Carnegie Mellon University
Heinz College

Unstructured Data Analysis

Lecture 12: Wrap up classical classification methods, model evaluation

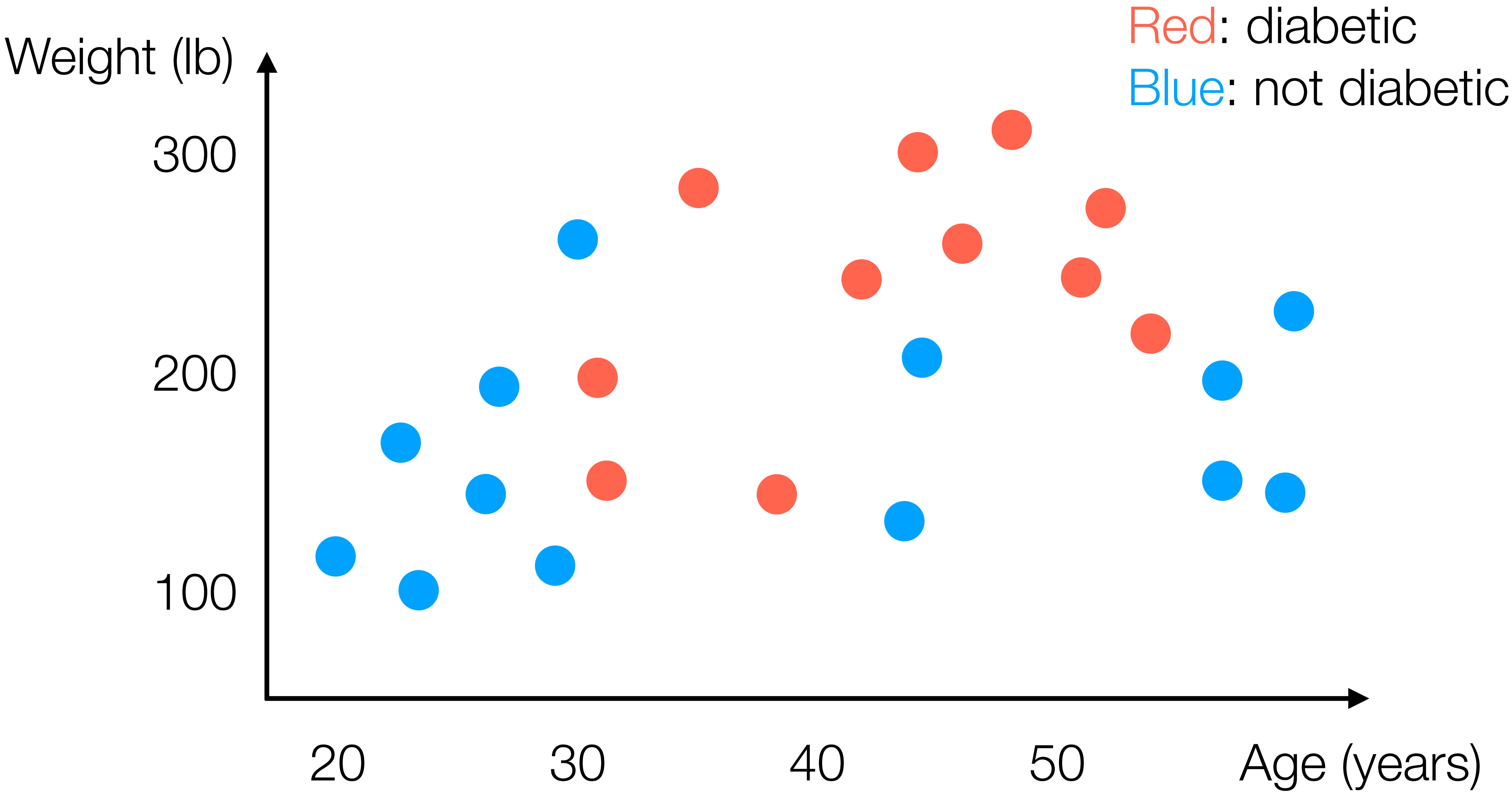
George Chen

Schedule

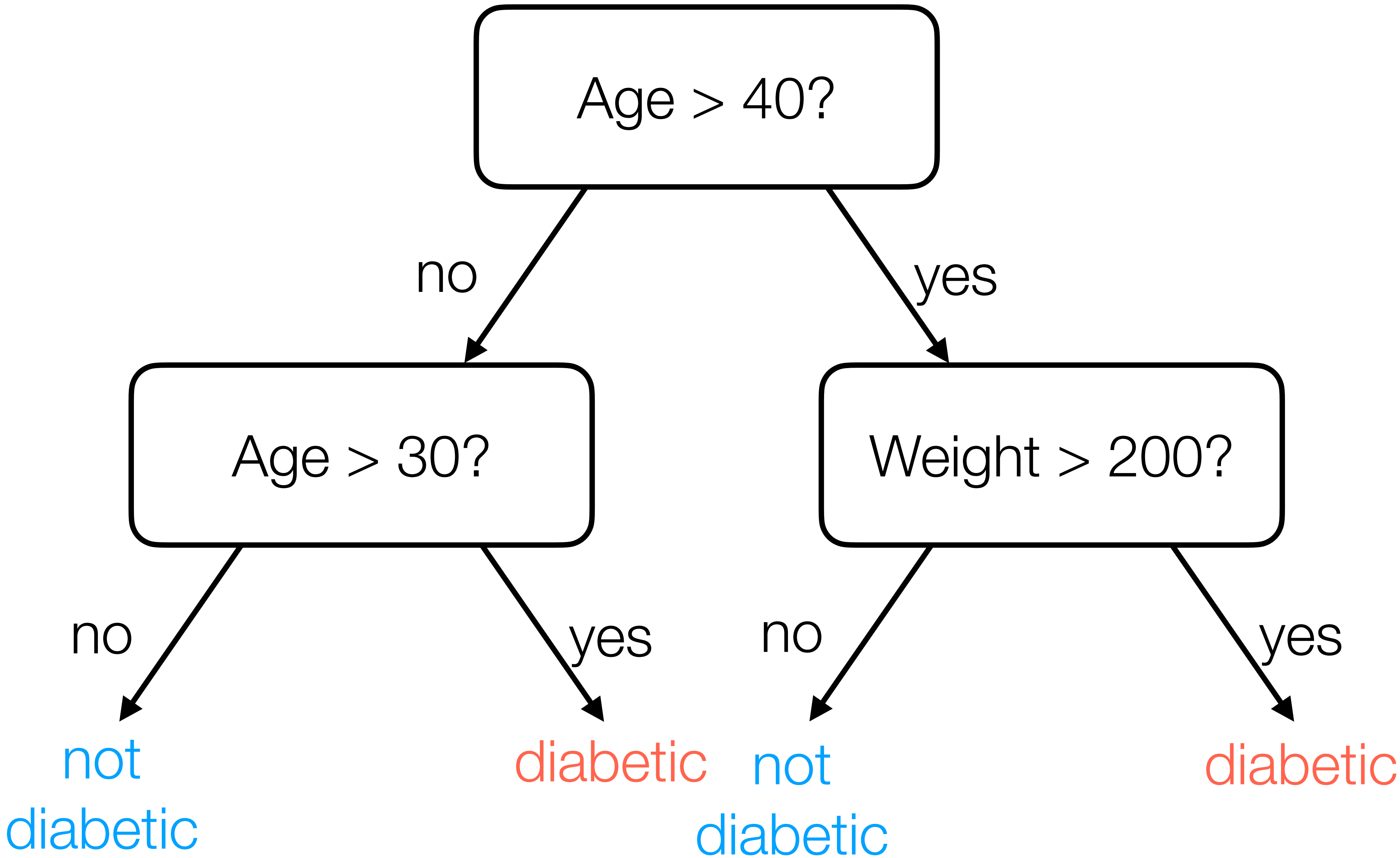
- Today: Wrap up *classical* classifiers and model evaluation
  - We'll discuss a family of classical classifiers that can at times work well also for unstructured data (decision-tree-based methods)
  - We'll talk more in depth about model evaluation for *binary* classification
    - One of your TA's will give more details in this Friday's recitation
- I mentioned that I slowed down lecture pacing
  - Unfortunately we are at risk of not covering everything needed for HW3, so I will teach a lecture during next Friday's recitation slot
(the Quiz 2 review will be rescheduled)

Decision Trees & Forests

Example Made-Up Data



Example Decision Tree



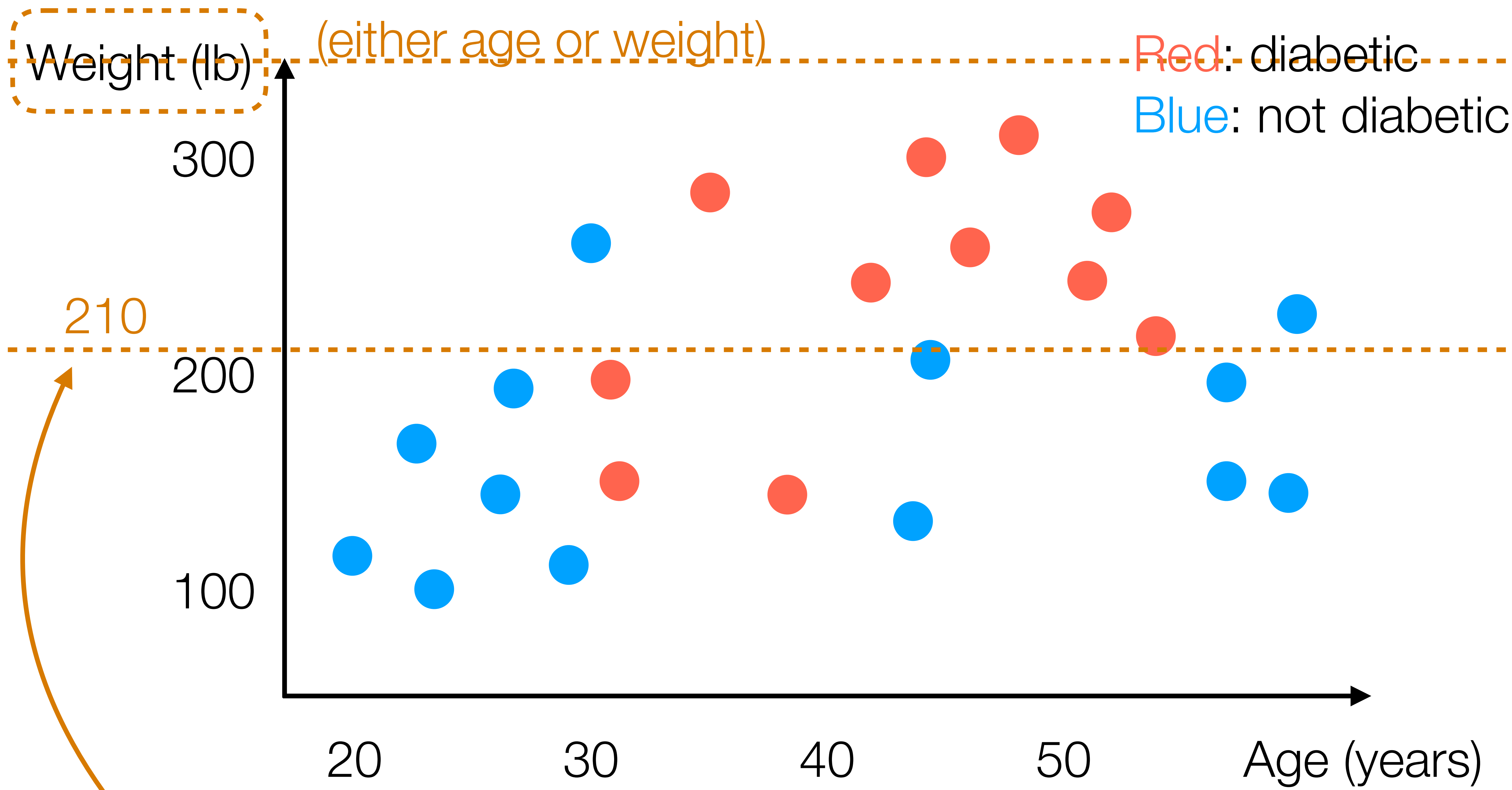
Learning a Decision Tree

- Many ways: general approach actually looks a lot like top-down hierarchical clustering *but accounts for label information* (if you haven't seen hierarchical clustering don't worry about this)

- I'll show one way (that nobody actually uses in practice) but it's easy to explain

Learning a Decision Tree

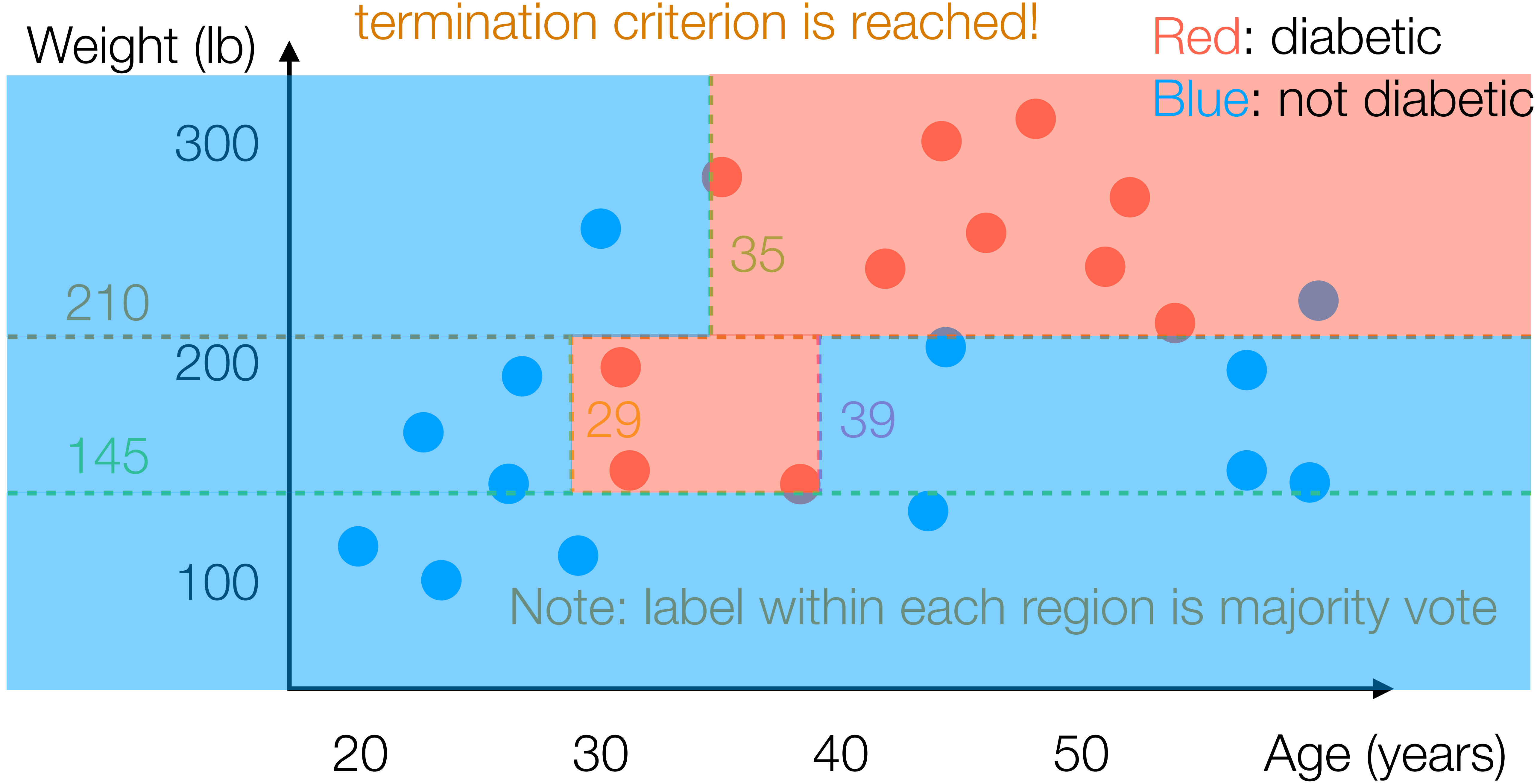
1. Pick a random feature
(either age or weight)



2. Find threshold for which red and blue are as “separate as possible” (on one side, mostly red; on other side, mostly blue)

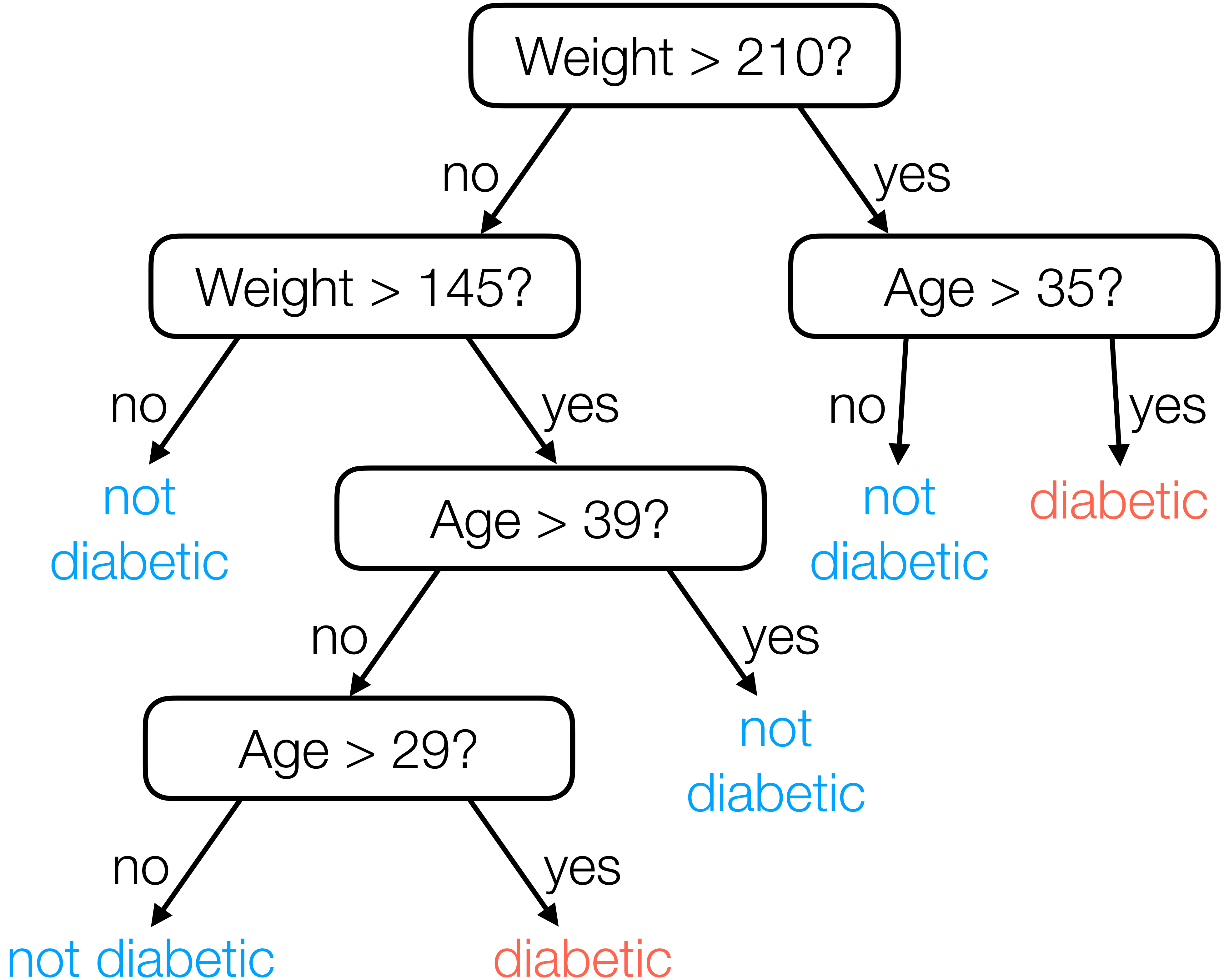
Learning a Decision Tree

Within each side, recurse until a termination criterion is reached!



Example termination criteria: ≥90% points within region has same label, number of points within region is <5

Decision Tree Learned



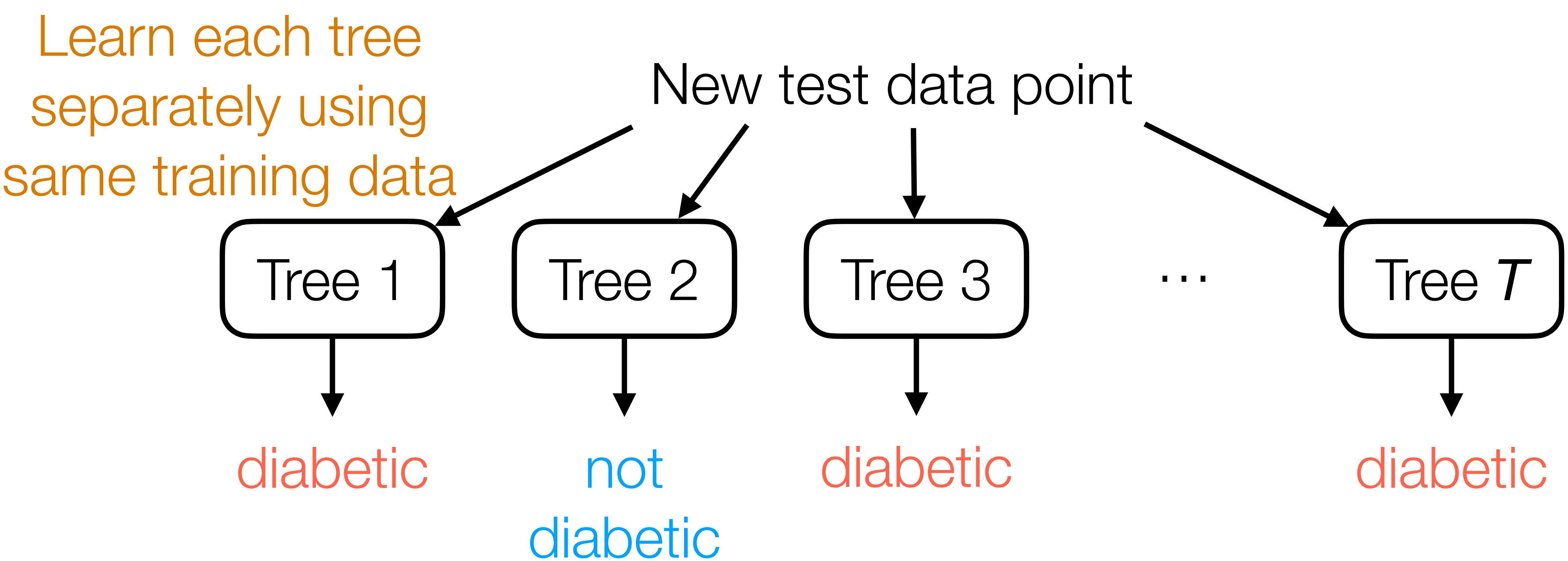
For a new person with feature vector (age, weight), easy to predict!

Decision Forest for Classification

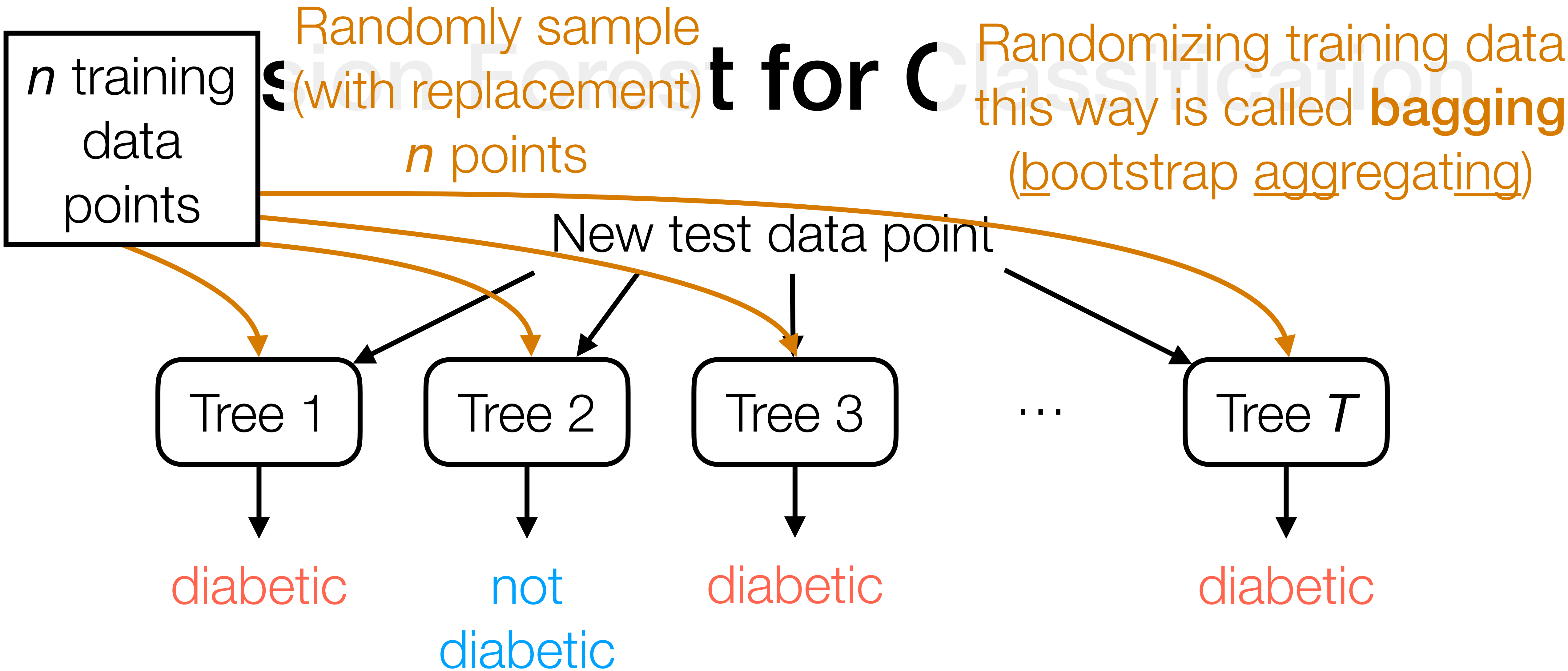
- Typically, a decision tree is learned with randomness (e.g., we randomly chose which feature to threshold)

  ➔ by re-running the same learning procedure, we can get different decision trees that make different predictions!

- For a more stable prediction, use many decision trees



Final prediction: majority vote of the different trees' predictions



Question: What happens if all the trees are the same?

Adding randomness can make trees more different!

- **Random Forest:** randomize training data used for each tree, randomly choose a few features to try to split on (and among these features, choose the best one to split on)

Demo

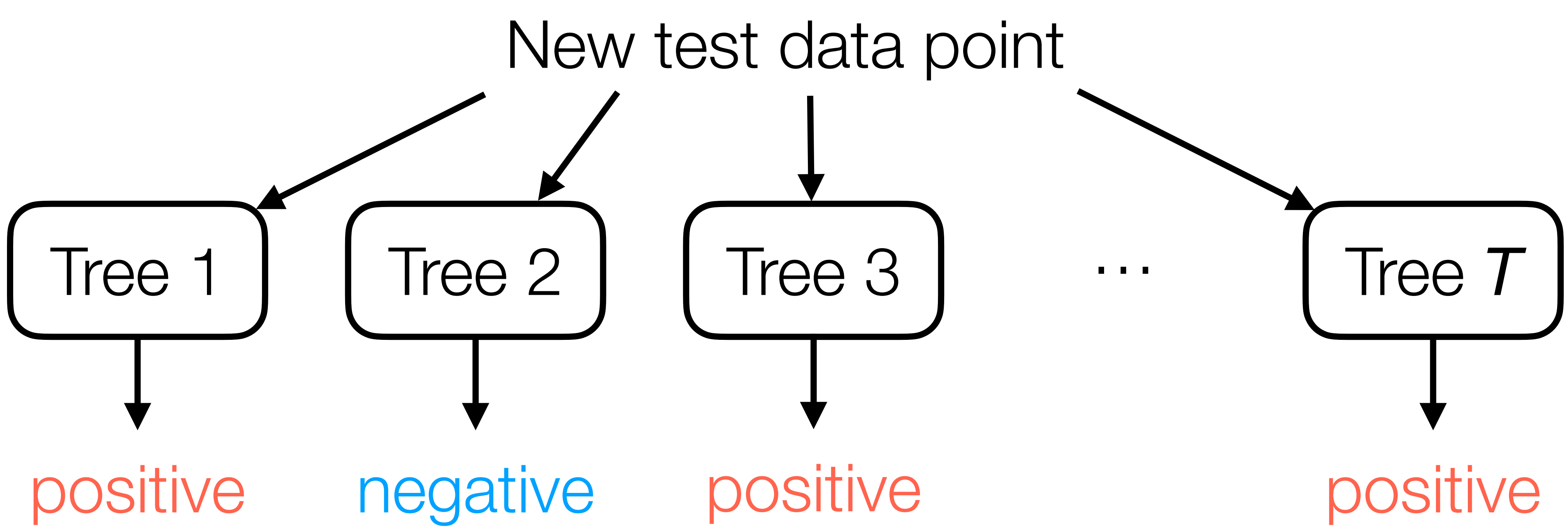
Another Way to Benchmark

- In the demo: we just saw that we can compare test set prediction accuracy across different algorithms and also look at confusion matrices

- For binary classification, we can do a more detailed analysis

Binary Classification: ROC Curves

For simplicity, think of the random forest for now

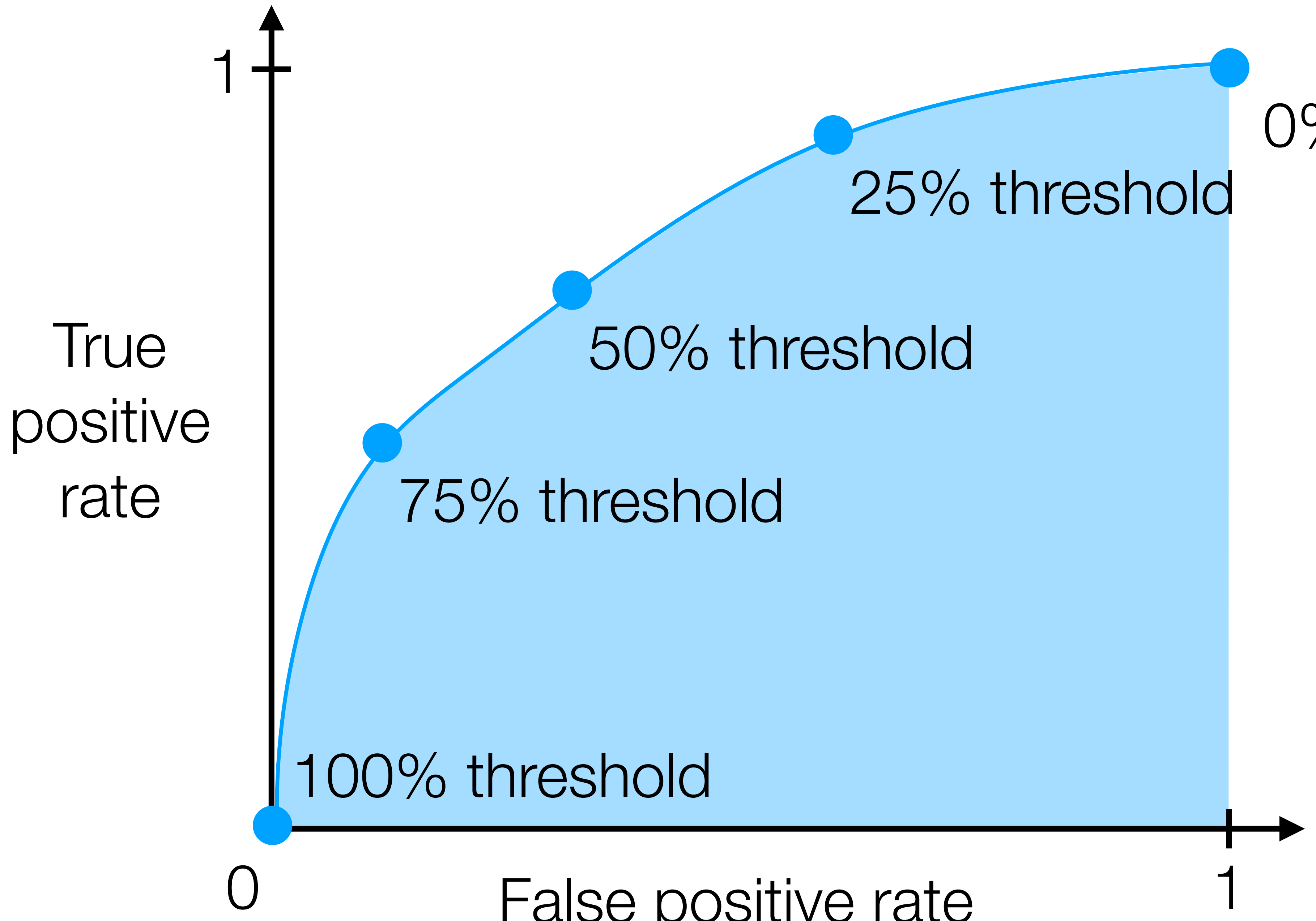


Final prediction: majority vote of the different trees' predictions

$\geq$ 50% of trees need to say **positive** for final prediction to be **positive**

We can vary this 50% threshold!

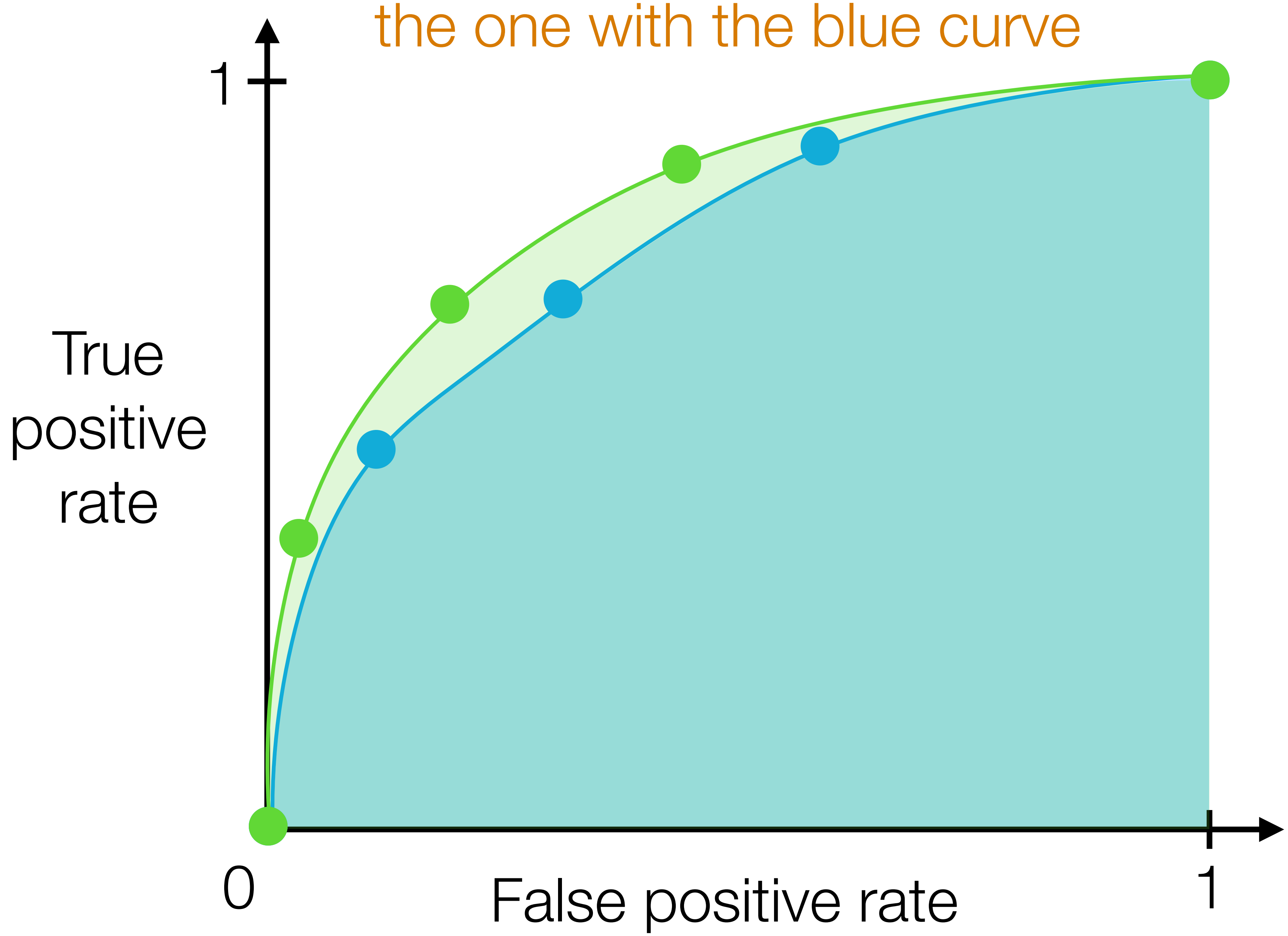
Binary Classification: ROC Curves



Error rates are computed on test data

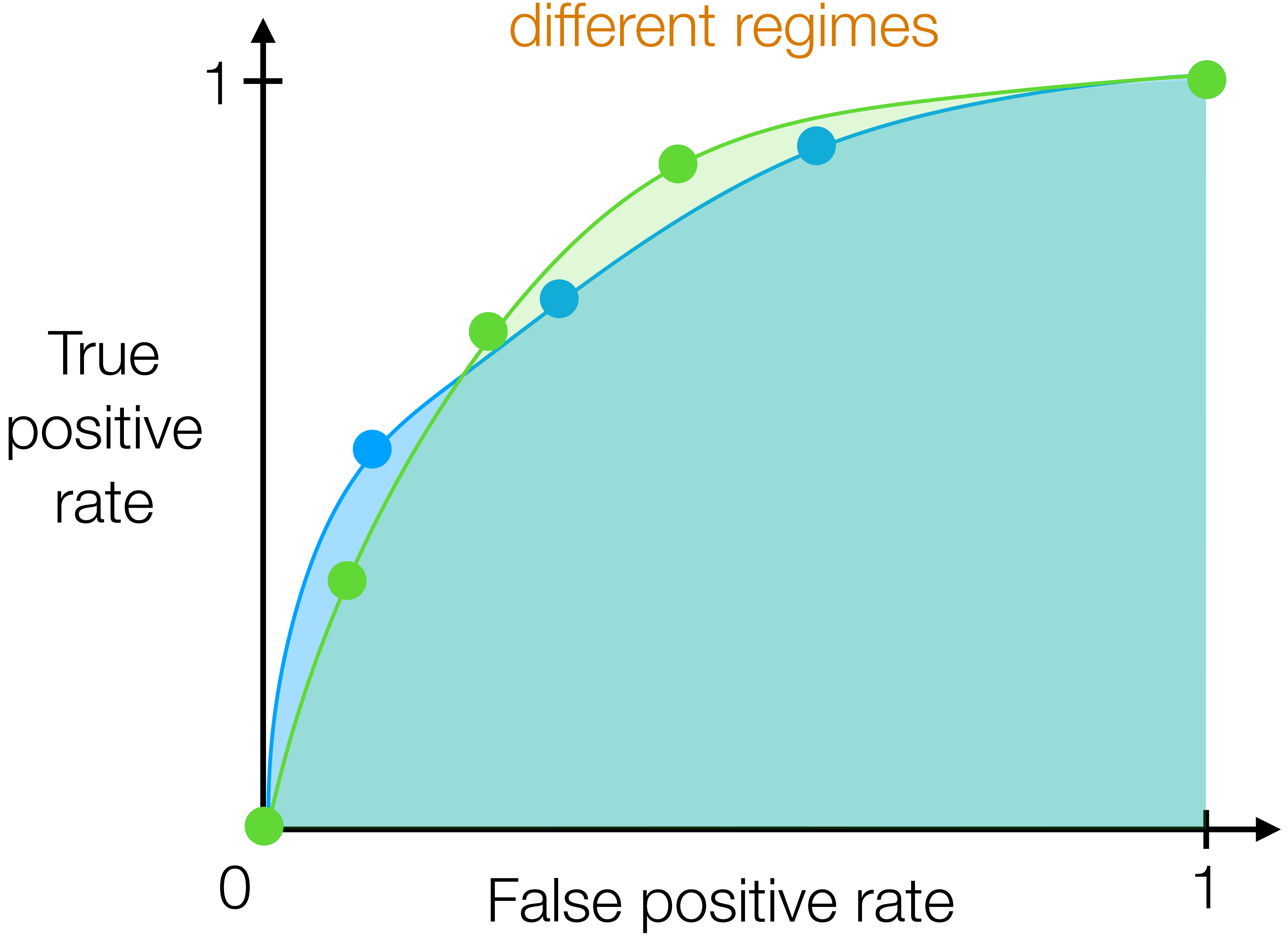
Binary Classification: ROC Curves

A classifier with the green curve is better than the one with the blue curve

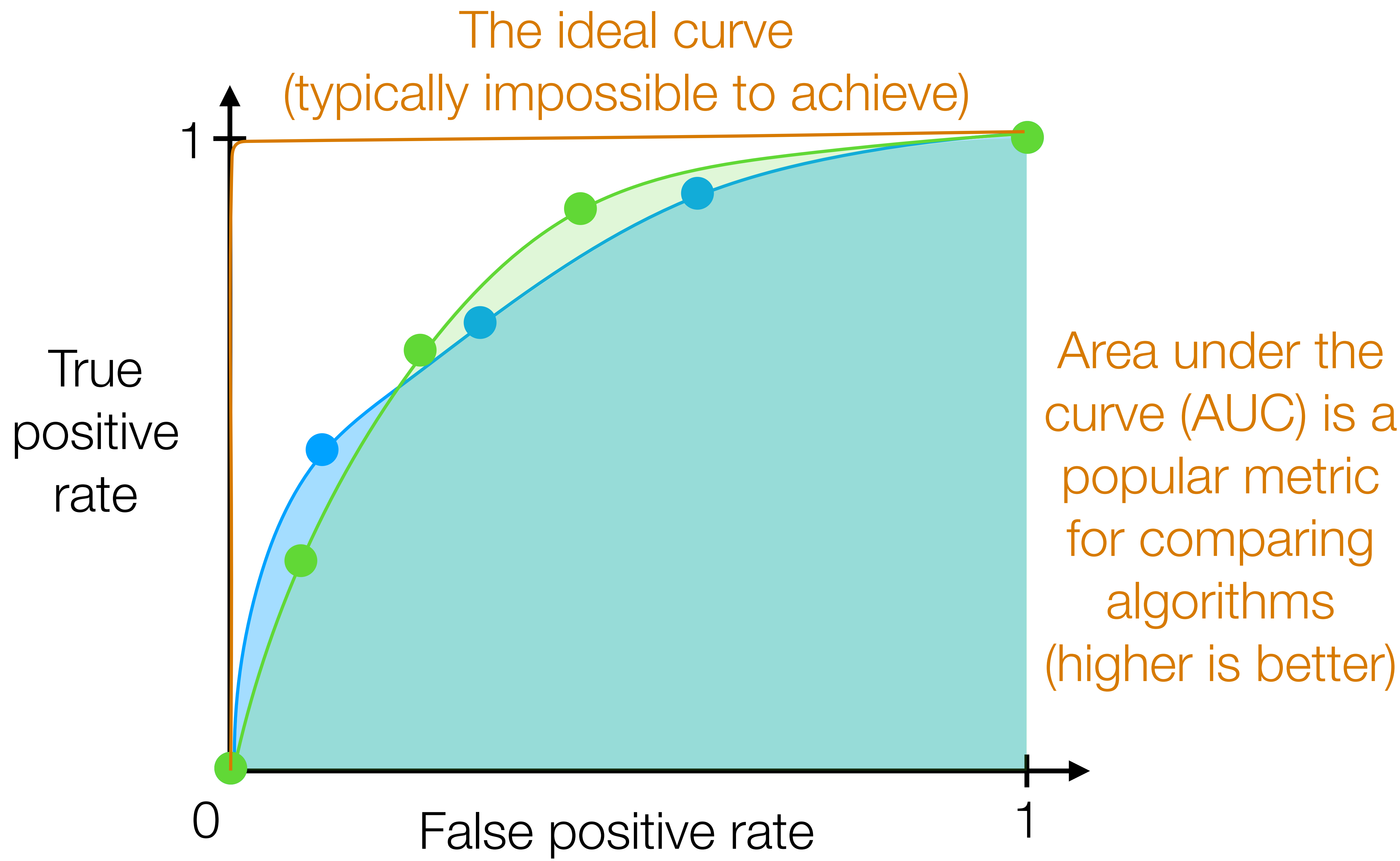


Binary Classification: ROC Curves

It's possible that algorithms are better in different regimes



Binary Classification: ROC Curves



Binary Classification: ROC Curves

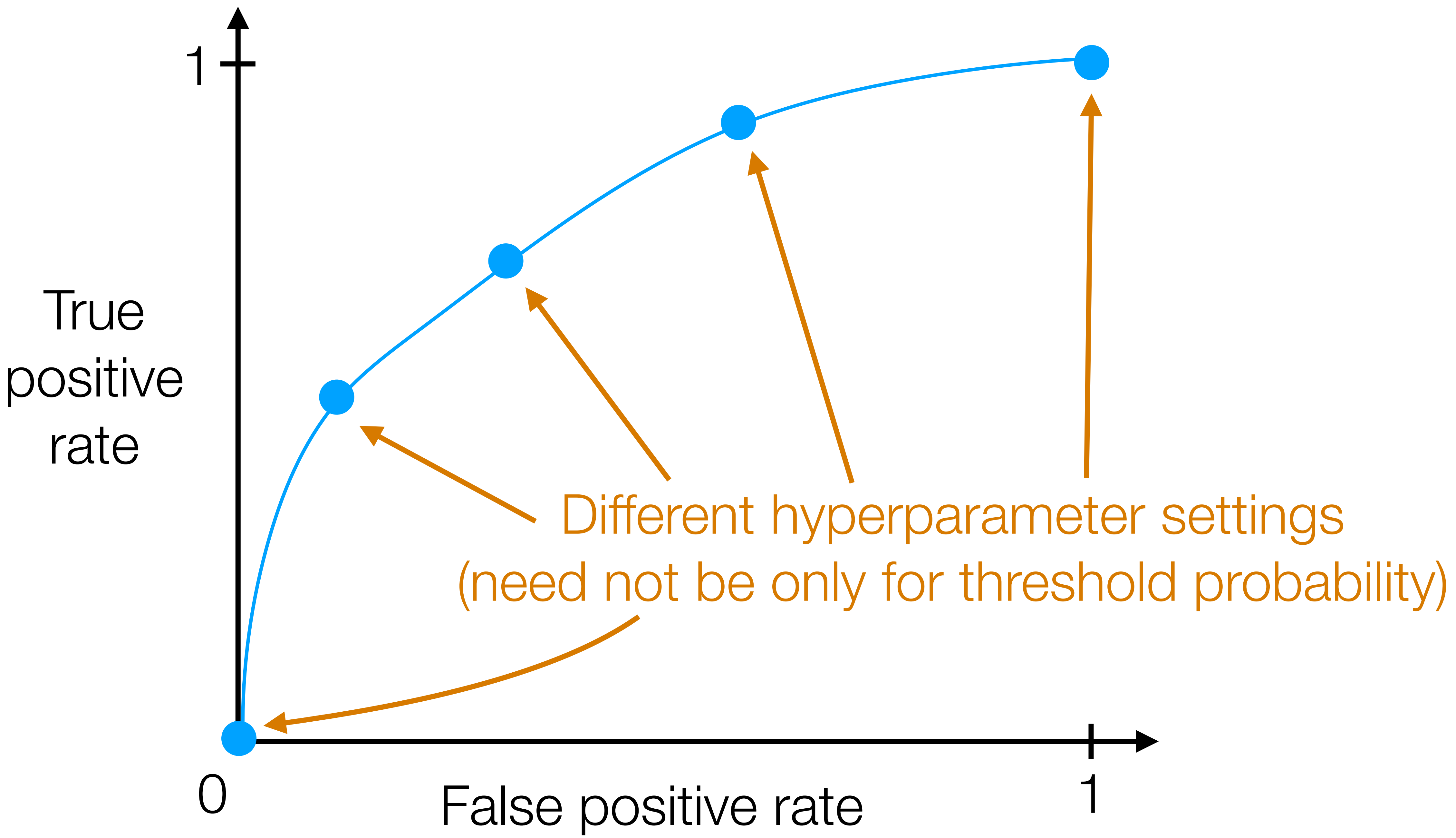
What we just saw:

- For a classifier that we can set the threshold probability to different values, we can plot an ROC curve
- True positive rate (TPR) and false positive rate (FPR) are evaluated on test data

Other variants are possible:

- Plot precision vs recall instead of TPR vs FPR
- Can actually plot ROC/precision-recall curves sweeping over hyperparameters aside from threshold probability!
- For ROC/precision-recall, rather than evaluating on test data, can evaluate on validation data during training *to help choose hyperparameters*

Binary Classification: ROC Curves



Can also be computed on validation data instead of test data!