GENERAL ASSEMBLY

DECISION TREES AND RANDOM FORESTS

Adam Jones, PhD

Data Scientist @ Critical Juncture

LEARNING OBJECTIVES

- 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

COURSE

PRE-WORK

PRE-WORK REVIEW

- Use Seaborn to create plots
- Knowledge of a bootstrap sample
- Explain the concepts of cross-validation, logistic regression, and overfitting
- Know how to build and evaluate *some* classification model in scikit-learn using cross-validation and AUC

OPENING

DECISION TREES AND RANDOM FORESTS

ACTIVITY: KNOWLEDGE CHECK

EXERCISE

ANSWER THE FOLLOWING QUESTIONS

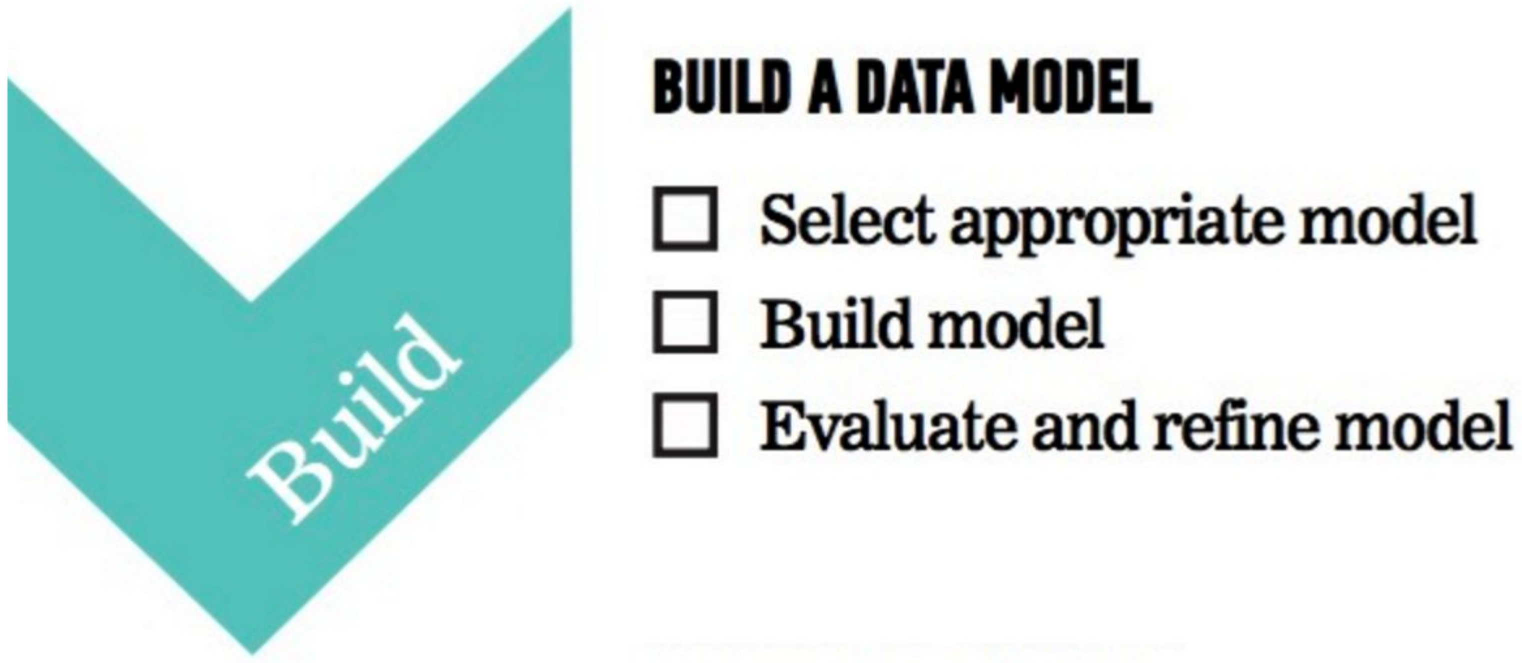
1. Define the difference between the **precision** and **recall** of a model
2. What are some use cases for **logistic regression**?

DELIVERABLE

Answers to the above questions

OVERVIEW OF THE DATA SCIENCE WORKFLOW

- In this lesson, we will focus on mining the dataset and building a model
    - We will focus on refining our model for the best predictive ability



GUIDED PRACTICE

EXPLORE THE DATASET

ACTIVITY: EXPLORE THE DATASET

EXERCISE

DIRECTIONS (25 minutes)

We will be using a dataset from **StumpleUpon**, a service that recommends web pages to users based upon their interests.
They like to recommend always-relevant "*evergreen*" sites, i.e websites that avoid topical content and focus on recipes, how-to guides, art projects, etc.
We want to determine important characteristics for "evergreen" websites.

1. Break into groups
2. Prior to looking at the data, brainstorm 3-5 characteristics that would be useful for predicting evergreen websites
3. Individually inspect the dataset- can you model or quantify any of the characteristics you wanted? (See the notebook for data dictionary and starter code)
4. Does being a news site affect evergreen-ness?
    - Compute or plot the percent of evergreen news sites

ACTIVITY: EXPLORE THE DATASET

EXERCISE

DIRECTIONS (25 minutes)

5. In general, does category affect evergreen-ness?
    - Plot the rate of evergreen sites for all Alchemy categories
6. How many articles are there per category?
7. Create a feature for the title containing “recipe”
    - Is the percentage of evergreen websites higher or lower on pages that have “recipe” in the title?

Check:

- Were you able to plot the requested features?
- Can you explain how you would approach this type of dataset?

DELIVERABLE

Requested features and answers to questions

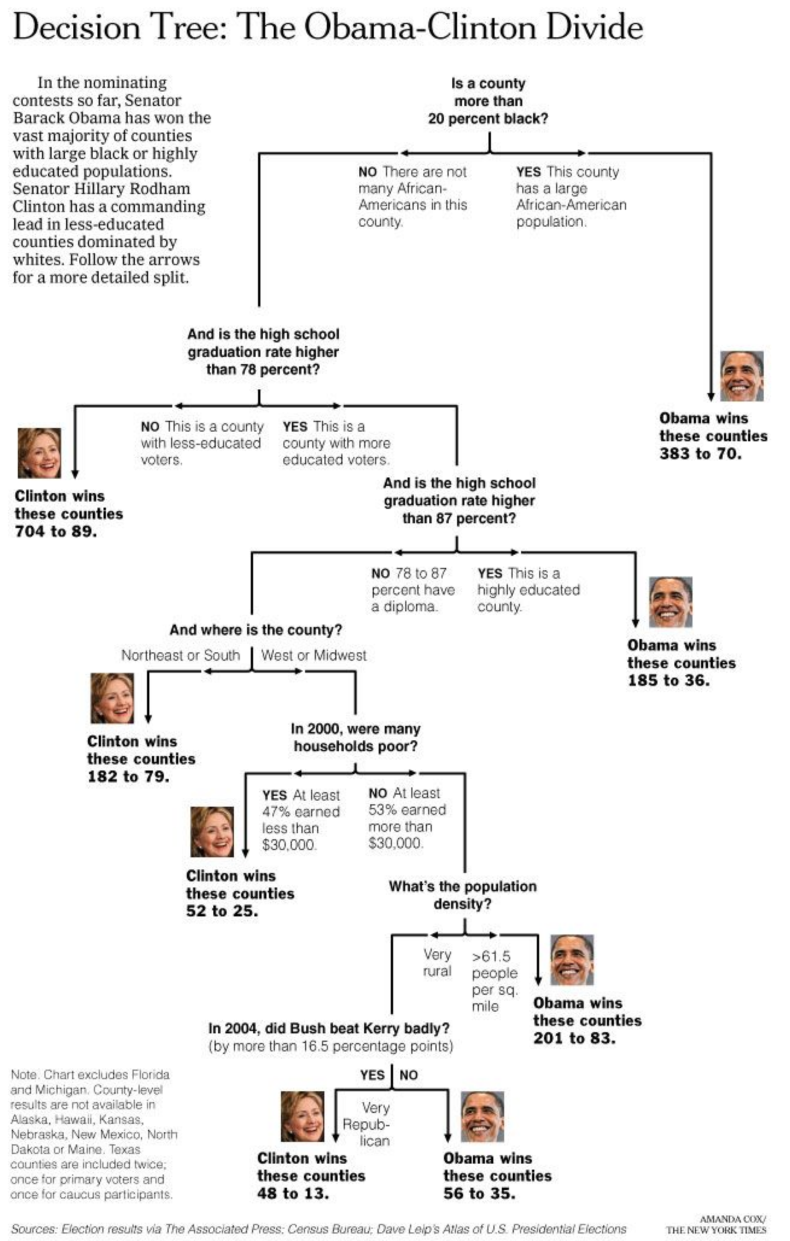
TRAINING DECISION TREES

INTUITION BEHIND DECISION TREES

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

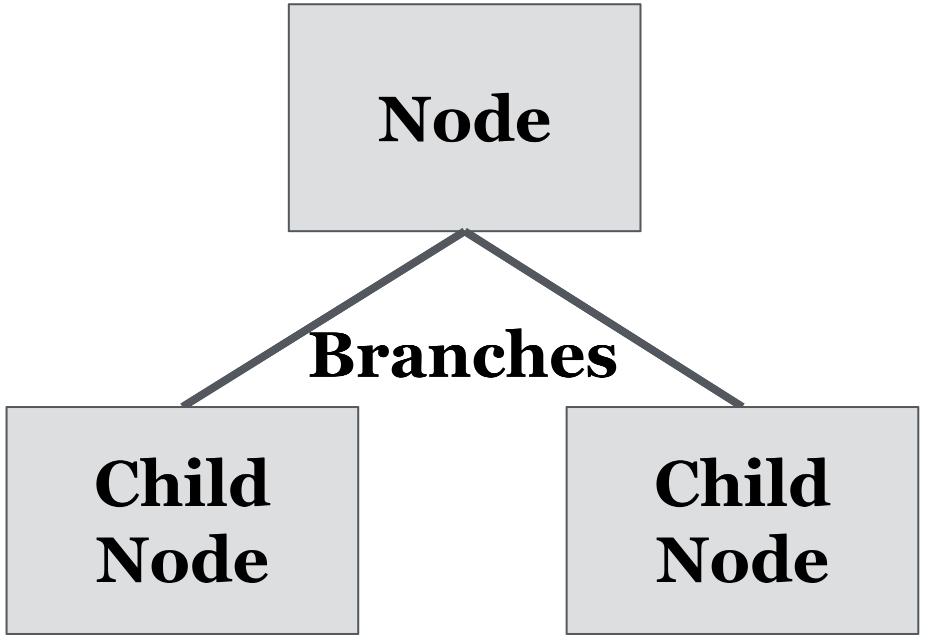
- We want the smallest set of questions to get to the right answer

- Each questions should reduce the search space as much as possible



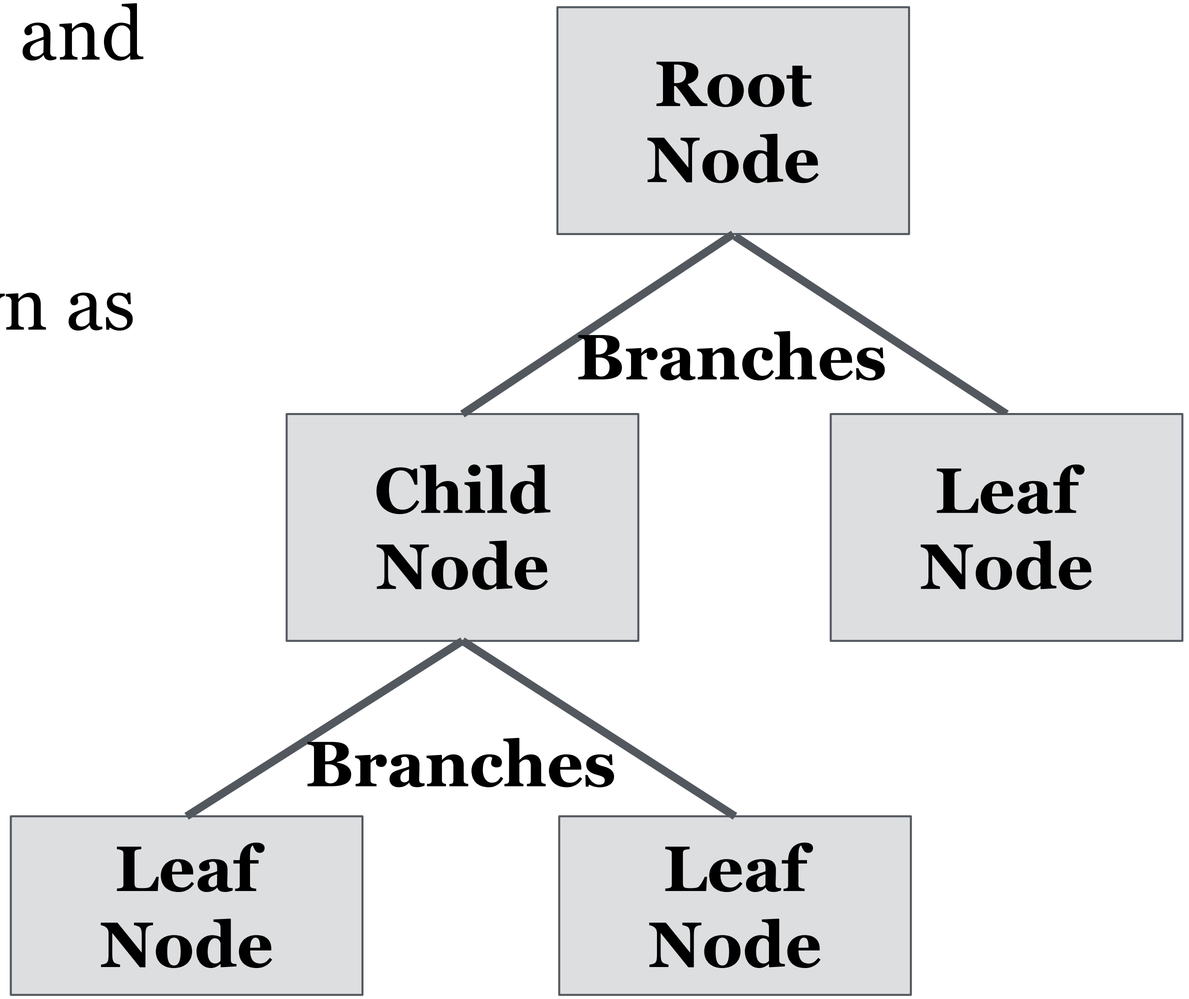
TREES

- Trees are a data structure made up of *nodes* and *branches*
- Each node typically has two or more branches that connect it to its *children*



TREES

- Each child is another node in the tree and contains its own *subtree*
- Nodes without any children are known as *leaf* nodes



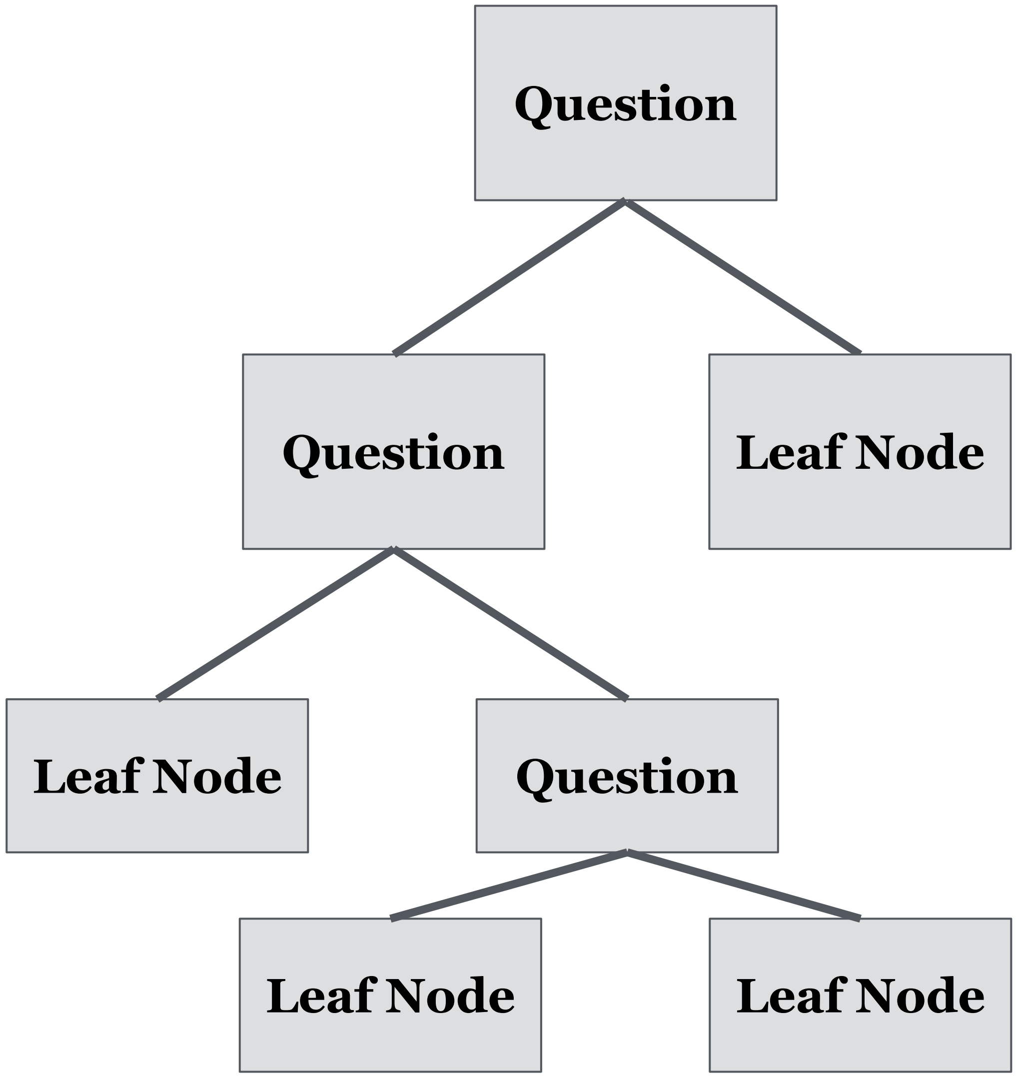
DECISION TREES

- A *decision tree* contains a question at every node

- Depending upon the answer to the question, we proceed down the left or right branch of the tree and ask another question

- Once we don't have any more questions (at the *leaf* nodes), we make a prediction

- *Note*: The next question is always dependent on the last



DECISION TREES

- Let's suppose we want to predict if an article is a *news* article
  - What questions should we ask to make a prediction?
  - How many questions should we ask?

DECISION TREES

- We may start by asking: does it mention a President?
  - If it does, it must be a news article
- If not, let’s ask another question: does the article contain other political features?
- If not, does the article contain references to political topics?
- We could keep going on in this manner until we were satisfied

ACTIVITY: KNOWLEDGE CHECK

EXERCISE

ANSWER THE FOLLOWING QUESTIONS

Let’s work as a class to accomplish the following:

1. Using our **StumpleUpon dataset**, try building a decision tree to predict whether a given article is *evergreen*

DELIVERABLE

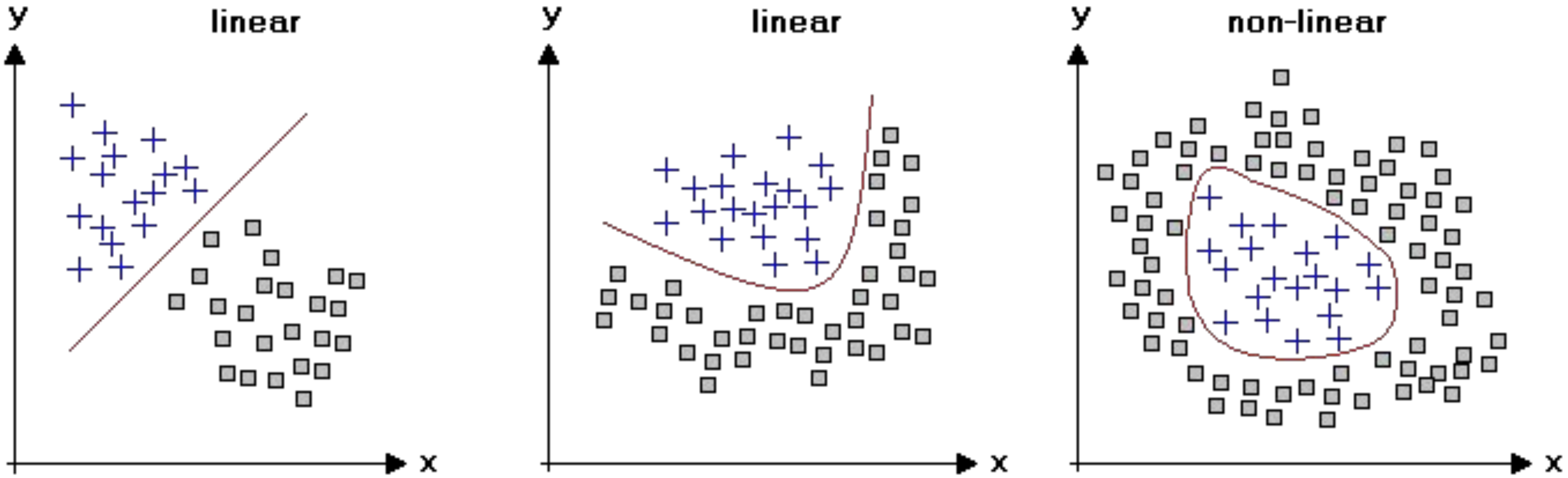
Our decision tree

COMPARISON TO PREVIOUS MODELS

- Decision trees are *non-linear*, an advantage over logistic regression
    - A *linear* model is one in which a change in an input variable has a constant change on the output variable

COMPARISON TO PREVIOUS MODELS

- Linear vs. non-linear classification models



COMPARISON TO PREVIOUS MODELS

- Example: the relationship between years of education and salary
  - In a *linear* model, the increase in salary from 10 to 15 years of education would be the same as the increase from 15 to 20 years
  - In a *non-linear* model, salary can change dramatically for years 0-15 and negligibly from years 15-20

- Trees automatically contain <u>interaction of features</u>, since each question is dependent on the last

TRAINING A DECISION TREE MODEL

- Decision models are useful for **feature selection**
- i.e. Training a decision model is deciding the best set of questions to ask
- A *good question* will be one that best segregates the positive group from the negative group and then narrows in on the correct answer
- For example, in our news article decision tree, the best question is one that creates two groups:
    - news stories
    - non-news stories

TRAINING A DECISION TREE MODEL

- We can quantify the *purity* of the separation of groups using **Classification error, Entropy,** or **Gini Coefficient**

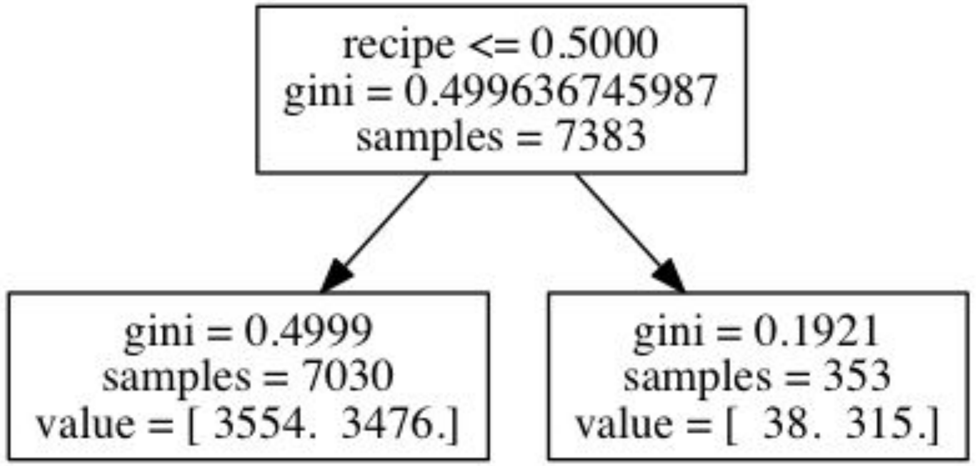
- We want to choose the question that gives us the best *change* in our purity measure
  - At each step, we can ask, "Given our current set of data points, which question will make the largest change in purity?"

- This is done *recursively** for each new set of two groups until we reach a stopping point

* See class_notes.ipynb for example of recursion

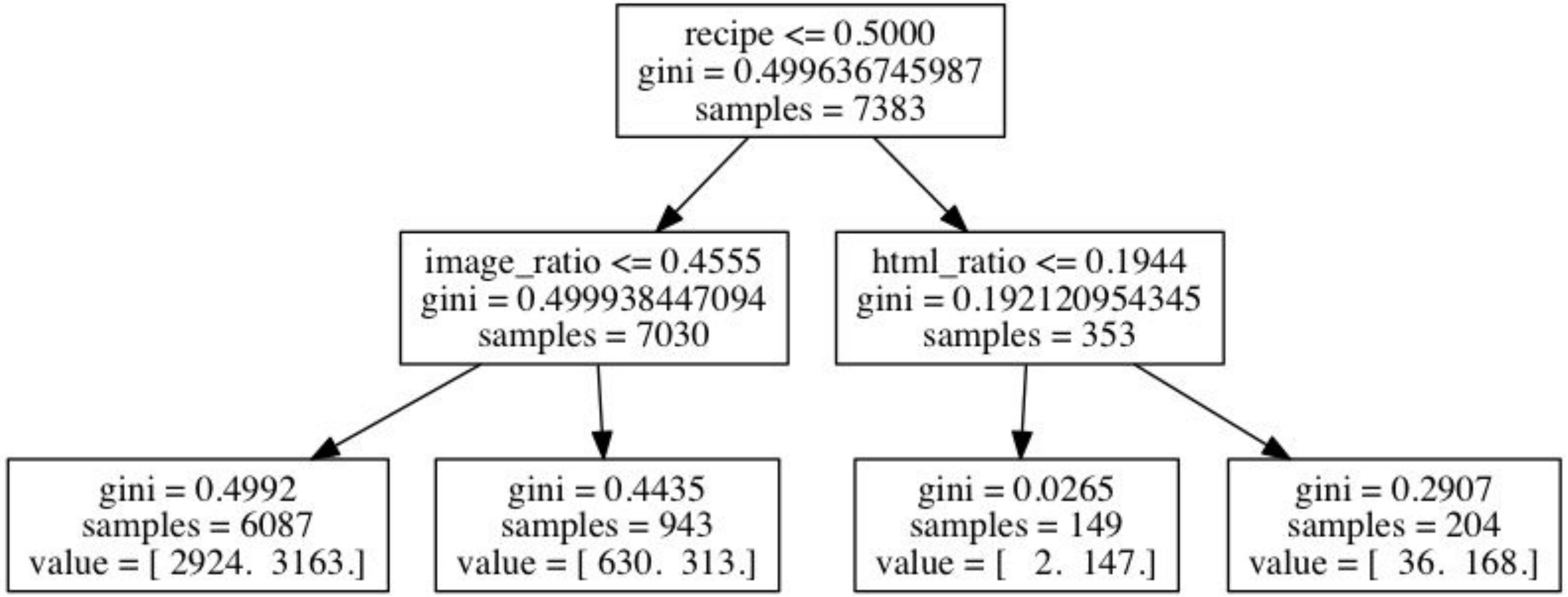
TRAINING A DECISION TREE MODEL

- Let's build a sample tree for our evergreen prediction problem
  - Assume our features are:
    - whether the article contains a recipe
    - the image ratio
    - the html ratio
- First, let's choose the feature that gives us the highest purity, the recipe feature



TRAINING A DECISION TREE MODEL

- We can take each side of the tree and repeat the process



- We can continue this process until we have asked as many questions as we want or until our leaf nodes are completely pure

MAKING PREDICTIONS FROM A DECISION TREE

- Predictions are made by answering each of the questions
- Once we reach a leaf node, our prediction is made by taking the majority label of the training samples that fulfill the questions
- In our sample tree, if we want to classify a new article, ask:
  - Does the article contain the word recipe?
  - If it doesn't, does the article have a lot of images?
  - If it does, then 630 / 943 article are evergreen
    - So we can assign a 0.67 probability for evergreen sites

ACTIVITY: KNOWLEDGE CHECK

EXERCISE

ANSWER THE FOLLOWING QUESTIONS

1. How do we classify a new article?
2. How do we make predictions from a decision tree?

DELIVERABLE

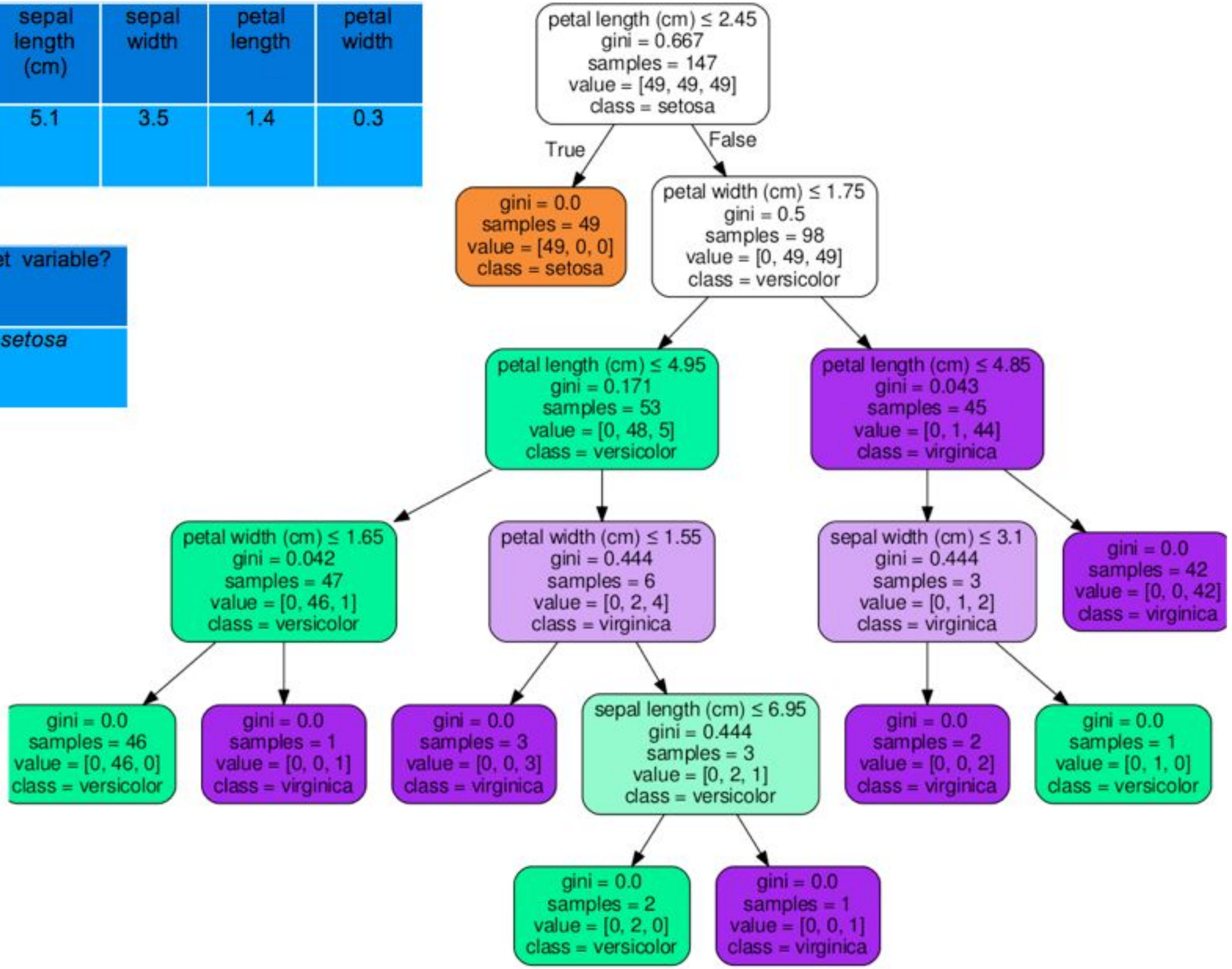
Answers to the above questions

ACTIVITY: KNOWLEDGE CHECK

EXERCISE

sample #	sepal length (cm)	sepal width	petal length	petal width
17	5.1	3.5	1.4	0.3

target variable?
setosa

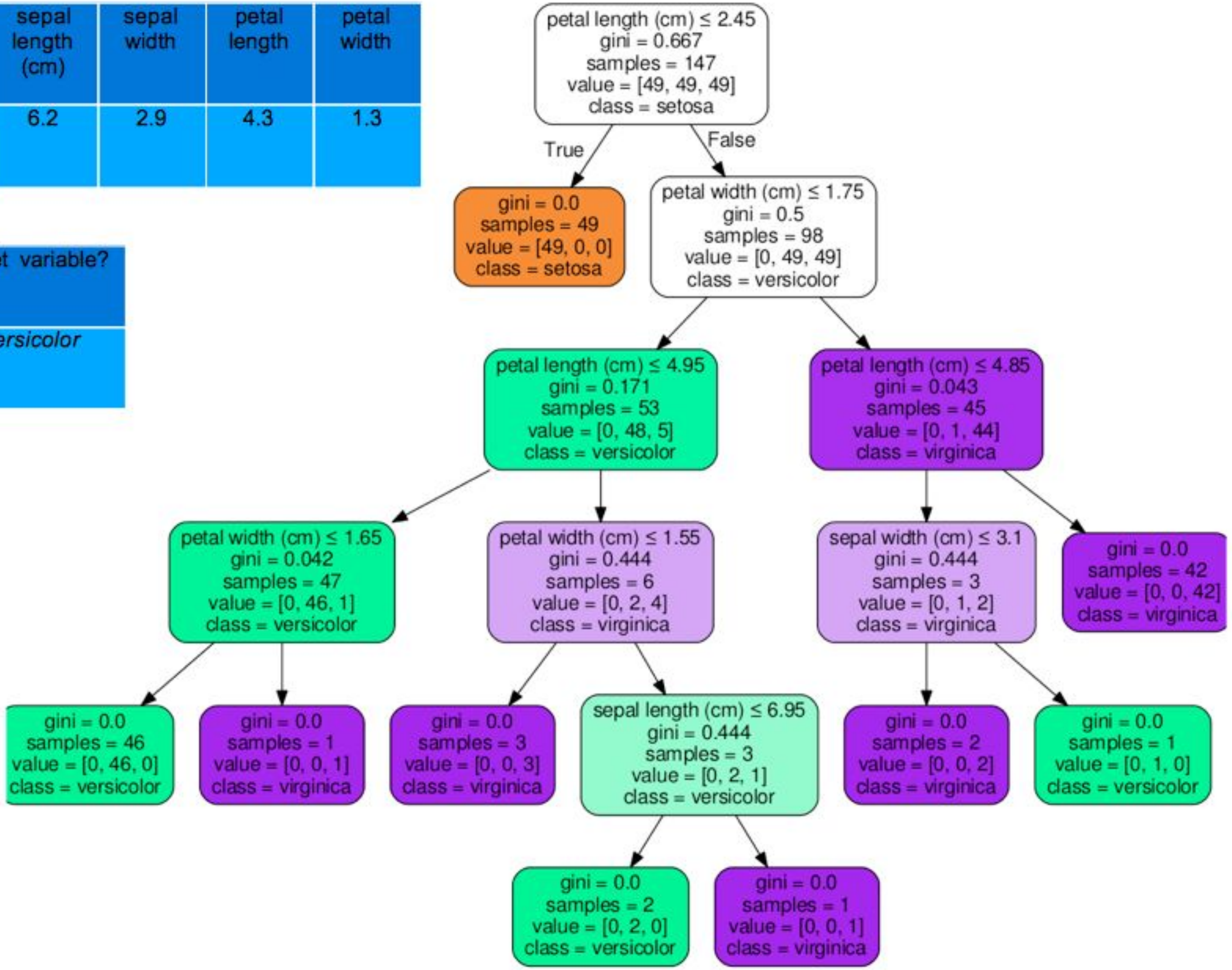


ACTIVITY: KNOWLEDGE CHECK

EXERCISE

sample #	sepal length (cm)	sepal width	petal length	petal width
97	6.2	2.9	4.3	1.3

target variable?
versicolor



GUIDED PRACTICE

DECISION TREES IN SCIKIT-LEARN

ACTIVITY: DECISION TREES IN SCIKIT-LEARN

EXERCISE

DIRECTIONS (15 minutes)

1. In the starter code notebook, work through the exercises in “12.5 Decision Trees in scikit-learn”
2. Work on evaluating the decision tree using cross-validation methods
3. What metrics would work best? Why?

Check: Are you able to evaluate the decision tree model using cross-validation methods?

DELIVERABLE

Completed exercises and answer to #3

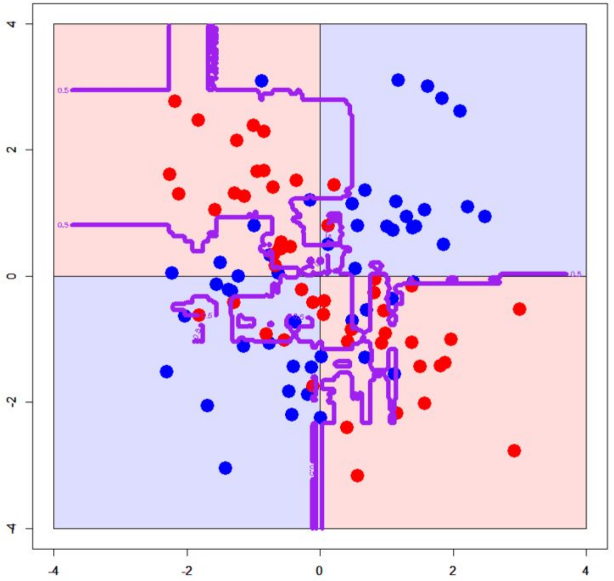
DEMO

OVERFITTING IN DECISION TREES

OVERFITTING IN DECISION TREES

- 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



OVERFITTING IN DECISION TREES

- An unconstrained decision tree can learn an extreme tree (e.g. one feature for each word in a news article)
- We can limit our decision trees using a few methods:
  - Limiting the number of questions (nodes) a tree can have
  - Limiting the number of samples in the leaf nodes

ACTIVITY: KNOWLEDGE CHECK

EXERCISE

ANSWER THE FOLLOWING QUESTIONS

1. Why are decision trees generally thought of as weak models?
2. How can we limit our decision trees?

DELIVERABLE

Answers to the above questions

GUIDED PRACTICE

ADJUSTING DECISION TREES TO AVOID OVERFITTING

ACTIVITY: ADJUSTING DECISION TREES TO AVOID OVERFITTING

EXERCISE

DIRECTIONS (15 minutes)

1. You can control for overfitting in decision trees by adjusting one of the following parameters:
   a. `max_depth`: Control the maximum number of questions
   b. `min_samples_in_leaf`: Control the minimum number of records in each node
2. Test each of these parameters in the starter code notebook

DELIVERABLE

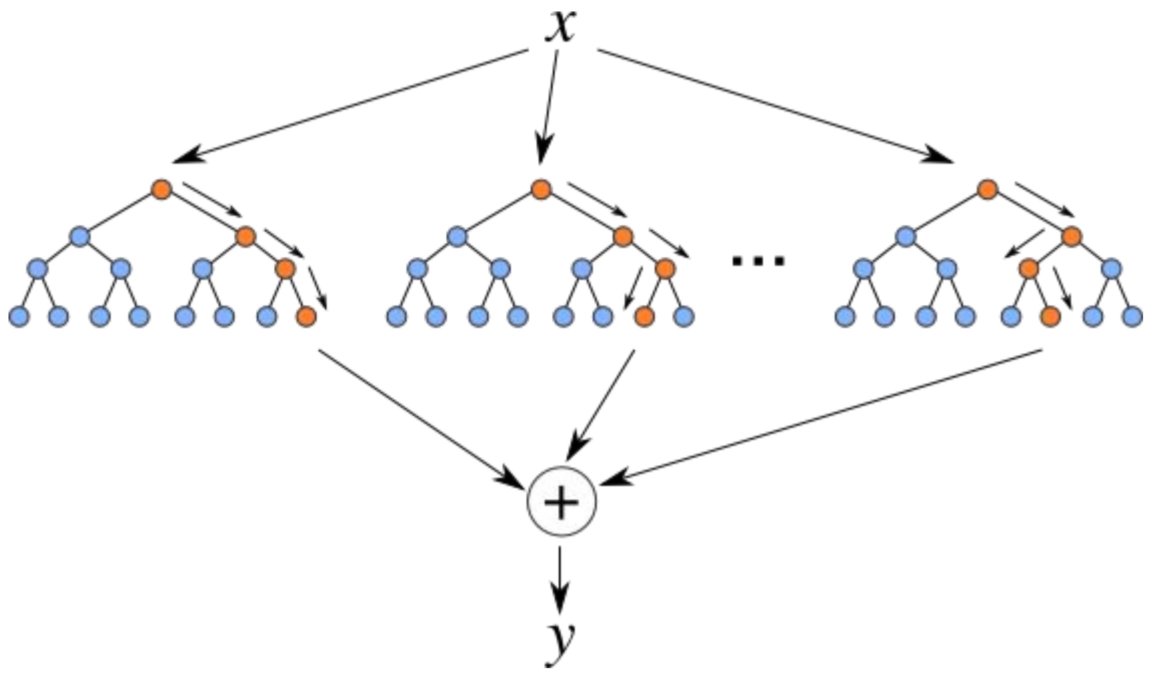
Code using the above parameters

INTRODUCTION

RUNNING THROUGH THE RANDOM FORESTS

RUNNING THROUGH THE RANDOM FORESTS

- Random forest models are one of the most widespread classifiers used
- They are relatively simple to use and help avoid overfitting
- Random Forests are an *ensemble* or collection of individual decision trees



PROS AND CONS OF RANDOM FORESTS

Advantages

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

Disadvantages

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

TRAINING A RANDOM FOREST

- Training a random forest model involves training many decision tree models

- Since decision trees overfit easily, we use many decision trees together and randomize the way they are created

TRAINING A RANDOM FOREST

- Random Forest Algorithm

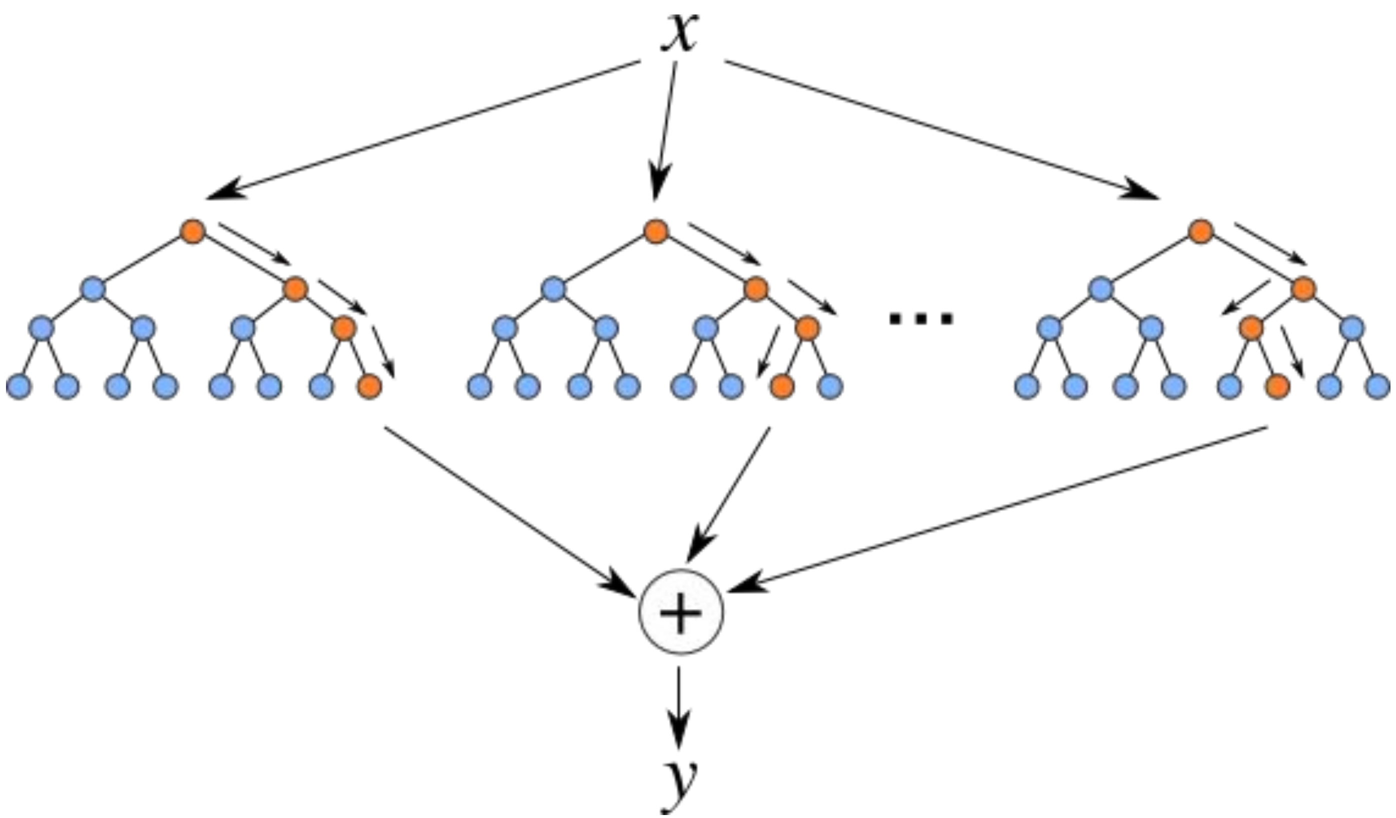
  a. Take a bootstrap sample of the dataset

  b. 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

  c. Repeat this for *N* trees

PREDICTIONS 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



GUIDED PRACTICE

REGRESSION WITH DECISION TREES AND RANDOM FORESTS

ACTIVITY: REGRESSION WITH DECISION TREES & RANDOM FORESTS

EXERCISE

DIRECTIONS (20 minutes)

1. Build a random forest model to predict the evergreen-ness of a website
    - Remember to use the parameter `n_estimators` to control the number of trees used in the model
2. Take note of the features that give the **best splits** to determine the most important features
    - See **Section 12.7** in the starter_code

DELIVERABLE

The models mentioned above

INDEPENDENT PRACTICE

EVALUATE RANDOM FOREST USING CROSS-VALIDATION

ACTIVITY: EVALUATE RANDOM FOREST USING CROSS-VALIDATION

EXERCISE

DIRECTIONS (25 minutes)

1. Building upon the previous Guided Practice, **add** any input variables to the model that you think may be relevant

2. For each feature:
   a. Evaluate the model for improved predictive performance using cross-validation
   b. Evaluate the importance of the feature
   - See **Section 12.8** in the starter_code

3. **Bonus**:
   - Just like the ‘recipe’ feature, add in similar text features and evaluate their performance

DELIVERABLE

Newly created features and models

CONCLUSION

TOPIC REVIEW

REVIEW Q&A

- 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?

COURSE

BEFORE NEXT CLASS

DUE DATE

- Final Project, Deliverable 2 due: *Next Thurs* (5/3)

LESSON

Q & A

LESSON

EXIT TICKET

DON'T FORGET TO FILL OUT YOUR EXIT TICKET