Carnegie Mellon University
Heinz College

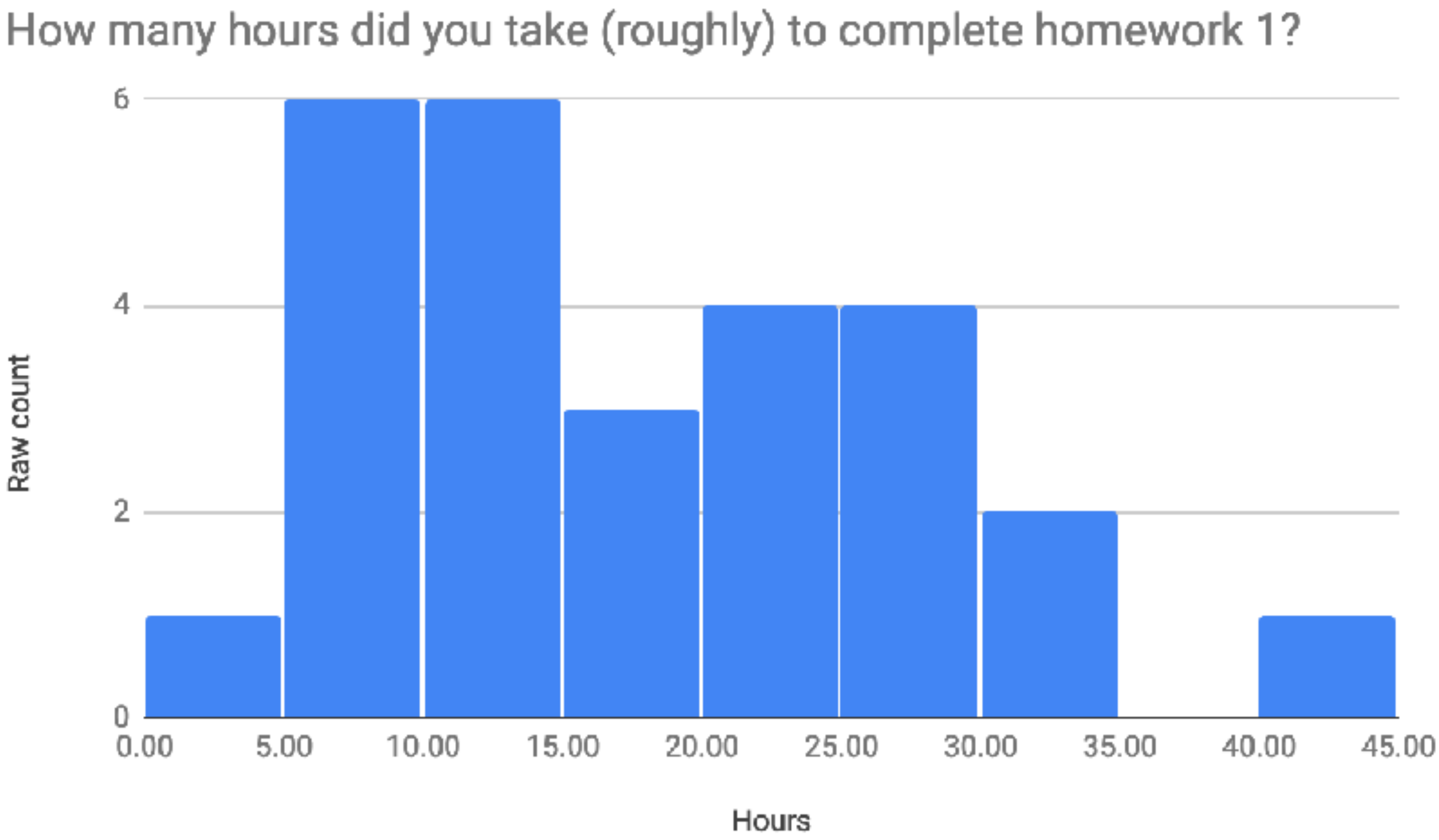
94-775/95-865 Lecture 9: Prediction and Model Validation, Decision Trees/Forests

George Chen

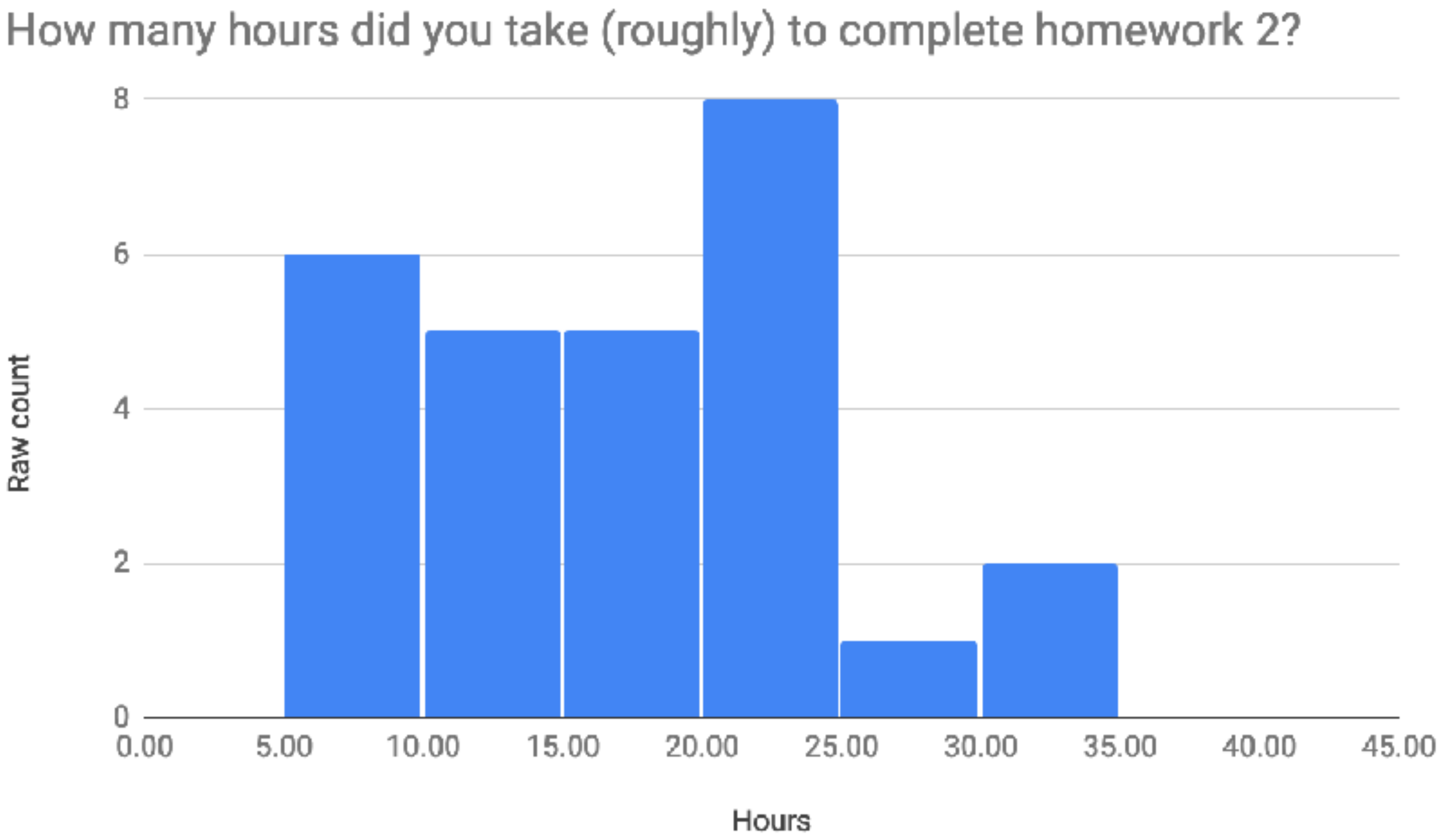
Announcements

- HW1 regrades are due by Friday 11:59pm:
  1. Carefully look over solutions
  2. If you think there’s a grading error, email me and say what you think the error is
  3. We will regrade your entire assignment and your score can go up, stay the same, or go down
  4. The regraded score is final
- All final project presentations will be on Tuesday 3/5:
  10 minutes per group
- No class on Thursday 3/7 (final project slide deck + code due Thursday 11:59pm)

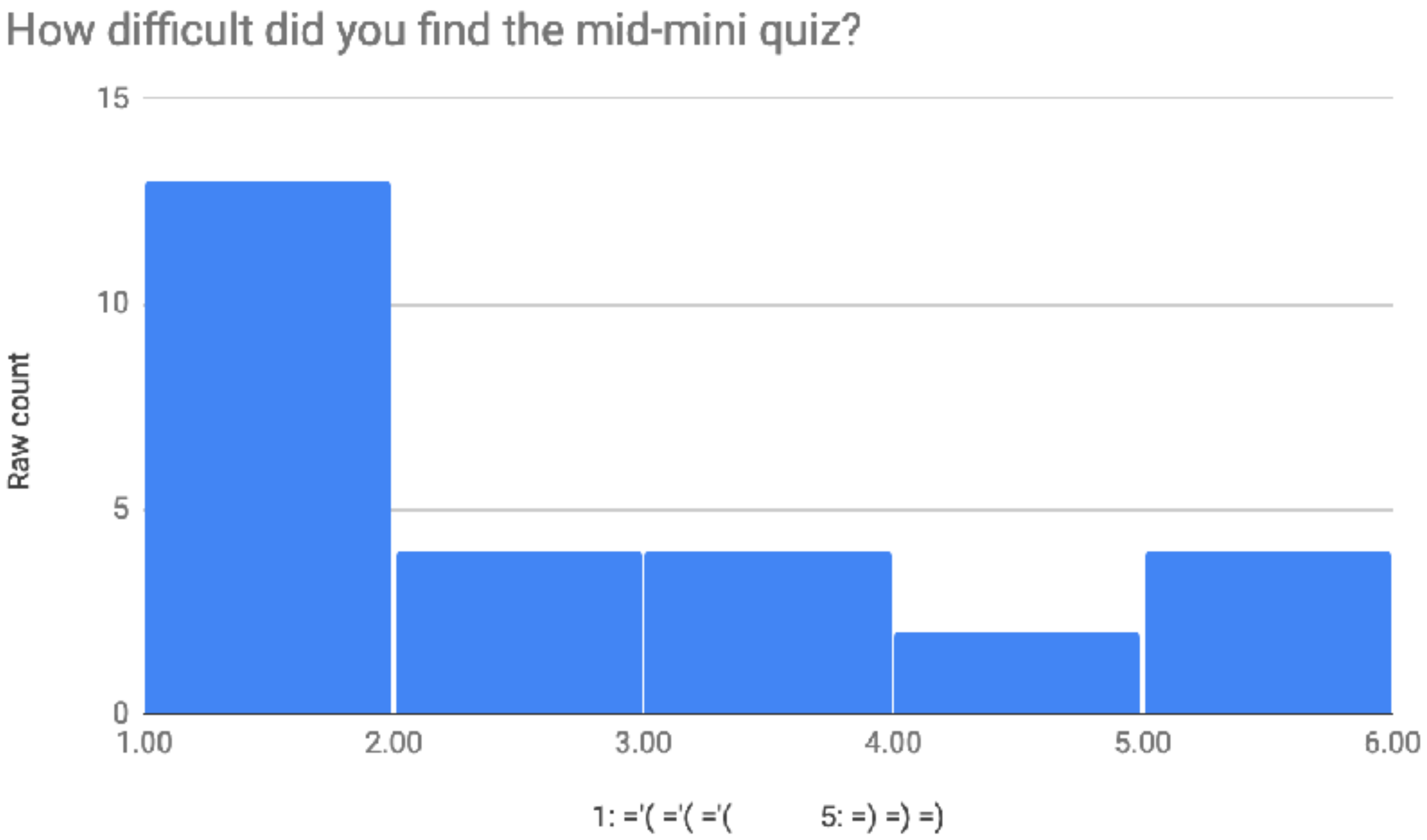
Questionnaire Results



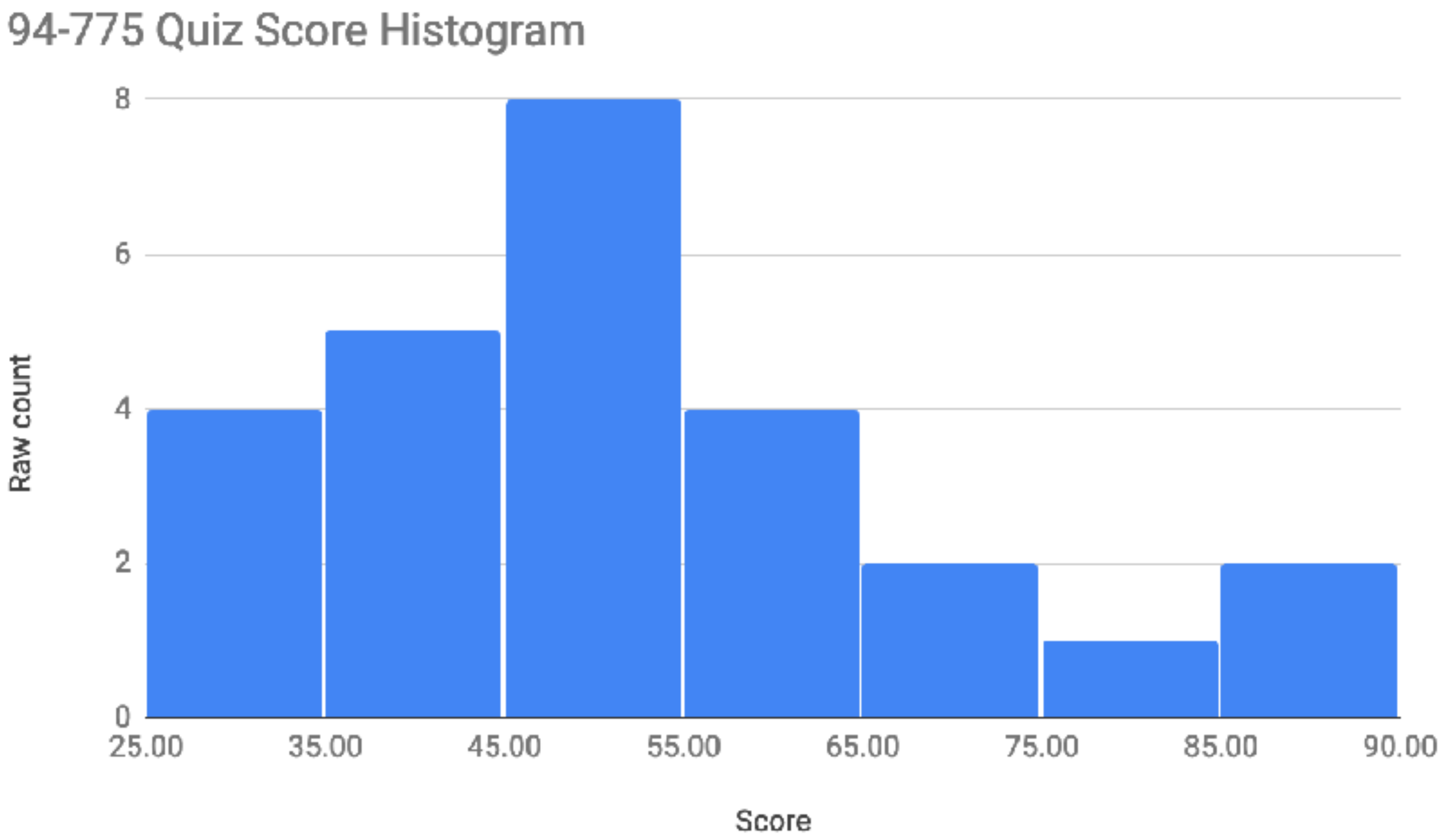
Questionnaire Results



Questionnaire Results



Quiz Results

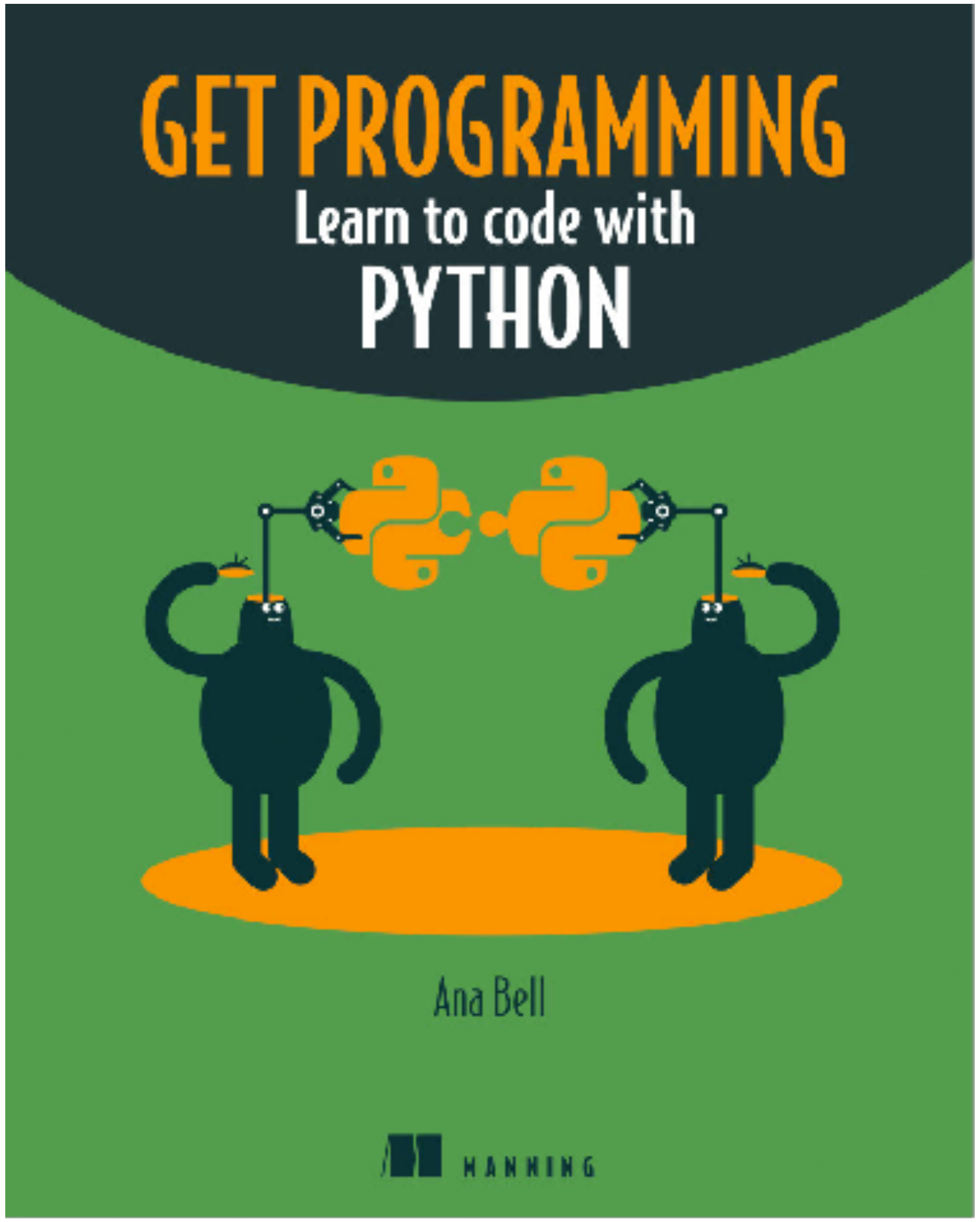


Mean: 51.7, standard deviation: 16.1, max: 87

Questionnaire Results

- Nearly all comments were on Python proficiency
- Some questions about how to learn Python faster
  - There isn’t some magic formula; need to practice!
  - It’s like learning to swim: you can’t just watch other people swim, you have to actually practice yourself
- If you want to get better at data analysis, improving your programming skills is helpful (you need not be a Jedi coder!)

Maybe This Book Can Help



Freely available online via CMU library

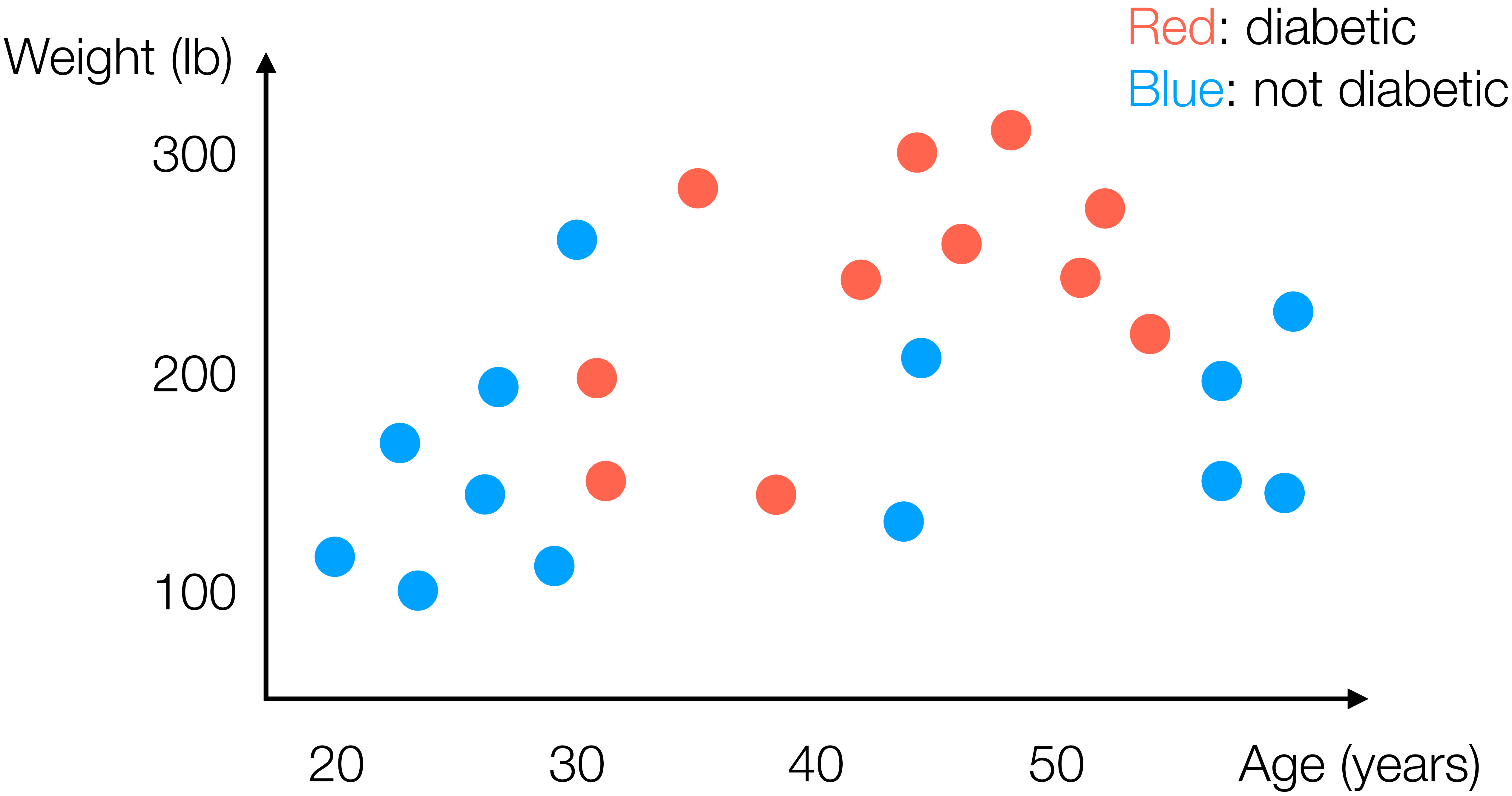
Back to Predictive Data Analysis

Prediction and Model Validation

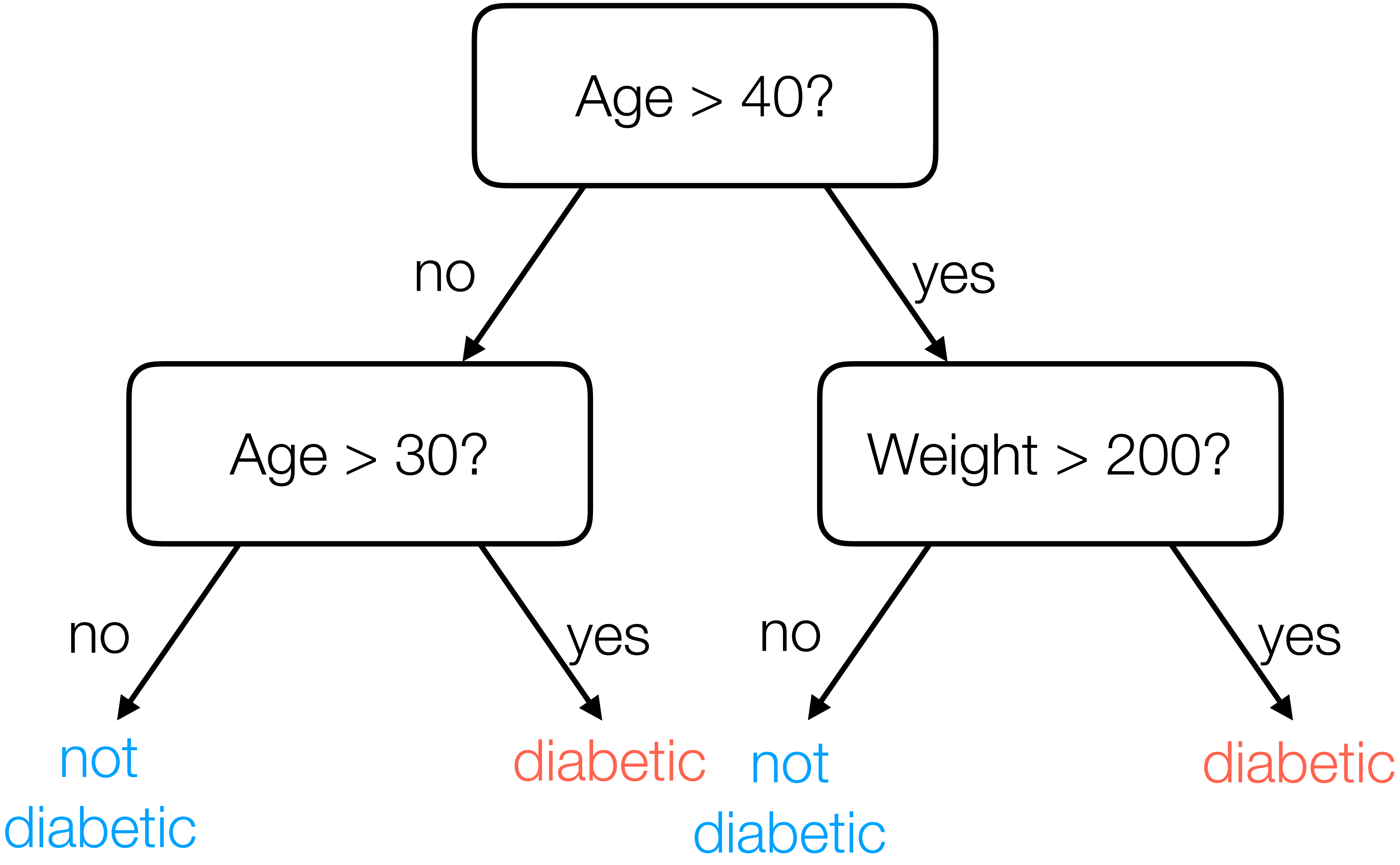
Demo

Decision Trees

Example Made-Up Data



Example Decision Tree



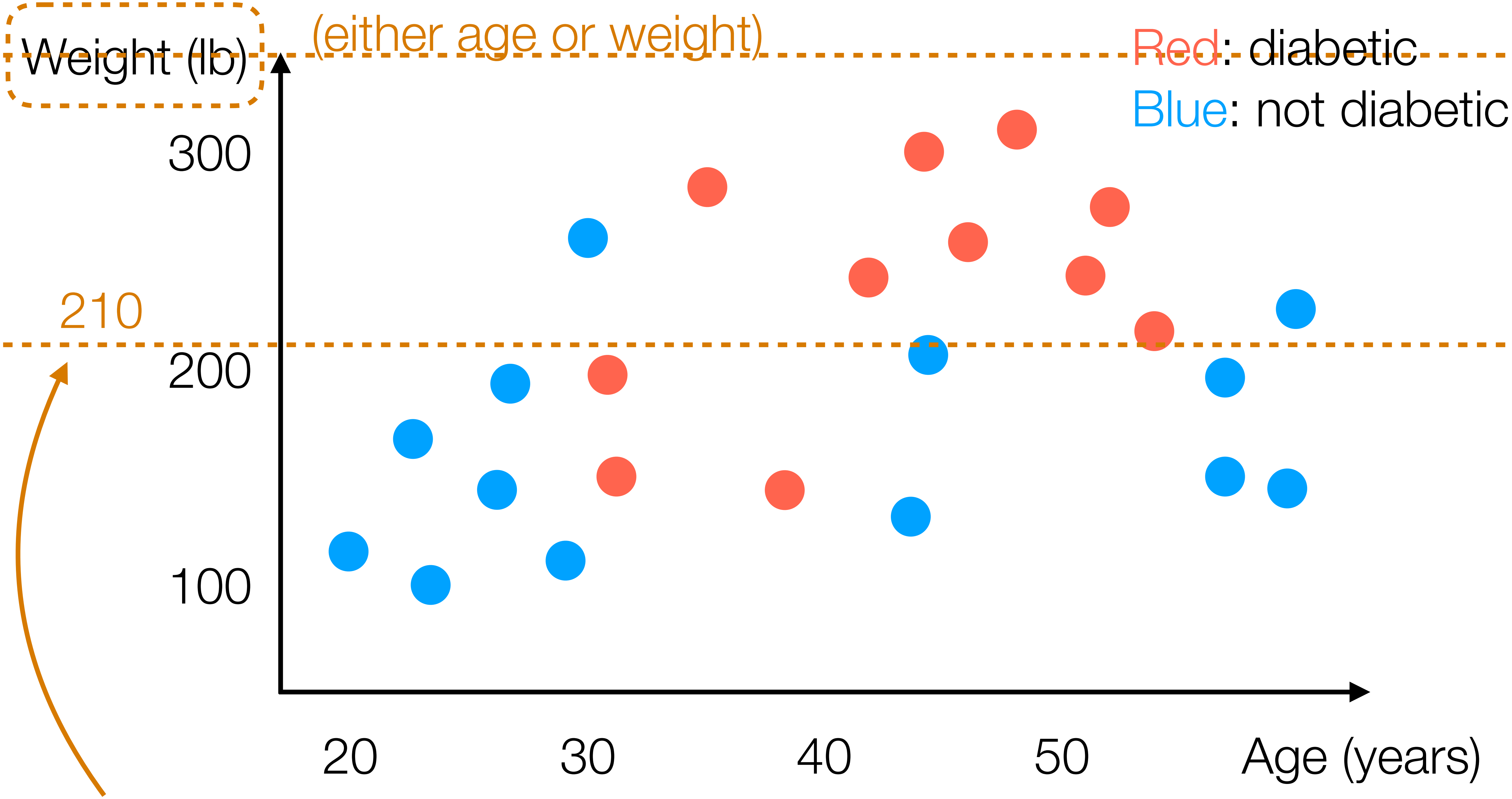
Learning a Decision Tree

- Many ways: general approach actually looks a lot like divisive clustering *but accounts for label information*

- 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)



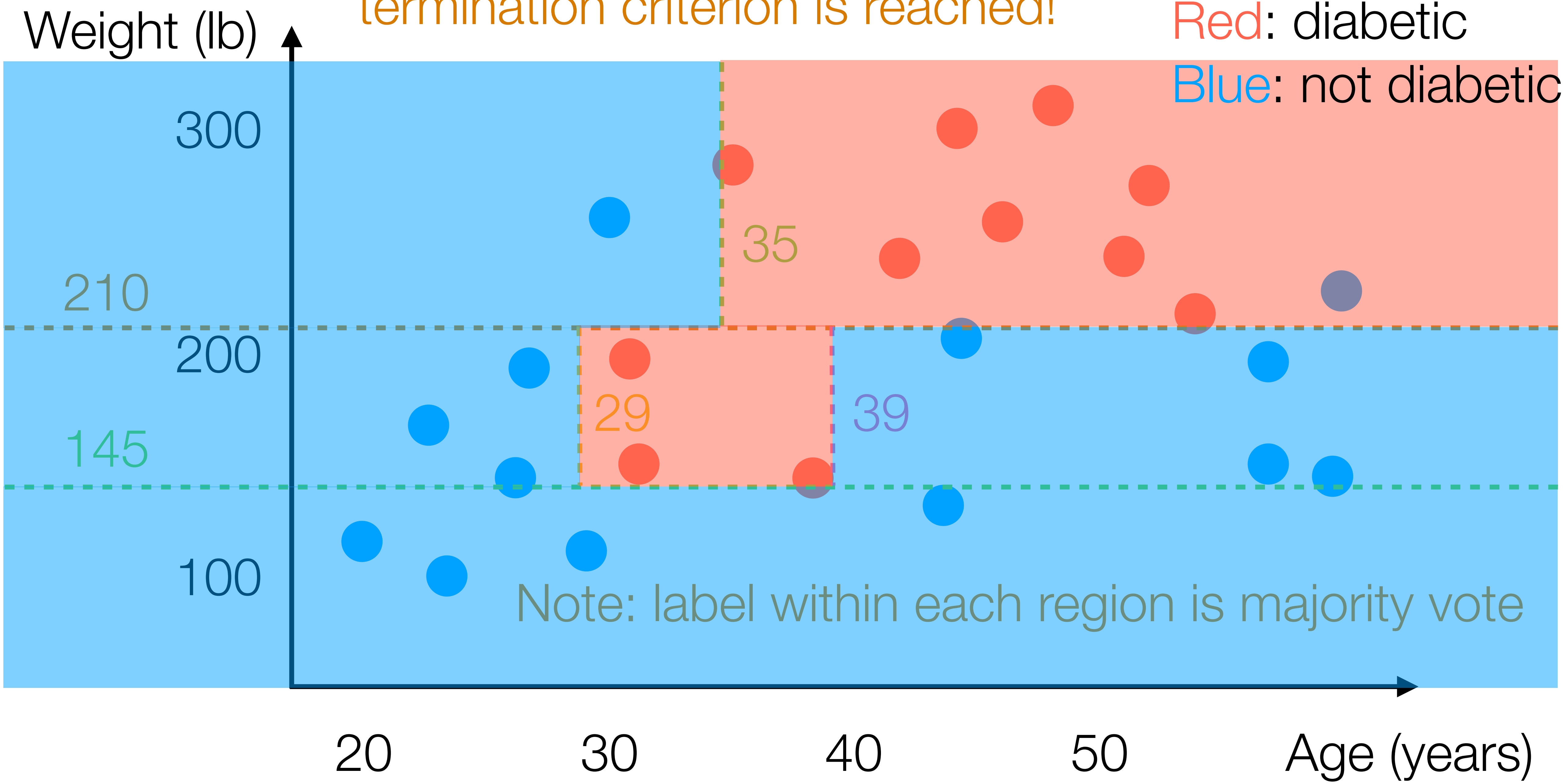
Red: diabetic

Blue: not diabetic

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

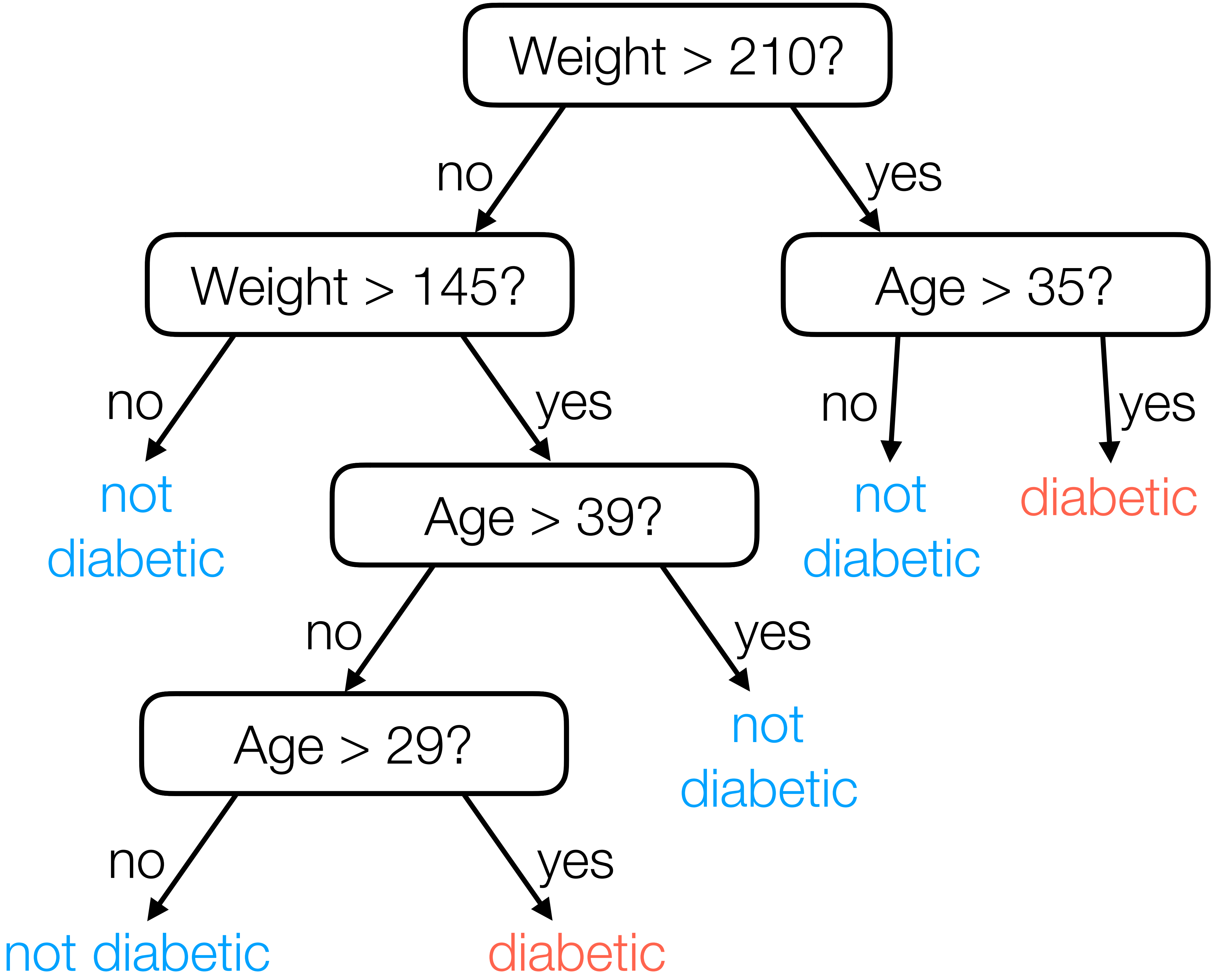
Learning a Decision Tree

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



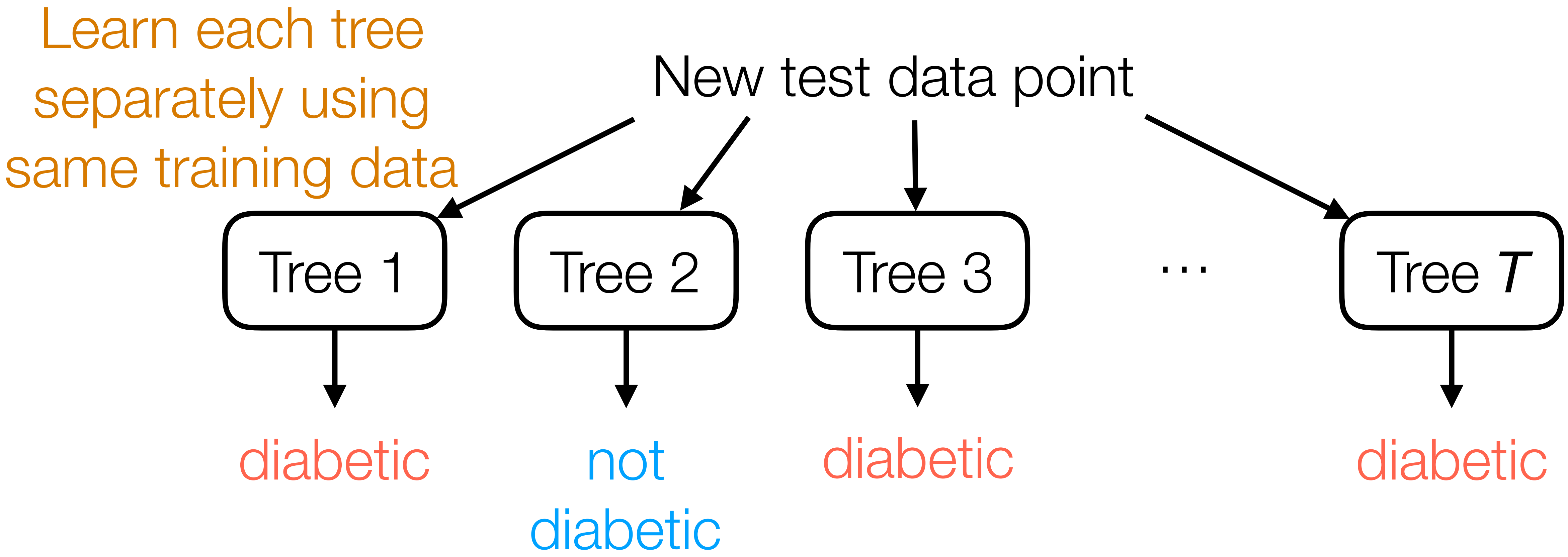
For a new person with feature vector (age, weight), easy to predict!

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

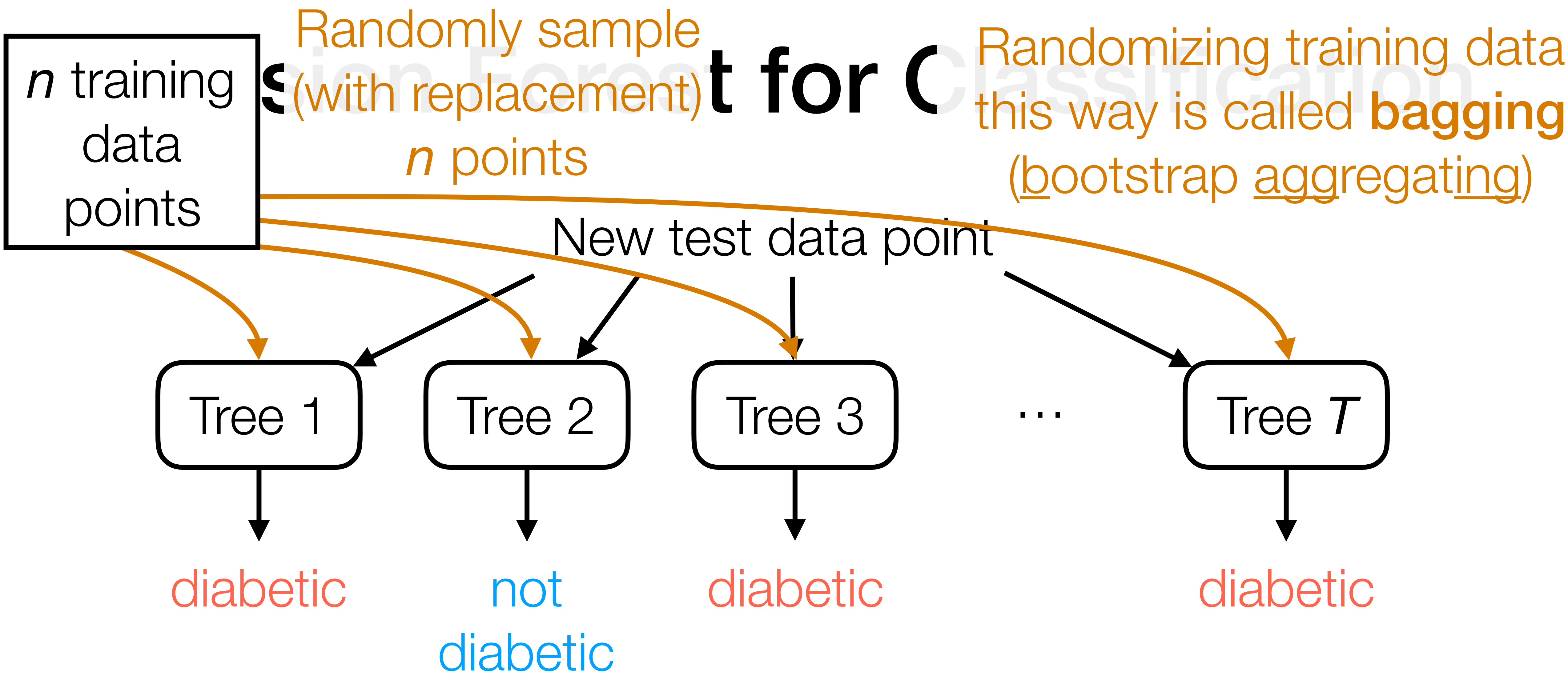


Learn each tree separately using same training data

New test data point

Final prediction: majority vote of the different trees’ predictions

sion Forest for Classification



n training data points
Randomly sample (with replacement) *n* points
Randomizing training data this way is called **bagging** (bootstrap aggregating)
New test data point
Tree 1
Tree 2
Tree 3
⋯
Tree *T*
diabetic
not diabetic
diabetic
diabetic

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)

Back to the demo