Predictive Analysis of Text:
Concepts, Features, and Instances

Jaime Arguello
jarguell@email.unc.edu

August 27, 2014

Predictive Analysis of Text

- Objective: developing and evaluating computer programs that automatically detect a particular concept in natural language text

Predictive Analysis
basic ingredients

1. Training data: a set of positive and negative examples of the concept we want to automatically recognize
2. Representation: a set of features that we believe are useful in recognizing the desired concept
3. Learning algorithm: a computer program that uses the training data to learn a predictive model of the concept

Predictive Analysis
basic ingredients

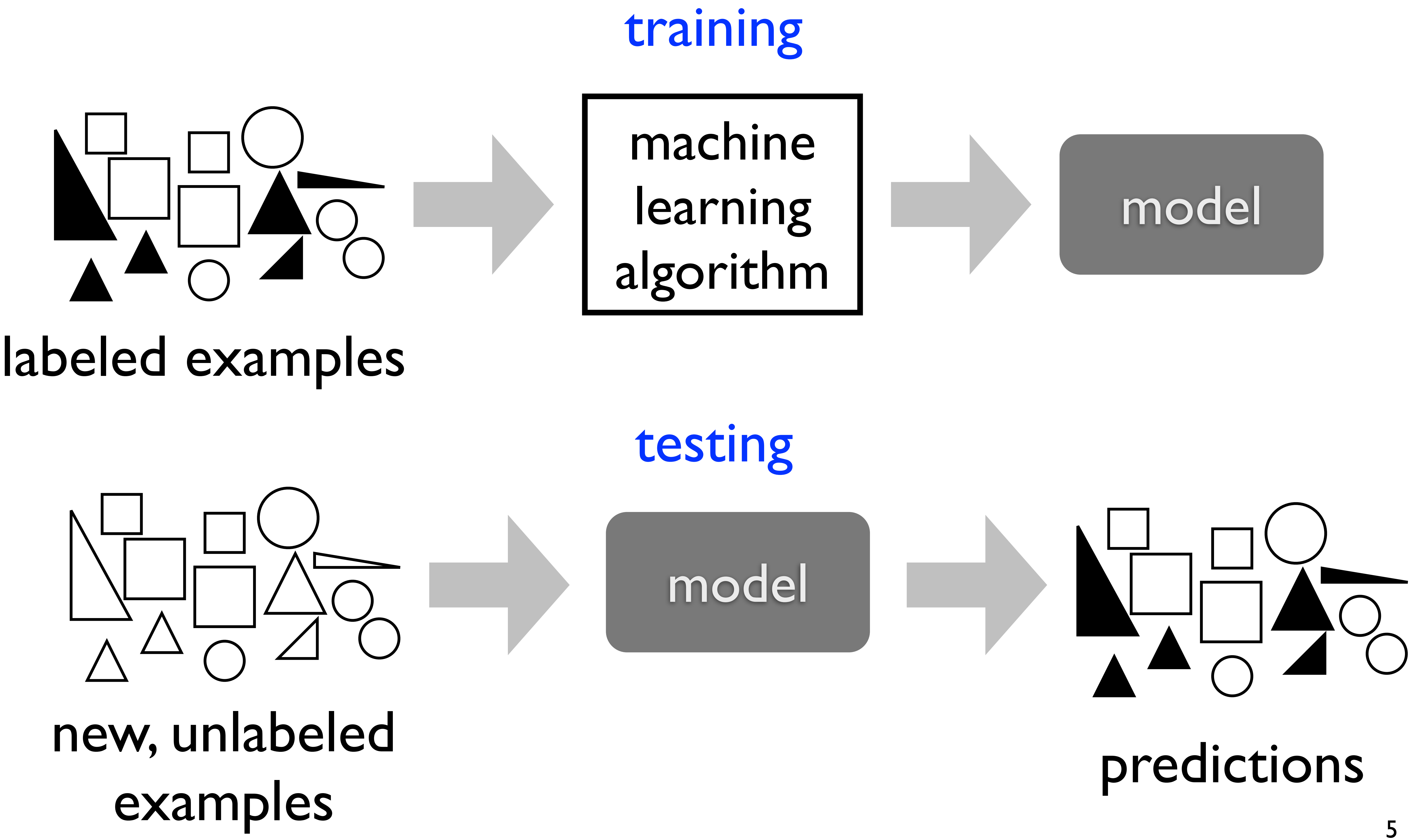
4. Model: a function that describes a predictive relationship between feature values and the presence/absence of the concept

5. Test data: a set of previously unseen examples used to estimate the model's effectiveness

6. Performance metrics: a set of statistics used to measure the predictive effectiveness of the model

Predictive Analysis

training and testing



Predictive Analysis

concept, instances, and features

features concept

instances

color	size	# slides	equal sides	...	label
red	big	3	no	...	yes
green	big	3	yes	...	yes
blue	small	inf	yes	...	no
blue	small	4	yes	...	no
⋮	⋮	⋮	⋮	⋮	⋮
red	big	3	yes	...	yes

Predictive Analysis

training and testing

training

color	size	sides	equal sides	...	label
red	big	3	no	...	yes
green	big	3	yes	...	yes
blue	small	inf	yes	...	no
blue	small	4	yes	...	no
⋮	⋮	⋮	⋮	⋮	⋮
red	big	3	yes	...	yes

labeled examples

machine learning algorithm

model

testing

color	size	sides	equal sides	...	label
red	big	3	no	...	???
green	big	3	yes	...	???
blue	small	inf	yes	...	???
blue	small	4	yes	...	???
⋮	⋮	⋮	⋮	⋮	???
red	big	3	yes	...	???

new, unlabeled examples

model

color	size	sides	equal sides	...	label
red	big	3	no	...	yes
green	big	3	yes	...	yes
blue	small	inf	yes	...	no
blue	small	4	yes	...	no
⋮	⋮	⋮	⋮	⋮	⋮
red	big	3	yes	...	yes

predictions

Predictive Analysis

questions

- Is a particular concept appropriate for predictive analysis?
- What should the unit of analysis be?
- How should I divide the data into training and test sets?
- What is a good feature representation for this task?
- What type of learning algorithm should I use?
- How should I evaluate my model's performance?

Predictive Analysis
concepts

- Learning algorithms can recognize some concepts better than others
- What are some properties of concepts that are easier to recognize?

Predictive Analysis

concepts

- Option 1: can a human recognize the concept?

Predictive Analysis
concepts

- Option 1: can a human recognize the concept?
- Option 2: can two or more humans recognize the concept independently and do they agree?

Predictive Analysis
concepts

- Option 1: can a human recognize the concept?
- Option 2: can two or more humans recognize the concept independently and do they agree?
- Option 2 is better.
- In fact, models are sometimes evaluated as an independent assessor
- How does the model's performance compare to the performance of one assessor with respect to another?
  - One assessor produces the "ground truth" and the other produces the "predictions"

Predictive Analysis

measures agreement: percent agreement

- Percent agreement: percentage of instances for which both assessors agree that the concept occurs or does not occur

	yes	no
yes	A	B
no	C	D

$$\frac{(? + ?)}{(? + ? + ? + ?)}$$

Predictive Analysis

measures agreement: percent agreement

- Percent agreement: percentage of instances for which both assessors agree that the concept occurs or does not occur

	yes	no
yes	A	B
no	C	D

$$\frac{(A + D)}{(A + B + C + D)}$$

Predictive Analysis

measures agreement: percent agreement

- Percent agreement: percentage of instances for which both assessors agree that the concept occurs or does not occur

	yes	no	
yes	5	5	10
no	15	75	90
	20	80	

% agreement = ???

Predictive Analysis

measures agreement: percent agreement

- Percent agreement: percentage of instances for which both assessors agree that the concept occurs or does not occur

	yes	no	
yes	5	5	10
no	15	75	90
	20	80	

% agreement = (5 + 75) / 100 = 80%

Predictive Analysis

measures agreement: percent agreement

- Problem: percent agreement does not account for agreement due to random chance.
- How can we compute the expected agreement due to random chance?
  - Option 1: assume unbiased assessors
  - Option 2: assume biased assessors

Predictive Analysis

kappa agreement: chance-corrected % agreement

- Option 1: unbiased assessors

	yes	no	
yes	??	??	50
no	??	??	50
	50	50	

Predictive Analysis

kappa agreement: chance-corrected % agreement

- Option 1: unbiased assessors

	yes	no	
yes	25	25	50
no	25	25	50
	50	50	

Predictive Analysis

kappa agreement: chance-corrected % agreement

- Option 1: unbiased assessors

	yes	no	
yes	25	25	50
no	25	25	50
	50	50	

random chance % agreement = ???

Predictive Analysis

kappa agreement: chance-corrected % agreement

- Option 1: unbiased assessors

	yes	no	
yes	25	25	50
no	25	25	50
	50	50	

random chance % agreement = (25 + 25)/100 = 50%

Predictive Analysis

kappa agreement: chance-corrected % agreement

- Kappa agreement: percent agreement after correcting for the expected agreement due to random chance

$$\mathcal{K} = \frac{P(a) - P(e)}{1 - P(e)}$$

- P(a) = percent of observed agreement
- P(e) = percent of agreement due to random chance

Predictive Analysis

kappa agreement: chance-corrected % agreement

- Kappa agreement: percent agreement after correcting for the expected agreement due to unbiased chance

	yes	no	
yes	5	5	10
no	15	75	90
	20	80	

	yes	no	
yes	25	25	50
no	25	25	50
	50	50	

$$P(a) = \frac{5+75}{100} = 0.80$$

$$P(e) = \frac{25+25}{100} = 0.50$$

$$\mathcal{K} = \frac{P(a)-P(e)}{1-P(e)} = \frac{0.80-0.50}{1-0.50} = 0.60$$

Predictive Analysis

kappa agreement: chance-corrected % agreement

- Option 2: biased assessors

	yes	no	
yes	5	5	10
no	15	75	90
	20	80	

biased chance % agreement = ???

Predictive Analysis

kappa agreement: chance-corrected % agreement

- Kappa agreement: percent agreement after correcting for the expected agreement due to <u>biased</u> chance

	yes	no	
yes	5	5	10
no	15	75	90
	20	80	

$$P(a) = \boxed{\frac{5+75}{100}} = 0.80 \qquad P(e) = \left(\boxed{\frac{10}{100}} \times \boxed{\frac{20}{100}}\right) + \left(\boxed{\frac{90}{100}} \times \boxed{\frac{80}{100}}\right) = 0.74$$

$$\mathcal{K} = \frac{P(a) - P(e)}{1 - P(e)} = \frac{0.80 - 0.74}{1 - 0.74} = 0.23$$

Predictive Analysis

data annotation process

- **INPUT**: unlabeled data, annotators, coding manual
- **OUTPUT**: labeled data

1. using the latest coding manual, have <u>all</u> annotators label some previously unseen portion of the data (~10%)
2. measure inter-annotator agreement (Kappa)
3. **IF** agreement < X, **THEN**:
    - refine coding manual using disagreements to resolve inconsistencies and clarify definitions
    - return to 1

   ELSE
    - have annotators label the remainder of the data <u>independently</u> and **EXIT**

Predictive Analysis
data annotation process

- What is good (Kappa) agreement?
- It depends on who you ask
- According to Landis and Koch, 1977:
  - 0.81 - 1.00: almost perfect
  - 0.61 - 0.70: substantial
  - 0.41 - 0.60: moderate
  - 0.21 - 0.40: fair
  - 0.00 - 0.20: slight
  - < 0.00: no agreement

Predictive Analysis

questions

- Is a particular concept appropriate for predictive analysis?
- What should the unit of analysis be?
- How should I divide the data into training and test sets?
- What is a good feature representation for this task?
- What type of learning algorithm should I use?
- How should I evaluate my model's performance?

Predictive Analysis

turning data into (training and test) instances

- For many text-mining applications, turning the data into instances for training and testing is fairly straightforward
- Easy case: instances are self-contained, independent units of analysis
  - text classification: instances = documents
  - opinion mining: instances = product reviews
  - bias detection: instances = political blog posts
  - emotion detection: instances = support group posts

Text Classification

predicting health-related documents

features | **concept**

instances

w_1	w_2	w_3	...	w_n	label
1	1	0	...	0	health
0	0	0	...	0	other
0	0	0	...	0	other
0	1	0	...	1	other
⋮	⋮	⋮	...	0	⋮
1	0	0	...	1	health

Opinion Mining

predicting positive/negative movie reviews

features

concept

instances

w_1	w_2	w_3	...	w_n	label
1	1	0	...	0	positive
0	0	0	...	0	negative
0	0	0	...	0	negative
0	1	0	...	1	negative
⋮	⋮	⋮	...	0	⋮
1	0	0	...	1	positive

Bias Detection

predicting liberal/conservative blog posts

features | **concept**

instances

w_1	w_2	w_3	...	w_n	label
1	1	0	...	0	liberal
0	0	0	...	0	conservative
0	0	0	...	0	conservative
0	1	0	...	1	conservative
⋮	⋮	⋮	...	0	⋮
1	0	0	...	1	liberal

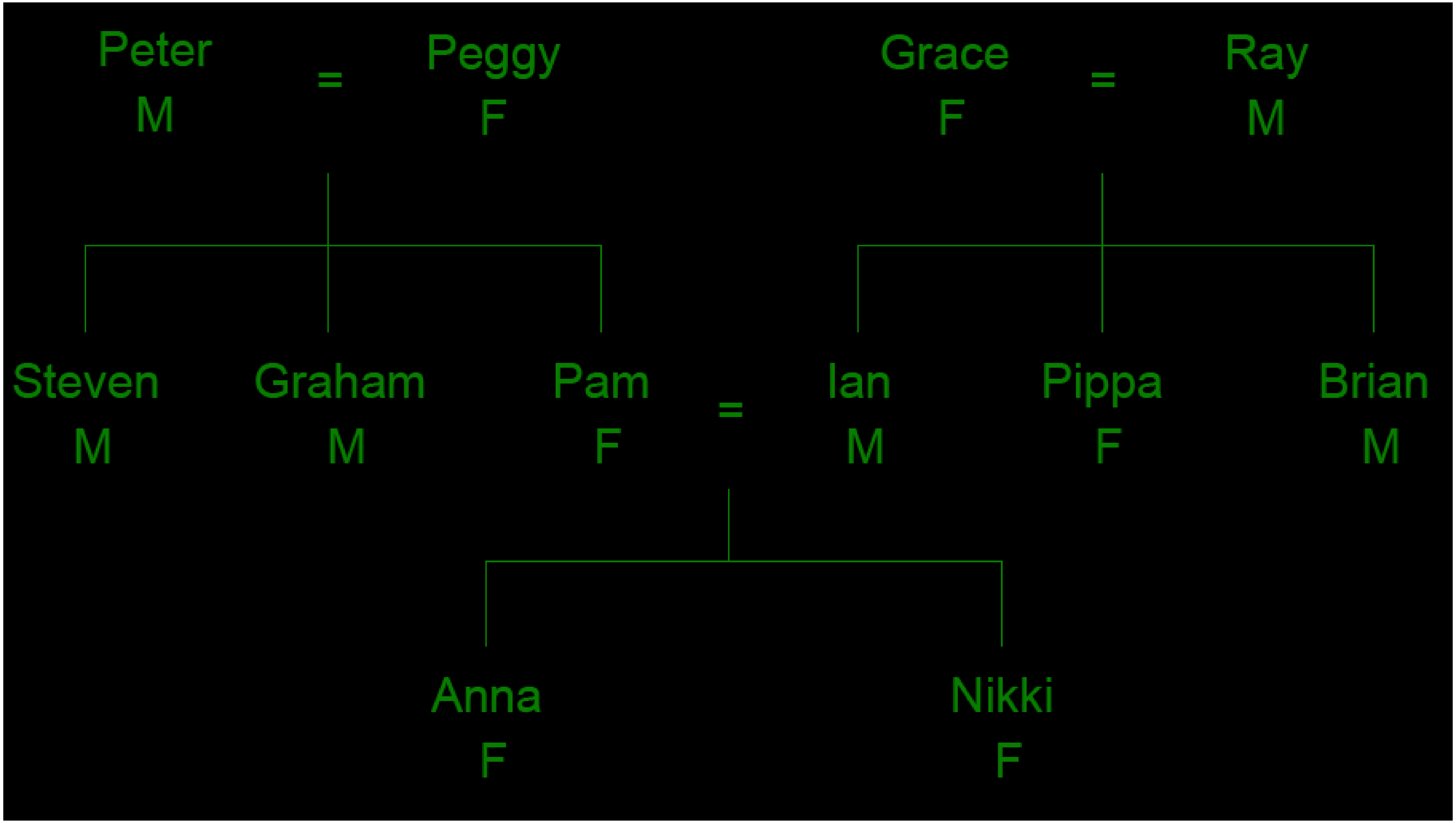
Predictive Analysis

turning data into (training and test) instances

- A not-so-easy case: relational data
- The concept to be learned is a <u>relation</u> between pairs of objects

Predictive Analysis

example of relational data: Brother(X,Y)



(example borrowed and modified from Witten *et al.* textbook)

Predictive Analysis

example of relational data: Brother(X,Y)

features **concept**

instances

name_1	gender_1	mother_1	father_1	name_2	gender_2	mother_2	father_2	brother
steven	male	peggy	peter	graham	male	peggy	peter	yes
Ian	male	grace	ray	brian	male	grace	ray	yes
anna	female	pam	ian	nikki	female	pam	ian	no
pippa	female	grace	ray	brian	male	grace	ray	no
steven	male	peggy	peter	brian	male	grace	ray	no
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
anna	female	pam	ian	brian	male	grace	ray	no

Predictive Analysis

turning data into (training and test) instances

- A not-so-easy case: relational data
- Each instance should correspond to an object pair (which may or may not share the relation of interest)
- May require features that characterize properties of the pair

Predictive Analysis

example of relational data: Brother(X,Y)

features **concept**

instances

name_1	gender_1	mother_1	father_1	name_2	gender_2	mother_2	father_2	brother
steven	male	peggy	peter	graham	male	peggy	peter	yes
Ian	male	grace	ray	brian	male	grace	ray	yes
anna	female	pam	ian	nikki	female	pam	ian	no
pippa	female	grace	ray	brian	male	grace	ray	no
steven	male	peggy	peter	brian	male	grace	ray	no
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
anna	female	pam	ian	brian	male	grace	ray	no

(can we think of a better feature representation?)

Predictive Analysis

example of relational data: Brother(X,Y)

features **concept**

instances

gender_1	gender_2	same parents	brother
male	male	yes	yes
male	male	yes	yes
female	female	no	no
female	male	yes	no
male	male	no	no
⋮	⋮	⋮	⋮
female	male	no	no

Predictive Analysis

turning data into (training and test) instances

- A not-so-easy case: relational data
- There is still an issue that we're not capturing! Any ideas?
- Hint: In this case, should the predicted labels really be independent?

Predictive Analysis

turning data into (training and test) instances

Brother(A,B) = yes

Brother(B,C) = yes

Brother(A,C) = no

Predictive Analysis

turning data into (training and test) instances

- In this case, what we would really want is:
  - a method that does <u>joint</u> prediction on the test set
  - a method whose joint predictions satisfy a set of known properties about the data as a whole (e.g., transitivity)

Predictive Analysis

turning data into (training and test) instances

- There are learning algorithms that incorporate relational constraints between predictions
- However, they are beyond the scope of this class
- We'll be covering algorithms that make independent predictions on instances
- That said, many algorithms output prediction confidence values
- Heuristics can be used to disfavor inconsistencies

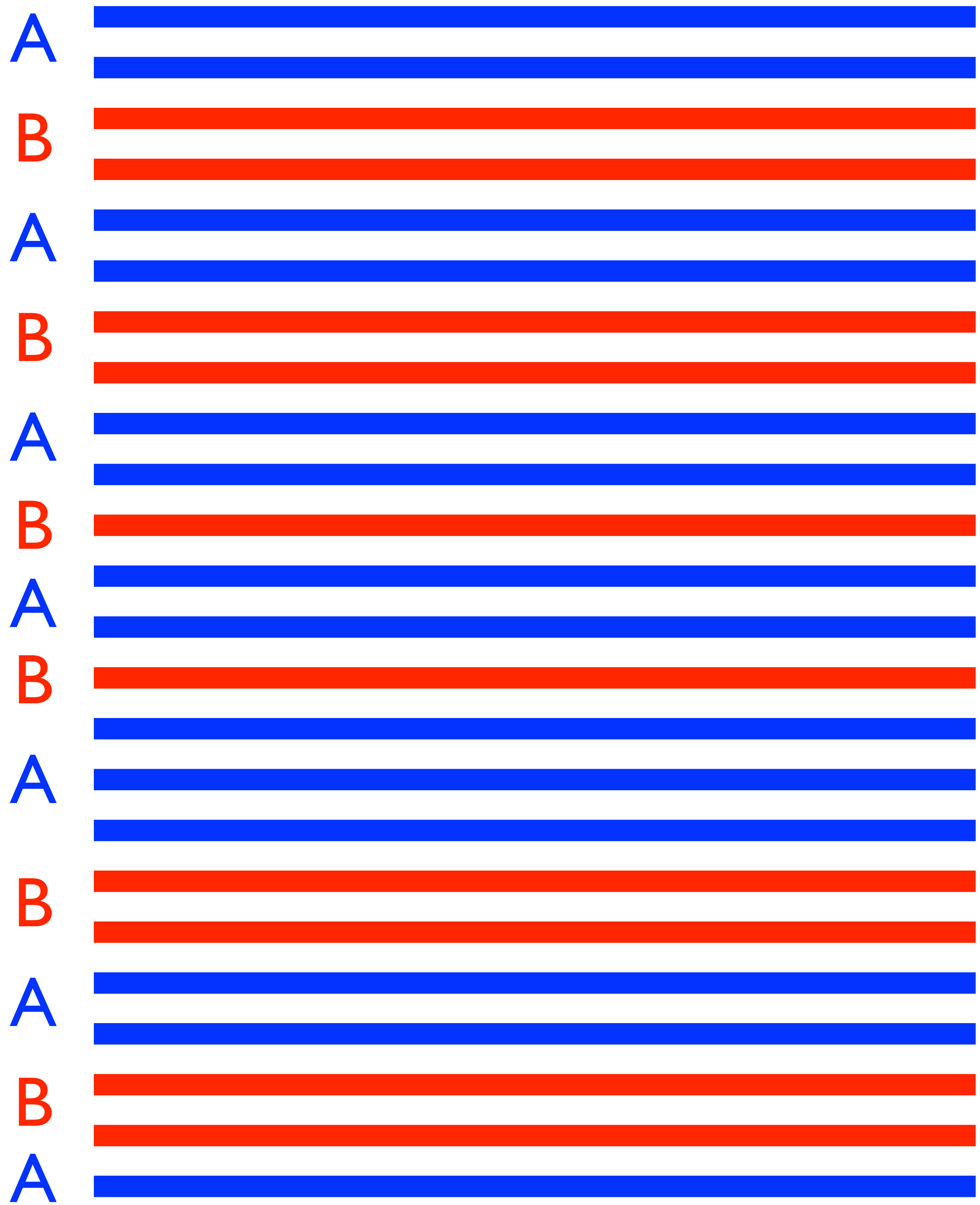
Predictive Analysis

turning data into (training and test) instances

- Examples of relational data in text-mining:
  - information extraction: predicting that a word-sequence belongs to a particular class (e.g., person, location)
  - topic segmentation: segmenting discourse into topically coherent chunks

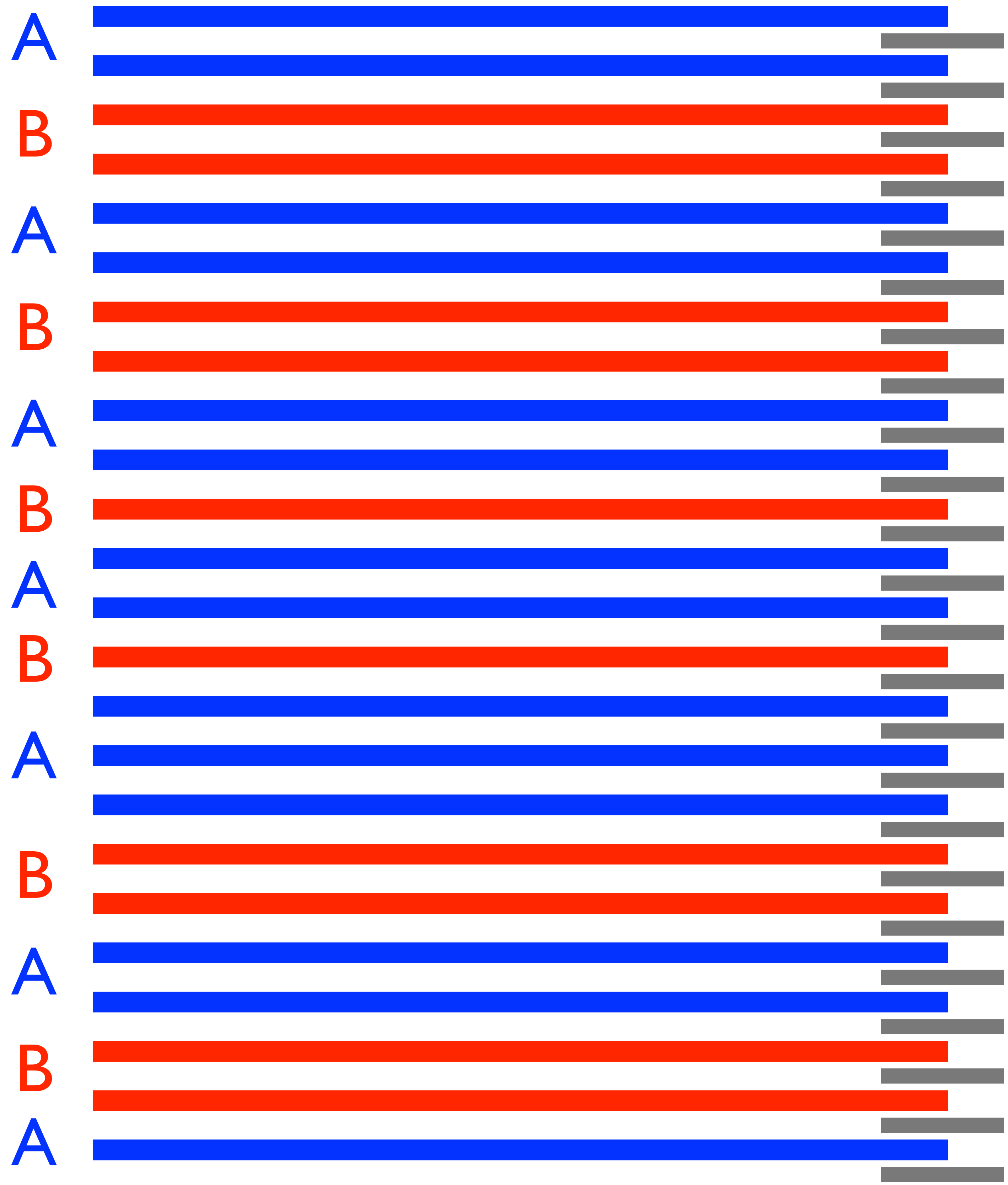
Predictive Analysis

topic segmentation example



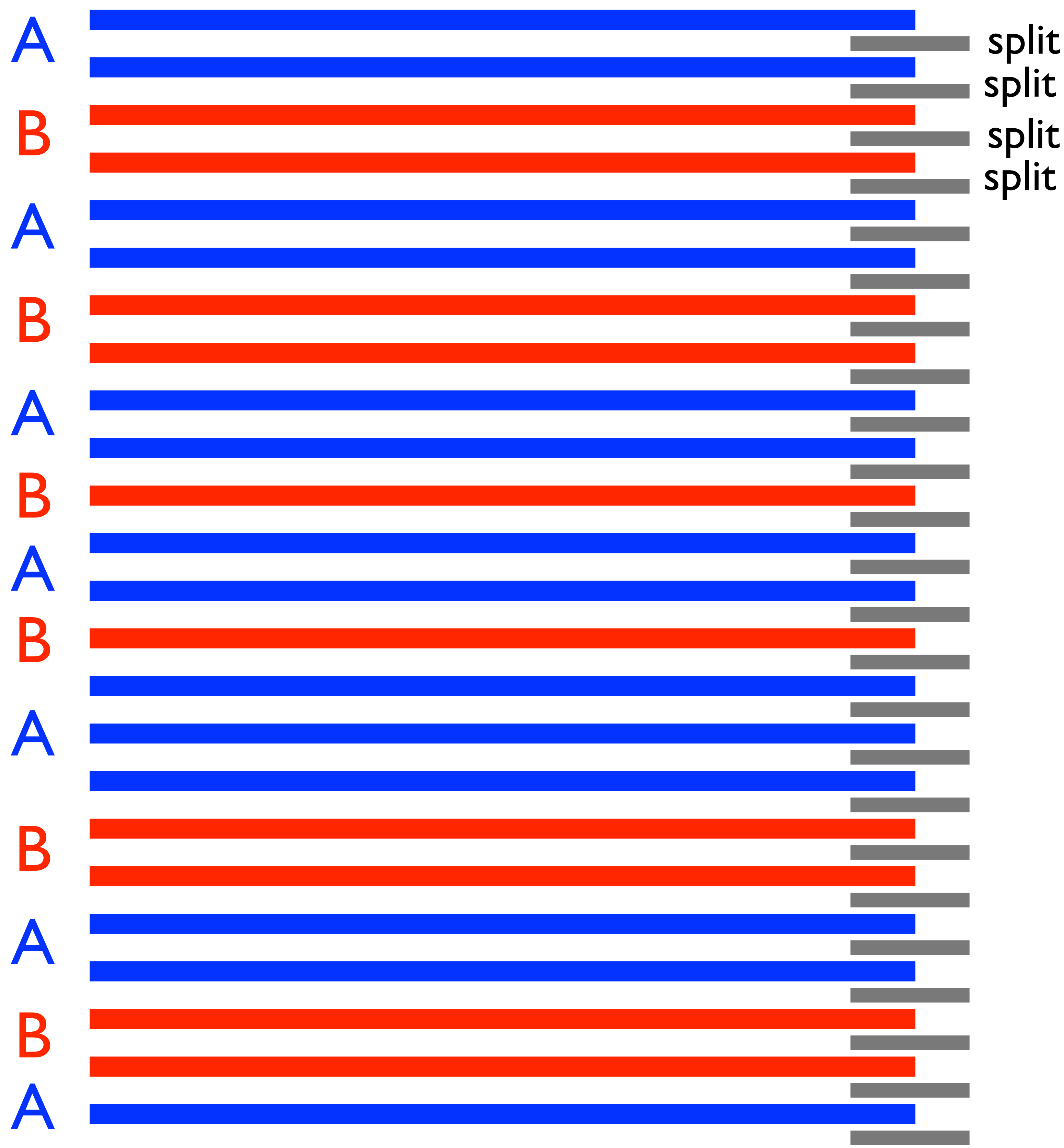
Predictive Analysis

topic segmentation example: instances



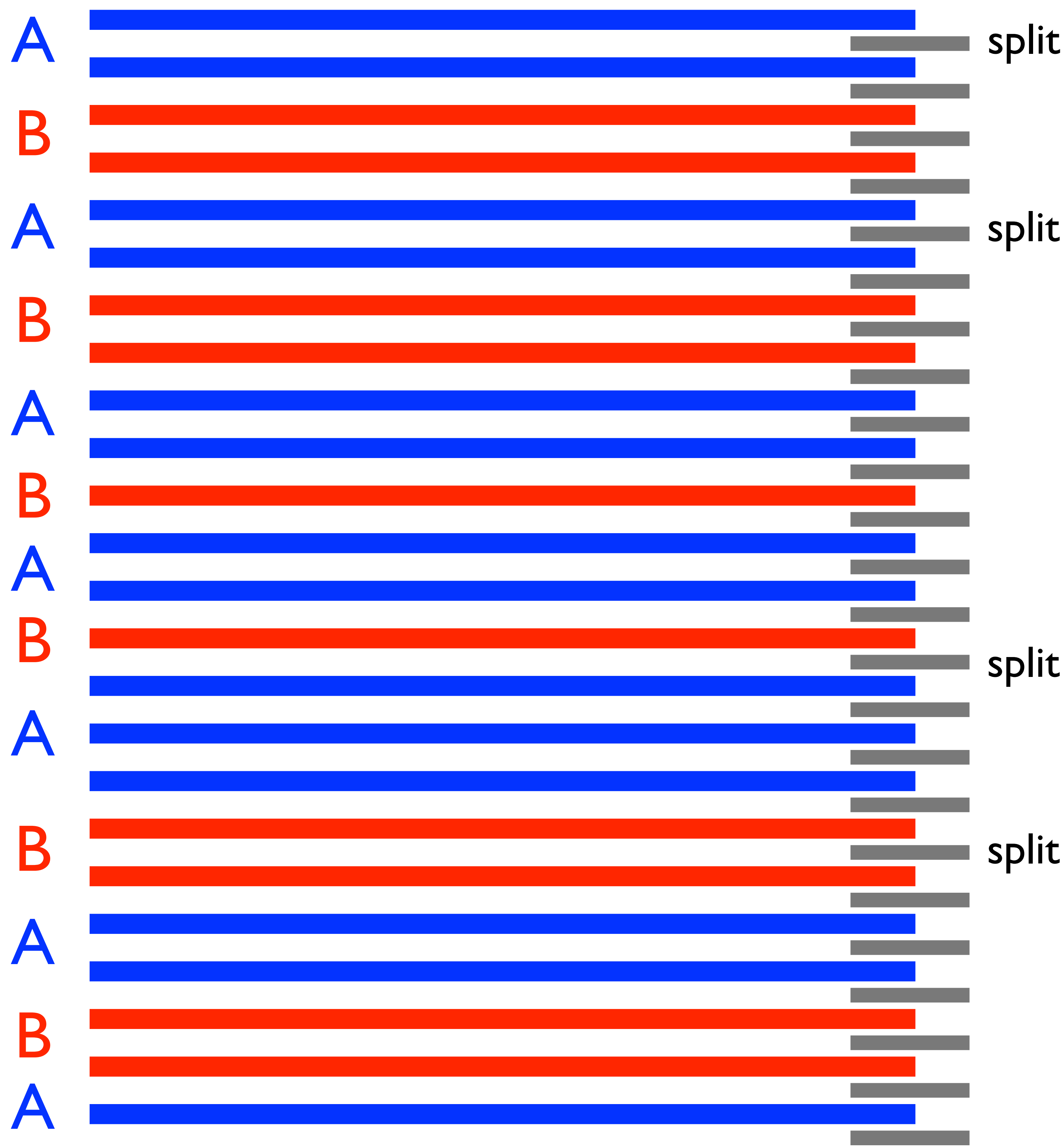
Predictive Analysis

topic segmentation example: independent instances?



Predictive Analysis

topic segmentation example: independent instances?



Predictive Analysis

questions

- Is a particular concept appropriate for predictive analysis?
- What should the unit of analysis be?
- How should I divide the data into training and test sets?
- What is a good feature representation for this task?
- What type of learning algorithm should I use?
- How should I evaluate my model's performance?

Predictive Analysis
training and test data

- We want our model to “learn” to recognize a concept
- So, what does it mean to <u>learn</u>?

Predictive Analysis
training and test data

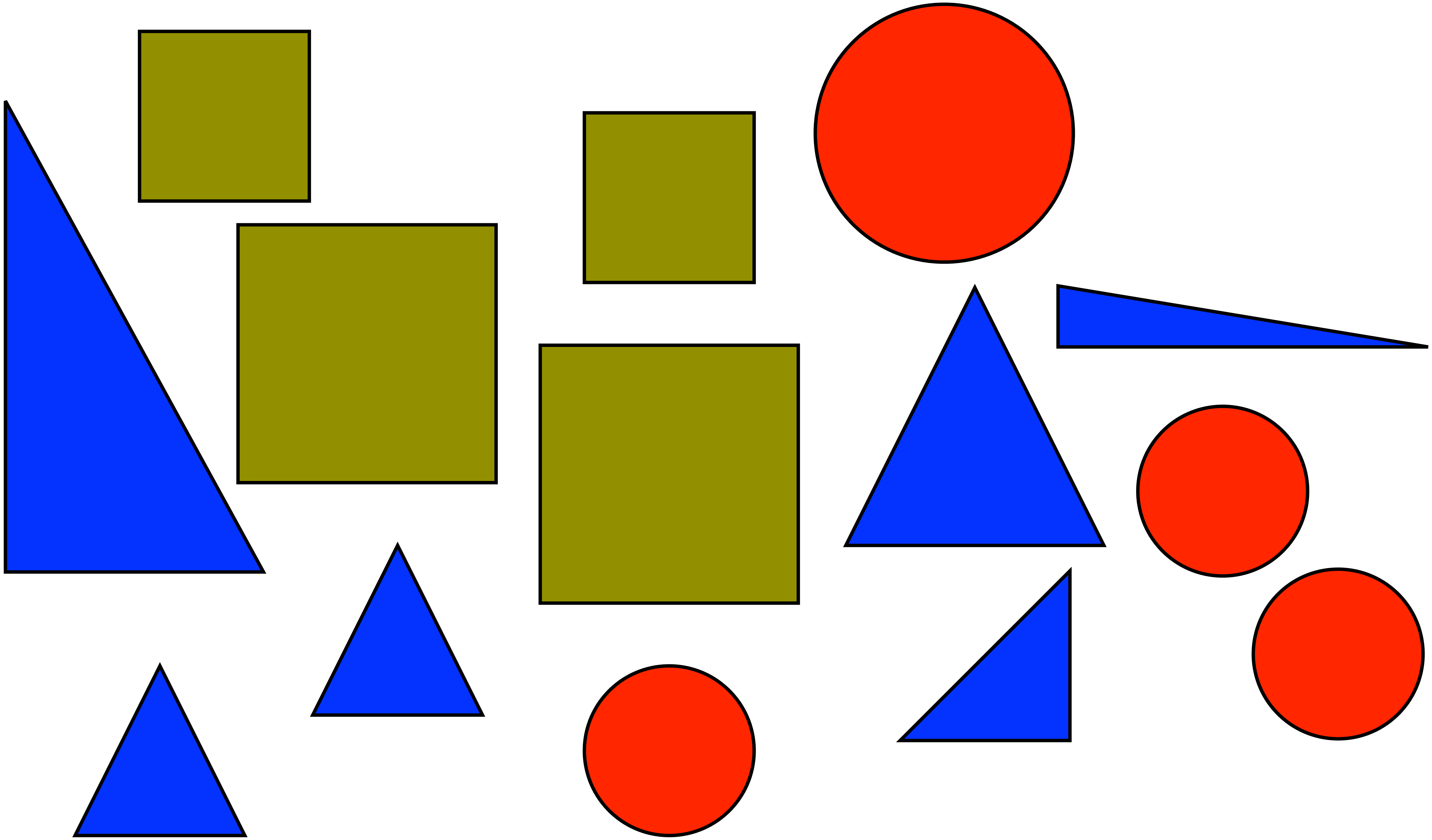
- The machine learning definition of *learning*:

  A machine *learns* with respect to a particular task T, performance metric P, and experience E, if the system improves its performance P at task T following experience E. -- Tom Mitchell

Predictive Analysis
training and test data

- We want our model to improve its generalization performance!
- That is, its performance on previously unseen data!
- Generalize: to derive or induce a general conception or principle from particulars. -- Merriam-Webster
- In order to test generalization performance, the training and test data cannot be the same.
- Why?

Training data + Representation
what could possibly go wrong?

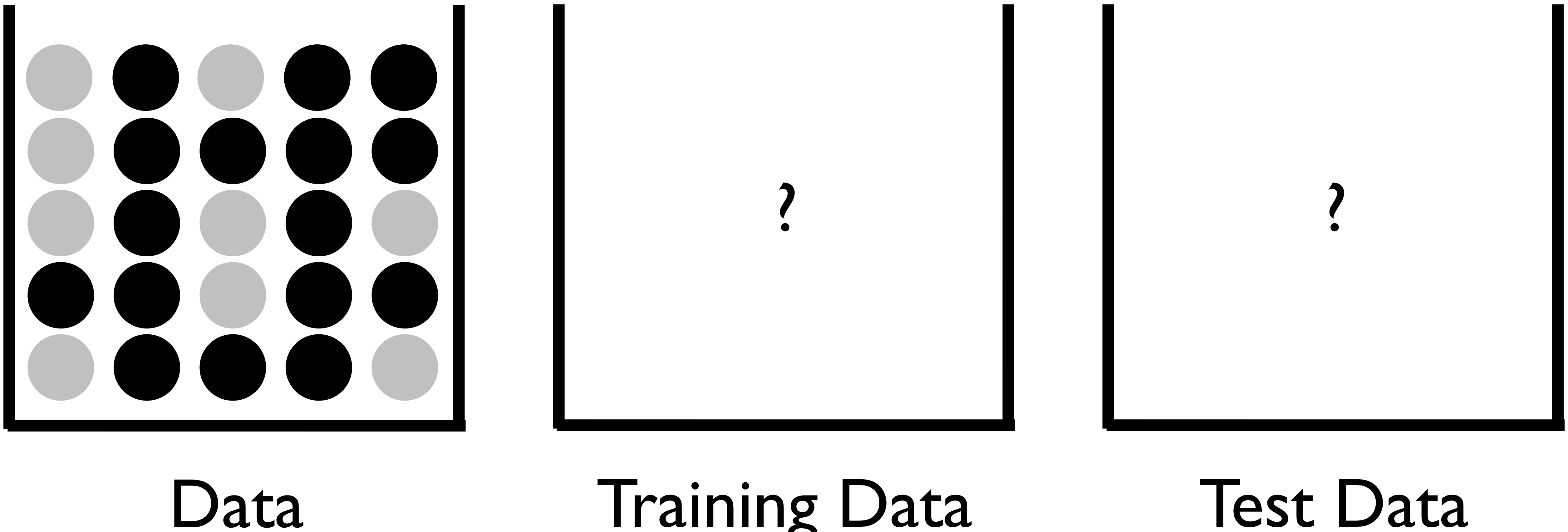


Predictive Analysis
training and test data

- While we don’t want to test on training data, models usually perform the best when the training and test set are derived from the same “probability distribution”.
- What does that mean?

Predictive Analysis

training and test data

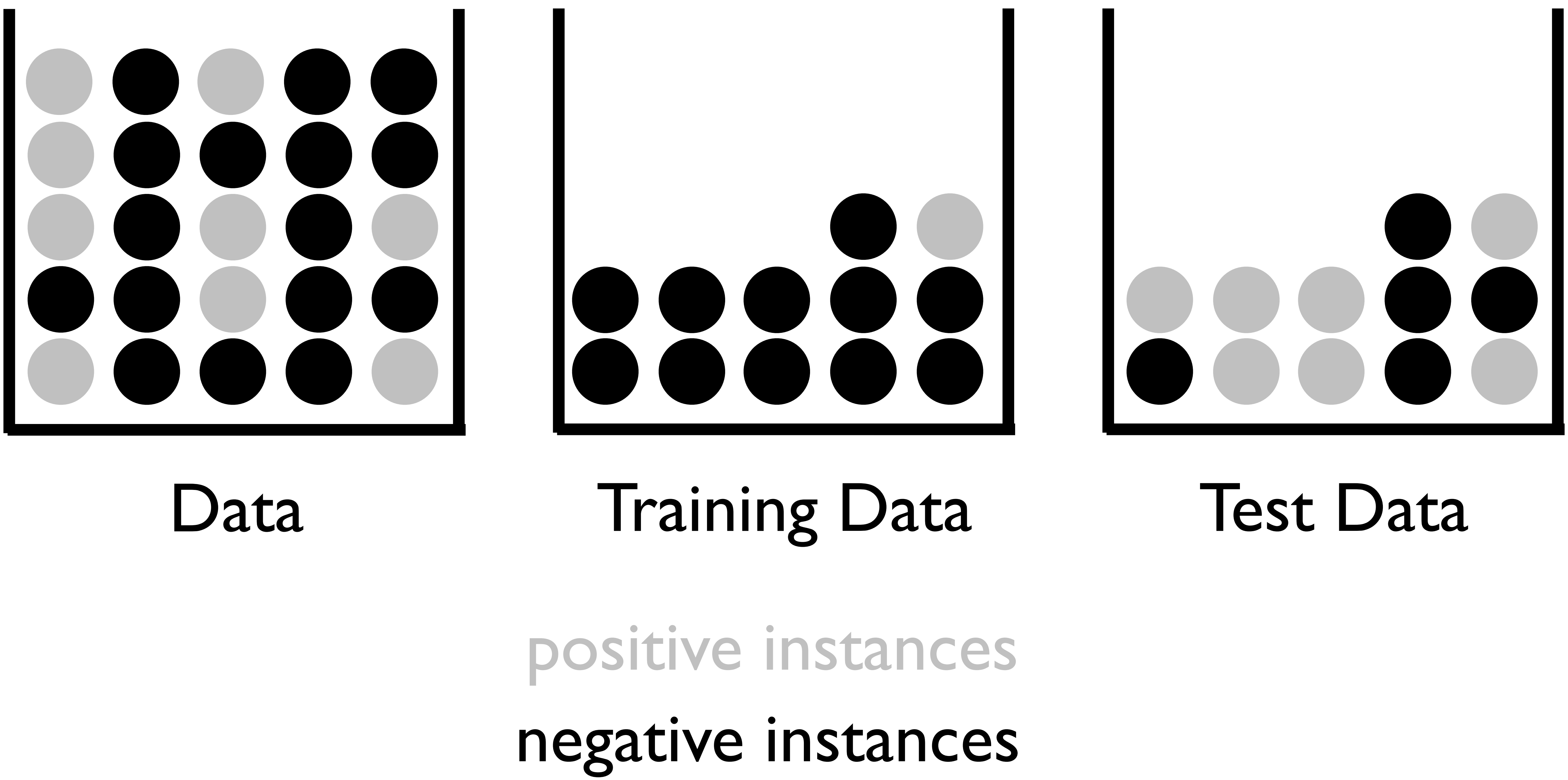


positive instances

negative instances

Predictive Analysis
training and test data

- Is this a good partitioning? Why or why not?



Predictive Analysis

training and test data

Data | Training Data | Test Data

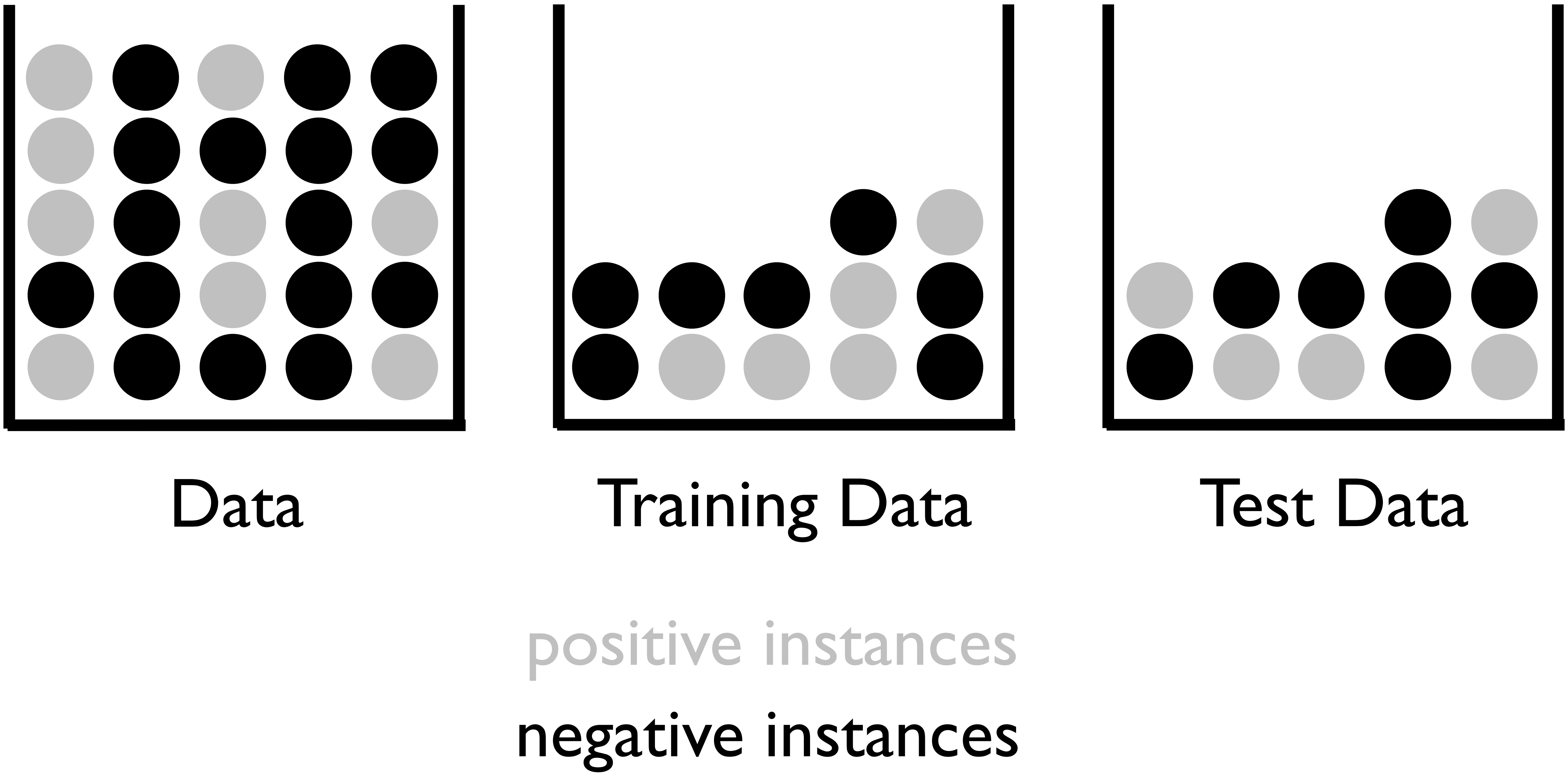
positive instances

negative instances

Predictive Analysis

training and test data

- On average, random sampling should produce comparable data for training and testing



Predictive Analysis
training and test data

- Models usually perform the best when the training and test set have:
    - a similar proportion of positive and negative examples
    - a similar co-occurrence of feature-values and each target class value

Predictive Analysis

training and test data

- Caution: in some situations, partitioning the data randomly might inflate performance in an unrealistic way!
- How the data is split into training and test sets determines what we can claim about generalization performance
- The appropriate split between training and test sets is usually determined on a case-by-case basis

Predictive Analysis
discussion

- Spam detection: should the training and test sets contain email messages from the <u>same sender</u>, <u>same recipient</u>, and/or <u>same timeframe</u>?
- Topic segmentation: should the training and test sets contain potential boundaries from the <u>same discourse</u>?
- Opinion mining for movie reviews: should the training and test sets contain reviews for the <u>same movie</u>?
- Sentiment analysis: should the training and test sets contain blog posts from the <u>same discussion thread</u>?

Predictive Analysis

questions

- Is a particular concept appropriate for predictive analysis?
- What should the unit of analysis be?
- How should I divide the data into training and test sets?
- What is a good feature representation for this task?
- What type of learning algorithm should I use?
- How should I evaluate my model's performance?

Predictive Analysis
three types of classifiers

- Linear classifiers
- Decision tree classifiers
- Instance-based classifiers

Predictive Analysis

three types of classifiers

- All types of classifiers learn to make predictions based on the input feature values
- However, different types of classifiers combine the input feature values in different ways
- Chapter 3 in the book refers to a trained model as knowledge representation

Predictive Analysis

linear classifiers: perceptron algorithm

$$y = \begin{cases} 1 & \text{if } w_0 + \sum_{j=1}^{n} w_j x_j > 0 \\ 0 & \text{otherwise} \end{cases}$$

Predictive Analysis

linear classifiers: perceptron algorithm

$$y = \begin{cases} \boxed{1} & \text{if } \boxed{w_0} + \sum_{j=1}^{n} \boxed{w_j} x_j > 0 \\ \boxed{0} & \text{otherwise} \end{cases}$$

parameters learned by the model

predicted value (e.g., 1 = positive, 0 = negative)

Predictive Analysis

linear classifiers: perceptron algorithm

test instance

f_1	f_2	f_3
0.5	1.0	0.2

model weights

w_0	w_1	w_2	w_3
2.0	-5.0	2.0	1.0

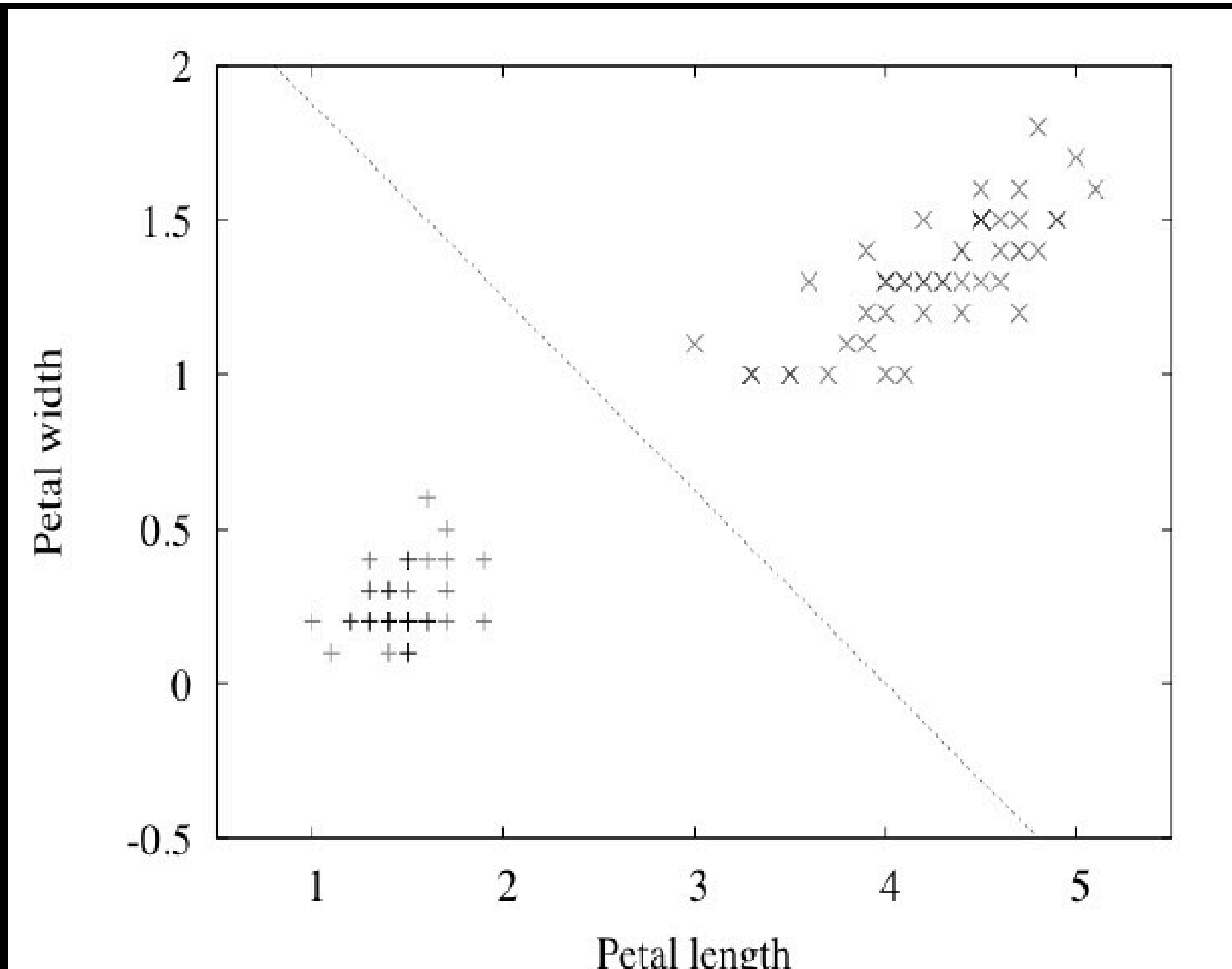
output = 2.0 + (0.50 x -5.0) + (1.0 x 2.0) + (0.2 x 1.0)

output = 1.7

output prediction = positive

Predictive Analysis

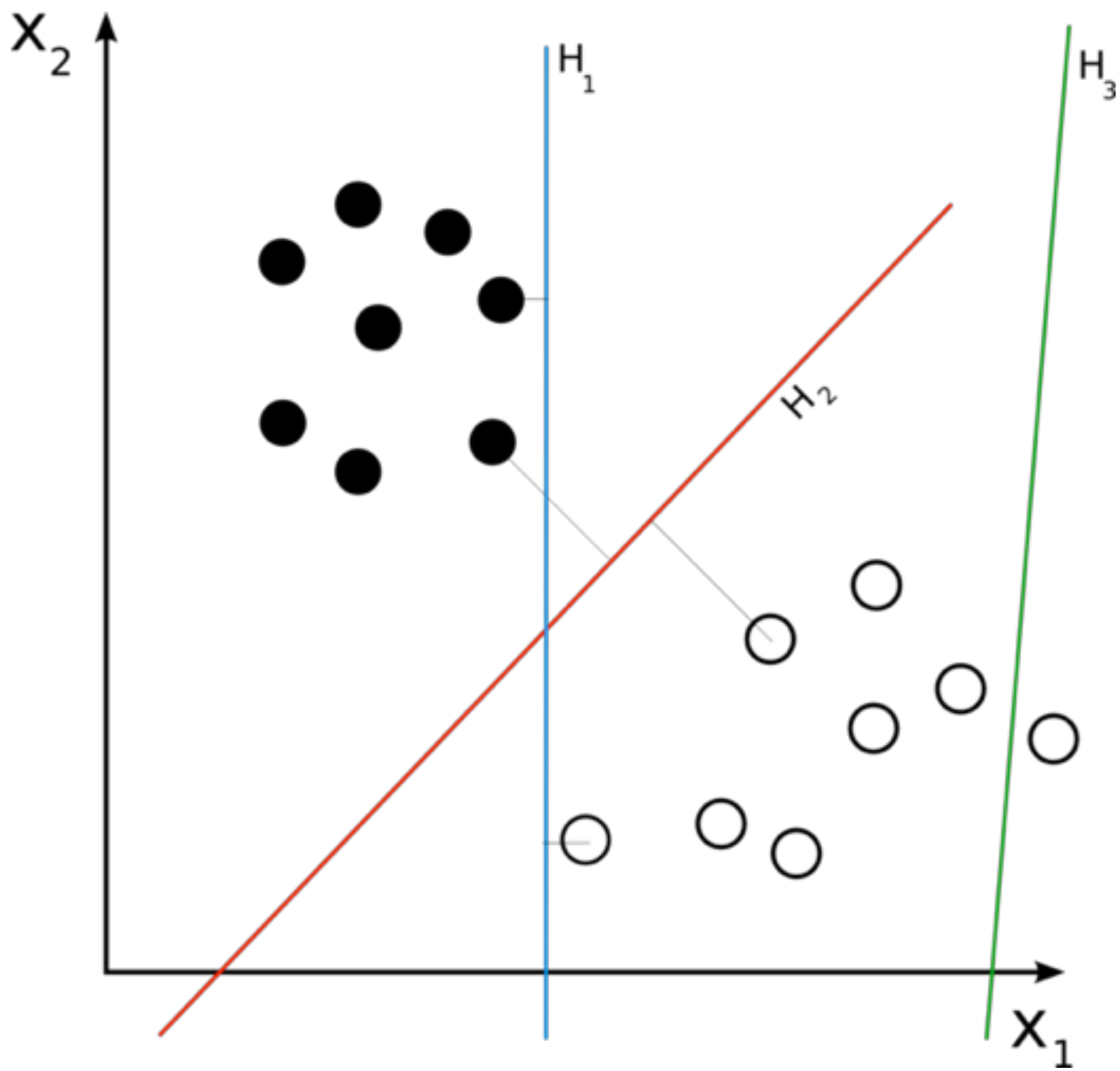
linear classifiers: perceptron algorithm



2.0 – 0.5PETAL-LENGTH – 0.8PETAL-WIDTH = 0

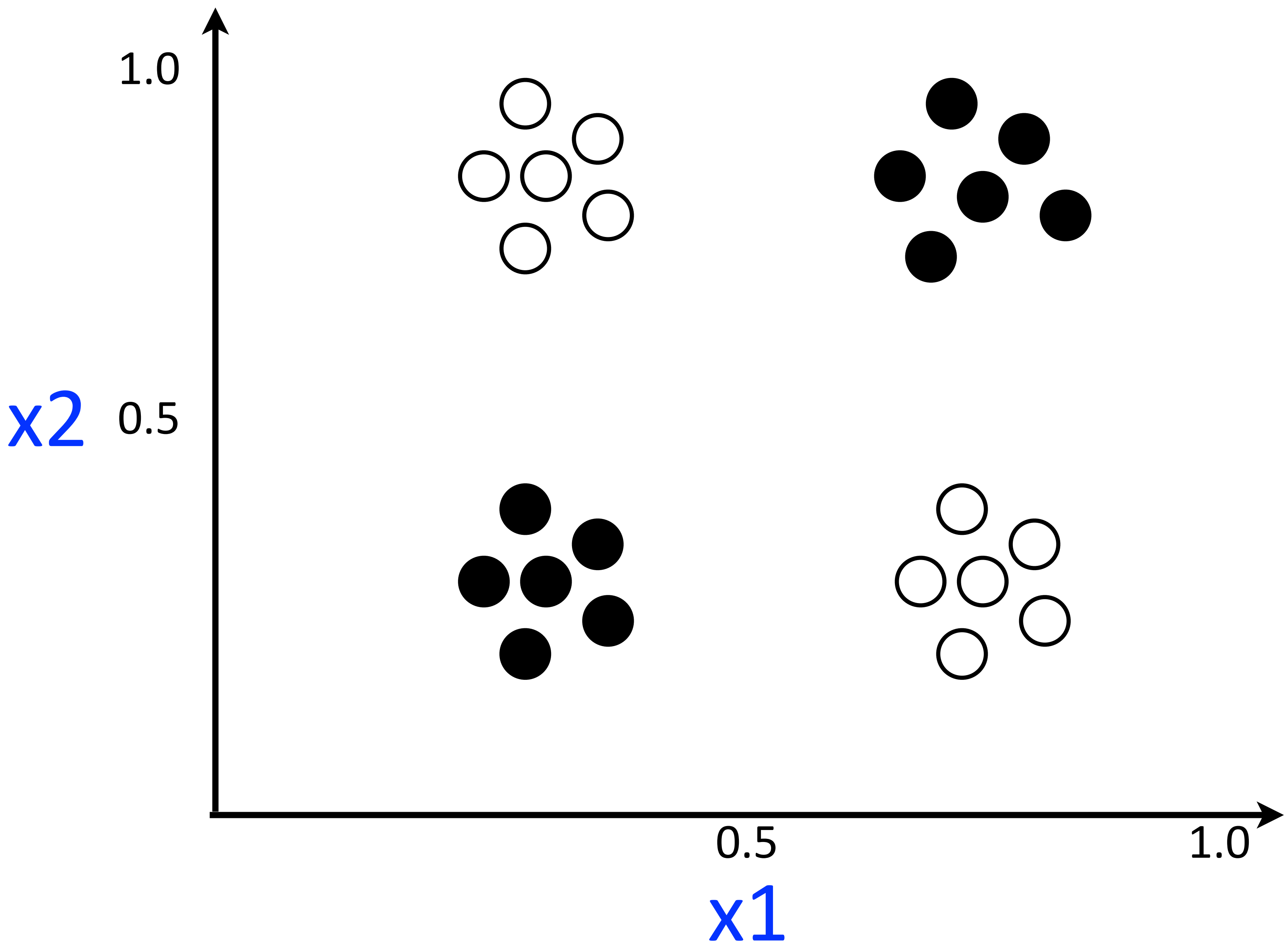
(two-feature example borrowed from Witten *et al.* textbook)

Predictive Analysis
linear classifiers: perceptron algorithm



(source: http://en.wikipedia.org/wiki/File:Svm_separating_hyperplanes.png)

Predictive Analysis
linear classifiers: perceptron algorithm



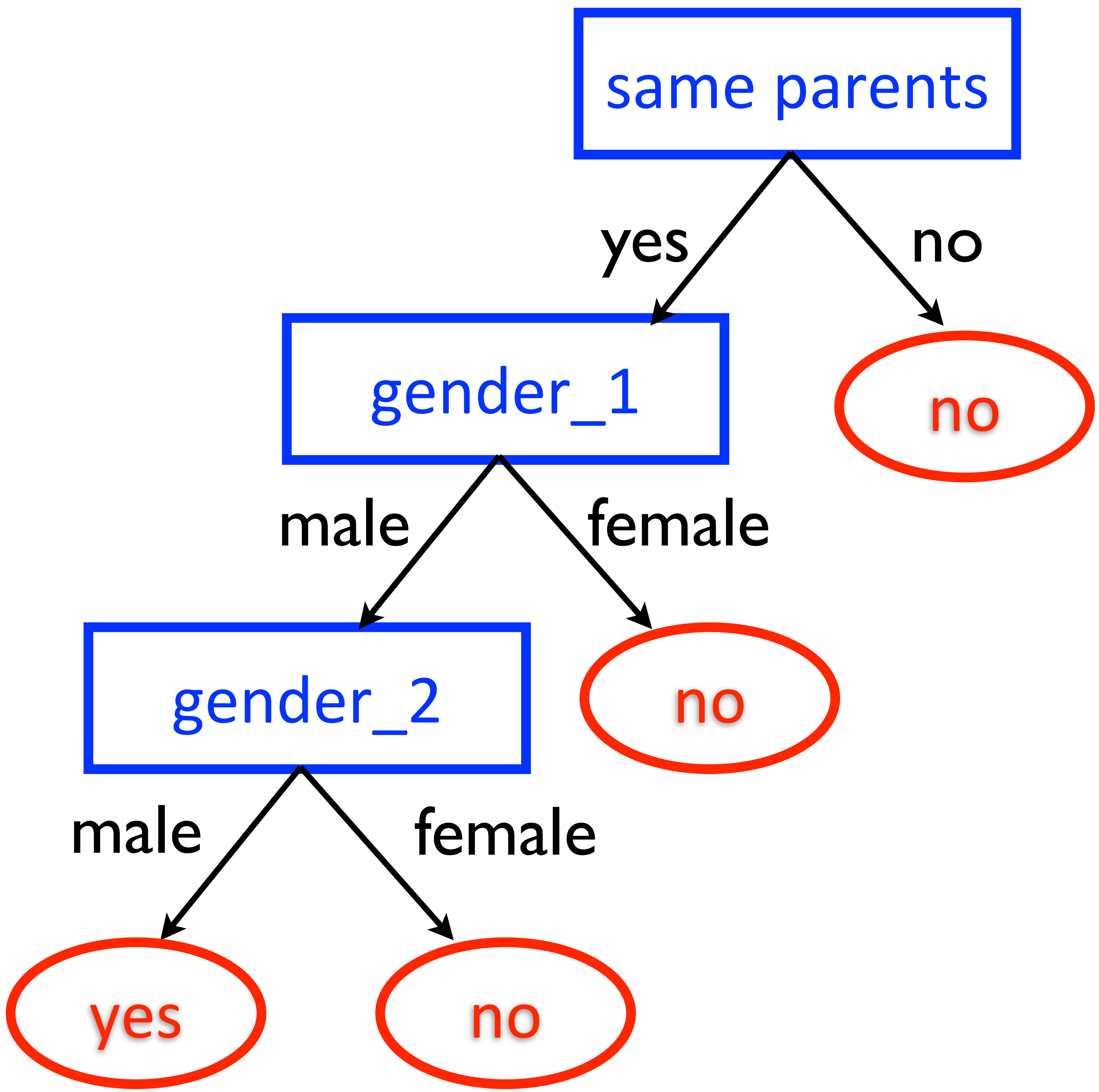
- Would a linear classifier do well on positive (black) and negative (white) data that looks like this?

Predictive Analysis
three types of classifiers

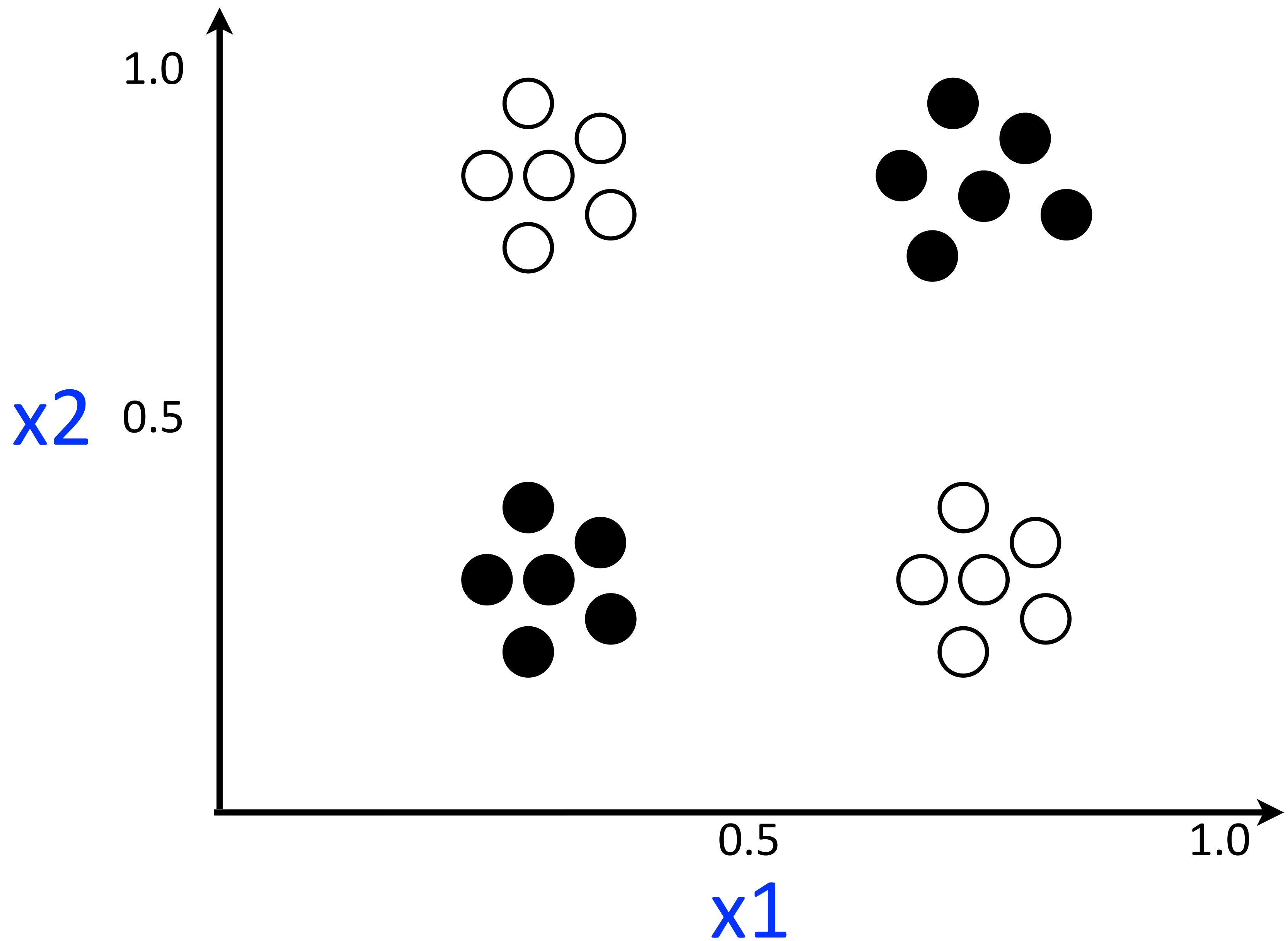
- Linear classifiers
- Decision tree classifiers
- Instance-based classifiers

Predictive Analysis

example of decision tree classifier: Brother(X,Y)



Predictive Analysis
decision tree classifiers



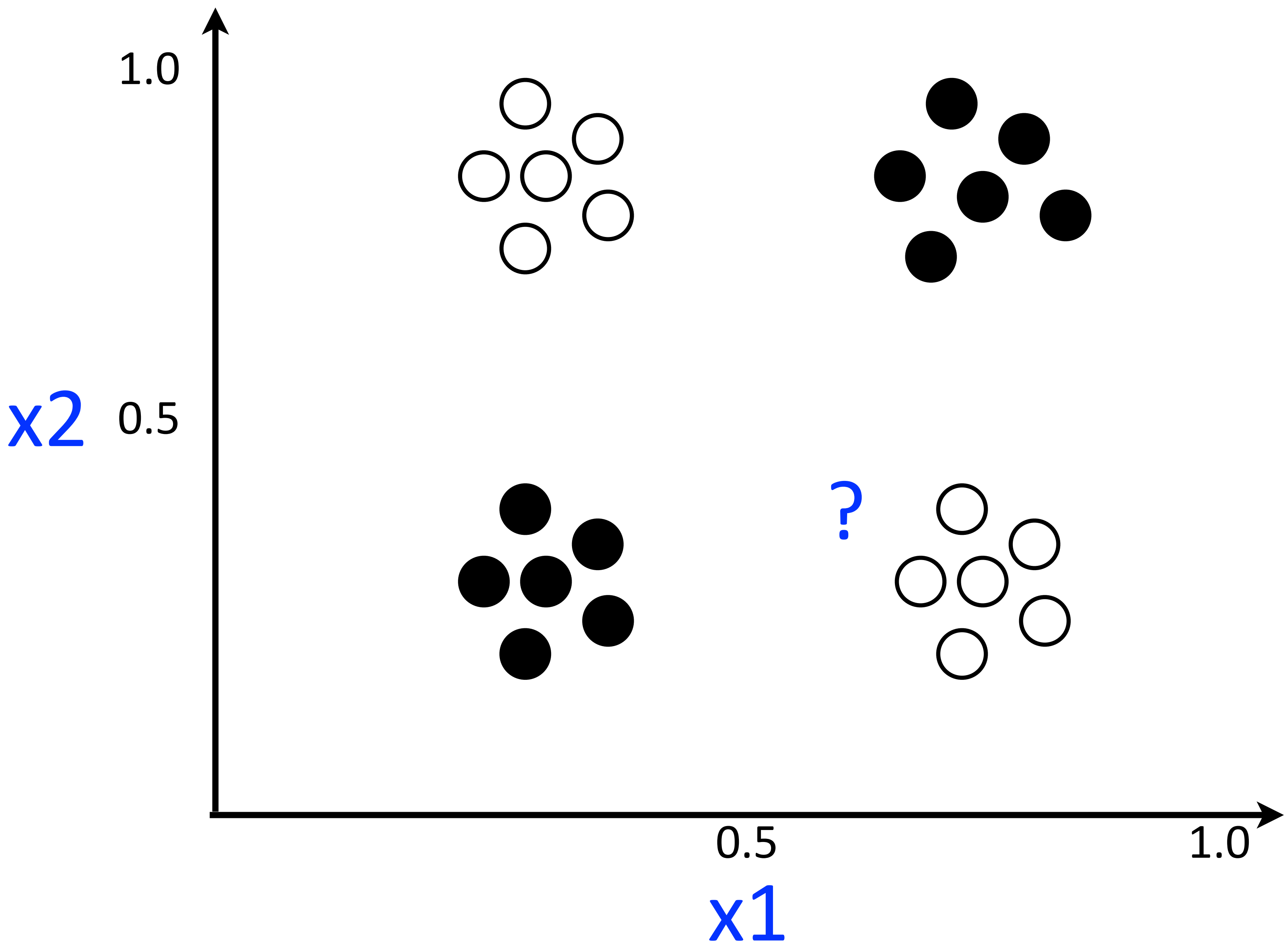
- Draw a decision tree that would perform perfectly on this training data!

Predictive Analysis

three types of classifiers

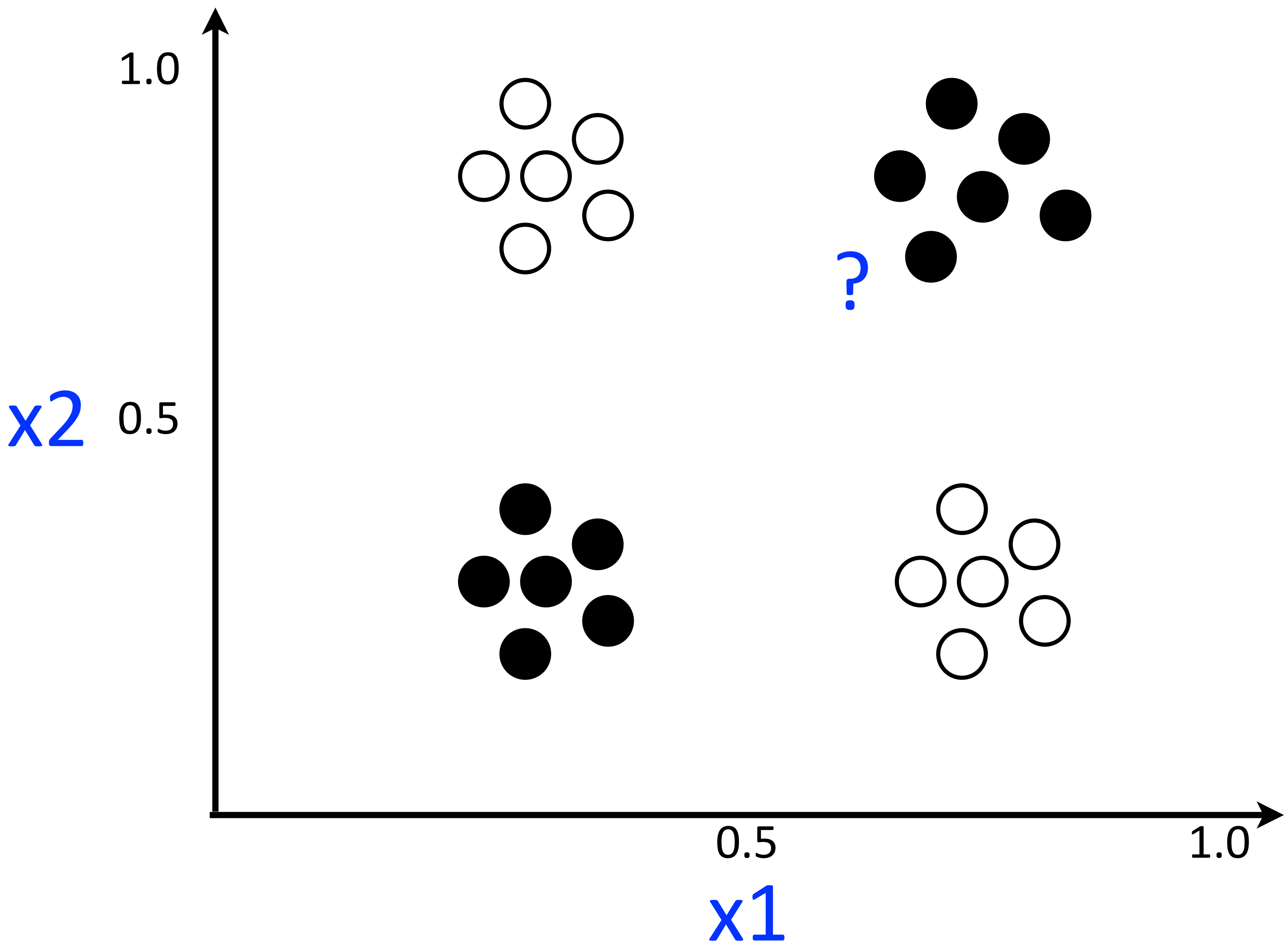
- Linear classifiers
- Decision tree classifiers
- Instance-based classifiers

Predictive Analysis
instance-based classifiers



- predict the class associated with the most similar training examples

Predictive Analysis
instance-based classifiers



- predict the class associated with the most similar training examples

Predictive Analysis
instance-based classifiers

- Assumption: instances with similar feature values should have a similar label
- Given a test instance, predict the label associated with its nearest neighbors
- There are many different similarity metrics for computing distance between training/test instances
- There are many ways of combining labels from multiple training instances

Predictive Analysis
questions

- Is a particular concept appropriate for predictive analysis?
- What should the unit of analysis be?
- How should I divide the data into training and test sets?
- What is a good feature representation for this task?
- What type of learning algorithm should I use?
- How should I evaluate my model's performance?