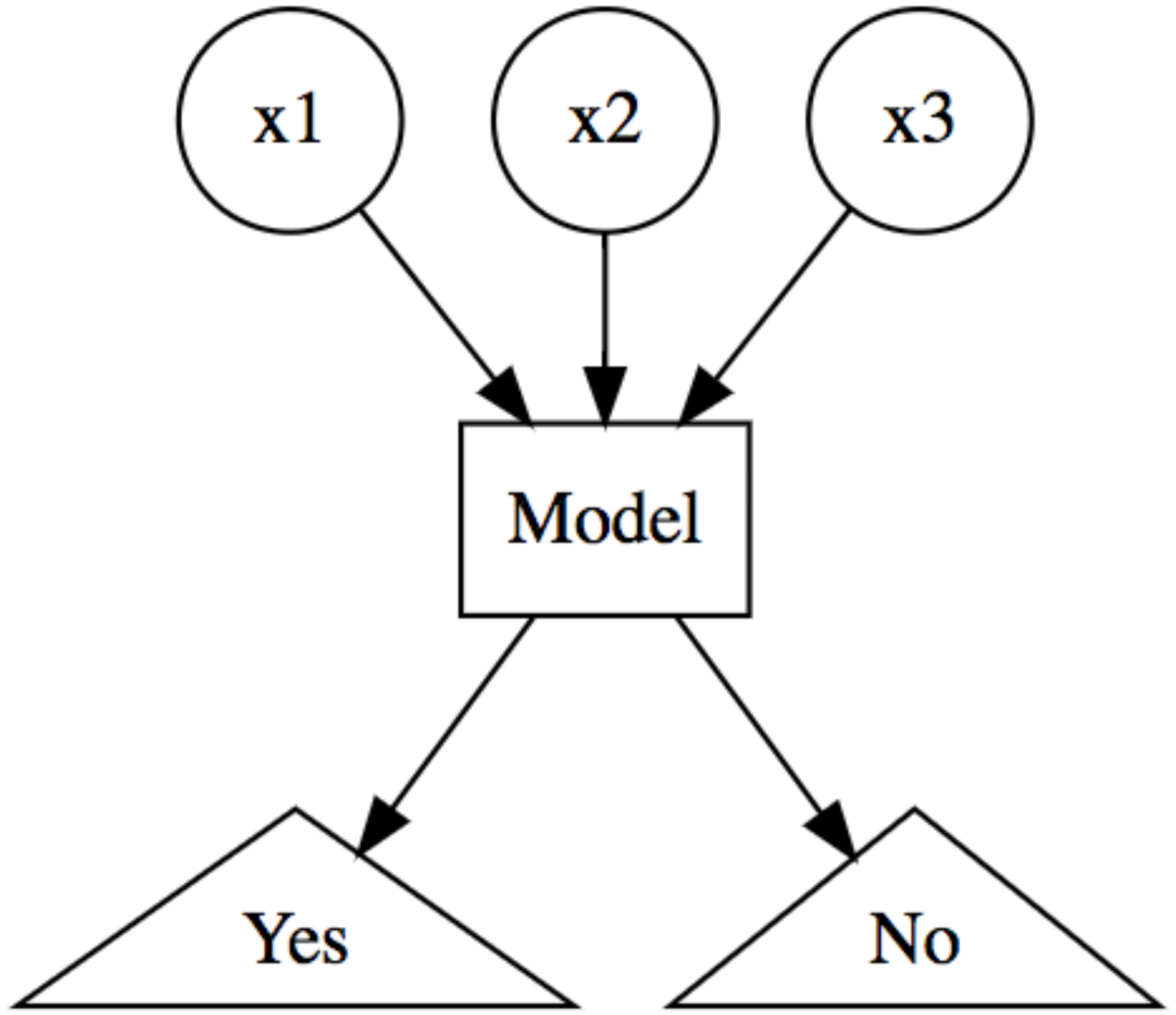
Classification

R workshops, 2021
Baruch College

Víctor Peña, January 2021

What is classification?

- Predicting a categorical outcome given a set of predictors
- Logistic and regularized logistic regression are examples of classification methods
- Today, we'll learn more terminology, how to evaluate model performance, and we'll learn some new methods



Titanic example

- **Goal:** Predicting whether a passenger survived (1 = survived, 0 = didn't) given their age, sex, passenger class (1st, 2nd, 3rd)
- Split the data into training and test sets.
- Train a model on the training set
- For the data in the test set, cross-tabulate the predictions against the actual outcomes

Age	Sex	Pclass	Survived
23	Female	1	1
32	Male	2	0
44	Female	1	1
...	...	...	...

Confusion matrices

Confusion matrix

- Simply tabulate predictions against actual values
- In green, correct classifications
- In red, incorrect classifications
- **Accuracy:**
    - proportion of observations that are classified correctly
    - (23+44)/(23+44+15+25) ~ 0.626

	Actually survived (1)	Actually died (0)
Predicted to survive (1)	23	15
Predicted to die (0)	25	44

Some terminology and metrics

- **Accuracy:** prop. obs. classified correctly
  - (TP+TN)/(TP+FP+FN+TN)
- **Sensitivity:** prop. of actual 1s that are classified correctly
  - TP/(TP+FN)
- **Specificity:** prop. of actual 0s that are classified correctly
  - TN/(TN+FP)

	Actually 1	Actually 0
Predicted 1	True positive (TP)	False positive (FP)
Predicted 0	False negative (FN)	True negative (TN)

Back to our example

- **Accuracy:** prop. obs. classified correctly
    - (23+44)/(23+44+15+25) ~ 0.626
- **Sensitivity:** prop. of actual 1s that are classified correctly
    - 23/(23+25) ~ 0.479
- **Specificity:** prop. of actual 0s that are classified correctly
    - 44/(44+15) ~ 0.746

	Actually survived (1)	Actually died (0)
Predicted to survive (1)	23	15
Predicted to die (0)	25	44

Conclusions

- **Accuracy:** prop. obs. classified correctly ~ **0.626**
- **Sensitivity:** prop. of actual 1s that are classified correctly ~ **0.479**
- **Specificity:** prop. of actual 0s that are classified correctly ~ **0.746**
- **Our model is better at detecting deaths than survivals**

	Actually survived (1)	Actually died (0)
Predicted to survive (1)	23	15
Predicted to die (0)	25	44

Comparing models

- In practice, we might be considering more than one model (for example, we might fit a logistic regression and a regularized one)
- We can use metrics such as **accuracy**, **sensitivity**, and **specificity** to compare them
- Suppose we have 2 models
  - Model A: 0.725 **accuracy**, 0.5 **sensitivity**, 0.95 **specificity**
  - Model B: 0.675 **accuracy**, 0.85 **sensitivity**, 0.5 **specificity**
- Which one is better? It depends on our goals!

Let's work with real data

- Fit logistic and regularized (elastic net) logistic regressions, find confusion tables, and compare the models
- Training set: vicpena.github.io/workshops/2021/titanic_train.csv
- Test set: vicpena.github.io/workshops/2021/titanic_test.csv

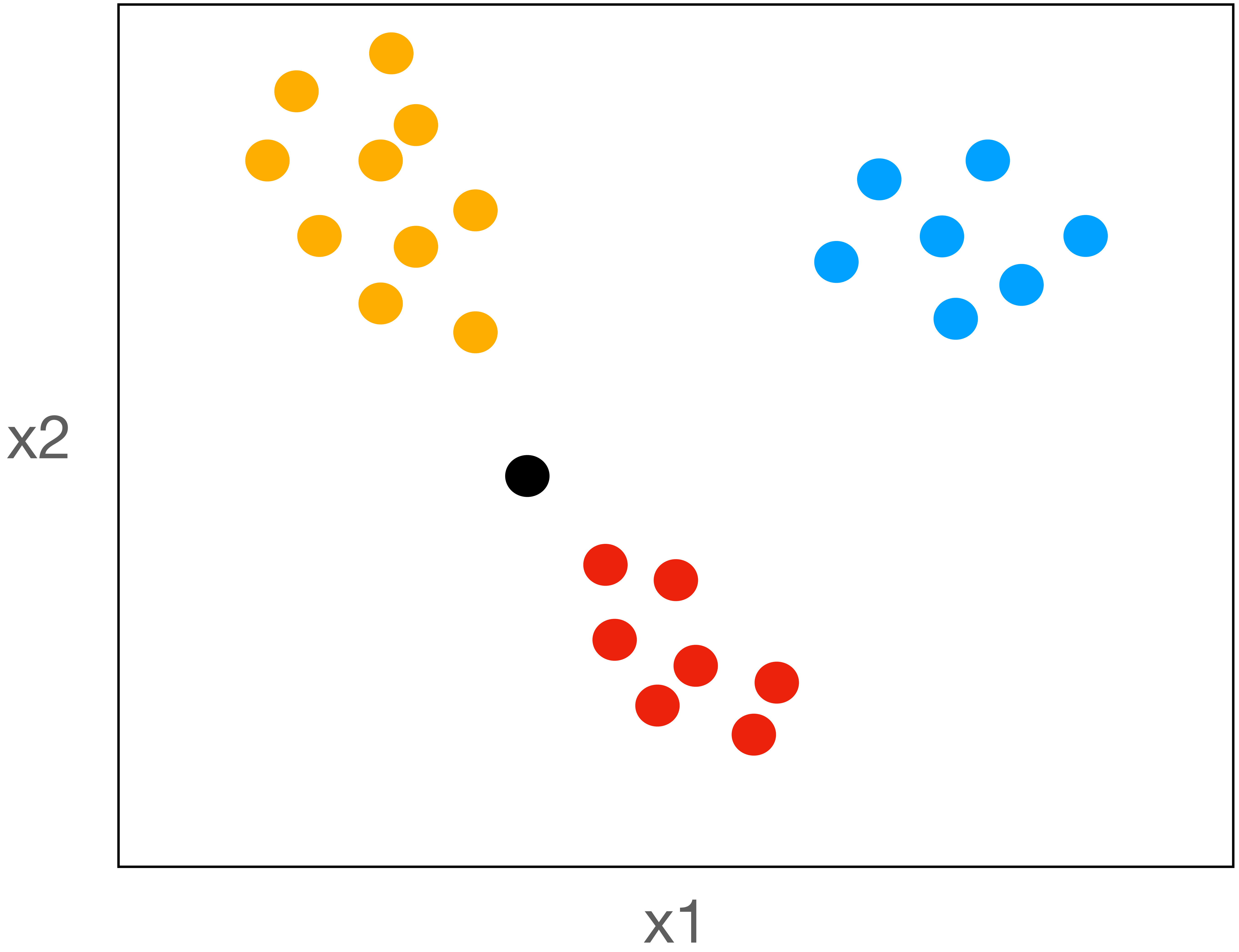
k-nn and regression trees

Classification with more than 2 categories

- Logistic regression (regularized or not) can only handle outcomes that have 2 categories
- Now, we'll see methods that allow us to deal with outcomes that have more
  - K-nearest neighbors (k-nn)
  - Regression trees

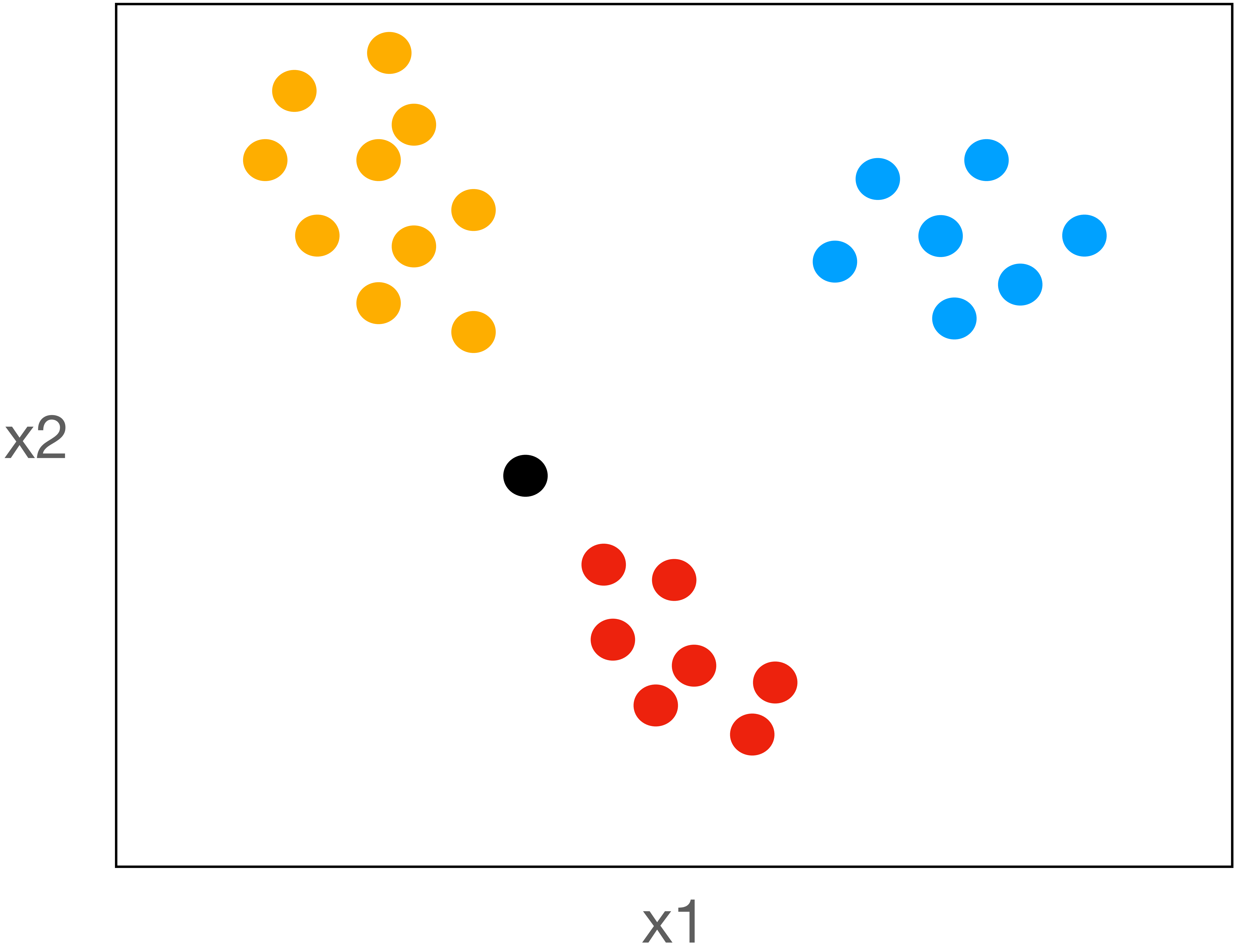
K-nearest neighbors (k-nn)

- Dataset with 3 variables
  - Outcome (y): can take on 3 values ●, ●, or ●
  - Predictors: x1 and x2, which are numeric
  - New observation ●...
  - Classify it as ●, ●, or ●?



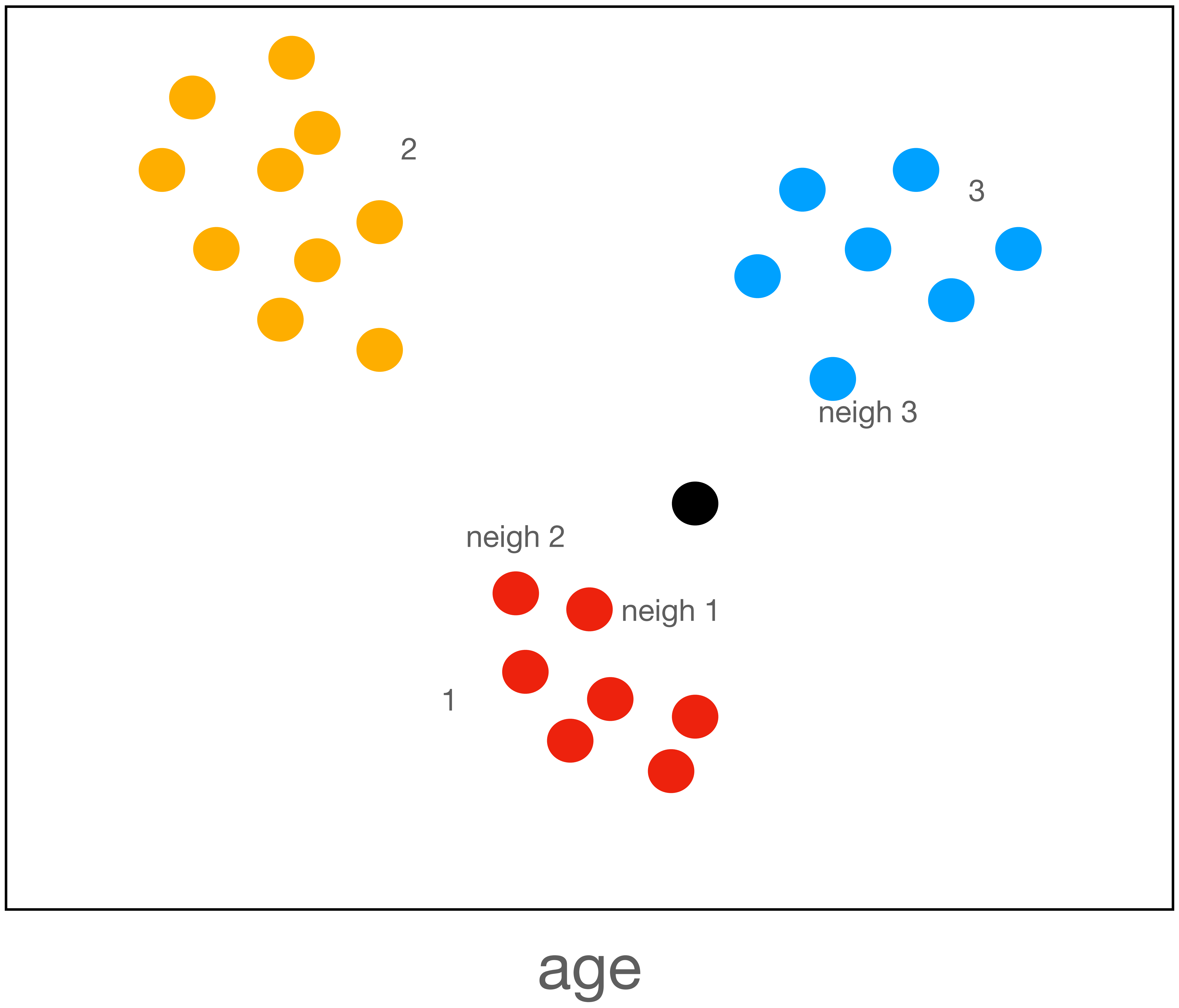
K-nearest neighbors (k-nn)

- New observation ●...
- Classify it as ●, ●, or ●?
- k = 1
  - Find closest observation to ●, classify it as such
  - In this case, we'd classify ● as ●



K-nearest neighbors (k-nn)

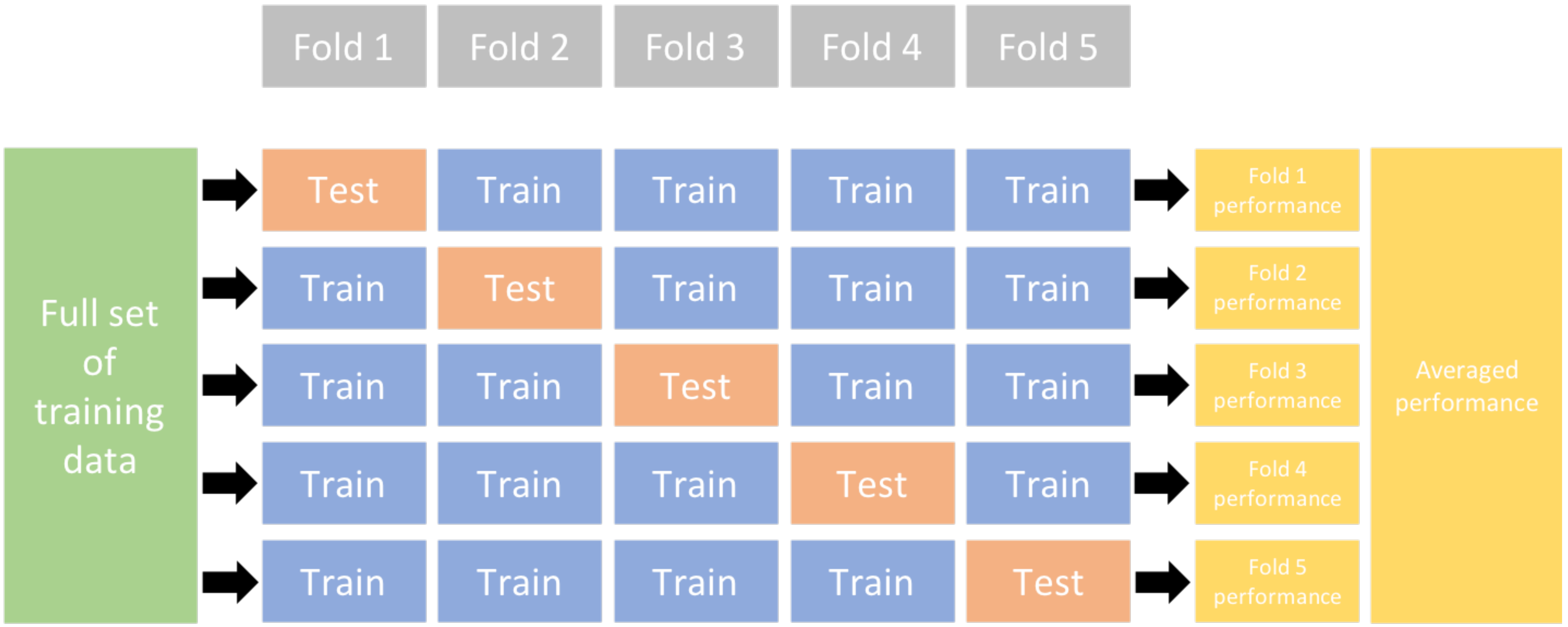
- New observation ⚫...
- Classify it as 🟠, 🔵, or 🔴?
- k = 3
    - Find 3 closest observations to ⚫
    - Here, it'd be 2 🔴 and 1 🟠
    - Assign to class with the most votes. Here, 🔴



How to pick k?

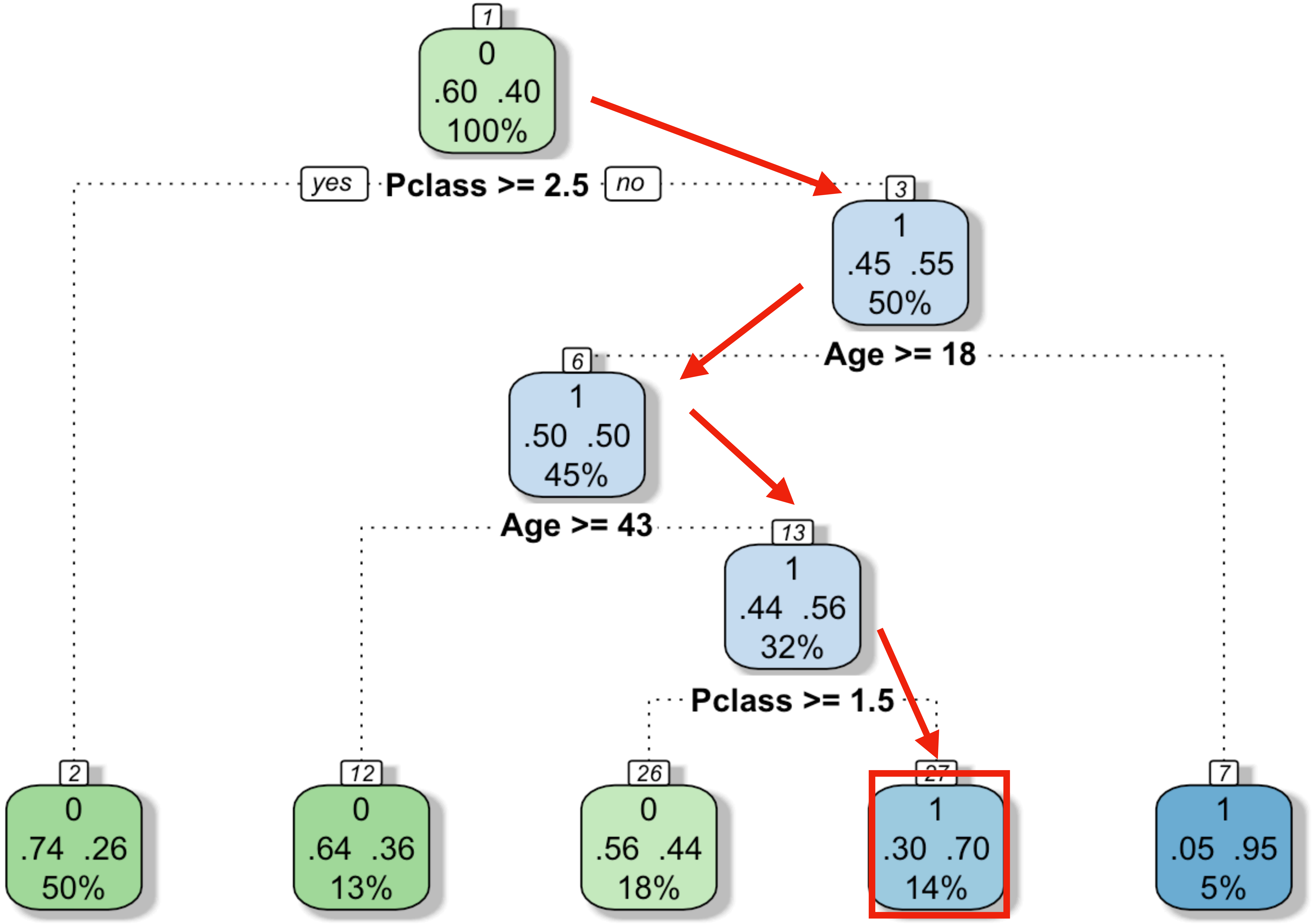
- Well...
  - In practice, library(caret) will pick it for us
- Without getting into too much detail, library(caret) can pick a "good" value of k using cross-validation
  - It estimates the predictive performance of the algorithm with different values of k and picks the one that seems to work best

k-fold cross validation



Regression trees with library(rpart)

- Let's start by looking at their output
- Simply go through the branches until you can't go further down
- **Titanic example:** Predict fate of a 31 year old that traveled 1st class
- Trees work with outcomes that have more than 2 categories (they'd look the same)



Predicted as 1 (survival!)
Prob(survival) = 0.7
14% of data in the training set
have this profile (18 <= Age < 43 & Pclass < 1.5)

How to split?

- How do we decide the variable we use for splitting at each step?
    - You pick a "sensible" criterion (there are many), and you pick the variable that optimizes it
        - **Example:** Gini impurity: Minimize the proportion of misclassified points you'd get if you were to use the tree to classify the points you already have

How are these trees fitted?

- Thankfully, library(rpart) and library(caret) will do everything for us
- The algorithm keeps partitioning the data into finer subsets until "it doesn't pay off" [Think about the usual trade-off of including more / fewer variables in regression. Same phenomenon occurs here]

Confusion tables with more than 2 categories

- Still working with the Titanic data, we build a model to predict the passenger class given the age and sex of the passengers
- **Accuracy: prop. of obs. correctly classified**
  - (15+12+20)/(15+6+3+10+12+1+5+7+20) ~ 0.595
- **Sensitivity? Prop. of 1s that I classify correctly (binary classifiers)**
- **Specificity? Prop. of 0s that I classify correctly (binary classifiers)**

	Actually 1st class	Actually 2nd class	Actually 3rd class
Predicted 1st class	15	6	3
Predicted 2nd class	10	12	1
Predicted 3rd class	5	7	20

Sensitivity

- Do it for each category separately
- **Sensitivity for 1st class: prop. of times that I classify actual 1st class correctly**
    - 15/(15+10+5) = 0.5
- **Sensitivity for 2nd class: prop. of times that I classify actual 2nd class correctly**
    - 12/(6+12+7) = 0.48
- **Sensitivity for 3rd class: prop. of times that I classify actually 3rd class correctly**
    - 20/(20+1+3) ~ 0.833

	Actually 1st class	Actually 2nd class	Actually 3rd class
Predicted 1st class	15	6	3
Predicted 2nd class	10	12	1
Predicted 3rd class	5	7	20

Specificity

- This one is a little confusing...
- **Specificity for 1st class**
    - Prop. of times that I classify ACTUAL NOT FIRST CLASS correctly
    - (12+20+1+7)/(6+12+7+3+1+20) ~ 0.816
- **Specificity for 2nd class**
    - Prop. of times that I classify ACTUAL not 2ND class correctly
    - (15+20+5+3)/(15+10+5+3+1+20) ~ 0.796
- **Specificity for 3rd class**
    - Prop. of times that I classify actual NOT in 3rd class correctly
    - (15+12+6+10)/(15+10+5+6+12+7) ~ 0.782

NOT FIRST CLASS

	Actually 1st class	Actually 2nd class	Actually 3rd class
Predicted 1st class	15	6	3
Predicted 2nd class	10	12	1
Predicted 3rd class	5	7	20

Other methods...

- There are many other classification methods that are easily implemented in library(caret)
- Some examples are random forests (method = "rf"), support vector machines, gradient boosting, neural nets, etc. All of these are implemented into caret. And they all share the same sort of syntax.

Let's go back to the titanic data

- Fit k-nn and regression trees for predicting the passenger class (3 category outcome), given the age, gender of the passengers, and whether they survived or not

- Training set: vicpena.github.io/workshops/2021/titanic_train.csv

- Test set: vicpena.github.io/workshops/2021/titanic_test.csv

References

- Hands-on Machine Learning with R, by Bradley Boehmke
- StatQuest, by Josh Starmer