GENERAL ASSEMBLY

DECISION TREES AND RANDOM FORESTS

Christoph Rahmede
GA DATA SCIENCE

LEARNING OBJECTIVES

- Build decision tree models for classification and regression
- Discriminate the differences between linear and non-linear models
- Build random forest models for classification and regression
- Extract the most important predictors in a random forest model

PRE-WORK

PRE-WORK REVIEW

- Use Seaborn to create plots
- 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

REVIEW

- Any questions from last class?

ACTIVITY: KNOWLEDGE CHECK

ANSWER THE FOLLOWING QUESTIONS

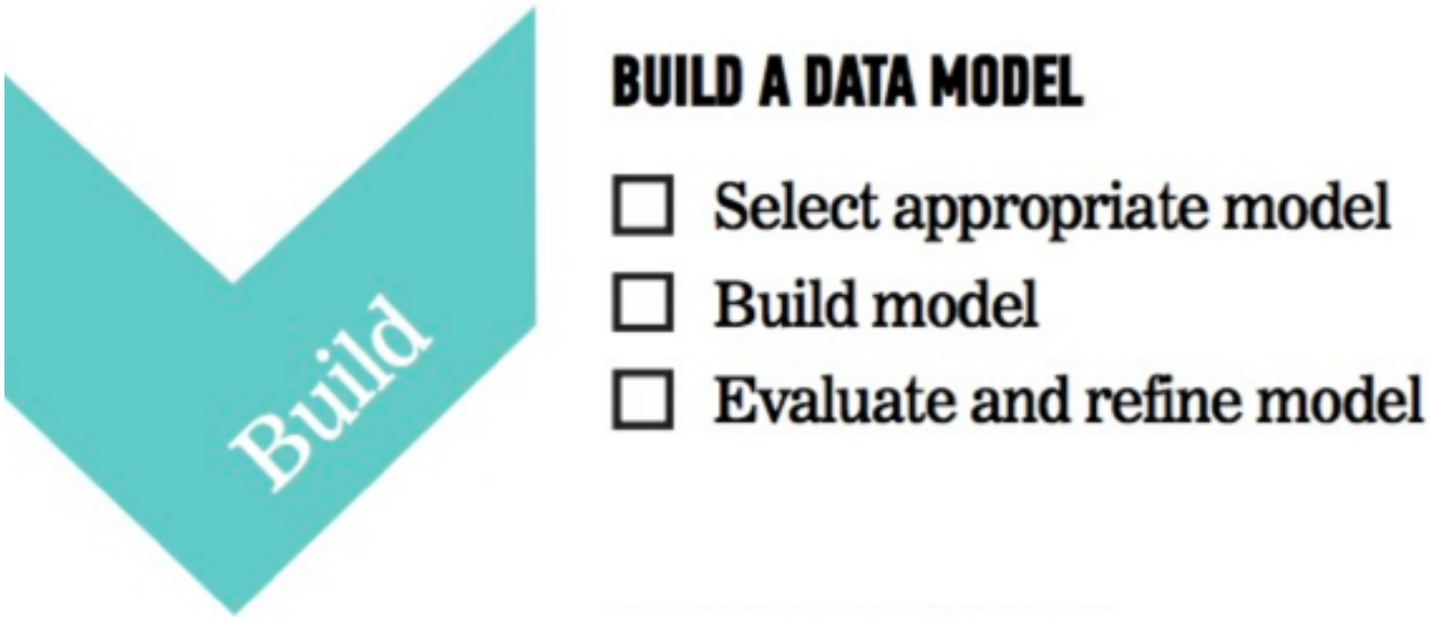
1. Define accuracy, precision and recall of a model. When would you use which?
2. What are use cases for logistic regression?
3. What does logistic regression predict?
4. What is the meaning of the coefficients in logistic regression?

OPENING

DECISION TREES AND RANDOM FORESTS

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

DIRECTIONS

We will be using a dataset from StumbleUpon, a service that recommends webpages to users based on their interests. They like to recommend "evergreen" sites, sites that are always relevant. This usually means websites that avoid topical content and focus on recipes, how-to guides, art projects, etc. We want to determine important characteristics for "evergreen" websites. Follow these prompts to get started:

1. Break into groups.
2. Prior to looking at the data, brainstorm 3-5 characteristics that would be useful for predicting evergreen websites.
3. After looking at 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 evergreeness? Compute or plot the percent of evergreen news sites.

ACTIVITY: EXPLORE THE DATASET

DIRECTIONS

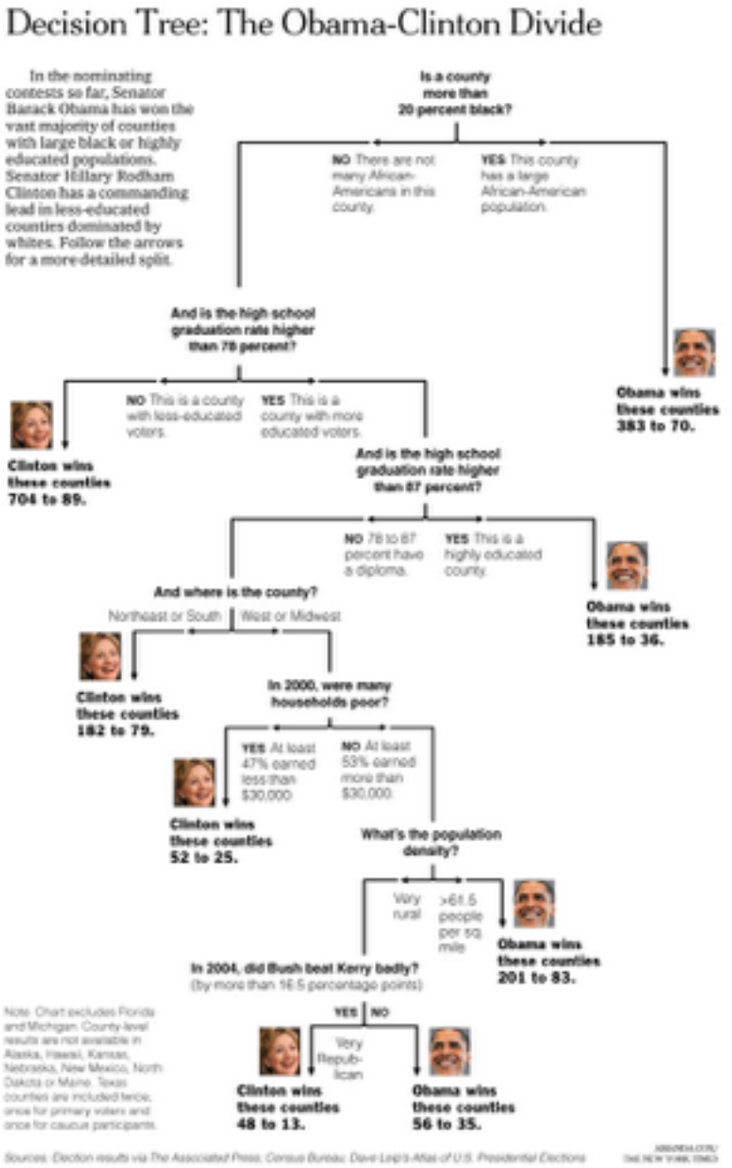
5. In general, does category affect evergreeness? Plot the rate of evergreen sites for all Alchemy categories.
6. How many articles are there per category?
7. Create a feature for the title containing “recipe”. Is the percentage of evergreen websites higher or lower on pages that have “recipe” in the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

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

TRAINING DECISION TREES

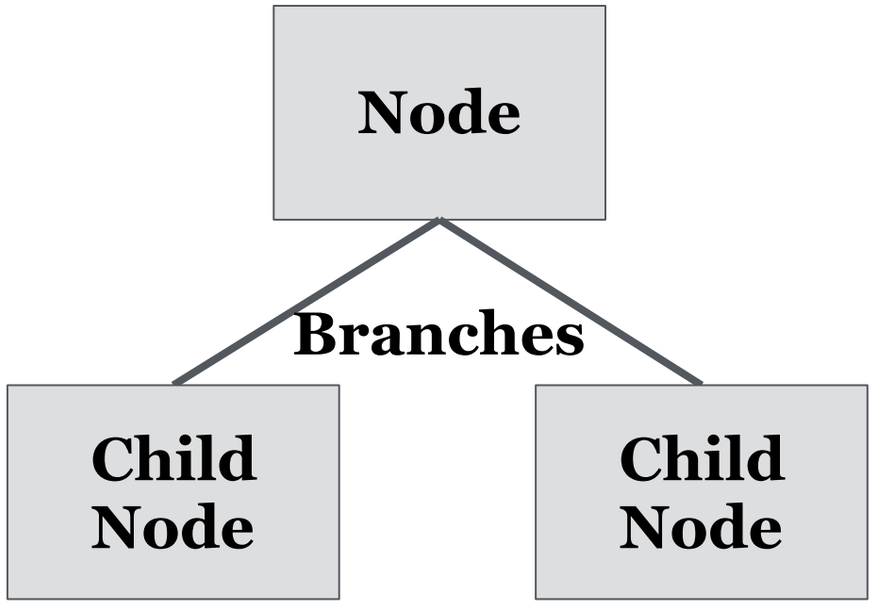
INTUITION BEHIND DECISION TREES

- Decision trees are like the game “20 questions”. They make decisions 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 question should reduce the search space as much as possible.



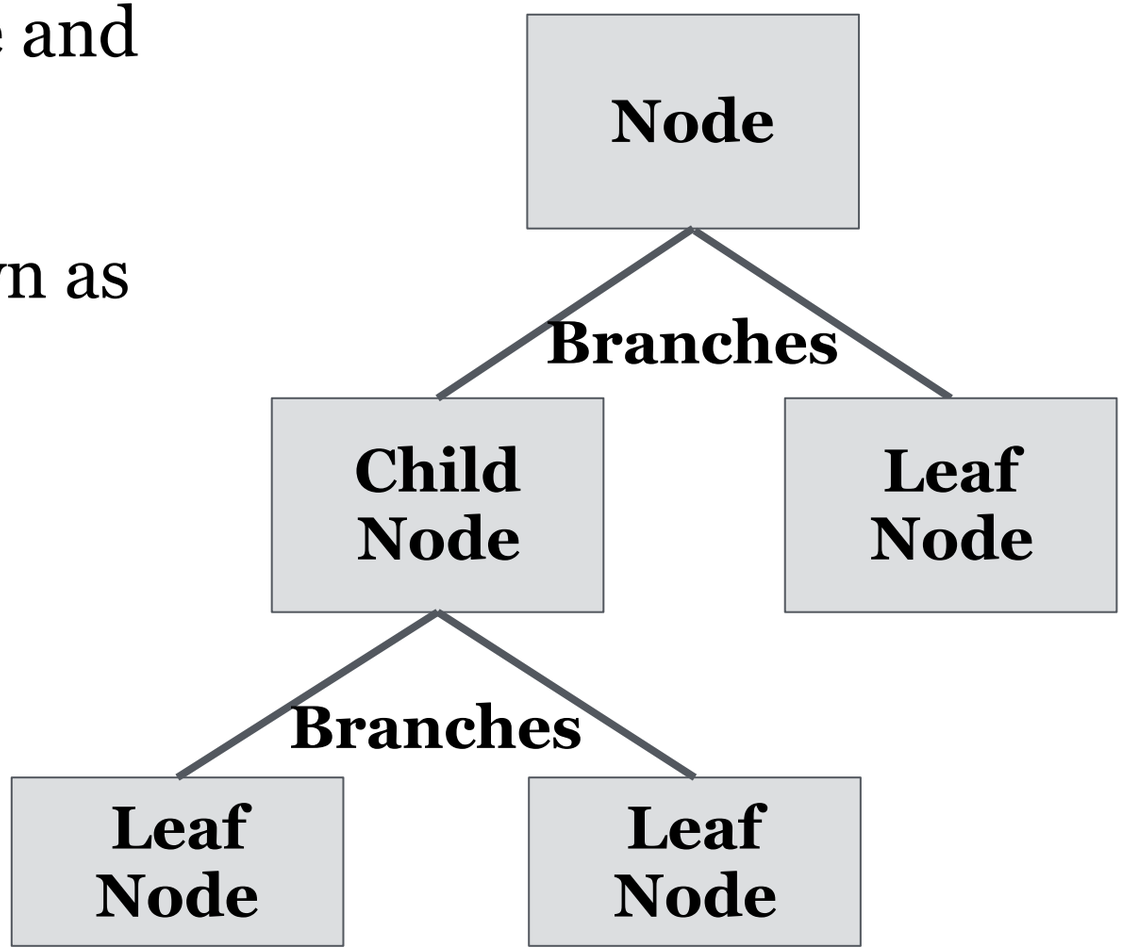
TREES

- 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.



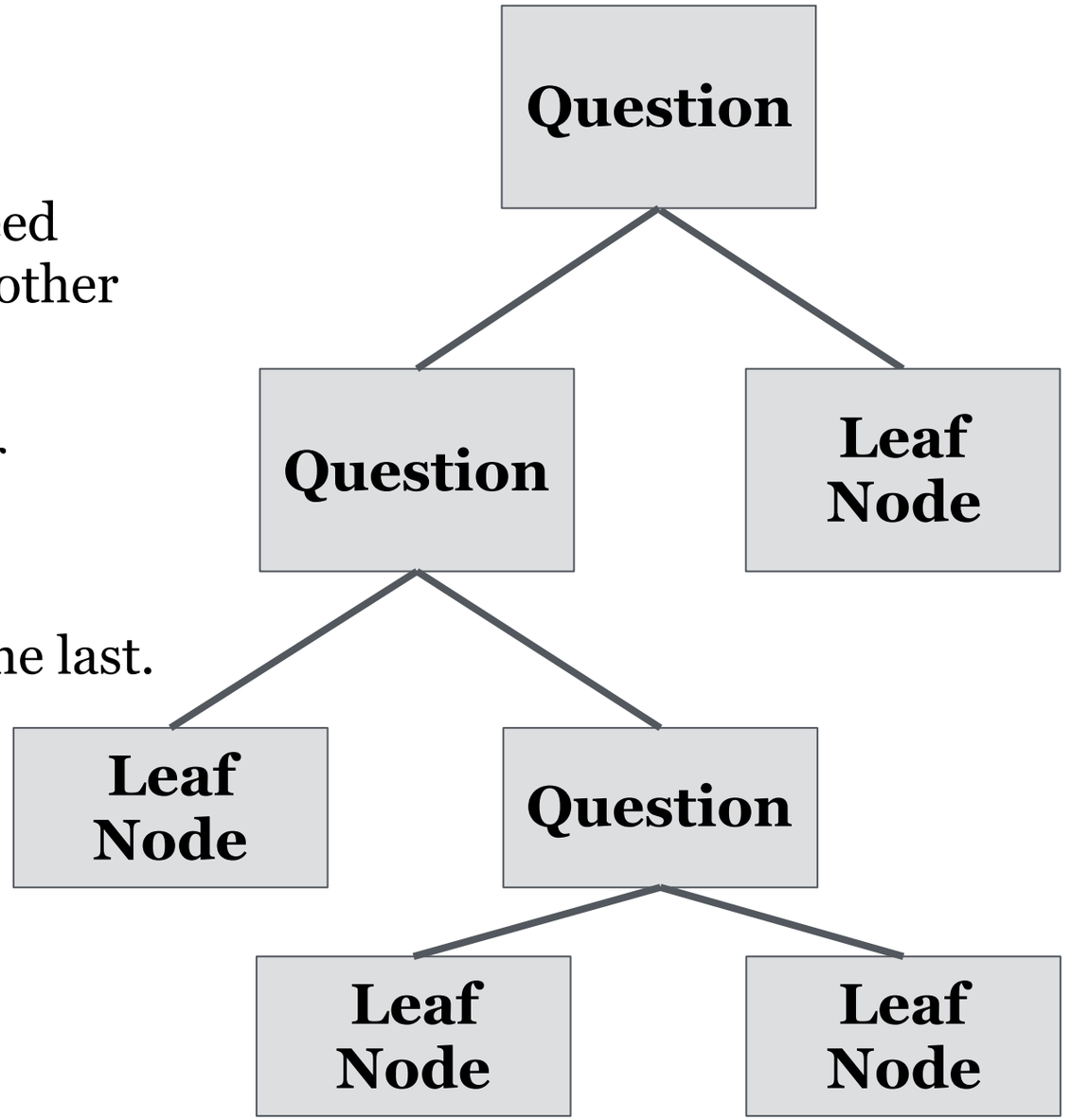
TREES

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



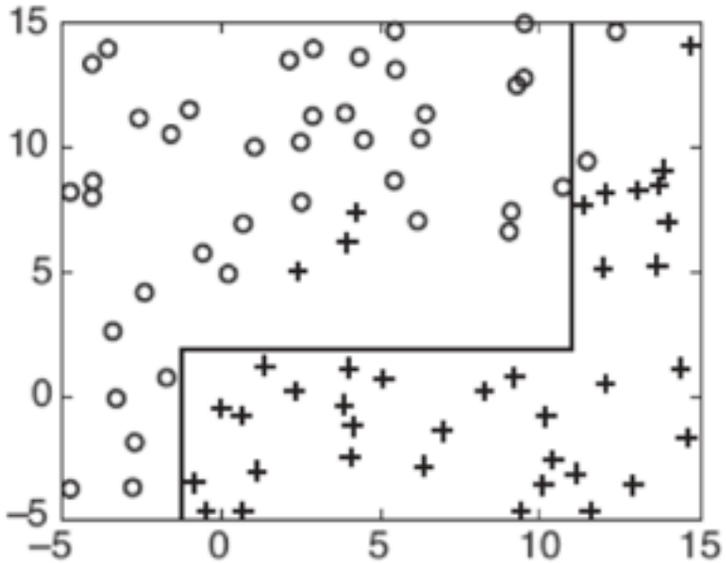
DECISION TREES

- A *decision tree* contains a question at every node.
- Depending on 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

- The questions partition our feature space into different rectangular regions.
- A point is assigned to the majority class of its region in feature space.



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

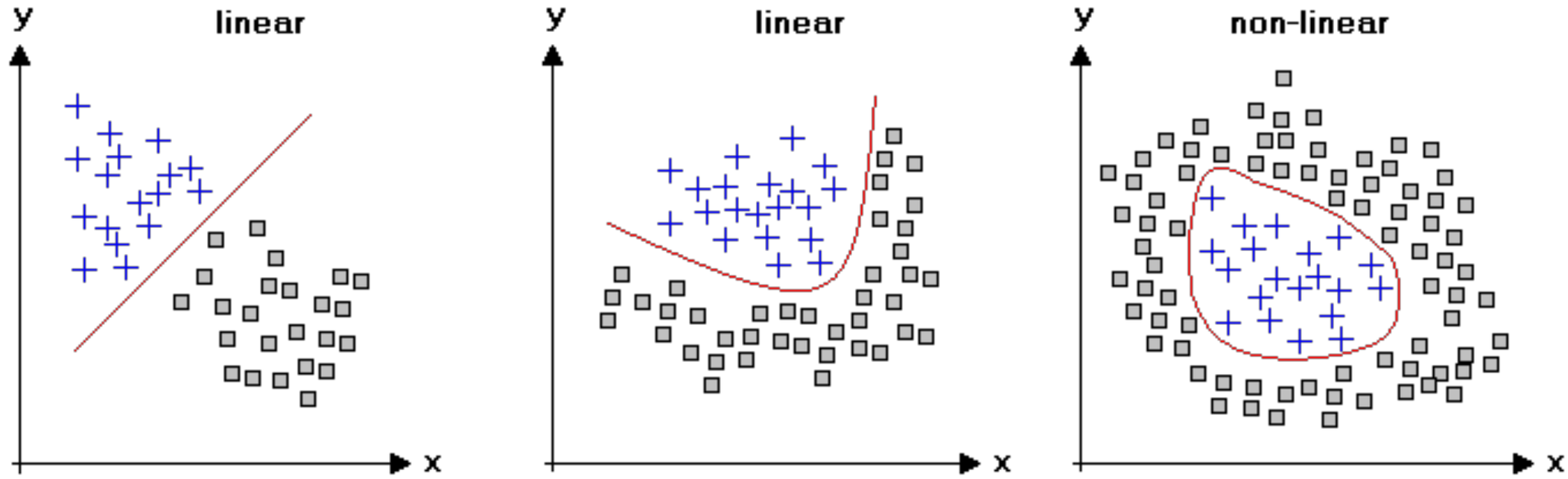
- We may start by asking: does it mention a President?
- If it does, it must be a news article.
- If not, let's ask another question: does the article contain other political features?
- If not, does the article contain references to political topics?
- We could keep going on in this manner until we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

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

- An example of this difference is the relationship between years of job experience and salary. In a *linear* model, the increase in salary from 10 to 15 years of job experience would be the same as the increase in salary from 15 to 20 years of job experience. In a *non-linear* model, salary can change dramatically for years 0-15 and negligibly from years 15-20.

- Trees automatically contain interaction of features, since each question is dependent on the last.

TRAINING A DECISION TREE MODEL

- 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, one that is mostly news stories and one that is mostly 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.
- Features gain importance the more often they are used to perform a split.

ACTIVITY: KNOWLEDGE CHECK

ANSWER THE FOLLOWING QUESTIONS

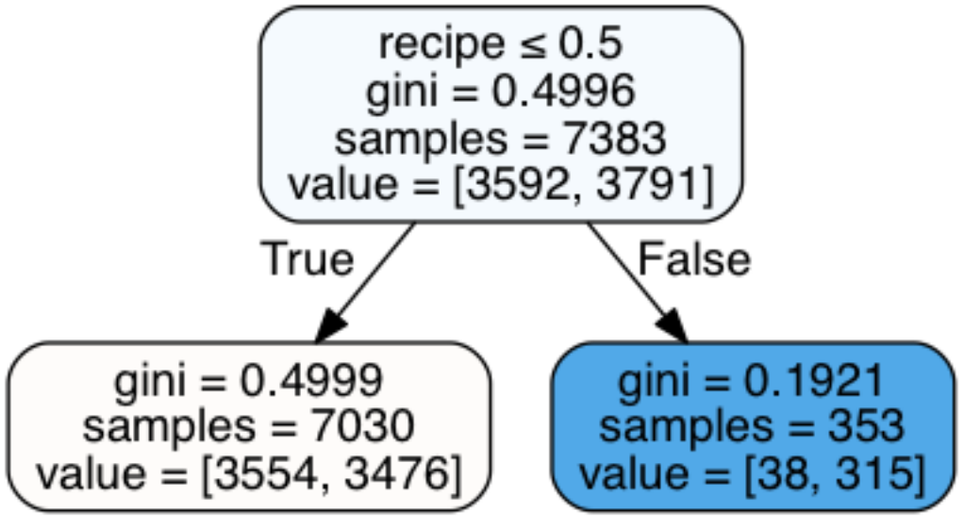
Let's work as a class to accomplish the following:

1. Using our StumbleUpon dataset, try to predict whether a given article is evergreen.
2. Build a decision tree to determine the above.

TRAINING A DECISION TREE MODEL

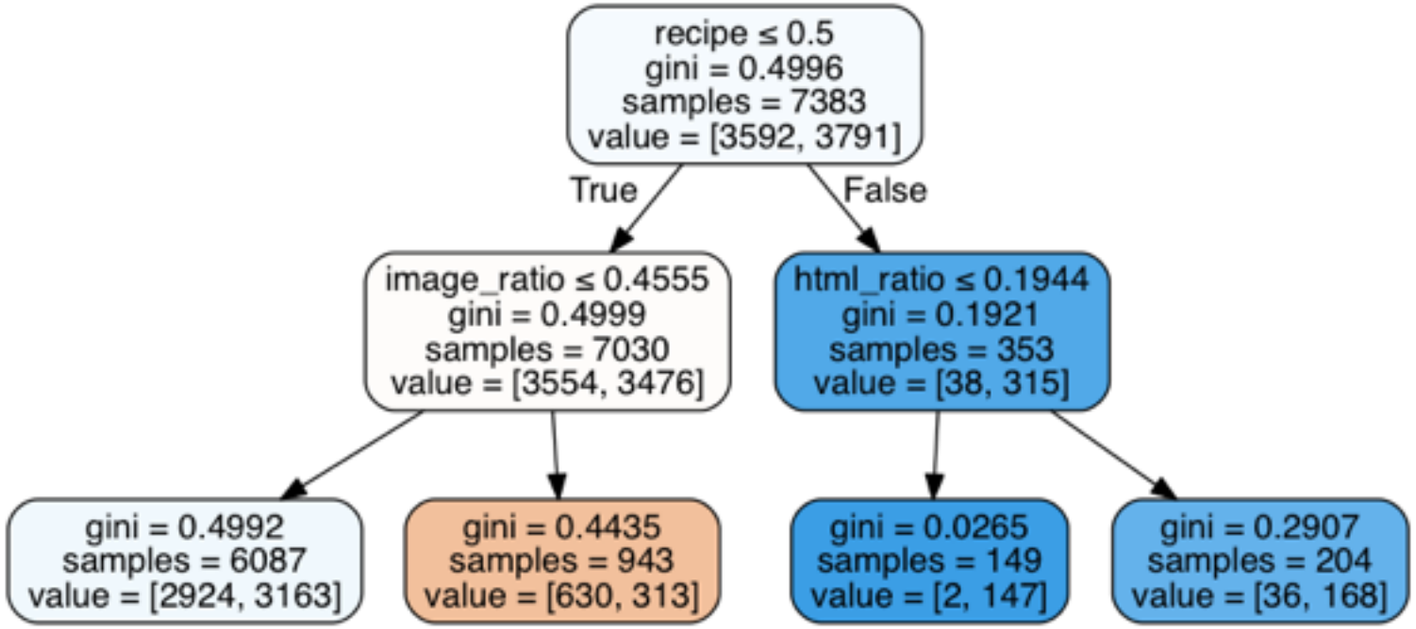
- Let's build a sample tree for our evergreen prediction problem. Assume our features are whether the article contains a recipe, the image ratio, the html ratio.

- 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 fulfil 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 313 / 943 articles are evergreen.
    - So we can assign a 0.33 probability for being evergreen.

ACTIVITY: KNOWLEDGE CHECK

ANSWER THE FOLLOWING QUESTIONS

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

GUIDED PRACTICE

DECISION TREES IN SCIKIT-LEARN

ACTIVITY: DECISION TREES IN SCIKIT-LEARN

DIRECTIONS

1. In the starter code notebook, work through the exercises titled "Decision Trees in scikit-learn".
2. In your groups from earlier, work on evaluating the decision tree using cross-validation methods.
3. What metrics work best? Why?

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

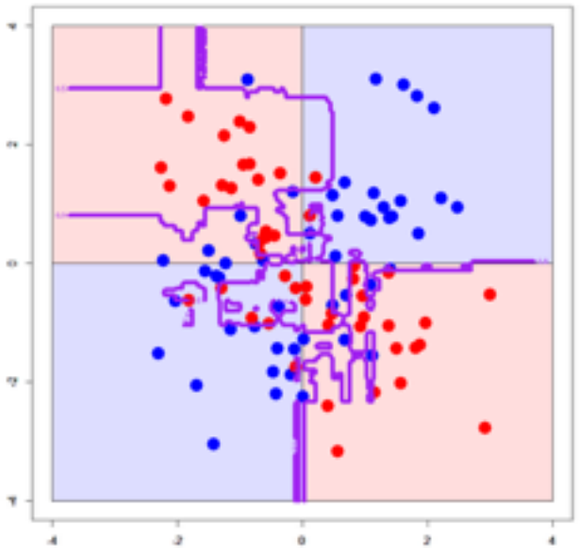
DEMO

OVERFITTING IN DECISION TREES

OVERFITTING IN DECISION TREES

- Decision trees tend to be weak models because they can easily memorise or overfit a dataset.

- A model is *overfit* when it memorises 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 constrain 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

ANSWER THE FOLLOWING QUESTIONS

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

GUIDED PRACTICE

ADJUSTING DECISION TREES TO AVOID OVERFITTING

ACTIVITY: ADJUSTING DECISION TREES TO AVOID OVERFITTING

EXERCISE

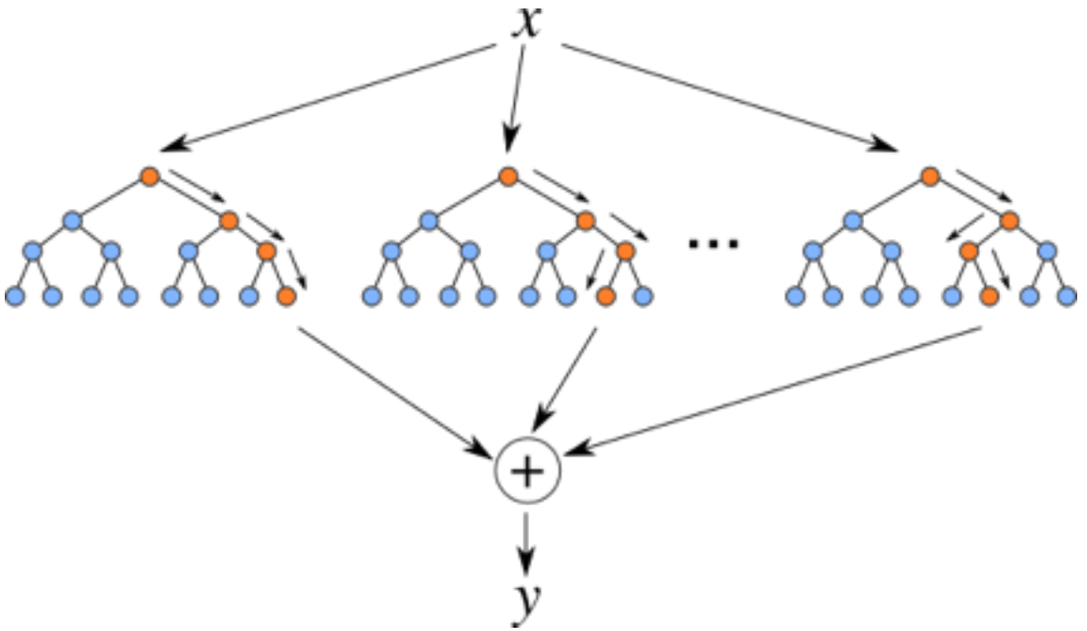
DIRECTIONS

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.

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 randomise the way they are created.

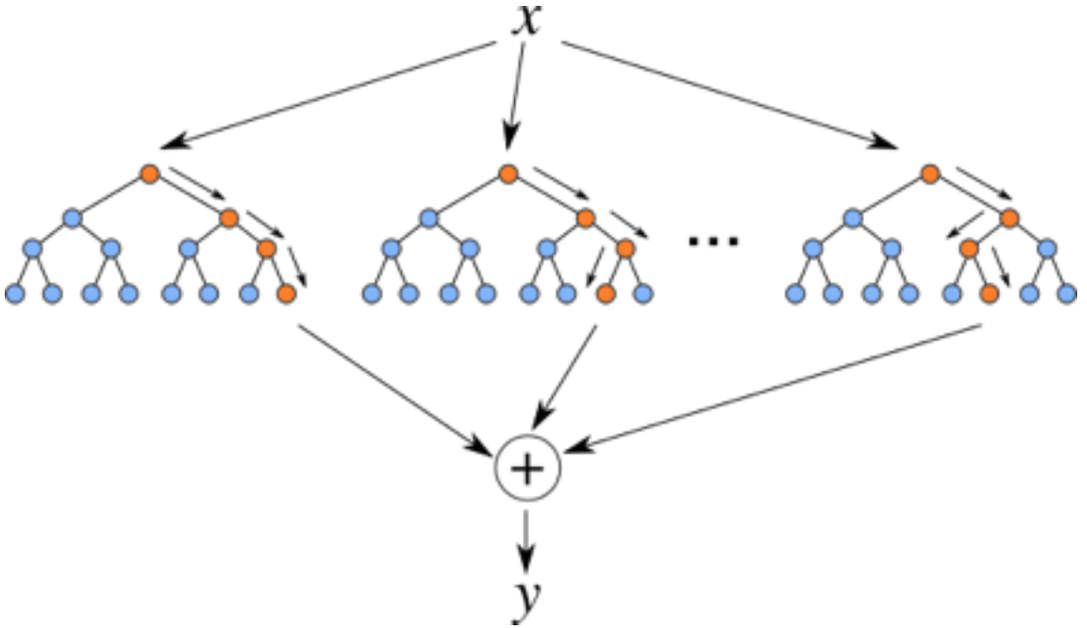
TRAINING A RANDOM FOREST

- 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.



REGRESSION USING A RANDOM FOREST

- Decision trees and random forests can be used for both classification and regression.
- In regression, predictions are made by taking the average value of the samples in the leaf node. You can take the average of the individual trees' predictions.

GUIDED PRACTICE

CLASSIFICATION WITH DECISION TREES AND RANDOM FORESTS

ACTIVITY: CLASSIFICATION WITH DECISION TREES & RANDOM FORESTS

DIRECTIONS

1. Build a random forest model to predict the evergreeness 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.

INDEPENDENT PRACTICE

EVALUATE RANDOM FOREST USING CROSS-VALIDATION

ACTIVITY: EVALUATE RANDOM FOREST USING CROSS-VALIDATION

DIRECTIONS

1. Building on 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.

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

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?

BEFORE NEXT CLASS

DUE DATE

- Project: Final Project, Deliverable 2