

Decision Trees and Random Forests

Ivan Corneillet

Data Scientist

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



DS

Announcements and Exit Tickets

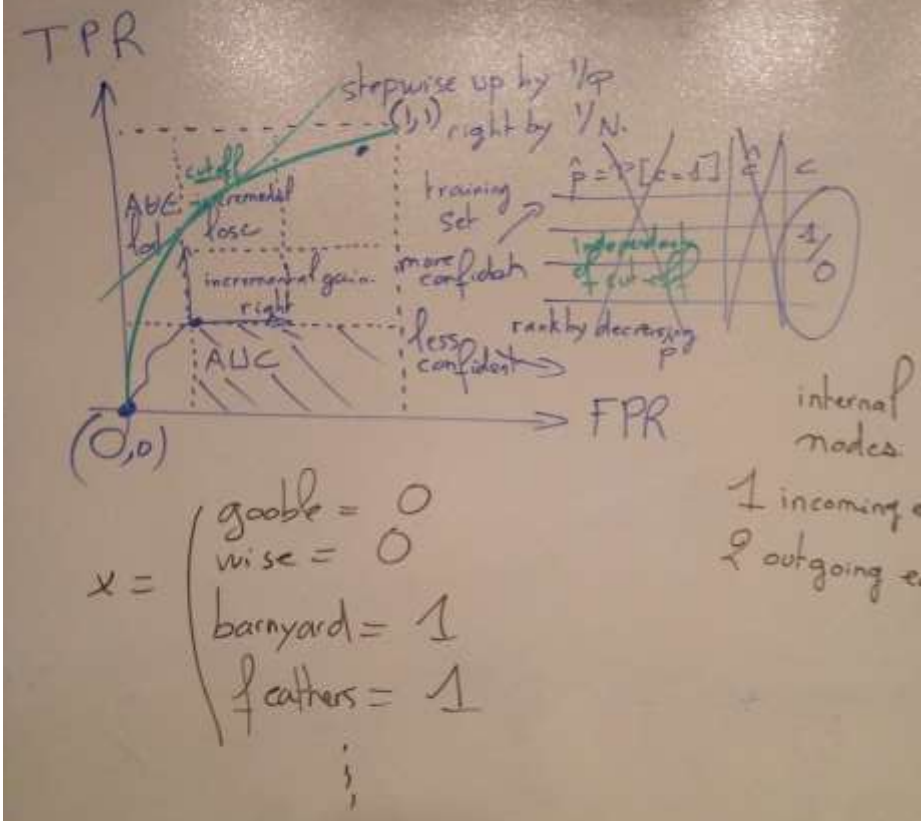
DS

Review

DS

Review

ROC/AUC





DS

Today

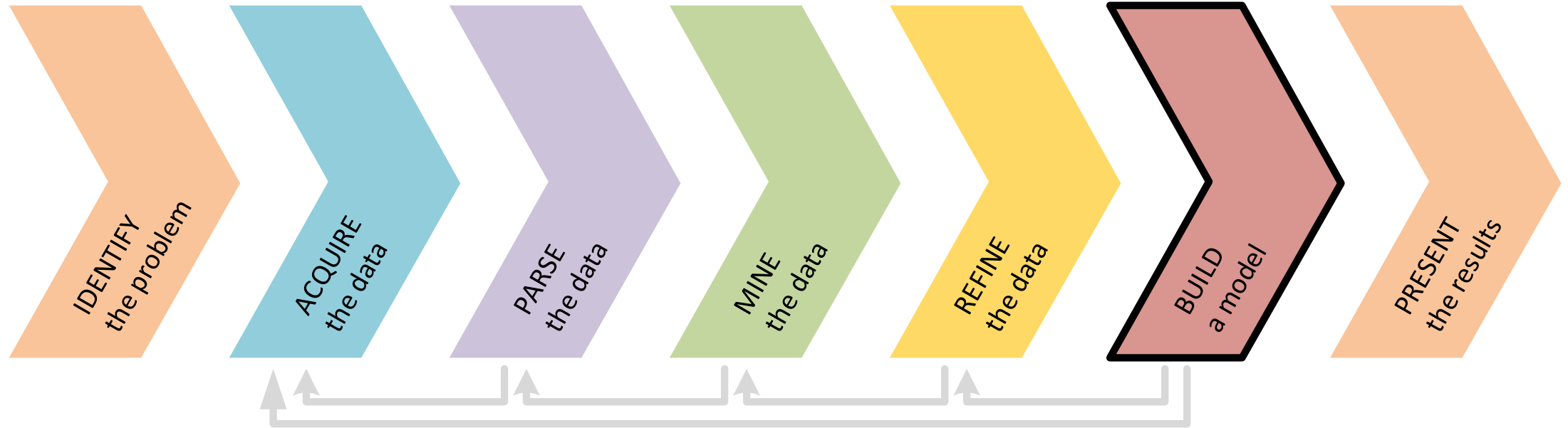
Here's what's happening today:

- Announcements and Exit Tickets
- Review
- **⑥ 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
- **⑥ Build a Model | Random Forests**
 - Pros and Cons
 - Training
 - Predicting
- Lab – Decision trees and random forests
- Review
- Exit Tickets

Today, we are starting Unit 3 by introducing two new machine learning algorithms: Decision Trees and Random Forests

Research Design and Data Analysis	Research Design	Data Visualization in <i>pandas</i>	Statistics	Exploratory Data Analysis in <i>pandas</i>
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 Models	Natural Language Processing	Databases

Therefore, today we are refocusing on **⑥ BUILD** a model step but with a focus on decision trees and random forests



DS

Decision Trees

A black circle containing the white text "DS".

DS

Decision Trees

Motivating Example and Activity

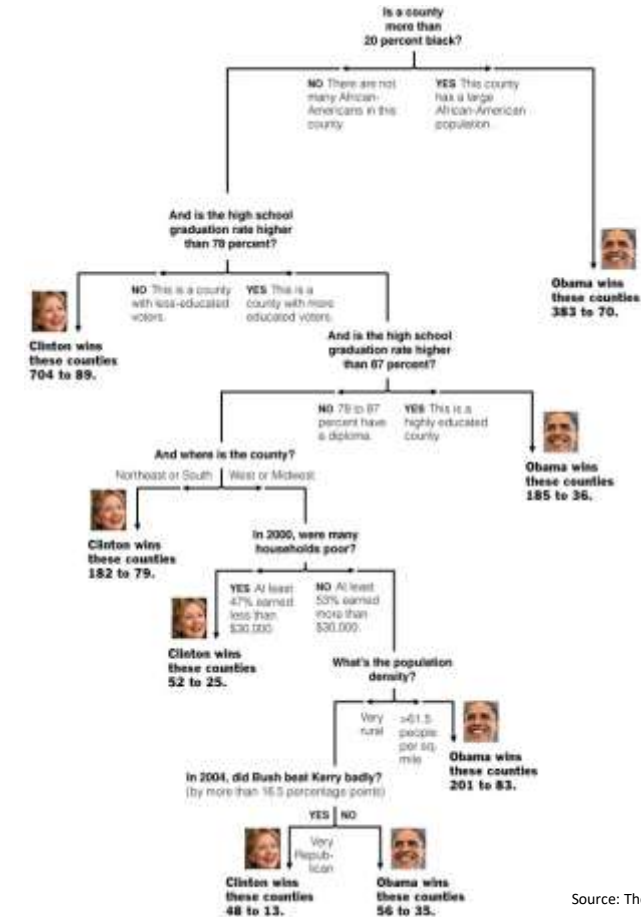
The 2008 Democratic Primaries

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



Source: The New York Times

Activity | The 2008 Democratic Primaries

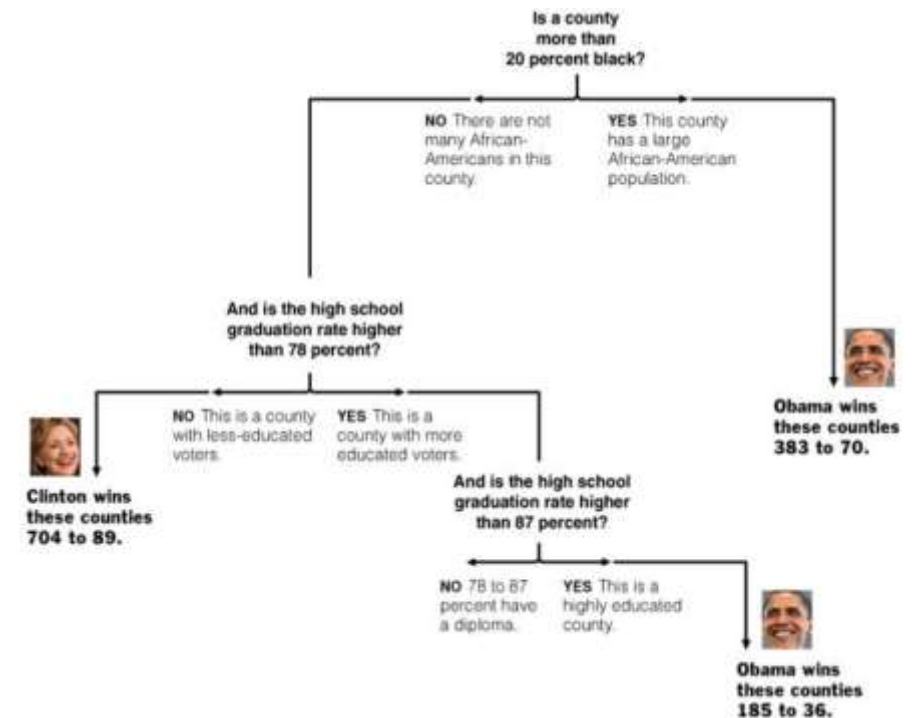
EXERCISE

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

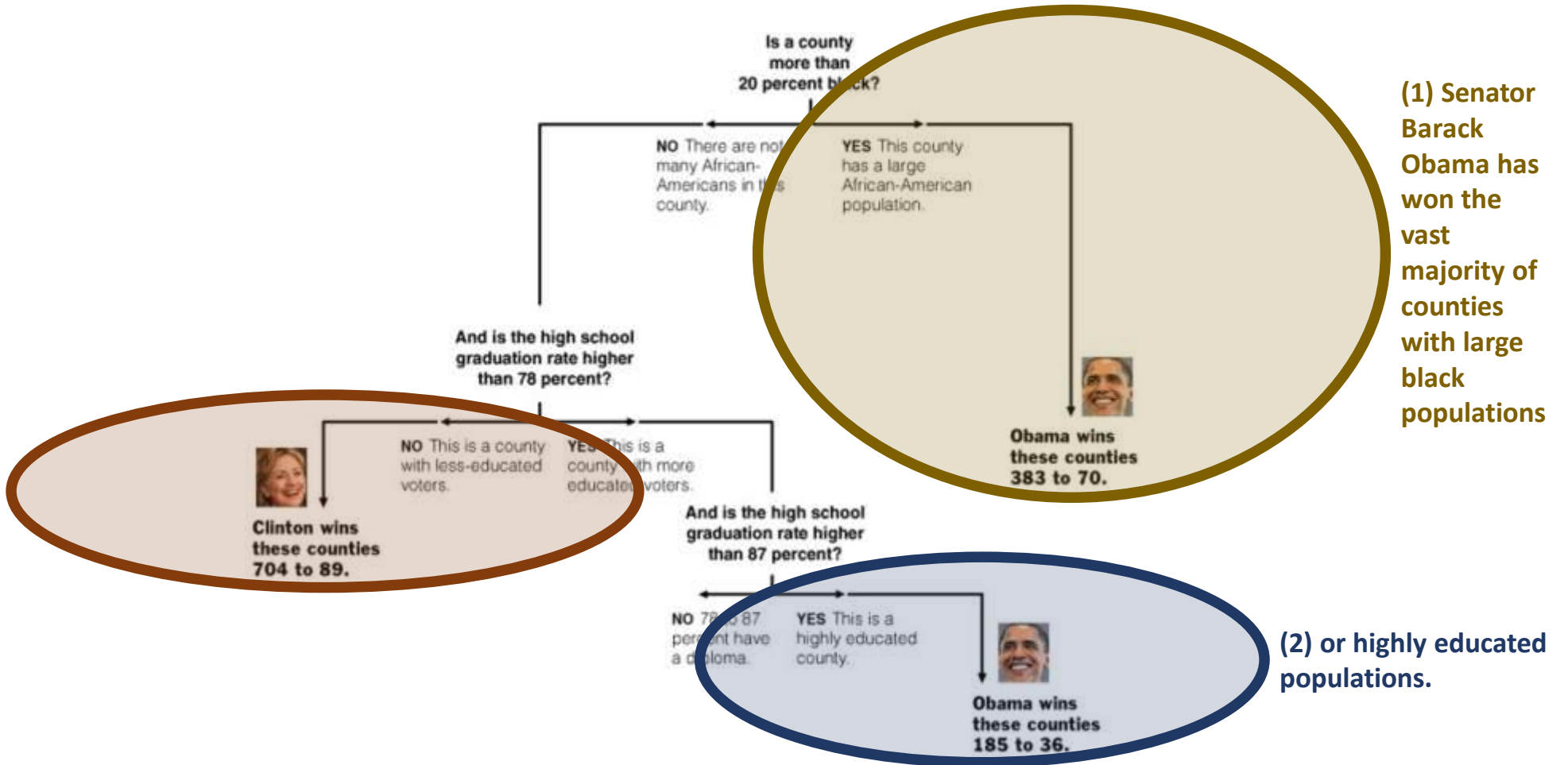
DELIVERABLE

Answers to the above questions



Activity | “In the nominating contests so far, ...

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



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

A black circle containing the white text "DS".

DS

Decision Trees

Codealong – Part A

The 2008 Democratic Primaries

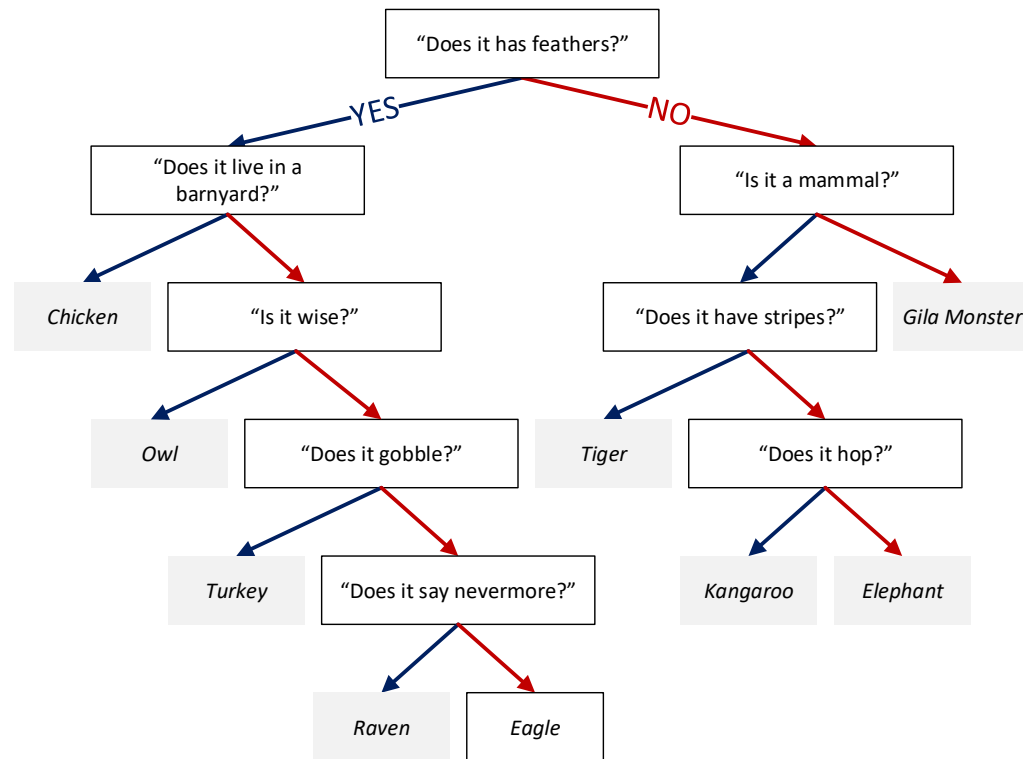


DS

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 question should reduce the search space as much as possible

DS

Decision Trees

Structure

DS

Decision Trees

Predicting

CLT

$$1 - \frac{1}{e}$$

class Tree

def predict (x)

← one observation/sample

① Set the root node as the parent node

② while the parent node is not a leaf node:

 evaluate the split condition of the parent node using

 if true:

 set the left node as the new parent node

 otherwise (if false)

 set the right node as the new parent node

③ return the class of the parent/leaf node



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?)

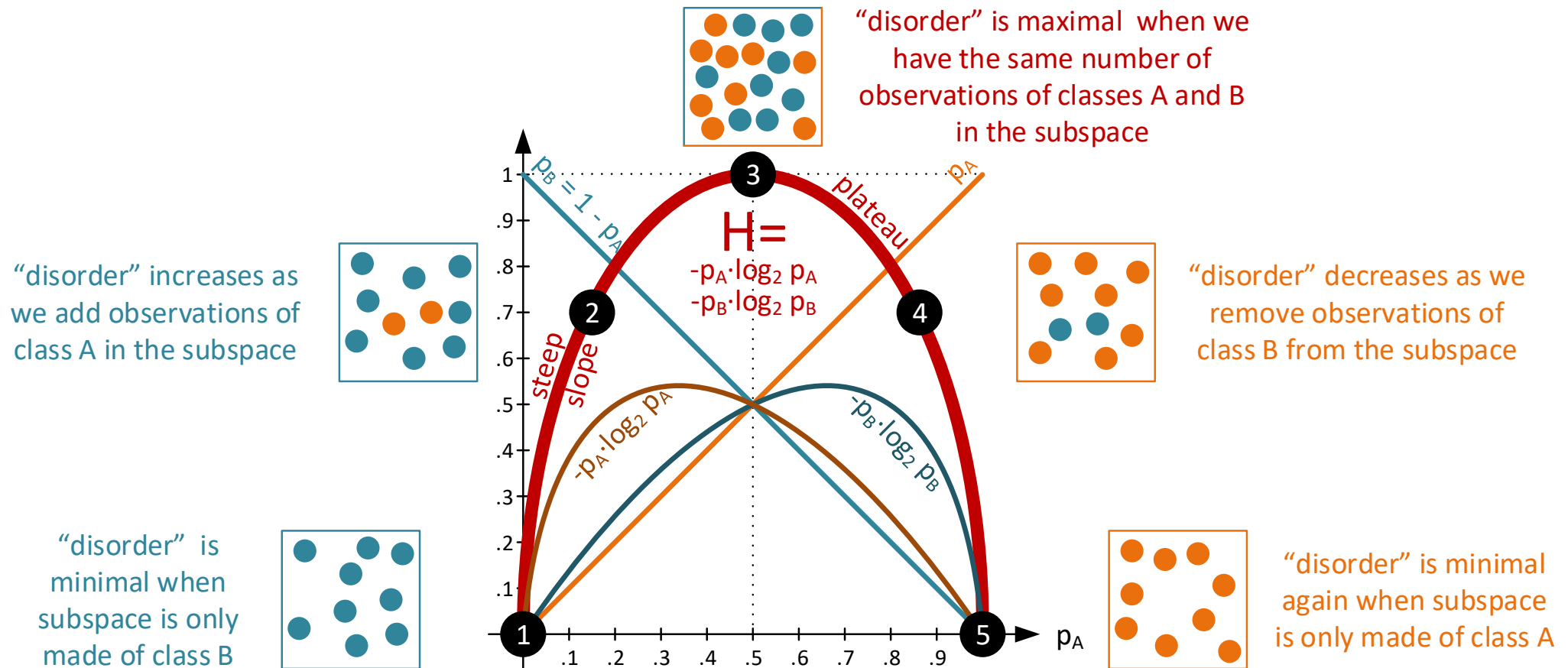
▸ We'll then move on to regression problems when addressing ❶ (How to choose the split conditions?)

DS

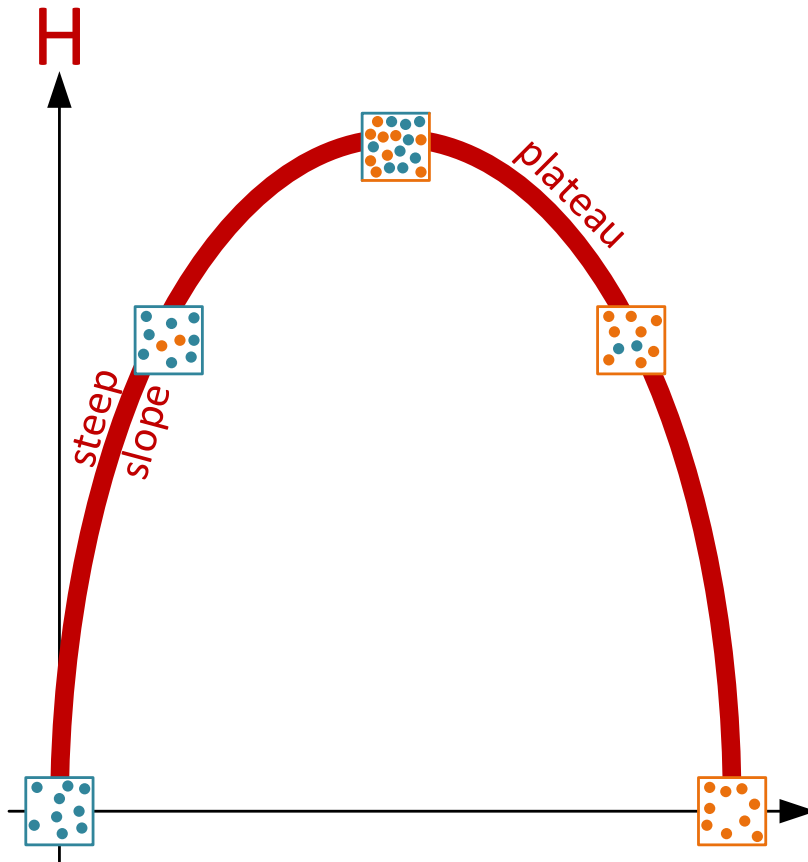
⑥ Build a Model

Entropy

Entropy (H) is a measure of disorder



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)

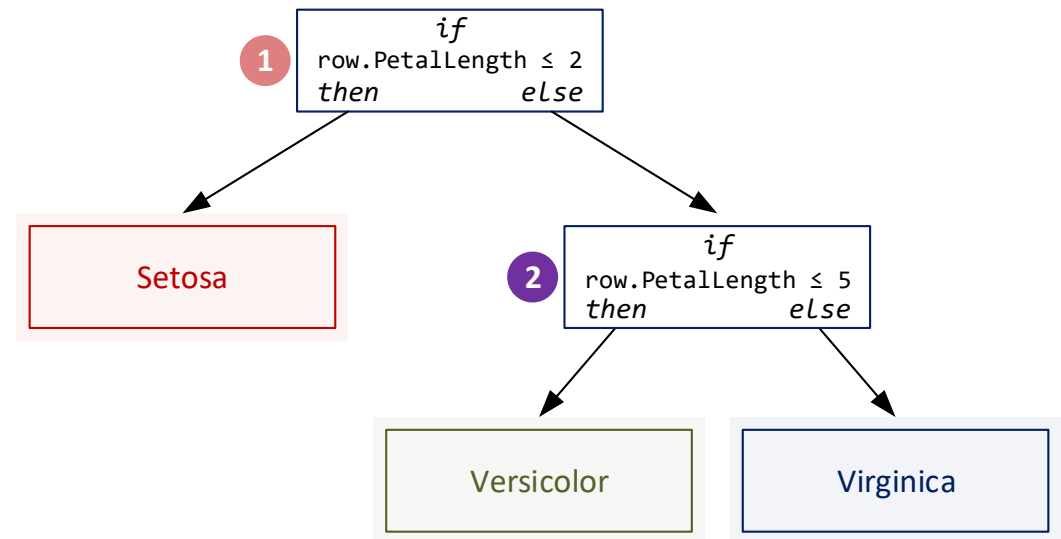
Decision Trees

Training a Classification Decision Tree

② *How to choose the order of the conditions?*

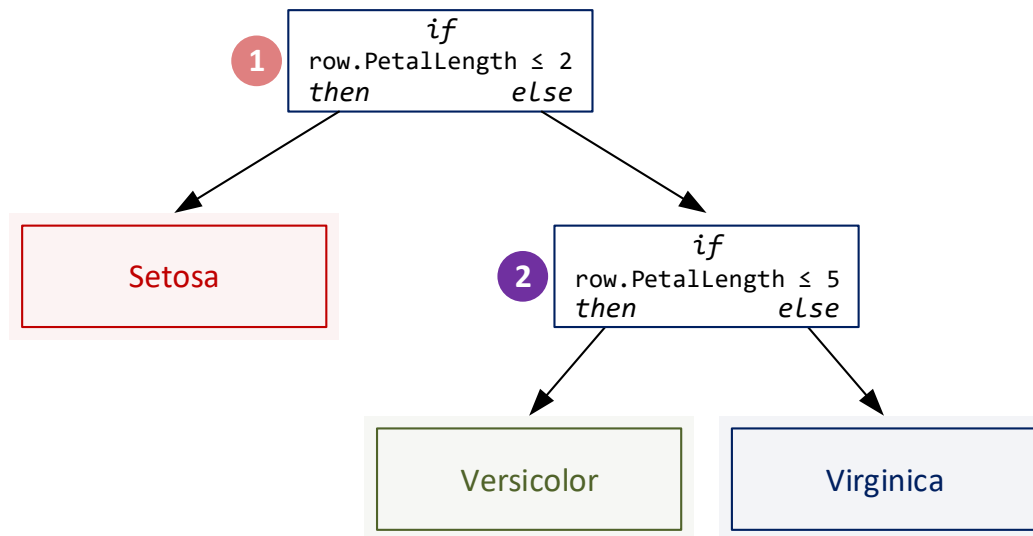
Our first classifiers (session 8) are decision trees

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

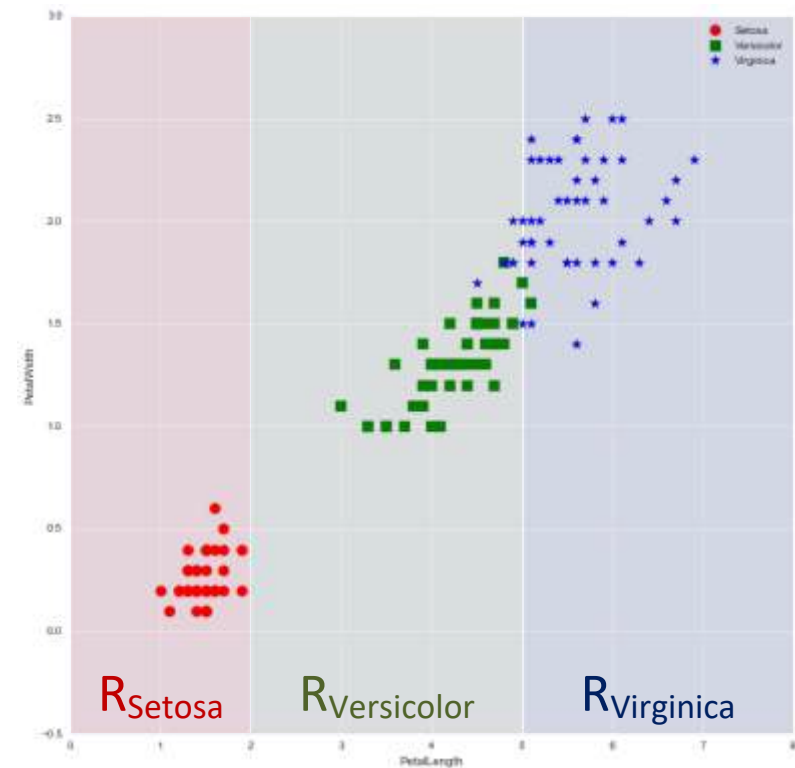


Our first classifiers (session 8) are decision trees (cont.)

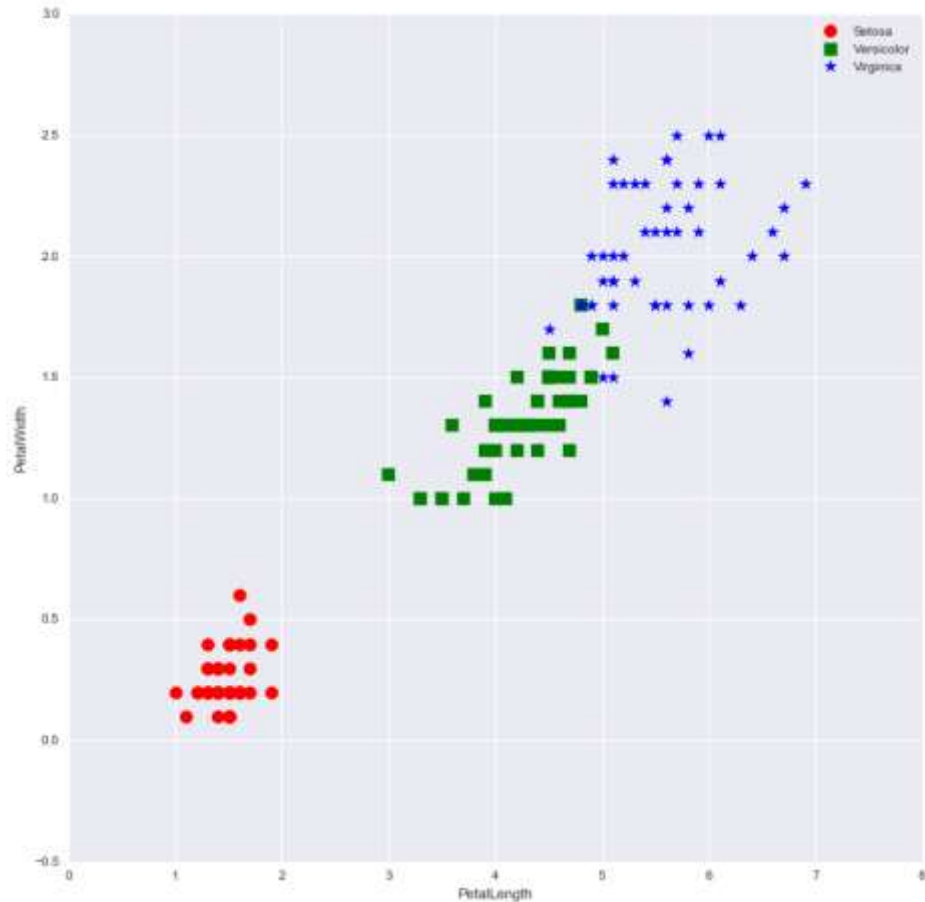
Decision Tree



Feature Space



① 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



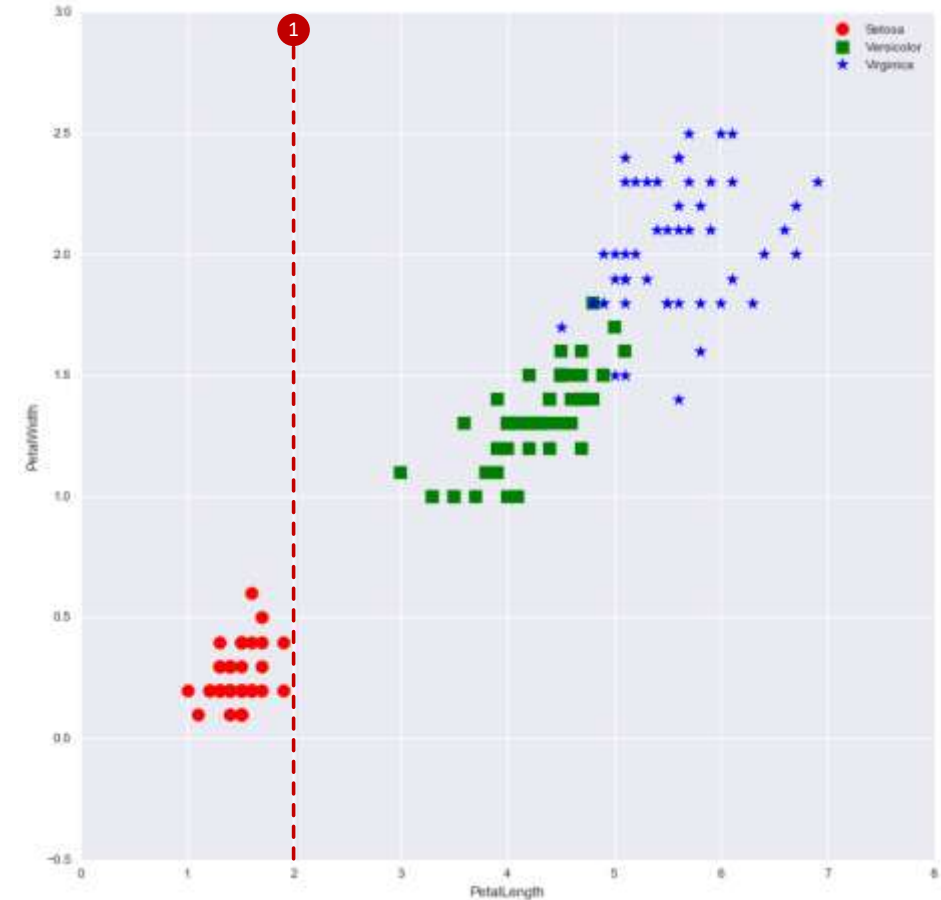
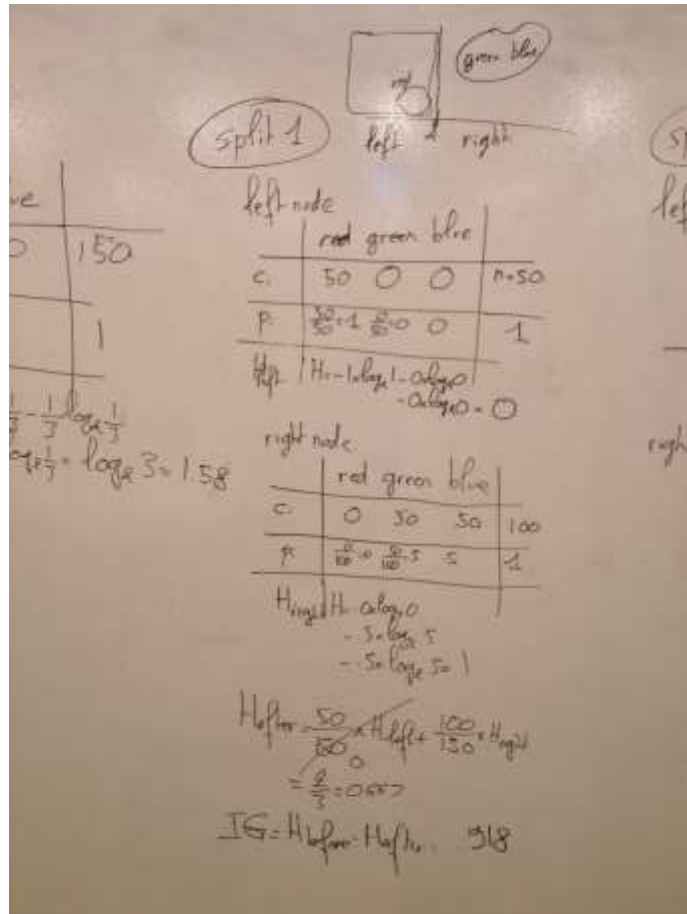
root node

	red	green	blue	
C_i	50	50	50	150
$p_i = \frac{C_i}{n}$	$\frac{50}{150} = \frac{1}{3}$	$\frac{1}{3}$	$\frac{1}{3}$	1

$H_{\text{before}} = H = -\frac{1}{3} \log_2 \frac{1}{3} - \frac{1}{3} \log_2 \frac{1}{3} - \frac{1}{3} \log_2 \frac{1}{3}$
 $= 3 \left(\frac{1}{3} \right) \log_2 \frac{1}{3} = -\log_2 \frac{1}{3} = \log_2 3 = 1.58$

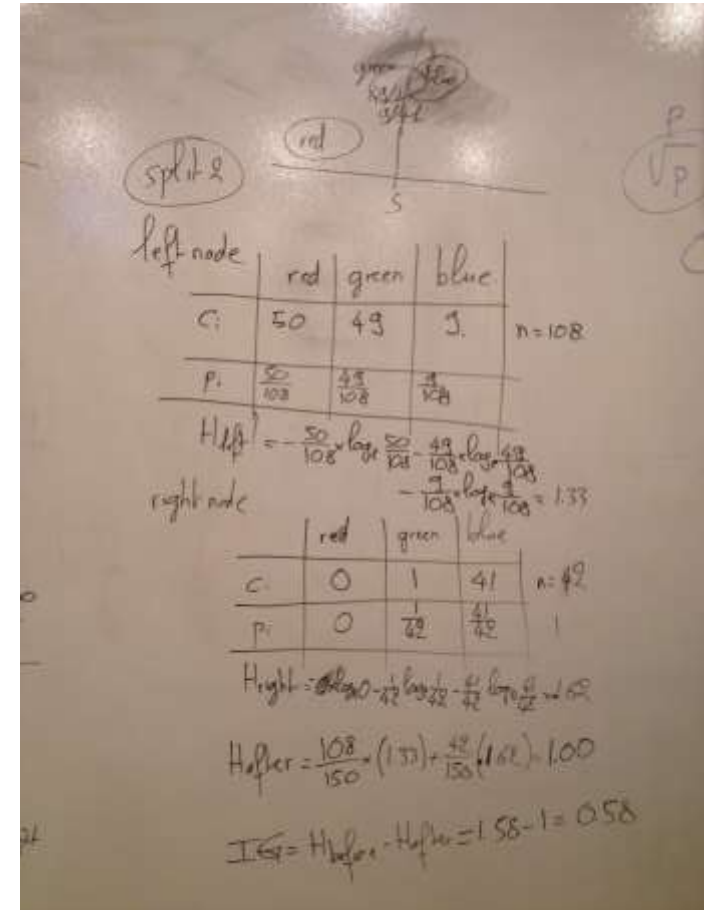
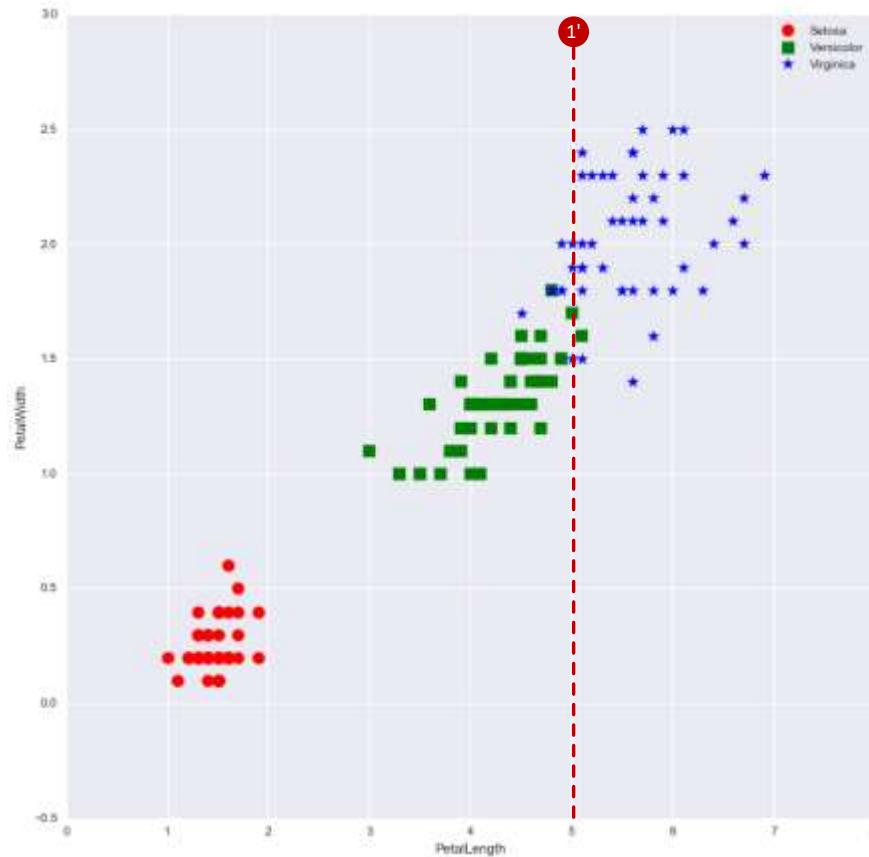
$H = -\sum p_i \times \log_2 p_i$

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



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

EXERCISE



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



DS

Decision Trees

Codealong – Part B

The 2008 Democratic Primaries

DS

⑥ Build a Model

Classification and Regression Decision Trees

Commonalities and differences between classification and regression decision trees

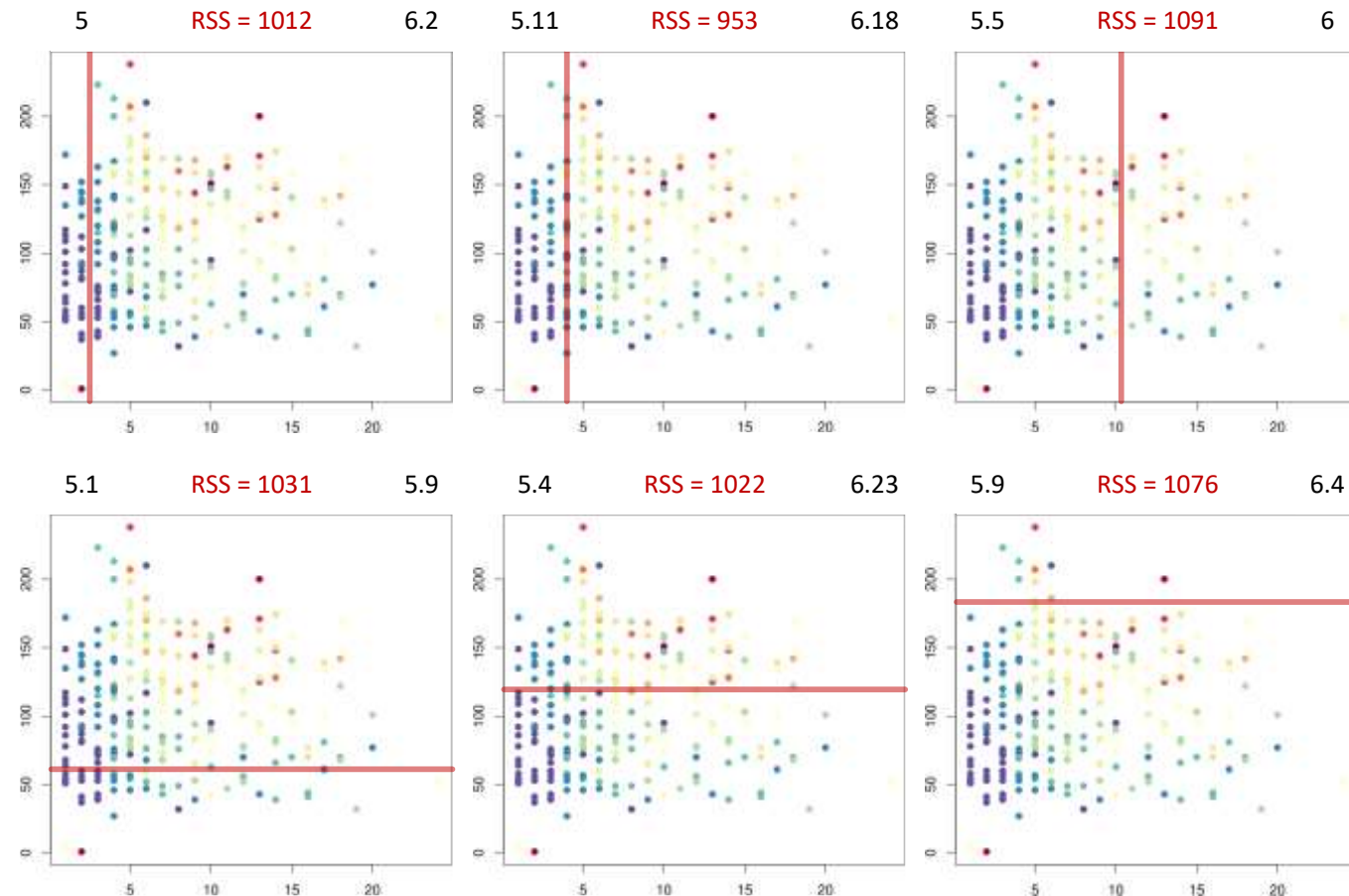
- Just as in the classification setting, we use recursive binary splitting to grow a regression tree

- 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)

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

We first select the feature and the cutpoint such that splitting the feature space into the regions $\{x \mid \textit{feature} \leq \textit{cutpoint}\}$ and $\{x \mid \textit{feature} > \textit{cutpoint}\}$ 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



DS

Decision Trees

Pros and Cons

Decision Trees | Pros and cons

▸ Pros

- Very easy to explain to people; even easier to explain than linear regression
- Mirror more closely human decision-making than do the regression and classification methods seen so far
- Can be displayed graphically and are easily interpreted even by non-experts
- Can easily handle qualitative predictors without the need to create binary variables

▸ Cons

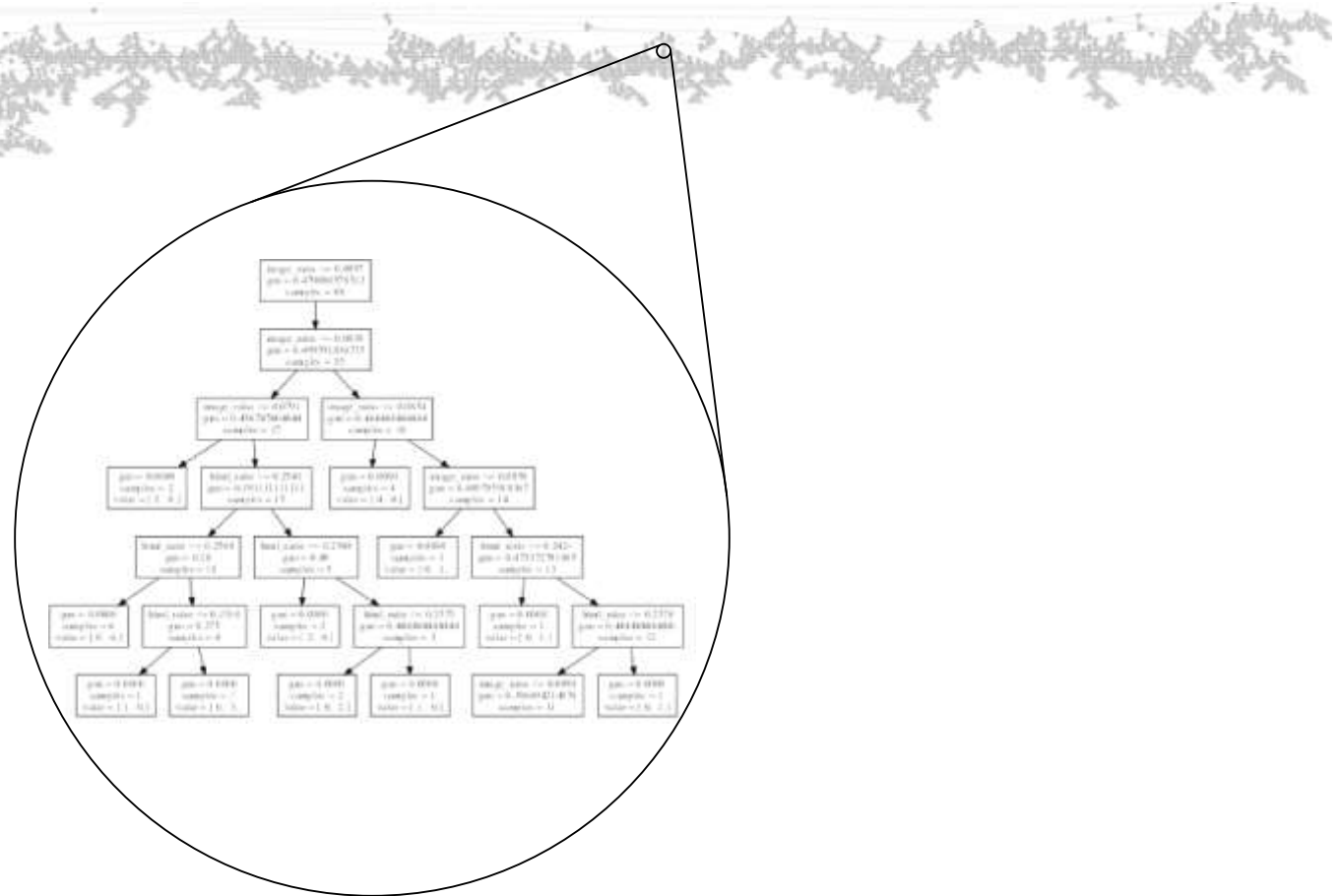
- Do not generally have the same level of predictive accuracy as some of the other regression and classification methods seen so far (higher variance). However, by aggregating many decision trees, the predictive performance of trees can be substantially improved

DS

⑥ Build a Model

Overfitting

An fully-grown (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 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

A black circle containing the white text "DS".

DS

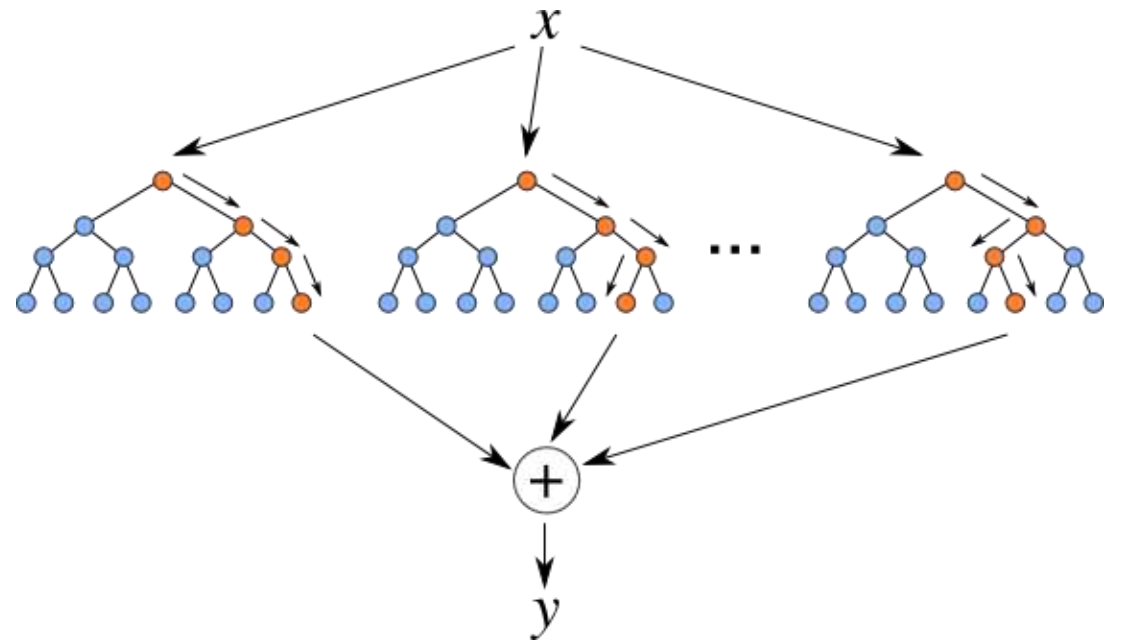
How can we avoid overfitting
and increase predictability?

DS

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



A black circle containing the white text "DS".

DS

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

A black circle containing the white text "DS".

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

A black circle containing the white text "DS".

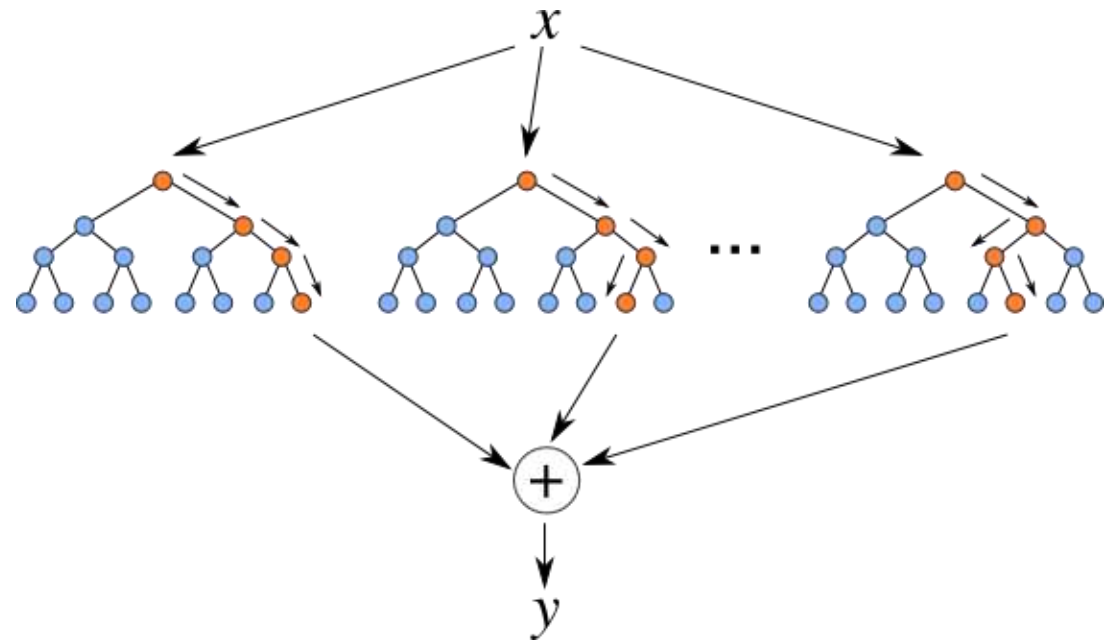
DS

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



DS

Decision Trees and Random Forests

Further Readings

Further Readings

- ISLR

- Basics of Decision Trees (section 8.1, pp. 303 – 316)
- Random Forests (section 8.2.2, pp. 319 – 321)

- ESLII

- Tree-Based Methods (section 9.2, pp. 305 – 317)
- Random Forests (section 15, pp. 587 – 604)

A black circle containing the white text "DS".

DS

Lab

Decision Trees and Random Forests

A black circle containing the white text "DS".

DS

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?

Next Class

Flexible Class Session #3 | Machine Learning Modeling, Part 2



DS

Exit Ticket

Don't forget to fill out your exit ticket [here](#)

Slides © 2016 Ivan Corneillet Where Applicable
Do Not Reproduce Without Permission