Decision Trees and Random Forests

1 Decision Trees

There might be multiple decision trees for deciding the same thing from different conditions. To decide which is best, we use Gini Impurity

$$\text{Gini Impurity} = 1 - (\text{the probability of Yes})^2 - (\text{the Probability of No})^2$$

A weighted average should be used if the sample size is different

The lower the value the better

From a raw table of data to a decision tree:

1. Calculate all of the Gini Impurity values
2. If a node itself has the lowest value, leave it as a Leaf node, don't further separate it
3. If separating the data results in an improvement, then pick the separation with the lowest Gini impurity value

1.1 Numeric Data

To get impurities

1. Sort the values lowest to highest
2. Calculate the average for adjacent values
3. Calculate the impurity values for each average weight
   - For each average, look at the yes and no instances on the greater than and less than sections, use these for the probabilities

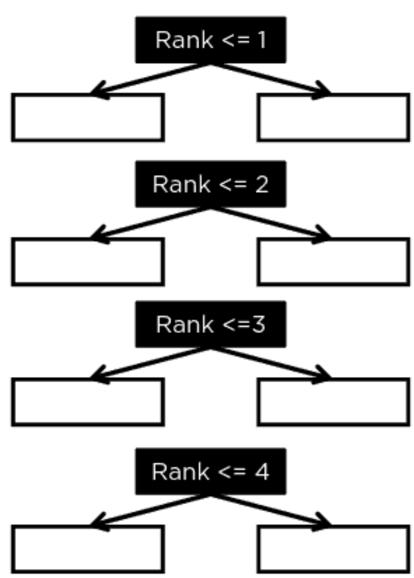
To Build a tree:

1. Yes/no questions at each step
2. Numeric data, like patient weight

1.2 Ranked Data and Multiple Choice Data

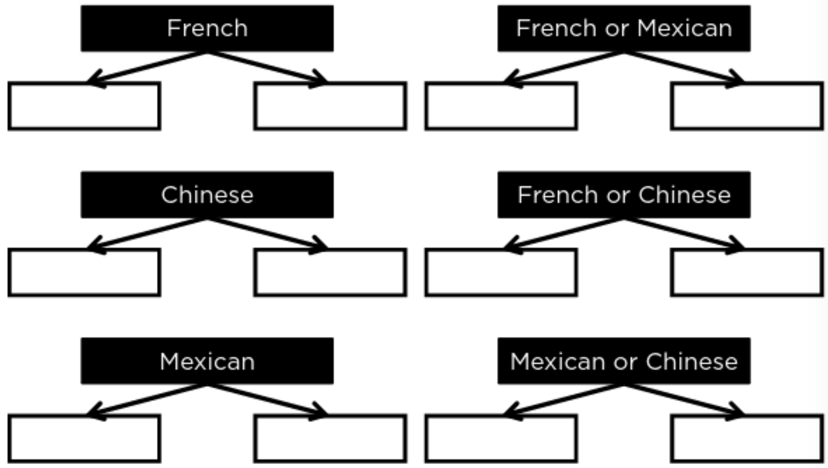
Ranked Data

Rank my ML lectures	Pass the exam
2	No
5	Yes
4	Yes
3	No
...	...



Multiple Choices Data

Cuisine Choice	Pass the exam
Mexican	Yes
Chinese	Yes
French	No
Chinese	Yes
...	...



1.3 Missing data

Options for boolean:

- Choose the most common value in the column
- Find another column that has the highest correlation with the feature and use that as a guide

Options for numbers:

- Use mean
- Use linear regression with another column with a good correlation

2 Random Forests

Why Random Forests:

- Decision Trees are easy to build, use and interpret, but not flexible when classifying new samples
- Random forests combine the simplicity of decision trees with flexibility for better accuracy

2.1 How to build a random forest

Step 1 - Create a ”bootstrapped” dataset:

- Same size as the original dataset
- Randomly selected samples from the original dataset
- Samples can be selected more than once

Step 2 - Build a decision tree using ”bootstrapped” dataset, but only use a random subset of variables, e.g. 2

Step 3 - Go back to step 1 and repeat: make a new bootstrap dataset and build a tree considering a subset of variables at each step (ideally 100’s of times)

- Using a bootstrapped sample and considering only a subset of the variables at each steps results in a wide variety of trees
- The variety makes random forests more effective than individual Decision Trees

2.2 How to use a random forest

- Take the data and run it down the first tree we built
- Keep track of the result
- Then run the next data down the second tree
- Then run the next data down all the trees and what the majority of the trees choose is the outcome

Definition: Bagging

Bootstrapping the data plus using the aggregate to make a decision

2.3 Performance

Definition: Out of bag dataset

Data that was not used in the bootstrapped dataset

- Use the data that doesn't end up in the bootstrapped dataset for testing
- Run the data on the trees and see if the outcome is correctly predicted
- Use the number that correctly predict vs incorrectly predict as the measure
- Repeat for all samples and trees

Definition: Out of bag error

The proportion of out of bag samples that were incorrectly classified