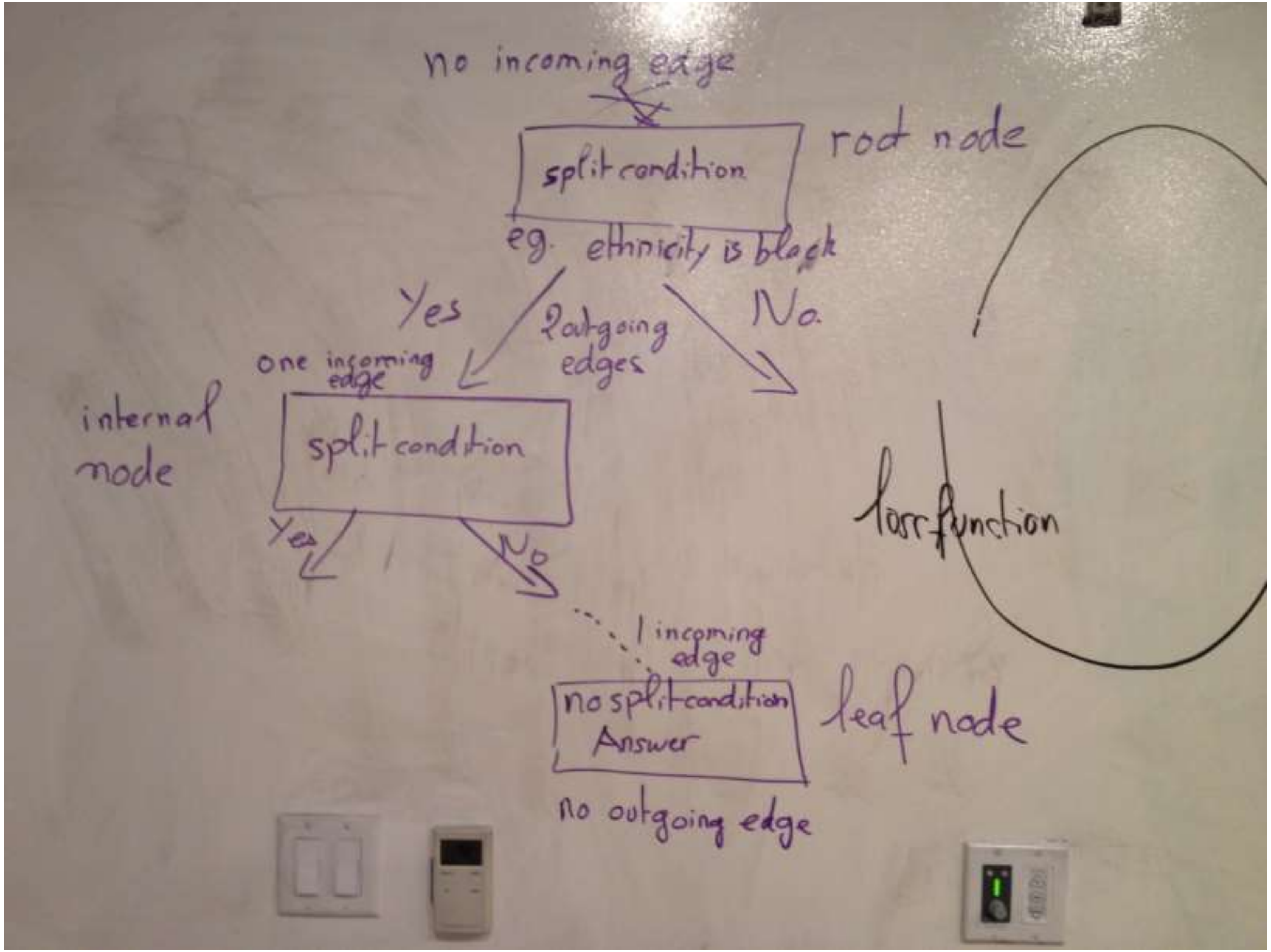
DS

Decision Trees

Structure

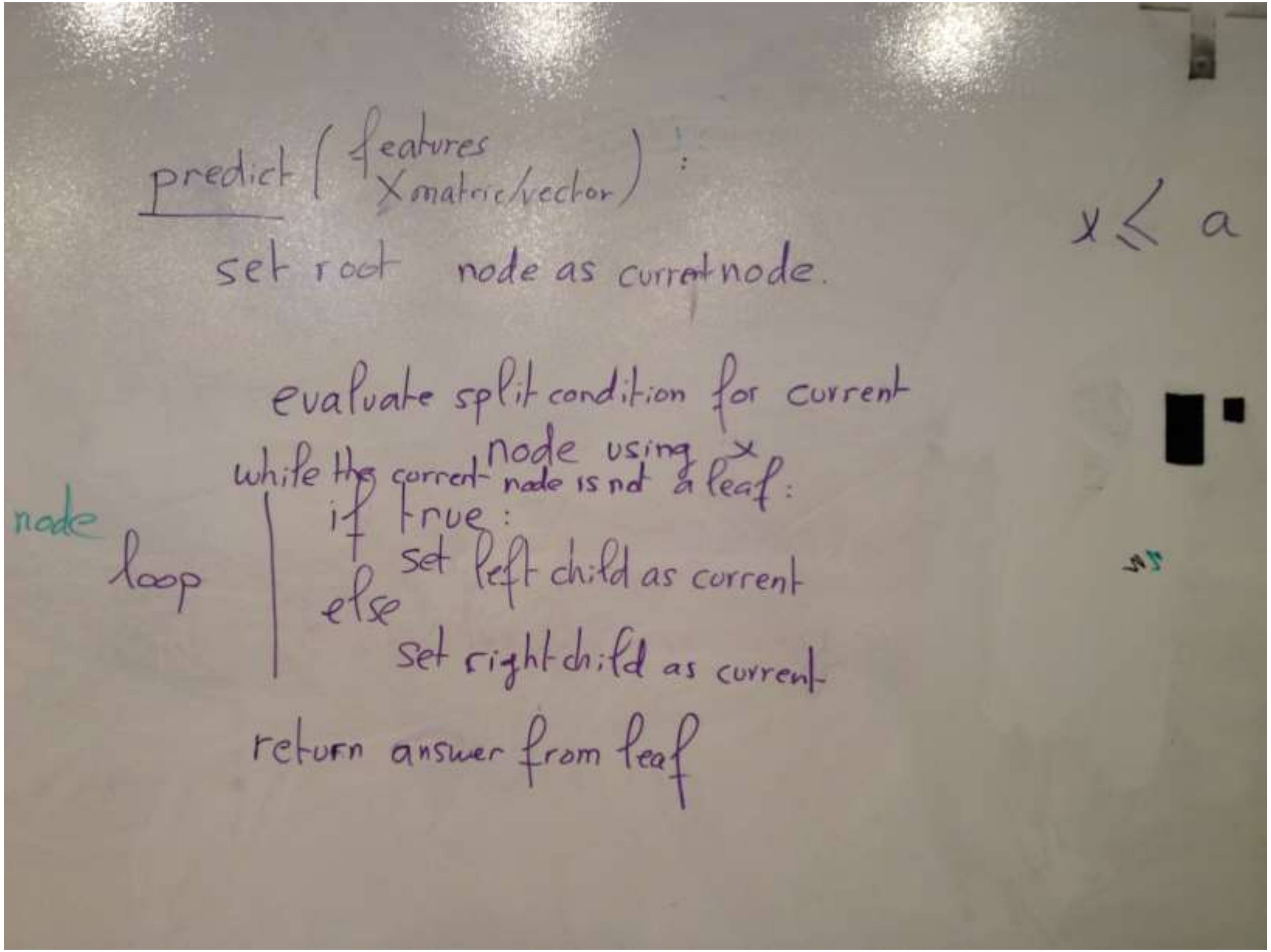


parent node
Vanessa
node
elationship
dherof"
vertex/vertices/edge
Yes.
directed edge
(information flow one way)
David.
Samantha.
child node
l
left child node
right child node.


no incoming edge
split condition
root node
eg. ethnicity is black
Yes
2 outgoing edges
No.
one incoming edge
internal node
split condition
Yes
No
1 incoming edge
no split condition
Answer
leaf node
no outgoing edge
loss function

Decision Trees

Predicting



```
predict( features
         X matric/vector ) :

    set root node as current node.

        evaluate split condition for current
                 node using x
    while the current node is not a leaf:
node
   loop |    if true:
        |       set left child as current
        |    else
        |       set right child as current

    return answer from leaf
```

$x \leq a$

DS

Using decision trees to make predictions is great but how do we build them in the first place?

Open questions from the previous activity

- ❶ How to choose the split conditions? (variables and threshold values)
  - E.g., why is the threshold for African-American population set at 20%?
- ❷ How to choose the order of the conditions?
  - E.g., why is the first split on African-American population vs. the voters' education level?
- ❸ When to stop?
  - E.g., why didn’t we include other factors such as voters’ age?

Because it isn't computationally feasible to consider every possible partition of the feature space, we take a *top-down*, *greedy* approach known as recursive binary splitting

Top-Down

- The approach begins at the top of the tree and then successively splits the predictor space; each split is indicated via two new branches further down on the tree

Greedy

- At each step of the tree-building process, the *best* split is made at that particular step, rather than looking ahead and picking a split that will lead to a better tree in some future step

Decision trees can be applied to both classification and regression problems

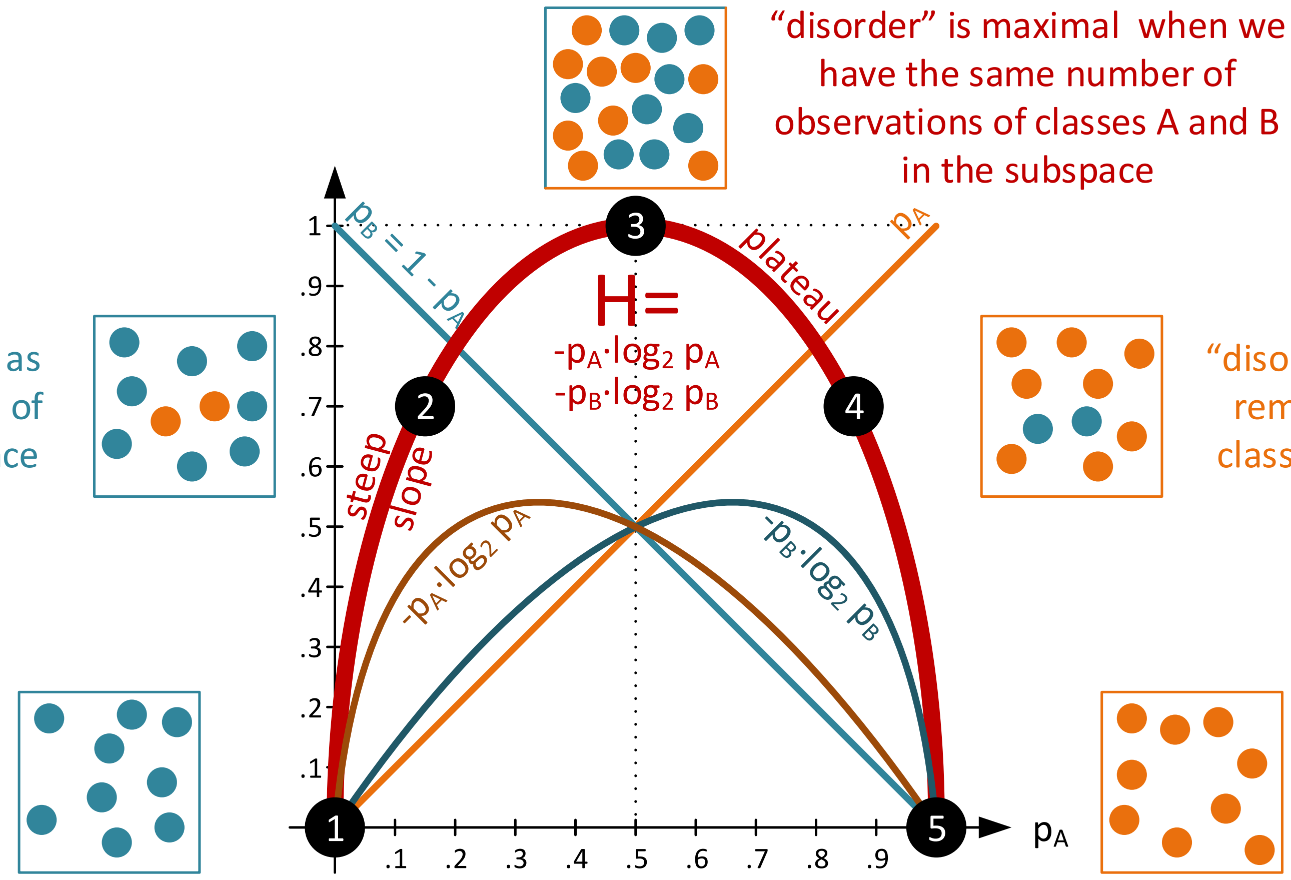
- We first consider classification problems to address ❷ (How to choose the order of the conditions?)

- We'll then move on to regression problems when addressing ❶ (How to choose the split conditions?)

❻ Build a Model

Entropy

Entropy (H) is a measure of disorder



“disorder” is maximal when we have the same number of observations of classes A and B in the subspace

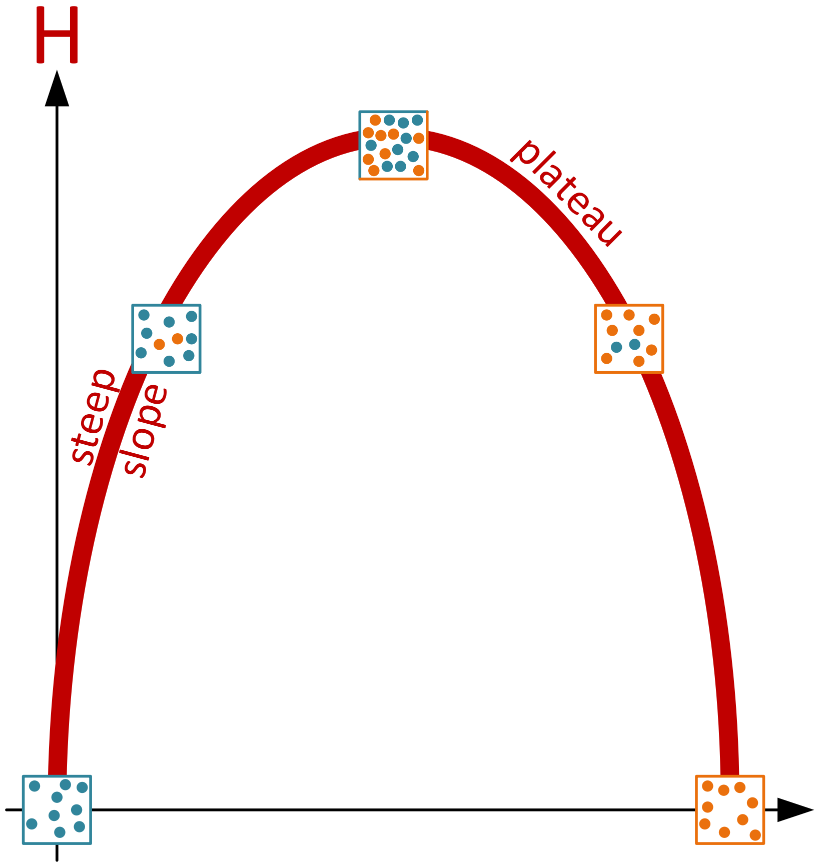
“disorder” increases as we add observations of class A in the subspace

“disorder” decreases as we remove observations of class B from the subspace

“disorder” is minimal when subspace is only made of class B

“disorder” is minimal again when subspace is only made of class A

Entropy | What to remember



$$H = -\sum_{i=1}^{k} p_i \cdot log_2(p_i)$$

(p_i represents the proportion of observations in the region that are from the i^{th} class)

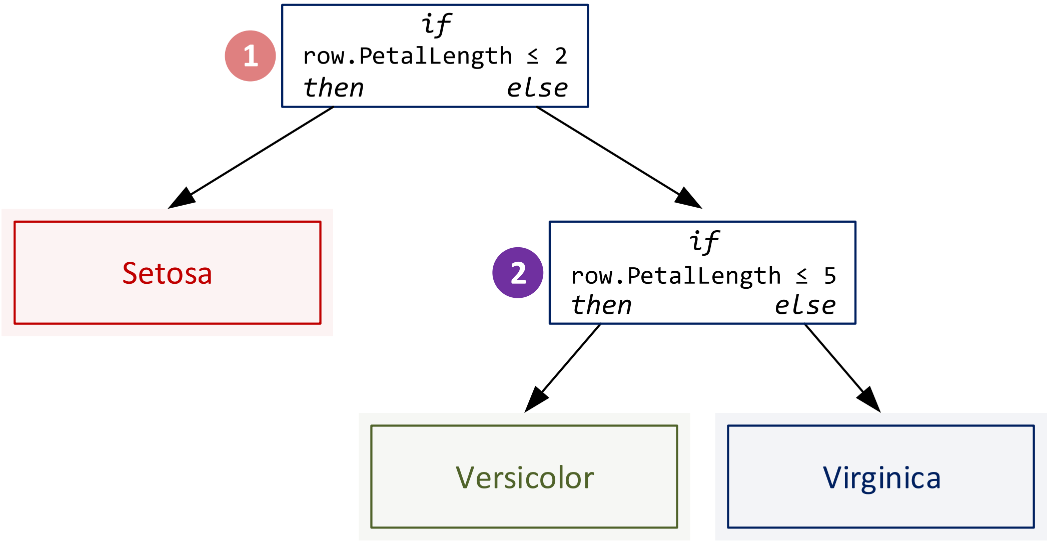
Decision Trees

Training a Classification Decision Tree

❷ *How to choose the order of the conditions?*

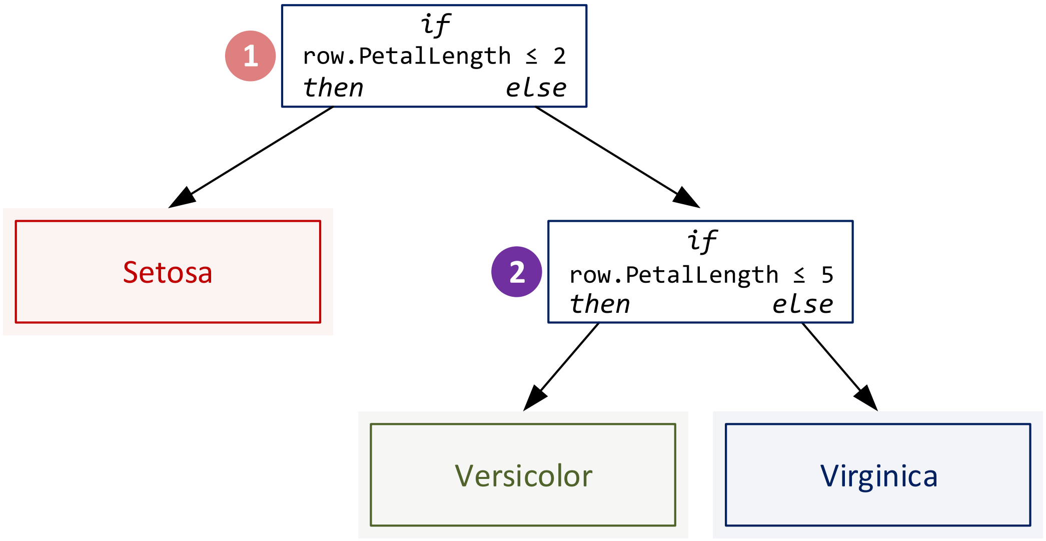
Our first classifiers (session 8) are decision trees

```python
def my_second_classifier(row):
  if row.PetalLength <= 2:
    return 'Setosa'
  elif row.PetalLength <= 5:
    return 'Versicolor'
  else:
    return 'Virginica'
```



Our first classifiers (session 8) are decision trees (cont.)

Decision Tree



1 *if* row.PetalLength ≤ 2 *then* *else*

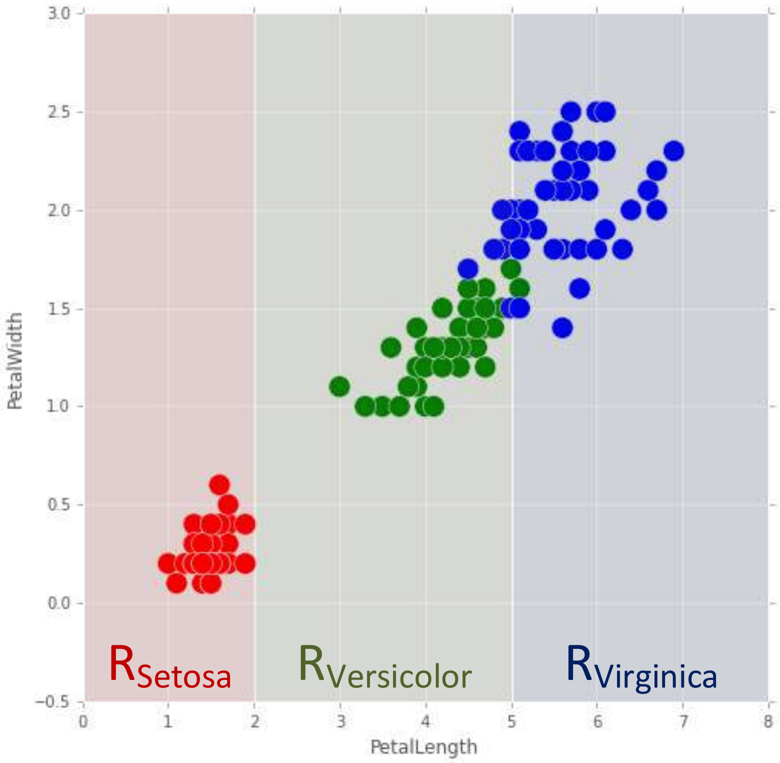
Setosa

2 *if* row.PetalLength ≤ 5 *then* *else*

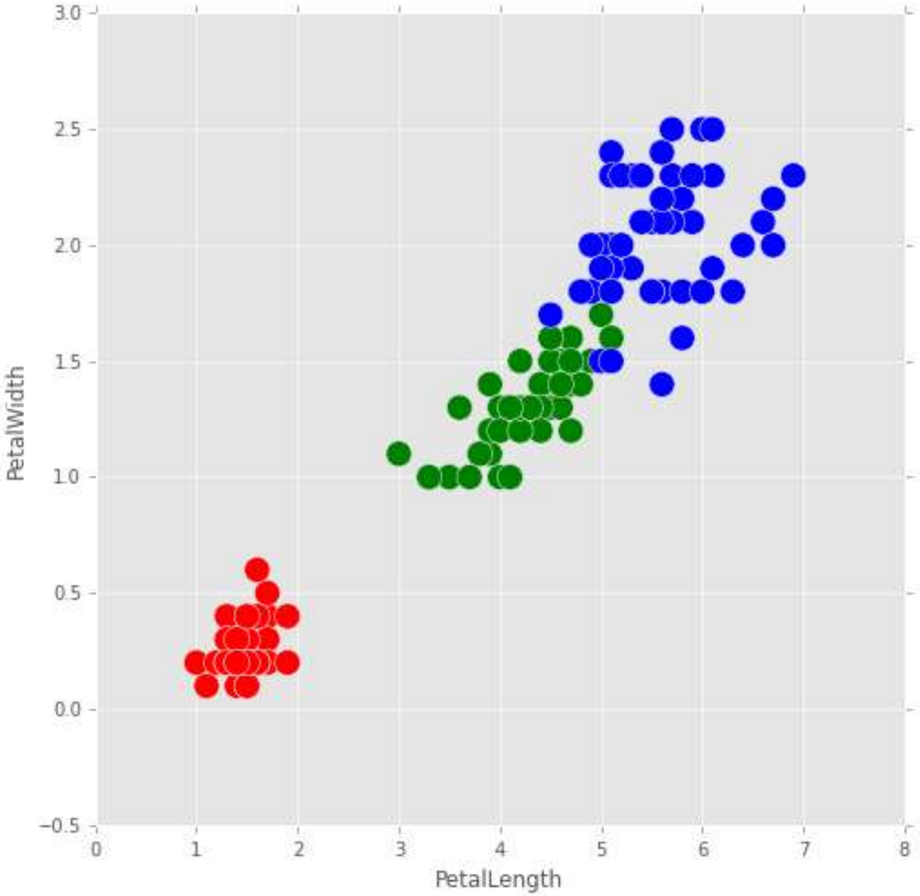
Versicolor

Virginica

Feature Space



⓿ What’s the entropy of the dataset/root node before the first split? What’s your intuition of its level?



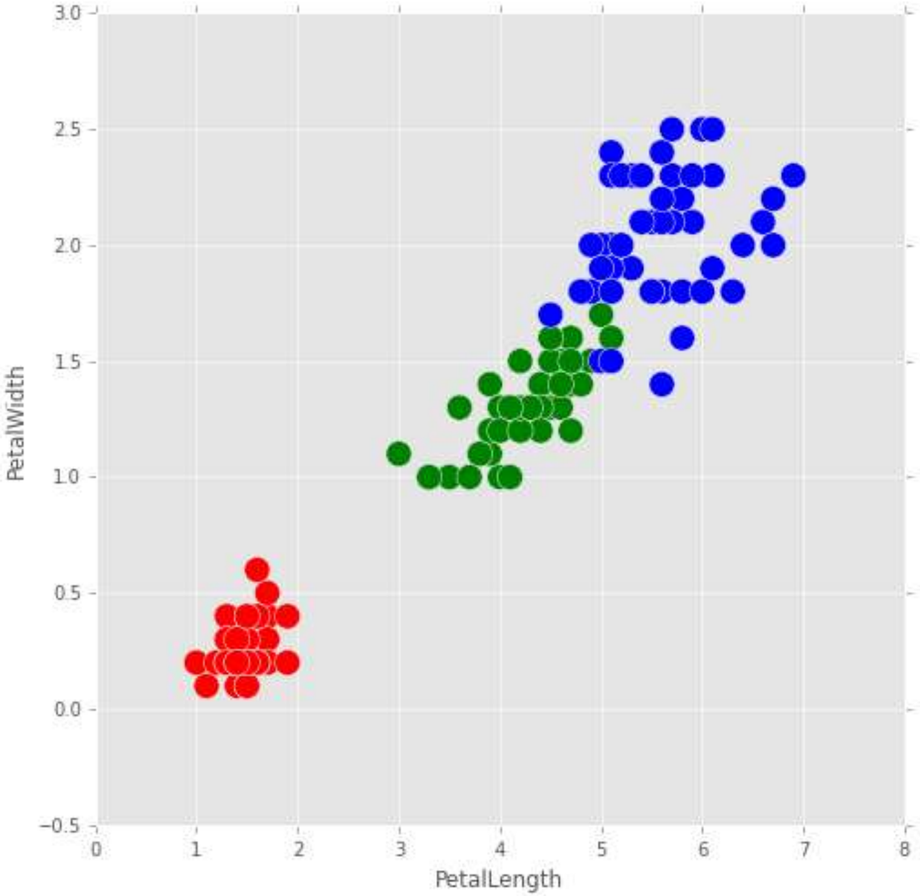
root node				
	red	green	blue	overall
c_i				
p_i				
H				

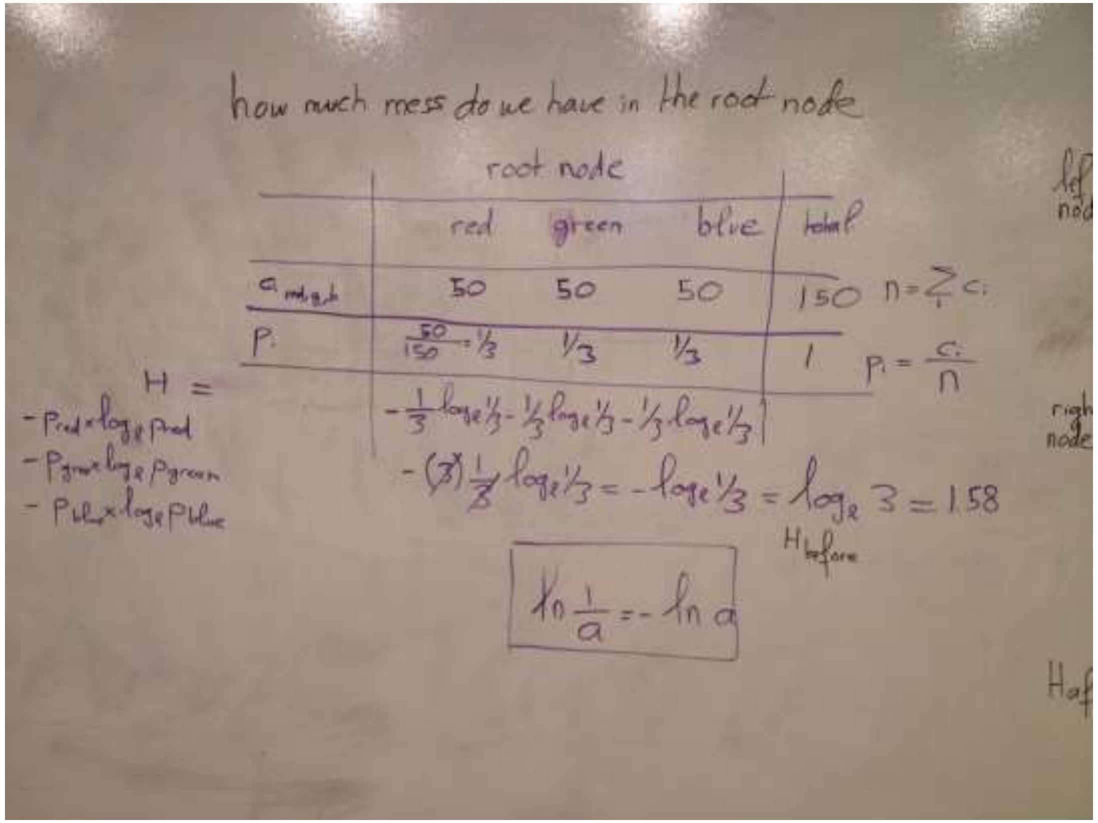
$$n = \sum_{i=1}^{k} c_i$$

$$p_i = \frac{c_i}{n}$$

$$H = -\sum_{i=1}^{k} p_i \cdot log_2(p_i)$$

⓿ The entropy of the dataset/root node is at the highest at 1.58; the intuition being that we have three classes in equal proportion





how much mess do we have in the root node

root node

	red	green	blue	total
c [illegible]	50	50	50	150
p_i	$\frac{50}{150} = 1/3$	$1/3$	$1/3$	1

$n = \sum_1 c_i$

$p_i = \frac{c_i}{n}$

$H =$

$- p_{red} \times \log_e p_{red}$

$- p_{green} \times \log_e p_{green}$

$- p_{blue} \times \log_e p_{blue}$

$-\frac{1}{3} \log_e 1/3 - 1/3 \log_e 1/3 - 1/3 \log_e 1/3$

$-(\cancel{3}) \frac{1}{\cancel{3}} \log_e 1/3 = -\log_e 1/3 = \log_e 3 = 1.58$

H_{before}

$$\boxed{\ln \frac{1}{a} = -\ln a}$$

lef nod

righ node

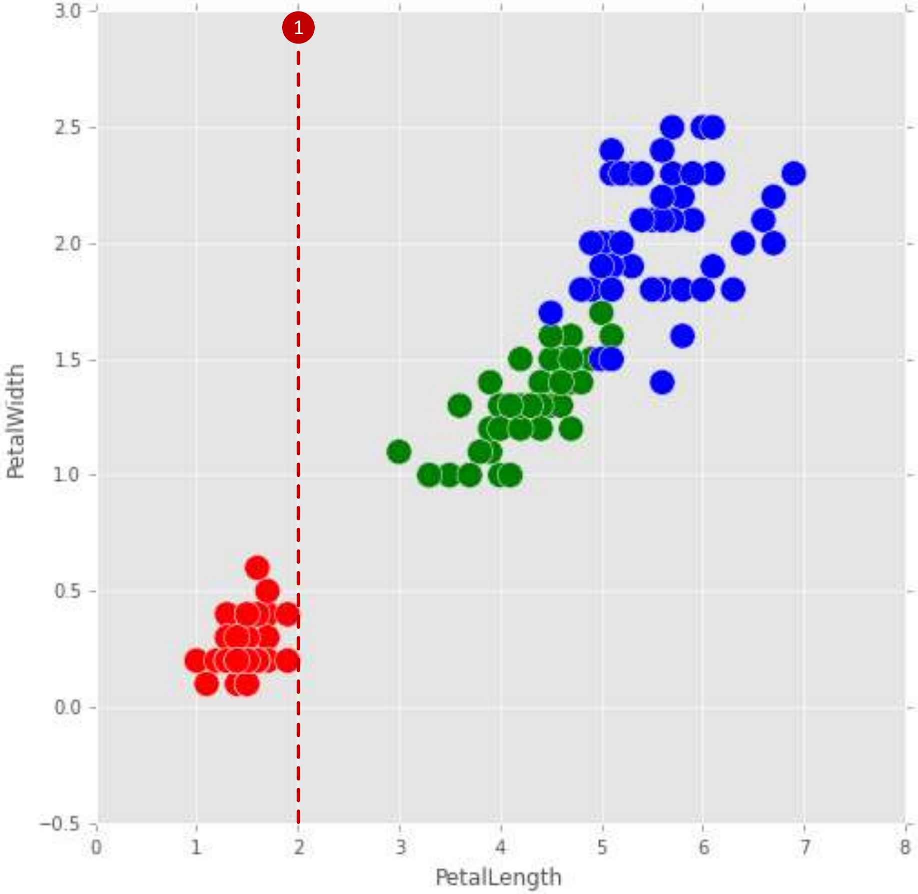
Haf

❶ What’s the entropy after split 1? Intuition?

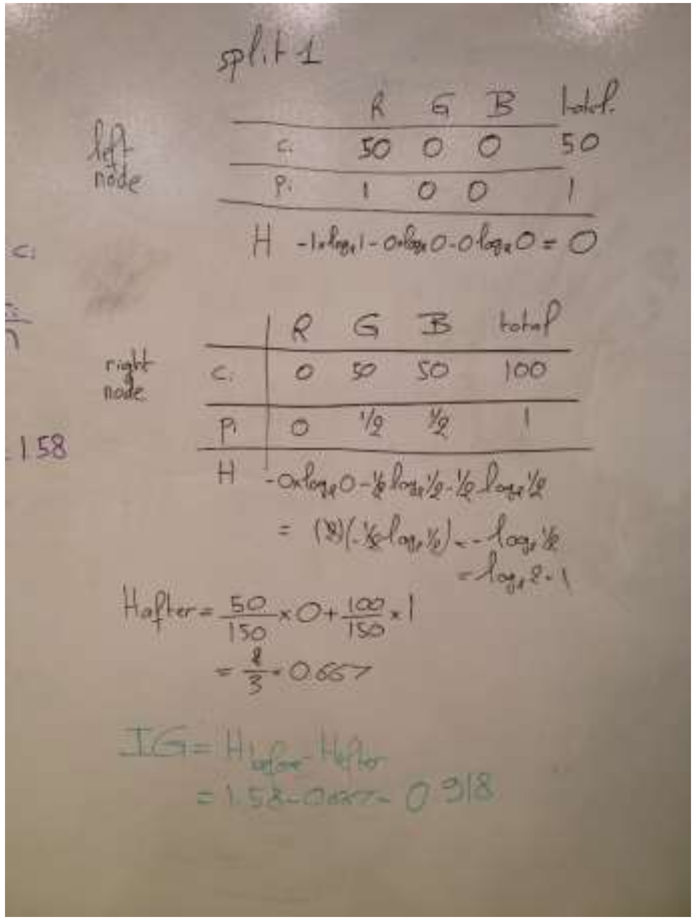
left node/lower subspace after split 1				
	red	green	blue	overall
c_i				
p_i				
H_{left}				

right node/higher subspace after split 1				
	red	green	blue	overall
c_i				
p_i				
H_{right}				

overall after split 1	
H_{after}	
IG	



The information gain is significant at .918. The subspace on the left is pure (0 entropy) bringing down significantly the weighted average entropy after the split



split 1

left node

	R	G	B	total
c_i	50	0	0	50
p_i	1	0	0	1

$H \quad -1 \times \log_2 1 - 0 \times \log_2 0 - 0 \log_2 0 = 0$

right node

	R	G	B	total
c_i	0	50	50	100
p_i	0	½	½	1

$H \quad -0 \times \log_2 0 - \frac{1}{2}\log_2 \frac{1}{2} - \frac{1}{2}\log_2 \frac{1}{2}$

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

$= \log_2 2 = 1$

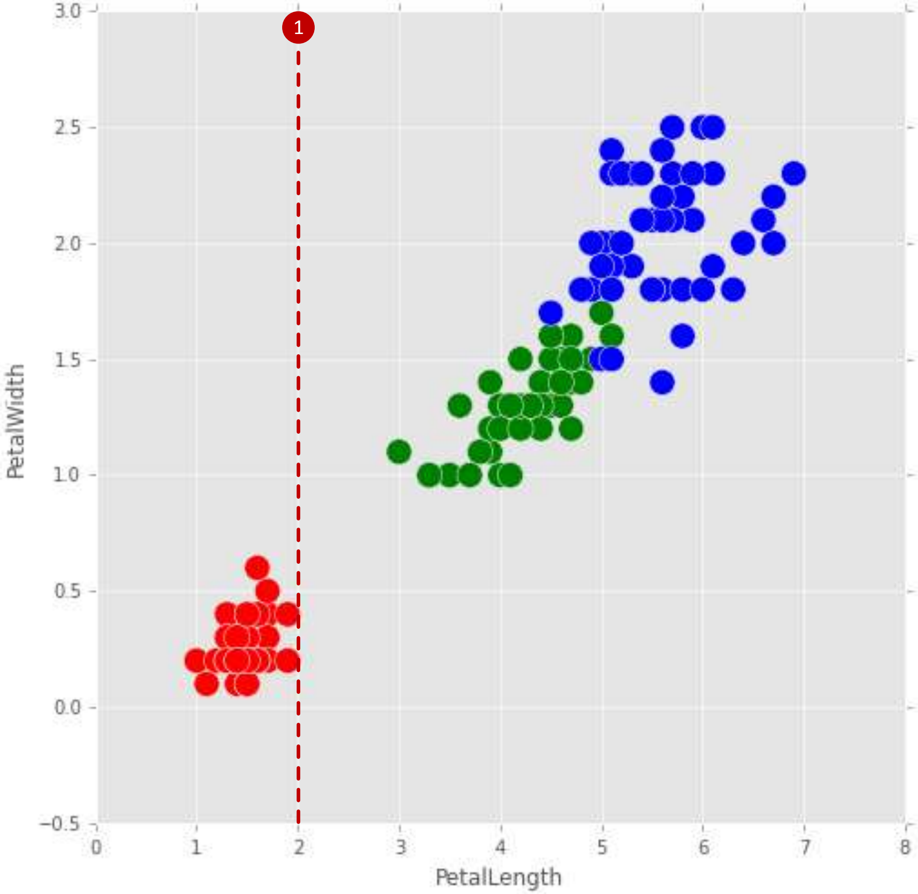
1.58

$H_{after} = \frac{50}{150} \times 0 + \frac{100}{150} \times 1$

$= \frac{2}{3} = 0.667$

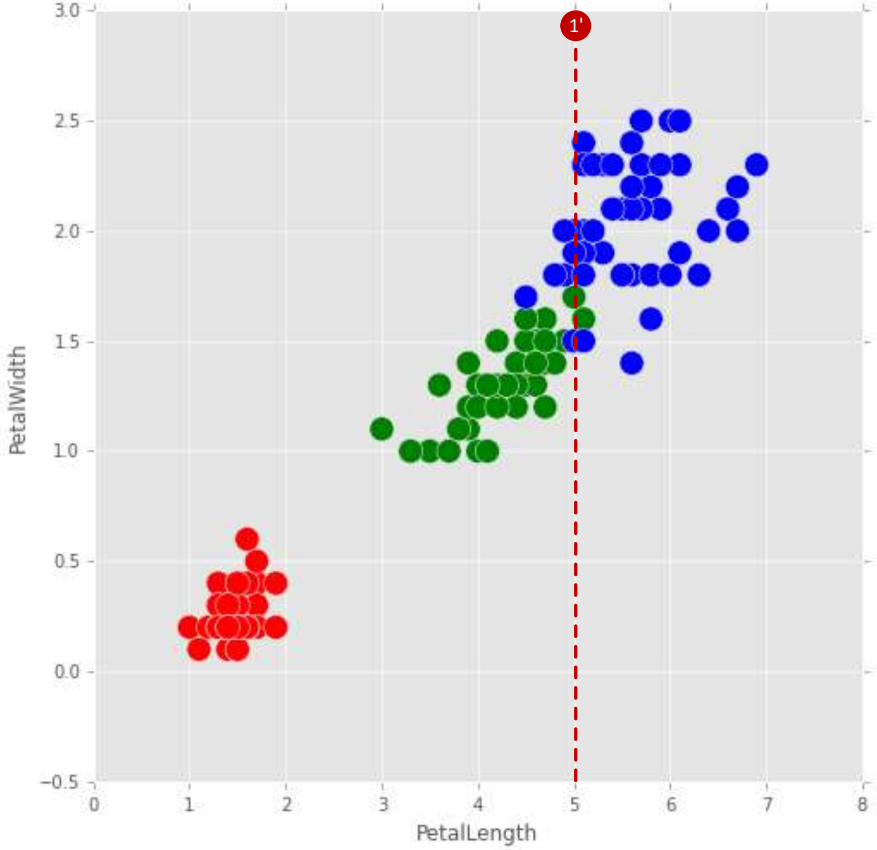
$IG = H_{before} - H_{after}$

$= 1.58 - 0.667 = 0.918$



Activity | ❷ What’s the entropy after split 1’? Intuition? (Note: 9 “blues” on the left, 1 “green” on the right)

EXERCISE

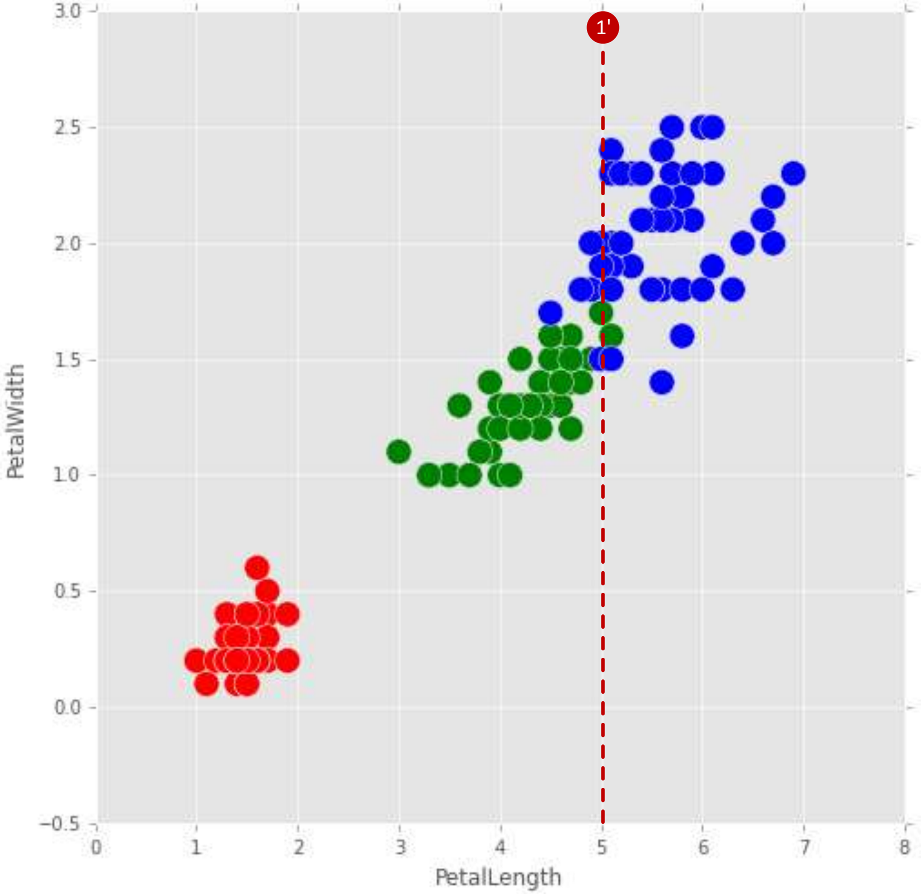


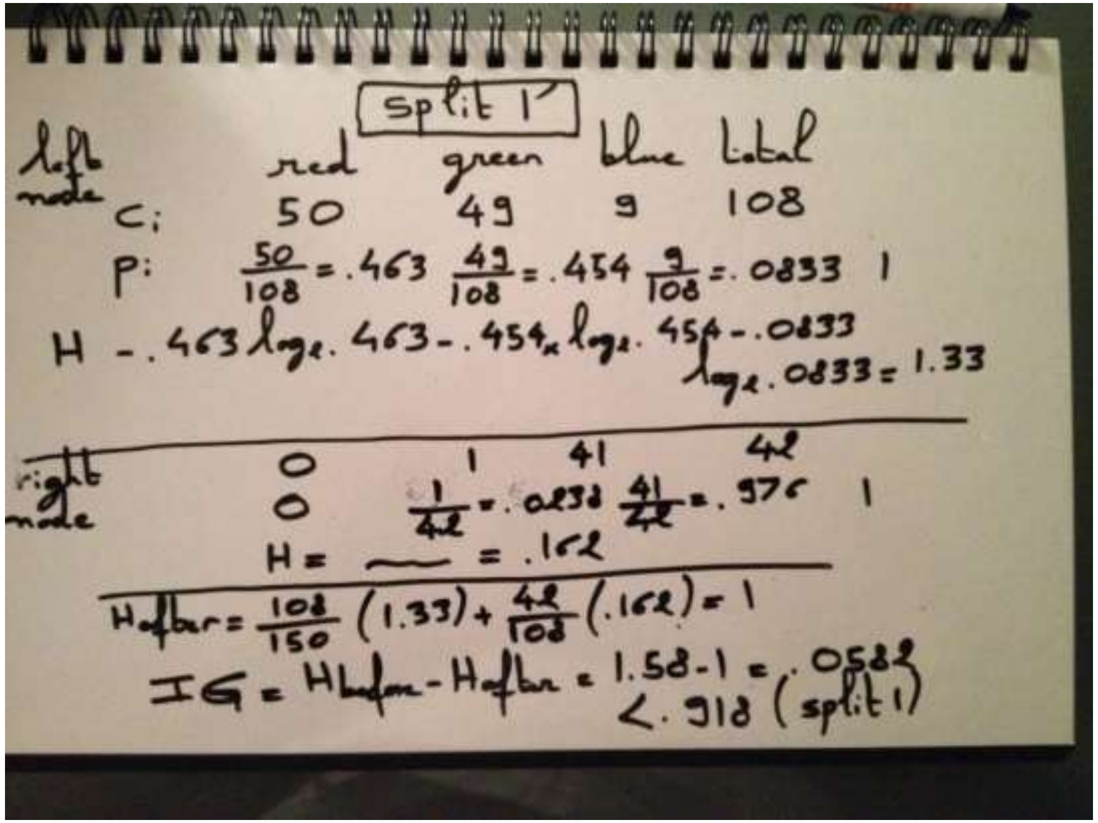
left node after split 1’				
	red	green	blue	overall
c_i				
p_i				
H_{left}				

right node after split 1’				
	red	green	blue	overall
c_i				
p_i				
H_{right}				

overall after split 1’	
H_{after}	
IG	

The information gain for split 1’ is modest at .582 and less than split 1’s. Split 1 wins





split 1'

left node	red	green	blue	total
C:	50	49	9	108
P:	$\frac{50}{108}$ = .463	$\frac{49}{108}$ = .454	$\frac{9}{108}$ = .0833	1

H $-.463 \log_2 .463 - .454 \times \log_2 .454 - .0833 \log_2 .0833 = 1.33$

right node				
	0	1	41	42
	0	$\frac{1}{42}$ = .0238	$\frac{41}{42}$ = .976	1

H = ~~ = .162

$H_{after} = \frac{108}{150}(1.33) + \frac{42}{108}(.162) = 1$

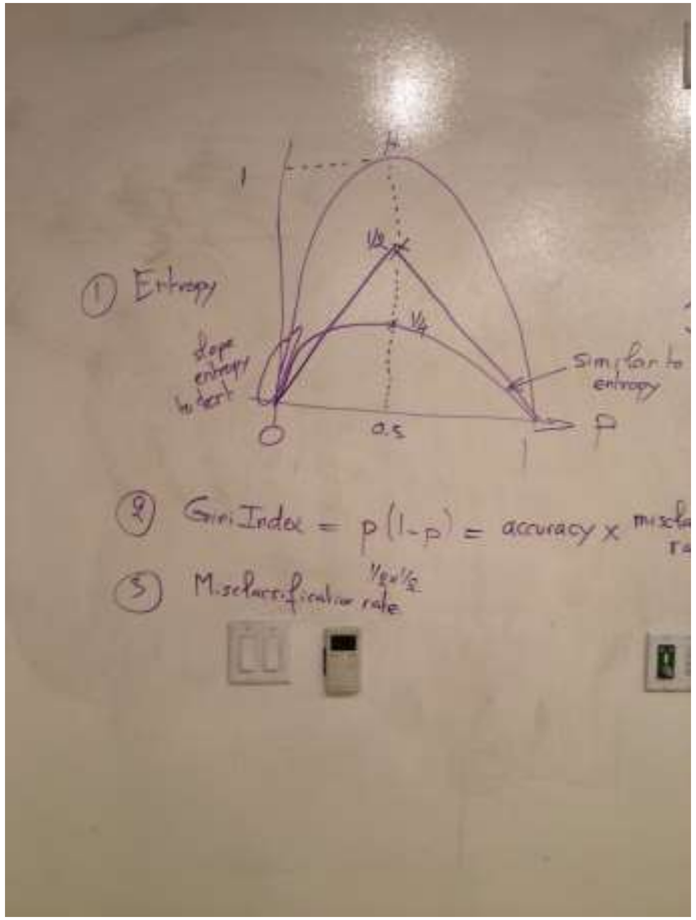
$IG = H_{before} - H_{after} = 1.58 - 1 = .0582$
$< .918$ (split 1)

Most common occurring class

- In practice, we don’t expect each terminal region to hold a single class

- Instead, we predict that each observation belongs to the most commonly occurring class of training observations in the region to which it belongs

Other measures | Gini index and misclassification rate



① Entropy

② Gini Index = p(1-p) = accuracy x miscla ra

③ Misclassification rate

❻ Build a Model

Classification and Regression Decision Trees

Commonalities and differences between classification and regression decision trees

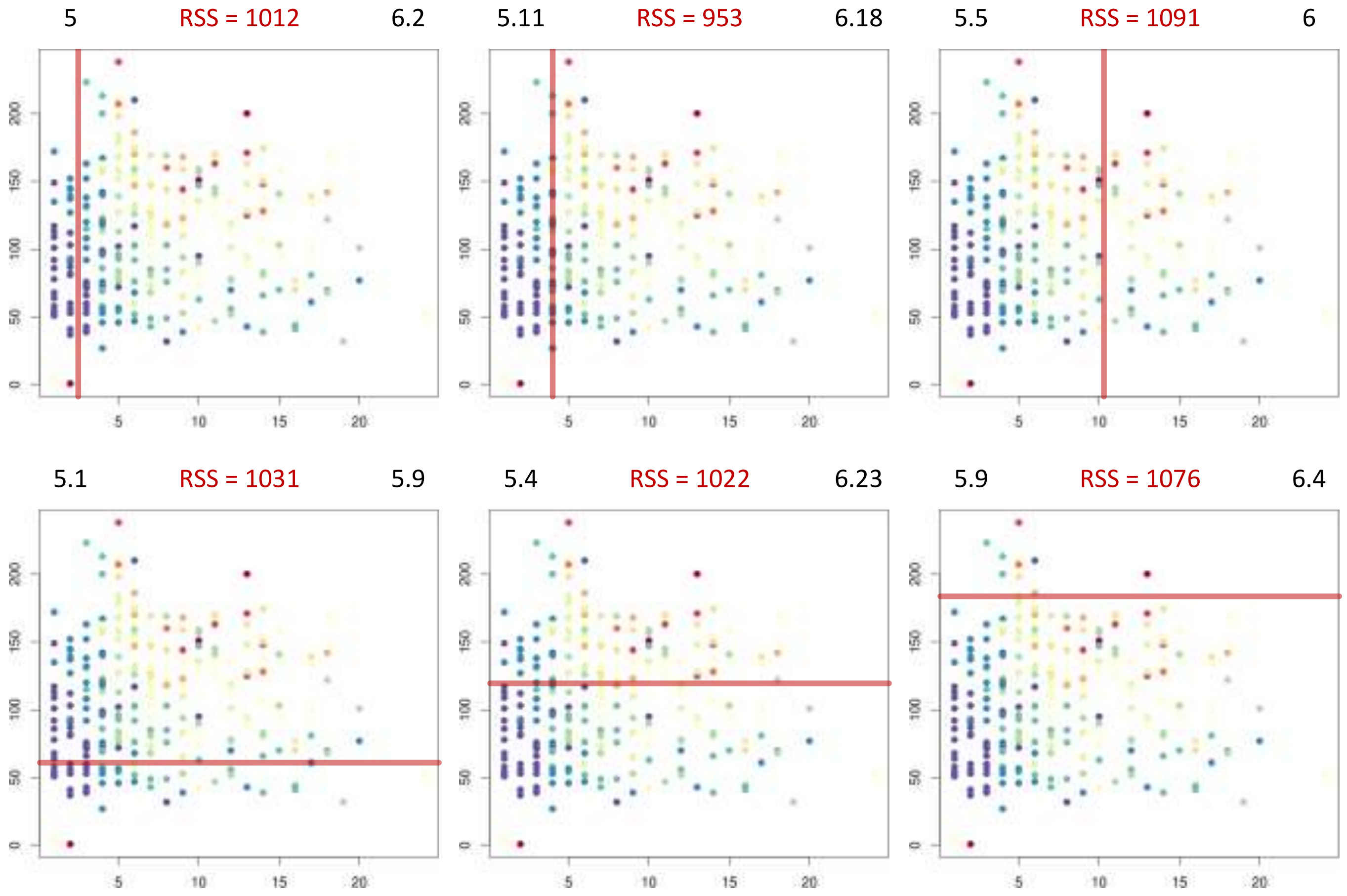
- Just as in the classification setting, we use recursive binary splitting to grow a regression tree

- For every observation that falls into the region R_j, we make the prediction $\hat{y}_{R_j}$, which is the mean of the response values for the training observations in R_j

- In the regression setting, we cannot use entropy for making the binary splits. A natural alternative to H is RSS (residual sum of squares)

$$\sum_{j=1}^{J} \sum_{i \in R_j} \left(y_i - \hat{y}_{R_j}\right)^2$$

We first select the feature and the cutpoint such that splitting the feature space into the regions $\{x \mid feature \leq cutpoint\}$ and $\{x \mid feature > cutpoint\}$ leads to the greatest possible reduction in RSS



Source: The Elements of Statistical Learning

Top-down greedy approach (a.k.a., recursive binary splitting)

- Once the first cut is made, we recursively repeat the process in the two previously identified regions

- For *regression* trees, we would be looking for the best predictor and the *best cutpoint* in order to split the data further so as to minimize the *RSS* within each of the resulting regions

- For *classification* trees, we would be looking for the best predictor and the *highest* information gain in order to split the data further so as to minimize the *entropy* within each of the resulting regions

Decision Trees

Pros and Cons

Decision Trees | Pros and cons

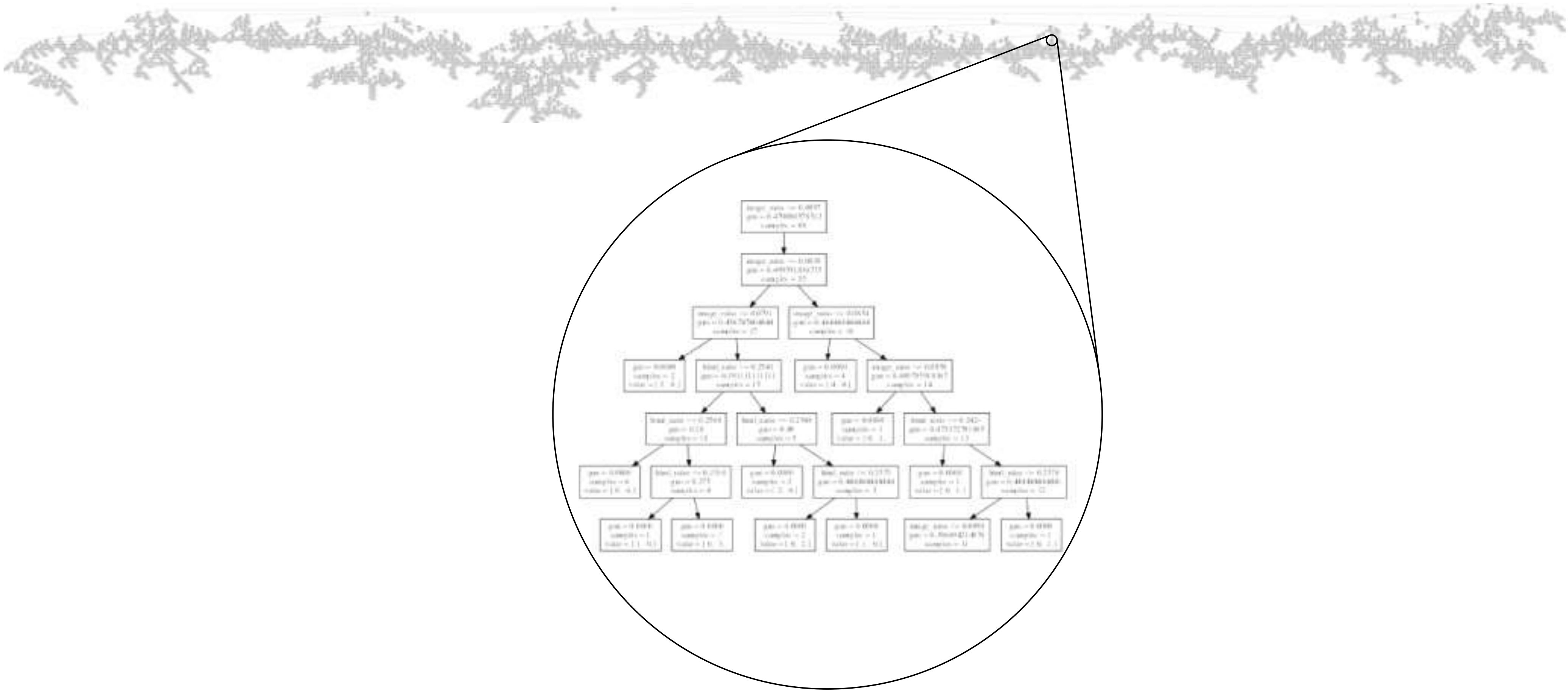
- Pros
  - Trees are very easy to explain to people. They are even easier to explain than linear regression
  - Decision trees more closely mirror human decision-making than do the regression and classification methods seen so far
  - Trees can be displayed graphically and are easily interpreted even by non-experts
  - Trees can easily handle qualitative predictors without the need to create binary variables

- Cons
  - Trees do not generally have the same level of predictive accuracy as some of the other regression and classification methods seen so far. However, by aggregating many decision trees, the predictive performance of trees can be substantially improved

❻ Build a Model

Overfitting

An unconstrained decision tree can learn an extreme tree (e.g., below)



Overfitting

- Decision trees tend to be weak models because they can easily memorize or overfit to a dataset
  - A model is overfit when it memorizes or bends to a few specific data points rather than picking up general trends in the data

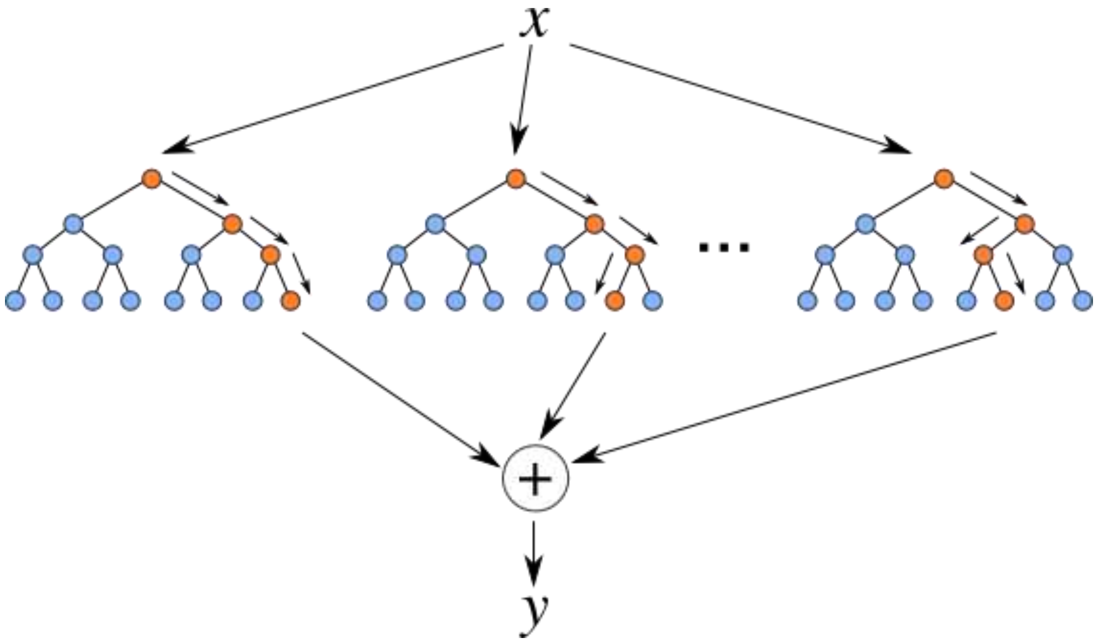
- We can limit our decision trees using a few methods
  - Limit the number of questions (nodes) a tree can have
  - Limit the number of samples in the leaf nodes

How can we avoid overfitting and increase predictability?

Random Forests

Random forests are an *ensemble* or collection of individual decision trees

- Random forest models are one of the most widespread classifiers used
- They are relatively simple to use and help avoid overfitting



Random Forests

Pros and Cons

Random Forests | Pros and cons

- Pros
  - Easy to tune
  - Built-in protection against overfitting
  - Non-linear
  - Built-in interaction effects

- Cons
  - Slow
  - No “coefficients”
  - Black-box
  - Harder to explain

DS

Random Forests

Training

Training a Random Forest

- Training a random forest model involves training many decision tree models
- Since decision trees easily overfit, we use many decision trees together and randomize the way they are created

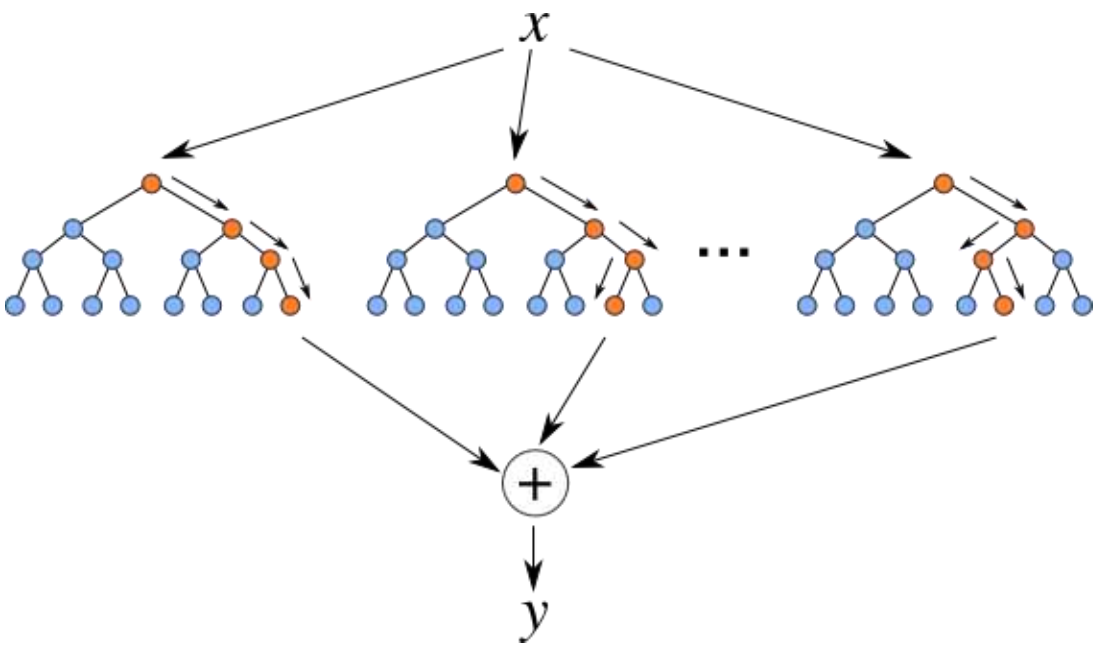
- Random Forest Training Algorithm
  - Take a bootstrap sample (random sample) of the dataset
  - Train a decision tree on the bootstrap sample
  - For each split/feature selection, only evaluate a *limited* number of features to find the best one
  - Repeat this for a number of trees

Random Forests

Predicting

Predicting using a Random Forest

- Predictions for a random forest model come from each decision tree
- Make an individual prediction with each decision tree
- Combine the individual predictions and take the majority vote



Lab

Decision Trees and Random Forests

Review

Review

- What are decision trees?
- What does training involve?
- What are some common problems with decision trees?
- What are random forests?
- What are some common problems with random forests?

Q & A

Before Next Class

Before Next Class

Before the next lesson, you should already be able to:

- Experience with *sklearn* classifiers, specifically random forests and decision trees
- Install the Python package spacy with `conda install spacy`
- Run the spacy download data command with `python -m spacy.en.download --force all`

Next Class

Natural Language Processing and Text Classification

Learning Objectives

After the next lesson, you should be able to:

- Define natural language processing
- List common tasks associated with
  - Use-cases
  - Tokenization
  - Stemming and lemmatization
  - Tagging and parsing
- Demonstrate how to classify text or documents using scikit-learn

Exit Ticket

Don’t forget to fill out your exit ticket here