GENERAL ASSEMBLY

DECISION TREES AND RANDOM FORESTS

Trevor Lindsay

LEARNING OBJECTIVES

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

COURSE

PRE-WORK

PRE-WORK: TAKE HOME PRACTICE

- Using models built from the flight data problem in last class, work through the same problems. Your data and models should already be accessible.

- Your goals:

  - There are many ways to manipulate this data set. Consider what is a proper "categorical" variable, and keep only what is significant. Aim to have a visual that clearly explain the relationship of variables you've used against the predicted flight delay.

  - Generate the AUC or precision-recall curve (based on which you think makes more sense), and have a statement that defines, compared to a baseline, how your model performs and any caveats.

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 do the coefficients in Logistic Regression represent?

GUIDED PRACTICE

EXPLORE THE DATASET

ACTIVITY: EXPLORE THE DATASET

DIRECTIONS (25 minutes)

We will be using a dataset from StumbleUpon, a service that recommends webpages to users based upon their interests. They like to recommend “evergreen” sites, ones 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 (25 minutes)

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?

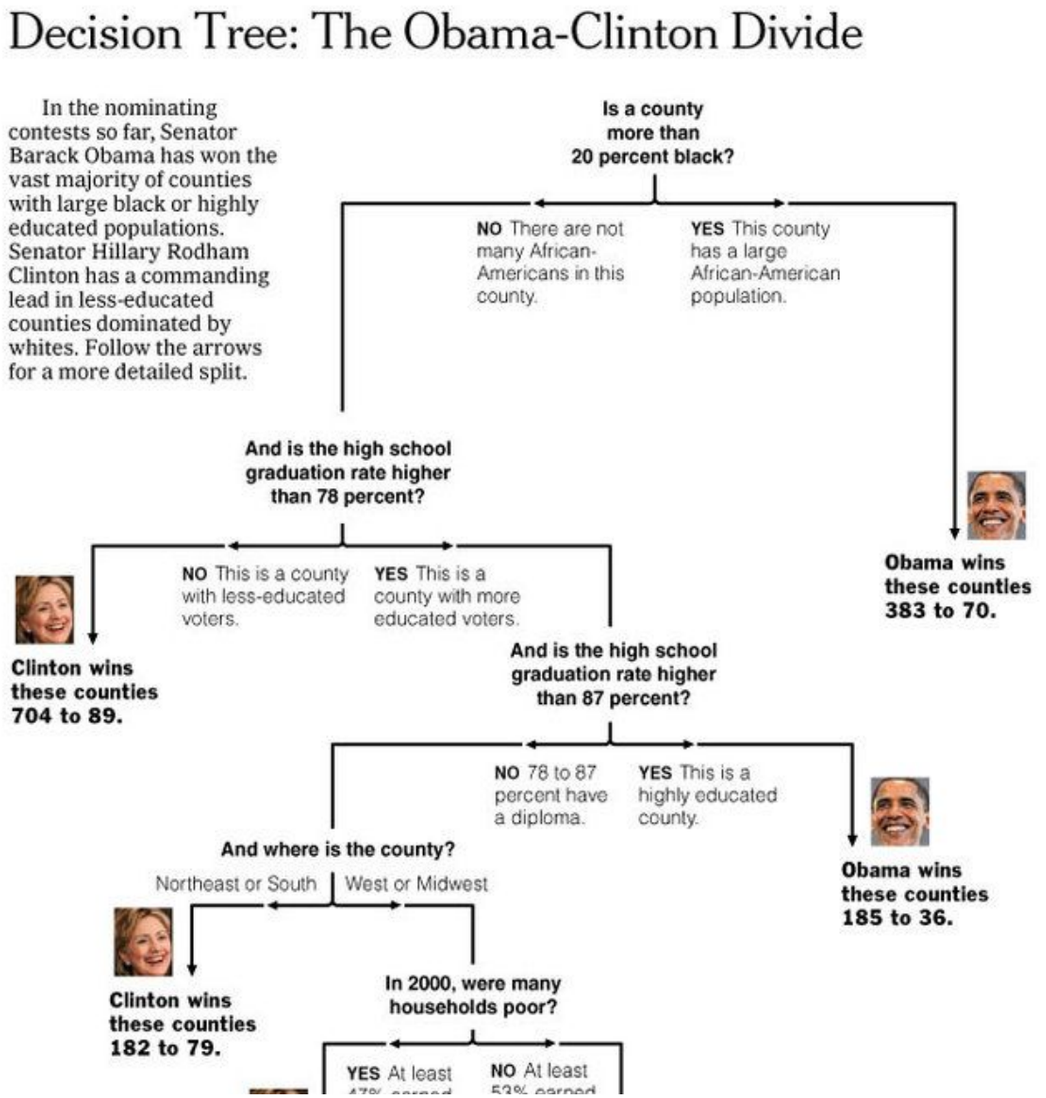
EXPLORE THE DATASET

```
import pandas as pd
import json

# Read in the dataset
data = pd.read_csv("../../data/stumbleupon.tsv", sep='\t')

# Parse out the title and body of the article
data['title'] = data['boilerplate'].map(lambda x: json.loads(x).get('title', ''))
data['body'] = data['boilerplate'].map(lambda x: json.loads(x).get('body', ''))

# Show a preview of the data
data.head()
```

TRAINING DECISION TREES

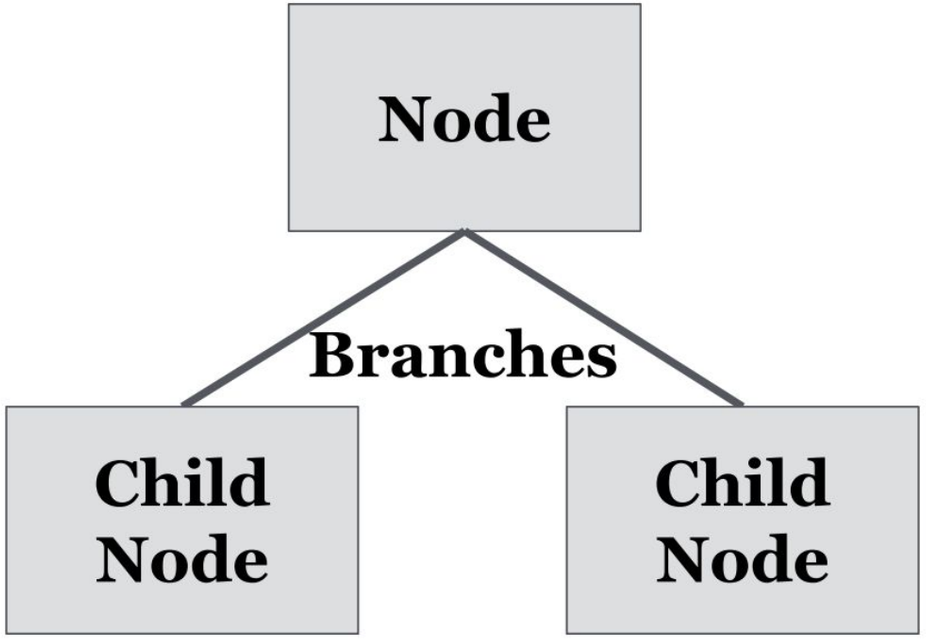
INTUITION BEHIND DECISION TREES

- Decision trees are like the game "20 questions" -- they make a decision by answering a series of questions (most often binary)
- We want the smallest set of questions to get to the right answer
- Each question should reduce the search space as much as possible



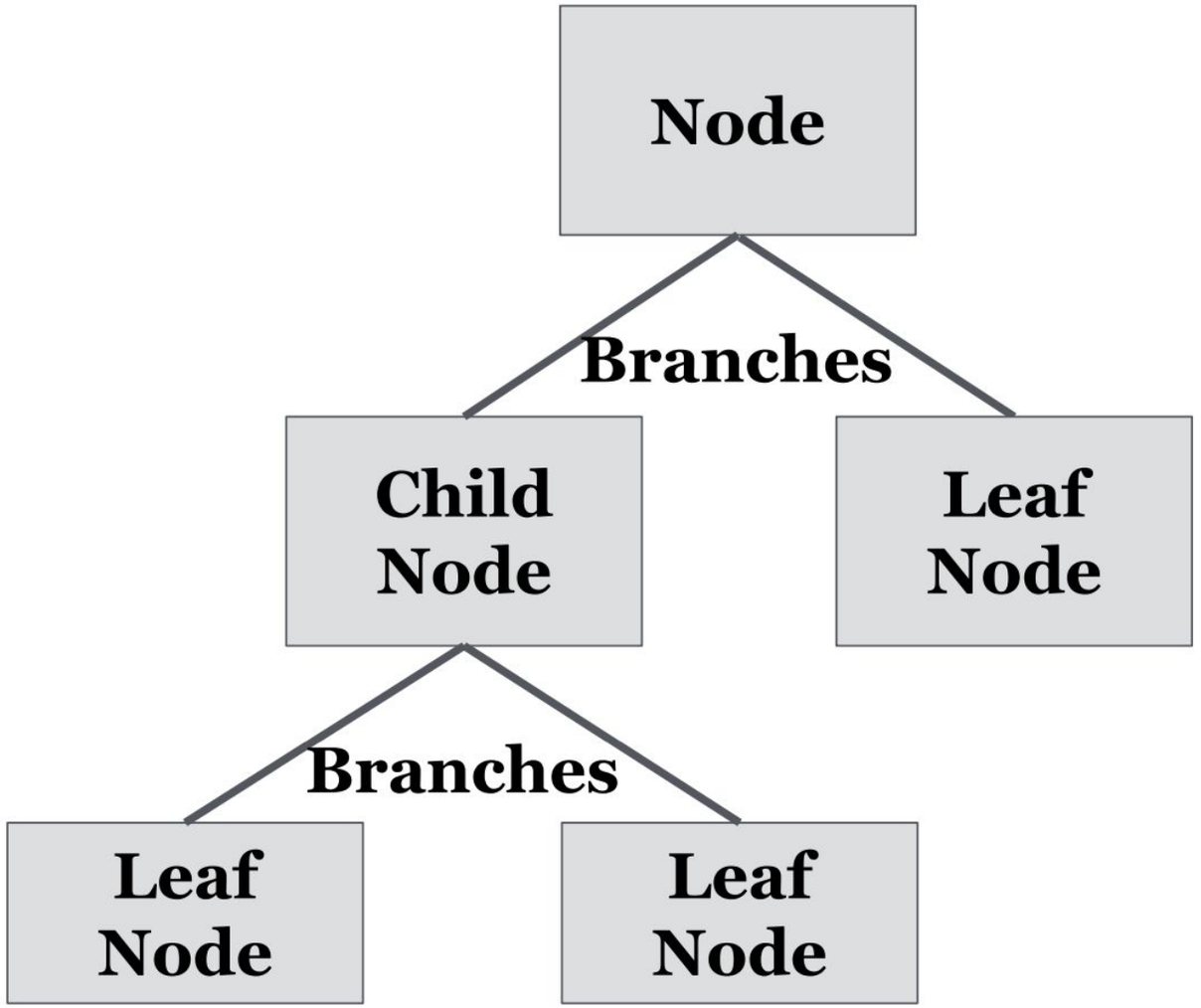
TREES

- 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



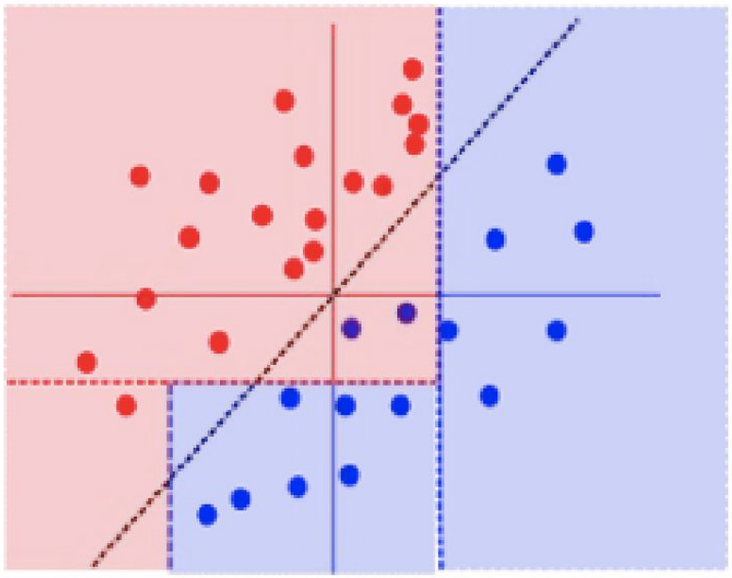
DECISION TREE

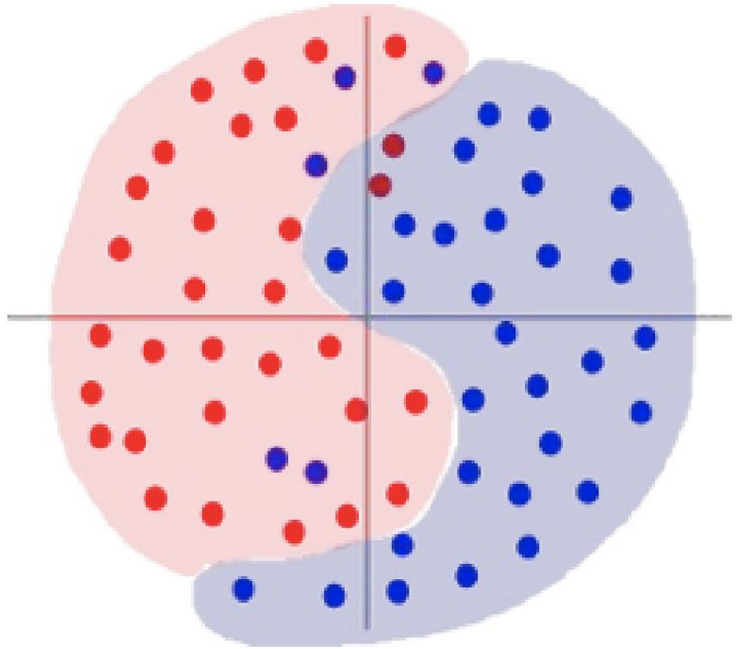
- A decision tree contains a *question* at every node
- Depending upon the answer to the question, we proceed down the left or right branch of the tree and ask another question
- Once we don’t have anymore questions (at the leaf nodes), we make a prediction

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





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 evergreen article decision tree, the best question is one that creates two groups, one that is mostly evergreen websites and one that is mostly non-evergreen websites

TRAINING A DECISION TREE MODEL

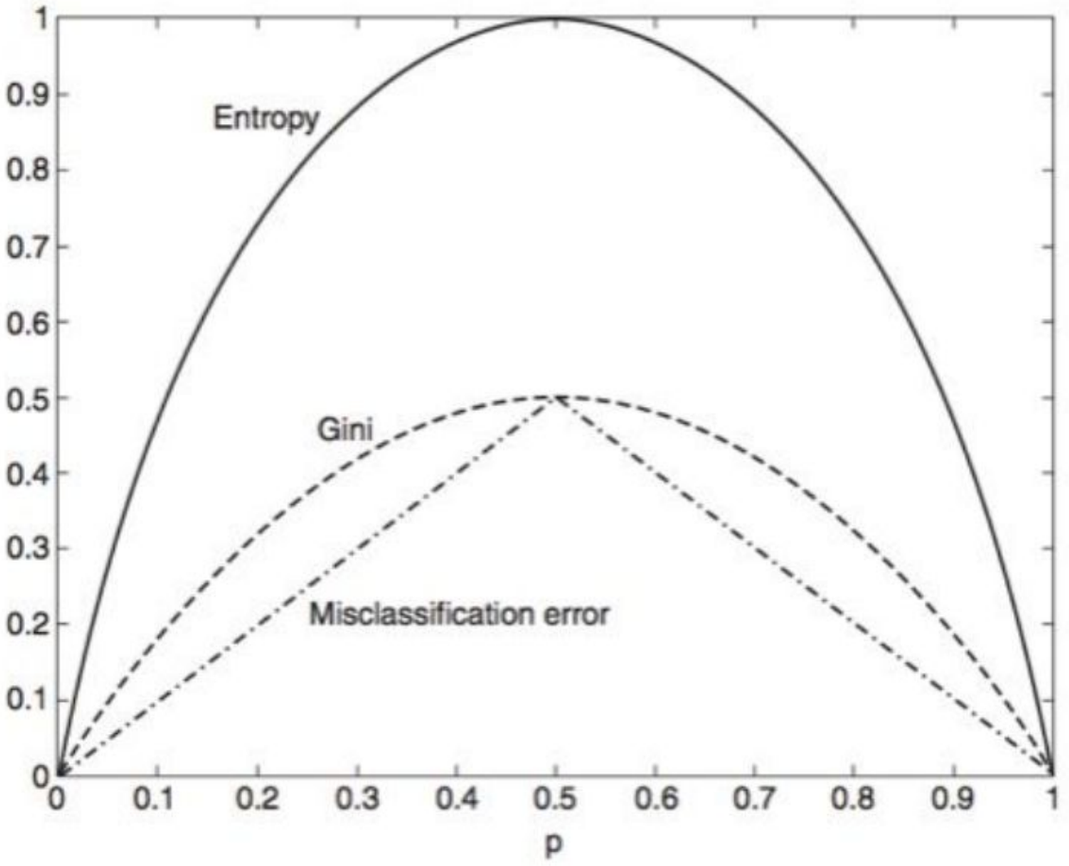
- We can quantify the *purity* of the separation of groups using Classification Error, Entropy, or Gini Coefficient

- 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

TRAINING A DECISION TREE MODEL

- Algorithm used to generate the tree:
  - Evaluate every threshold within each feature relative to a “purity” metric and find the feature / threshold combination that provides the greatest increase in “purity”
  - Do this until an exit criteria such as depth of tree or purity of leaves is met



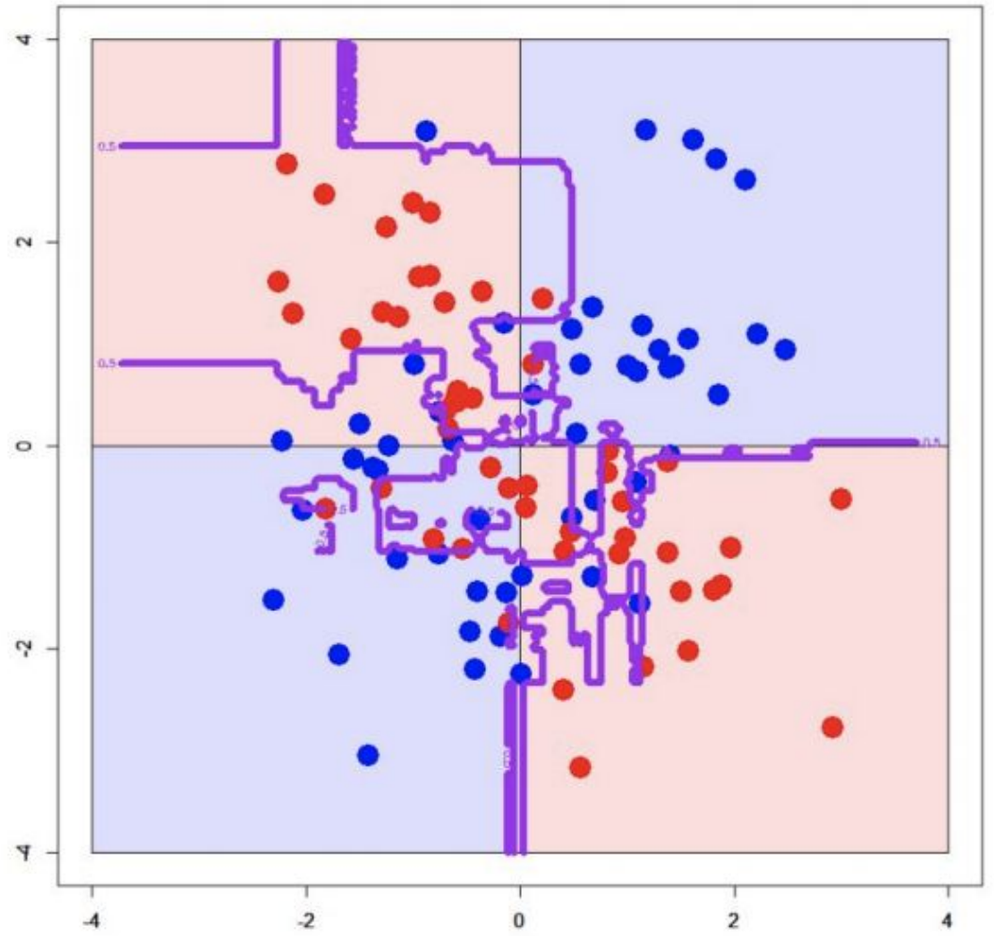
MAKING PREDICTIONS FROM A DECISION TREE

- 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.

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.

GUIDED PRACTICE

DECISION TREES IN SCIKIT LEARN

ACTIVITY: DECISION TREES IN SCIKIT LEARN

DIRECTIONS (15 minutes)

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 would work best? Why?

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

ACTIVITY: KNOWLEDGE CHECK

ANSWER THE FOLLOWING QUESTIONS

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

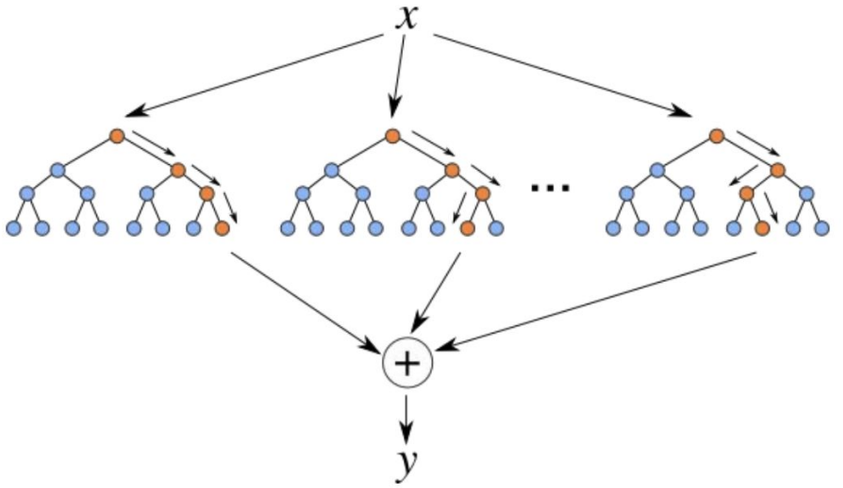
1. Using our StumbleUpon dataset, try to predict whether a given article is evergreen.
2. Build a decision tree to determine the above.
3. Explore different hyperparameters in the decision tree model

INTRODUCTION

RUNNING THROUGH RANDOM FORESTS

RUNNING THROUGH 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[er]
  - Black-box
  - No “coefficients”
  - Harder to explain

TRAINING A RANDOM FOREST

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

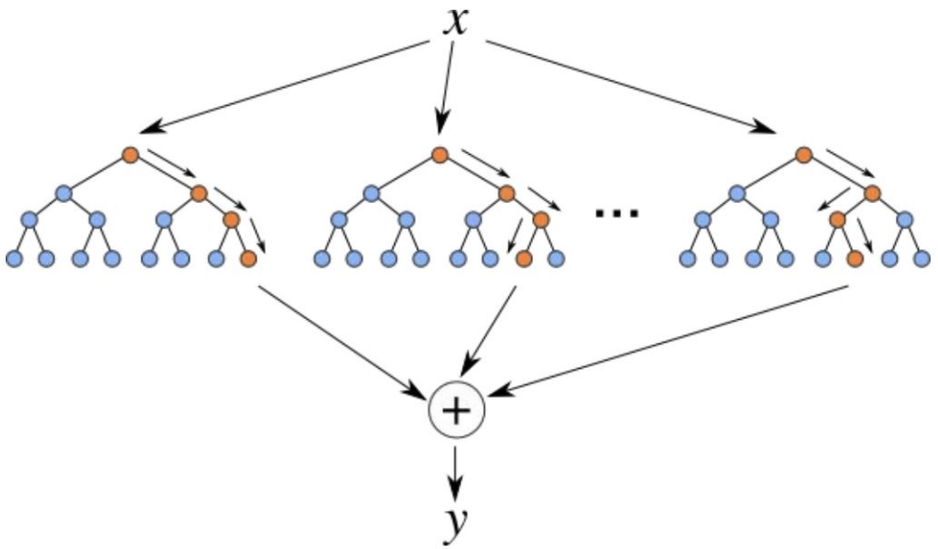
- Since decision trees overfit easily, we use many decision trees together and randomize the way they are created.

TRAINING A RANDOM FOREST

- Random Forest Algorithm
  - Take a bootstrap 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 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.



INDEPENDENT PRACTICE

EVALUATE RANDOM FORESTS USING CROSS-VALIDATION

ACTIVITY: RANDOM FORESTS

EXERCISE

DIRECTIONS (20 minutes)

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 most important features.

ACTIVITY: EVALUATE RANDOM FORESTS 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.

3. **Bonus:** Just like the 'recipe' feature, add in similar text features and evaluate their performance. Try to find the best possible model.

CONCLUSION

TOPIC REVIEW

TOPIC 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?