Ensemble Learning and Unsupervised Learning

Spring 2018

Review

- Last week:
  - Real-world community challenges
  - Feature Representation (categorical data, missing data, scaling data)
  - Dimensionality Reduction
  - Classification Evaluation (PR curves, ROC curves)

- Assignments (Canvas):
  - Problem set due yesterday
  - New lab assignment out

- Questions?

Today’s Topics

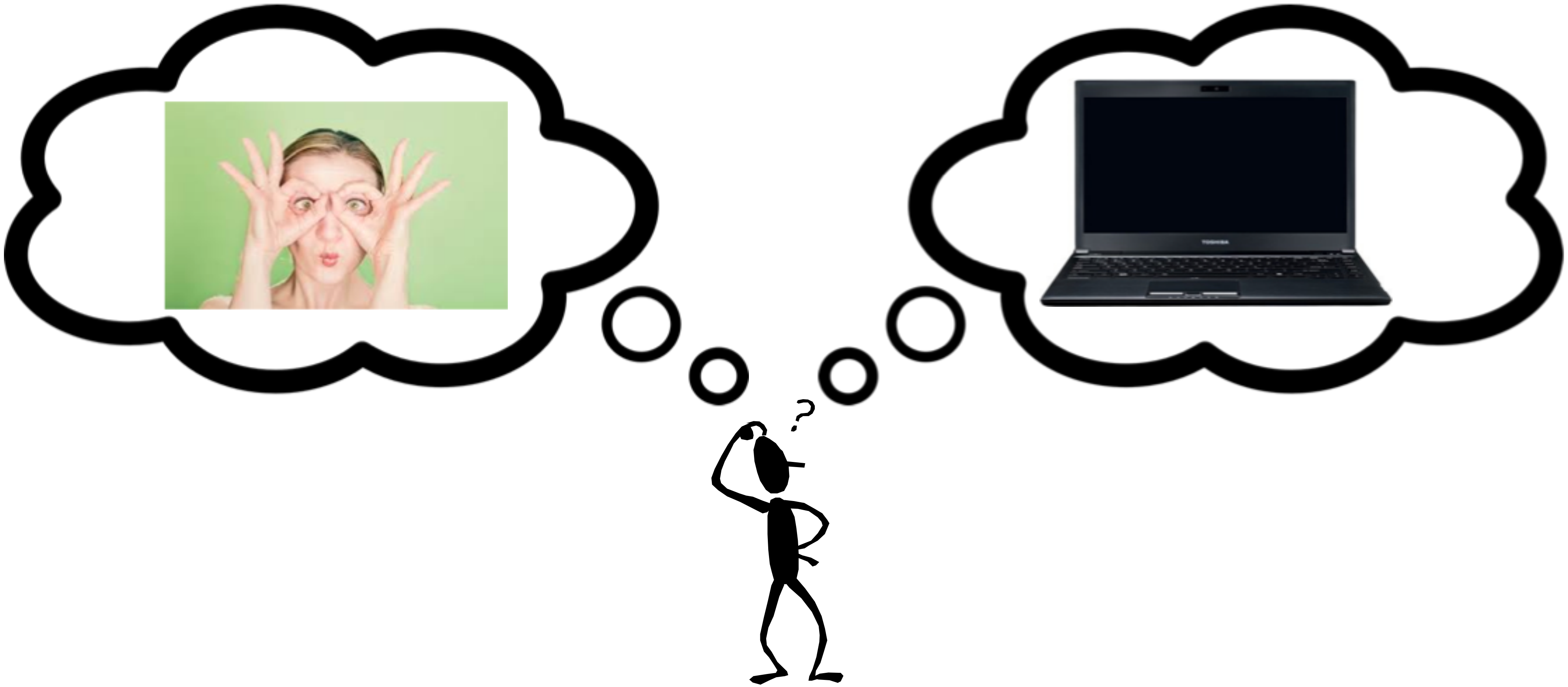
- Computer Vision
- Ensemble Learning
- Unsupervised Learning
- Lab

Today’s Topics

- Computer Vision
- Ensemble Learning
- Unsupervised Learning
- Lab

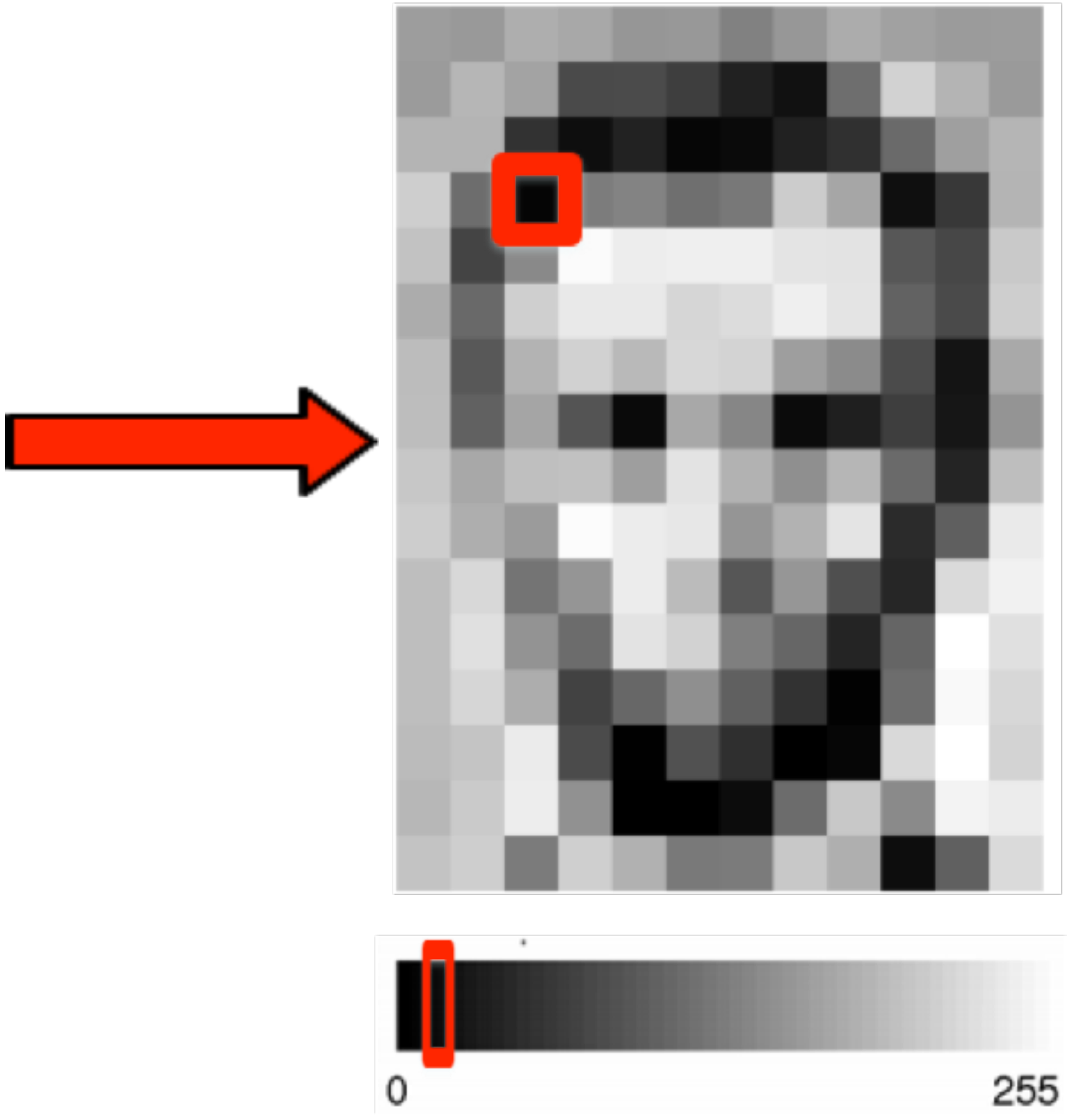
Computer Vision Goal (Turing Test)

Design computer vision that is indistinguishable from human vision



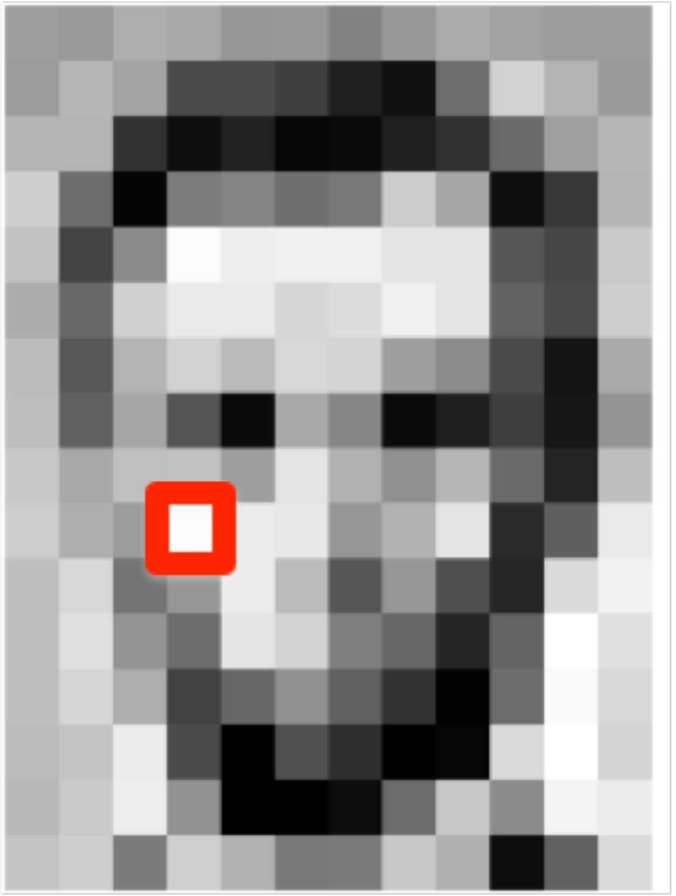
What a Computer Sees: Image

157	153	174	168	150	152	129	151	172	161	155	156
155	182	163	74	75	62	33	17	110	210	180	154
180	180	50	14	34	6	10	33	48	106	159	181
206	[illegible]	5	[illegible]	131	111	120	204	166	15	56	180
194	68	[illegible]	251	237	239	239	228	227	87	71	201
172	105	207	233	233	214	220	239	228	98	74	206
188	88	179	209	185	215	211	158	139	75	20	169
189	97	165	84	10	168	134	11	31	62	22	148
199	168	191	193	158	227	178	143	182	106	36	190
205	174	155	252	236	231	149	178	228	43	95	234
190	216	116	149	236	187	86	150	79	38	218	241
190	224	147	108	227	210	127	102	36	101	255	224
190	214	173	66	103	143	95	50	2	109	249	215
187	196	235	75	1	81	47	0	6	217	255	211
183	202	237	145	0	0	12	108	200	138	243	236
195	206	123	207	177	121	123	200	175	13	96	218



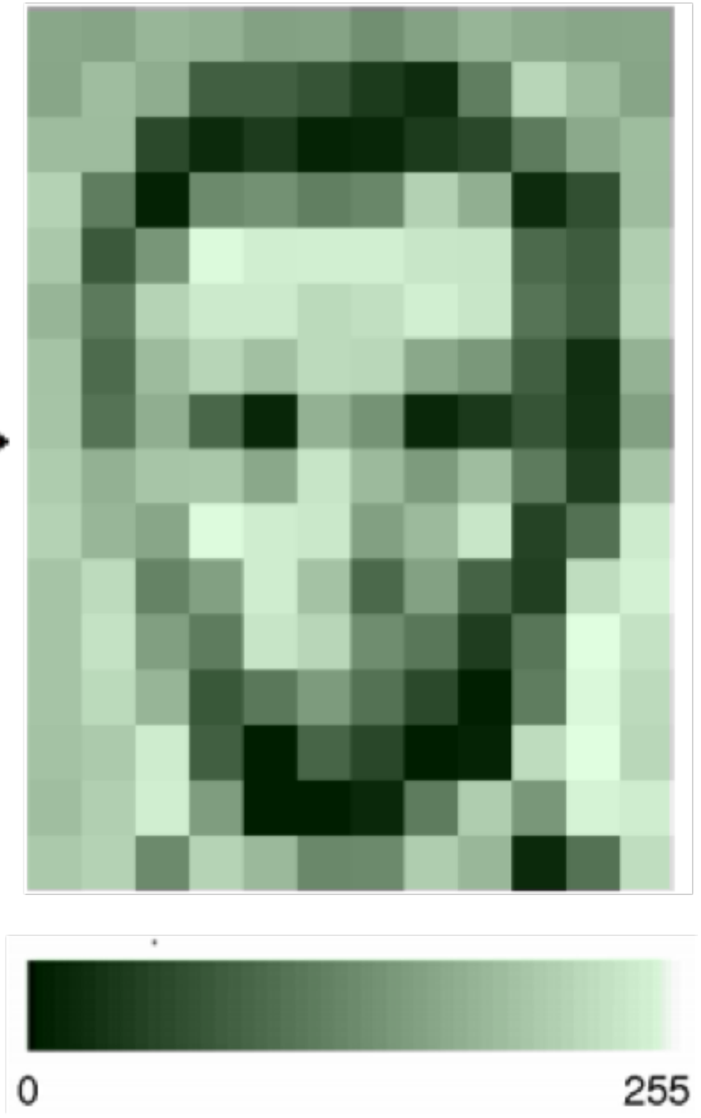
What a Computer Sees: Image

157	153	174	168	150	152	129	151	172	161	155	156
155	182	163	74	75	62	33	17	110	210	180	154
180	180	50	14	34	6	10	33	48	106	159	181
206	109	5	124	131	111	120	204	166	15	56	180
194	68	137	251	237	239	239	228	227	87	71	201
172	105	207	233	233	214	220	239	228	98	74	206
188	88	179	209	185	215	211	158	139	75	20	169
189	97	165	84	10	168	134	11	31	62	22	148
199	168	191	193	158	227	178	143	182	106	36	190
205	174	[illegible]	252	[illegible]	231	149	178	228	43	95	234
190	216	[illegible]	[illegible]	[illegible]	187	86	150	79	38	218	241
190	224	147	108	227	210	127	102	36	101	255	224
190	214	173	66	103	143	96	50	2	109	249	215
187	196	235	75	1	81	47	0	6	217	255	211
183	202	237	145	0	0	12	108	200	138	243	236
195	206	123	207	177	121	123	200	175	13	96	218



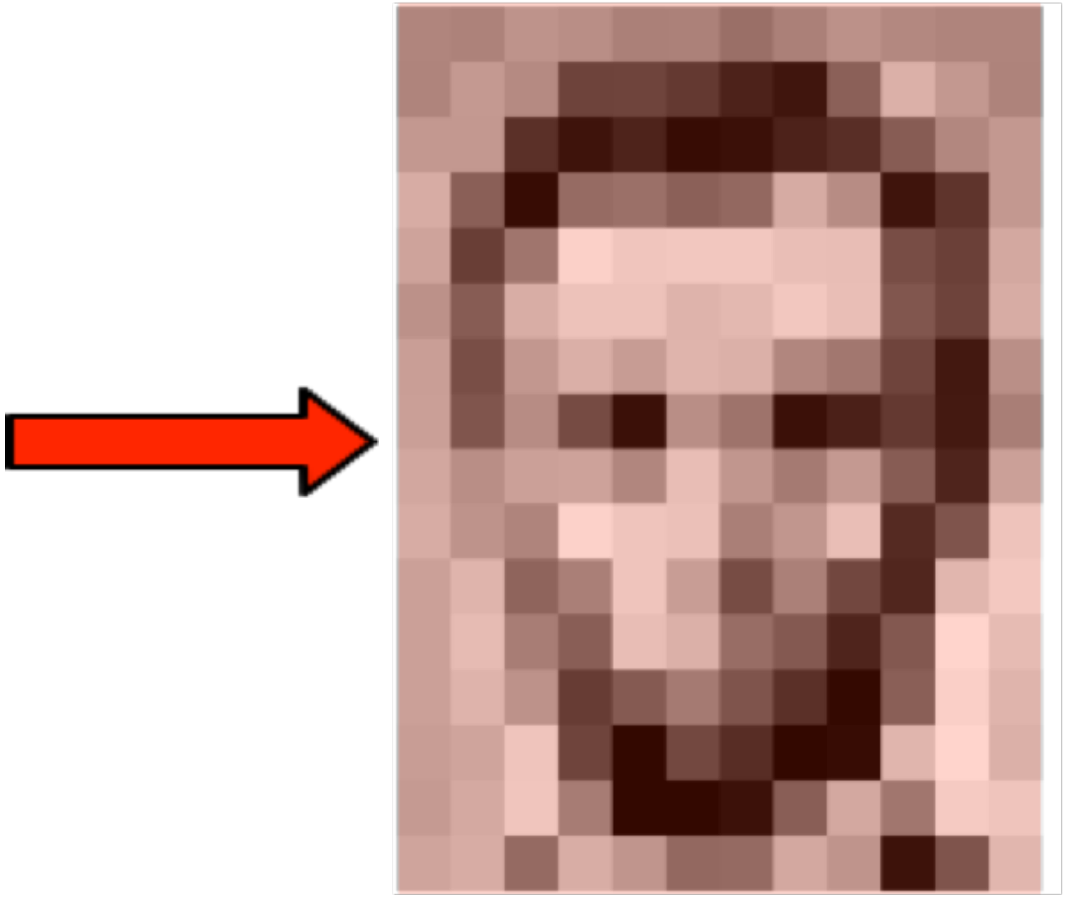
What a Computer Sees: Visualizing an Image

157	153	174	168	150	152	129	151	172	161	155	156
155	182	163	74	75	62	33	17	110	210	180	154
180	180	50	14	34	6	10	33	48	106	159	181
206	109	5	124	131	111	120	204	166	15	56	180
194	68	137	251	237	239	239	228	227	87	71	201
172	105	207	233	233	214	220	239	228	98	74	206
188	88	179	209	185	215	211	158	139	75	20	169
189	97	165	84	10	168	134	11	31	62	22	148
199	168	191	193	158	227	178	143	182	106	36	190
205	174	155	252	236	231	149	178	228	43	95	234
190	216	116	149	236	187	86	150	79	38	218	241
190	224	147	108	227	210	127	102	36	101	255	224
190	214	173	66	103	143	96	50	2	109	249	215
187	196	235	75	1	81	47	0	6	217	255	211
183	202	237	145	0	0	12	108	200	138	243	236
195	206	123	207	177	121	123	200	175	13	96	218



What a Computer Sees: Visualizing an Image

157	153	174	168	150	152	129	151	172	161	155	156
155	182	163	74	75	62	33	17	110	210	180	154
180	180	50	14	34	6	10	33	48	106	159	181
206	109	5	124	131	111	120	204	166	15	56	180
194	68	137	251	237	239	239	228	227	87	71	201
172	105	207	233	233	214	220	239	228	98	74	206
188	88	179	209	185	215	211	158	139	75	20	169
189	97	165	84	10	168	134	11	31	62	22	148
199	168	191	193	158	227	178	143	182	106	36	190
205	174	155	252	236	231	149	178	228	43	95	234
190	216	116	149	236	187	86	150	79	38	218	241
190	224	147	108	227	210	127	102	36	101	255	224
190	214	173	66	103	143	96	50	2	109	249	215
187	196	235	75	1	81	47	0	6	217	255	211
183	202	237	145	0	0	12	108	200	138	243	236
195	206	123	207	177	121	123	200	175	13	96	218



Understanding Images (Computer Vision)

157	153	174	168	150	152	129	151	172	161	155	156
155	182	163	74	75	62	33	17	110	210	180	154
180	180	50	14	34	6	10	33	48	106	159	181
206	109	5	124	131	111	120	204	166	15	56	180
194	68	137	251	237	239	239	228	227	87	71	201
172	105	207	233	233	214	220	239	228	98	74	206
188	88	179	209	185	215	211	158	139	75	20	169
189	97	165	84	10	168	134	11	31	62	22	148
199	168	191	193	158	227	178	143	182	106	36	190
205	174	155	252	236	231	149	178	228	43	95	234
190	216	116	149	236	187	86	150	79	38	218	241
190	224	147	108	227	210	127	102	36	101	255	224
190	214	173	66	103	143	95	50	2	109	249	215
187	196	235	75	1	81	47	0	6	217	255	211
183	202	237	145	0	0	12	108	200	138	243	236
195	206	123	207	177	121	123	200	175	13	96	218

What is this?
- A picture of a person

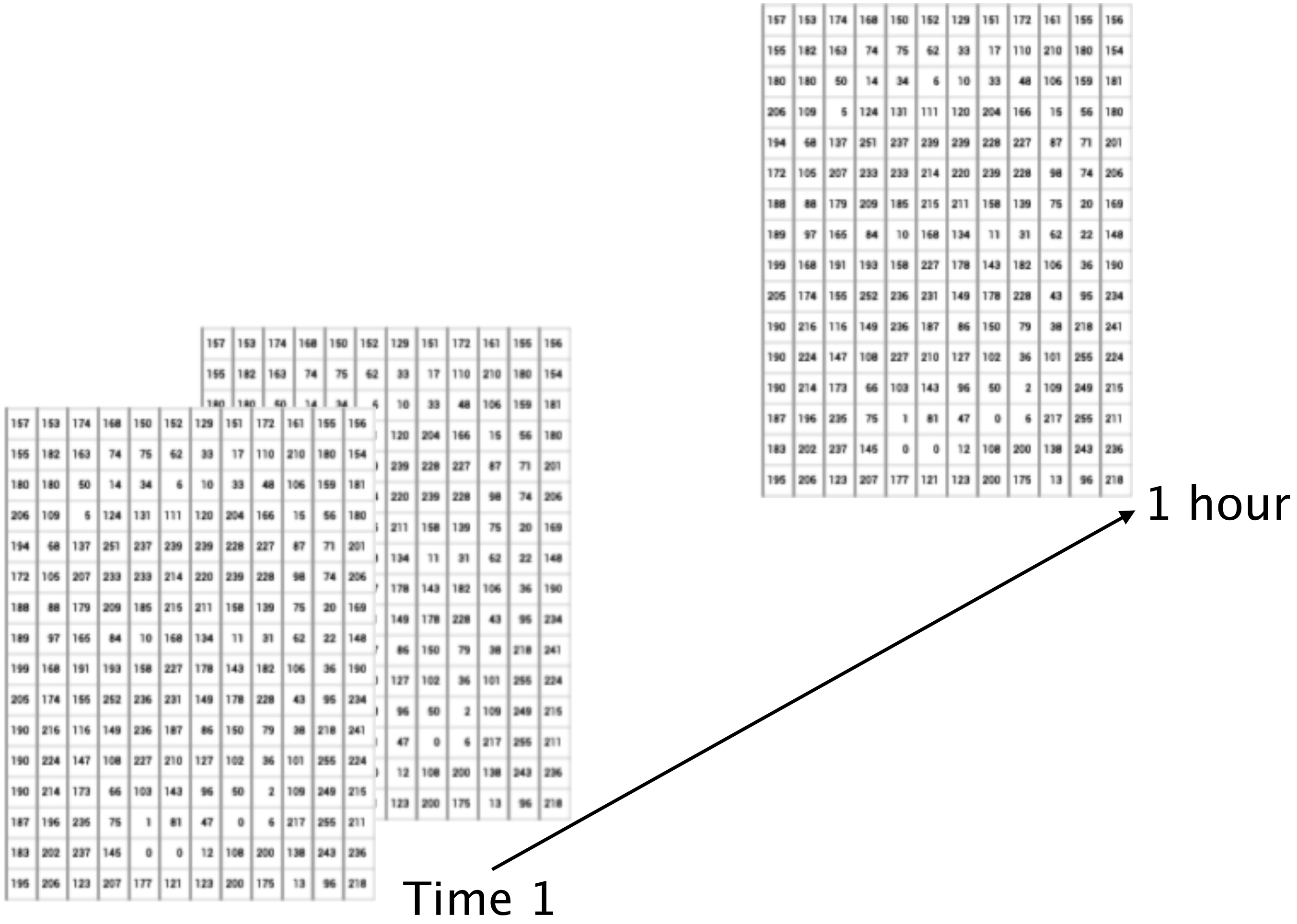
Could you describe this person?
- Long face
- Angular jaw
- Has a beard

Who is this person?
- Abraham Lincoln

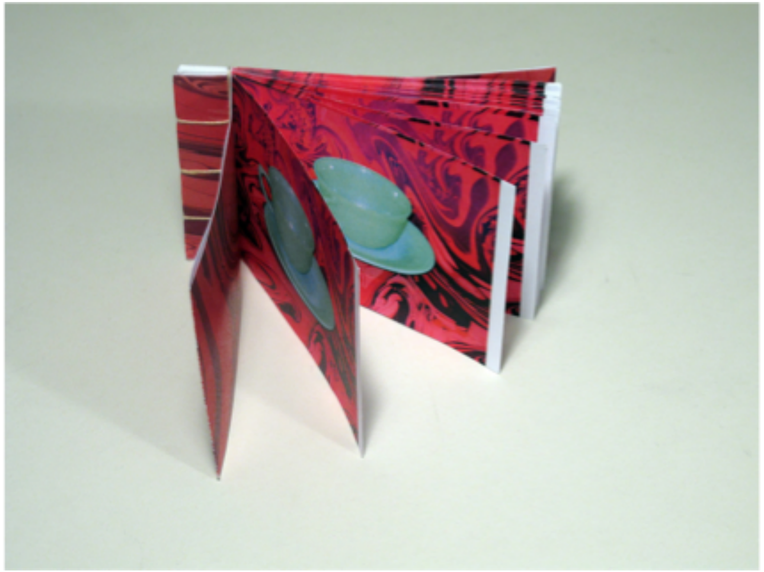
Is this person happy?
- I am not sure.

Is this person handsome?
- ~70% of people would say “yes”

What a Computer Sees: Video

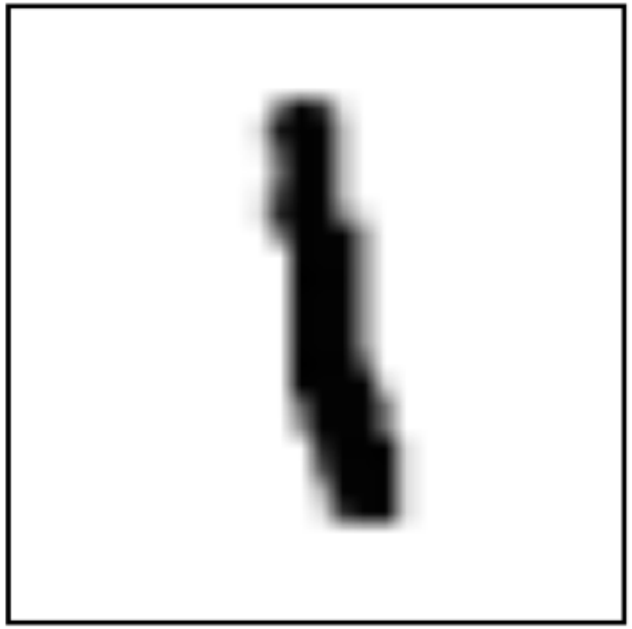


Analogous to:



How to Describe an Image to a Computer?

- Raw pixel values
  - e.g. MNIST: how many “features” would be in an image (28 x 28 pixels)



$$\approx \begin{bmatrix}
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & .6 & .8 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & .7 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & .7 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & .5 & 1 & .4 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & .4 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & .4 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & .7 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & .9 & 1 & .1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & .3 & 1 & .1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0
\end{bmatrix}$$

  - 784

http://colah.github.io/posts/2014-10-Visualizing-MNIST/

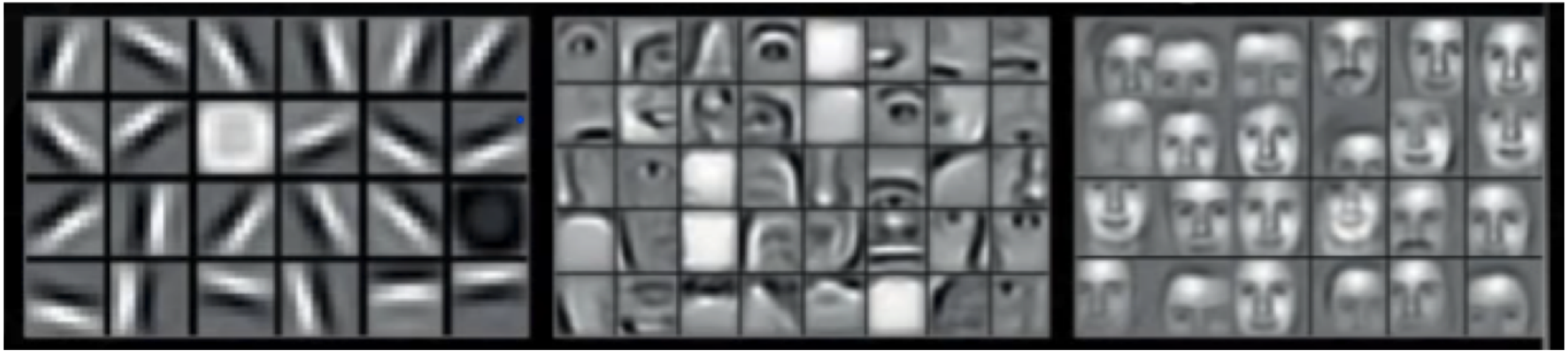
How to Describe an Image to a Computer?

- Raw pixel values
  - e.g. LFW: how many “features” would be in an image (50 x 37 pixels)

    - 1,850

http://vis-www.cs.umass.edu/lfw/

How to Describe an Image to a Computer? Low-Level to High-Level Representations



<u>Low-level features</u>
e.g., dots, edges, corners, lines, curves

<u>Mid-level features</u>
e.g., forms, colors

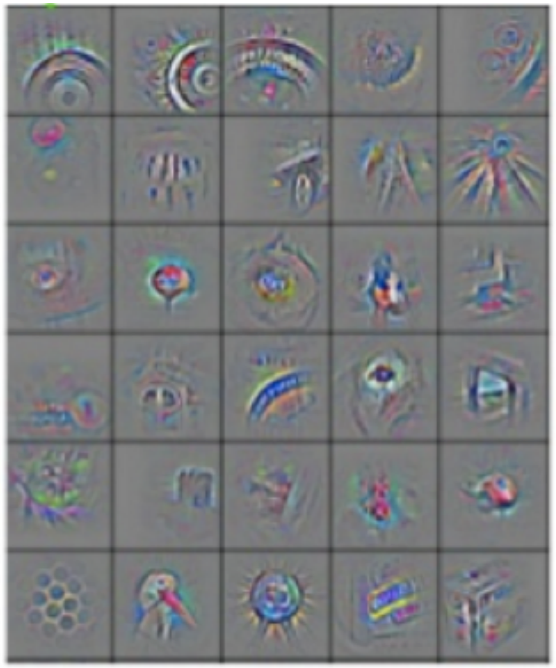
<u>High-level features</u>
e.g., objects, scenes, emotions

Adapted from https://dzone.com/articles/deep-learning-vs-machine-learning-the-hottest-topi

How to Describe an Image to a Computer? Low-Level to High-Level Representations

Low-level features

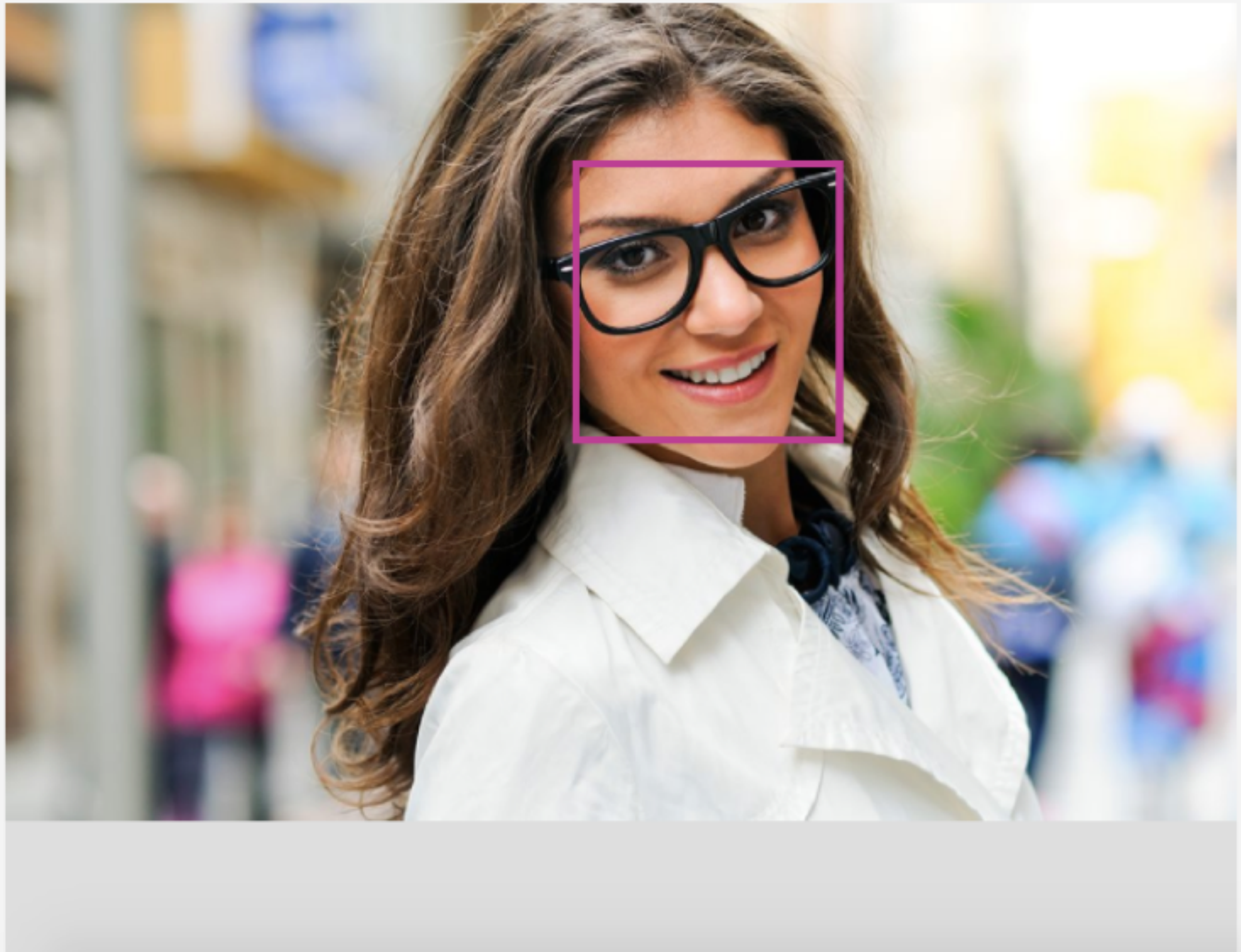
Mid-level features



High-level features

Adapted from https://www.slideshare.net/embeddedvision/01-am-keynotelecun

How to Describe an Image to a Computer? Microsoft Azure: Face API



Detection result:

JSON:


```
[
  {
    "faceId": "83c0f042-8c96-4b00-97dd-66bdeba3b6bc",
    "faceRectangle": {
      "top": 128,
      "left": 459,
      "width": 224,
      "height": 224
    },
    "faceAttributes": {
      "hair": {
        "bald": 0.0,
        "invisible": false,
        "hairColor": [
          {
            "color": "brown",
            "confidence": 1.0
          },
          {
            "color": "blond",
            "confidence": 0.69
```

How to Describe an Image to a Computer? Microsoft Azure: Computer Vision API

FEATURE NAME:	VALUE
Description	{ "tags": ["train", "platform", "station", "building", "indoor", "subway", "track", "walking", "waiting", "pulling", "board", "people", "man", "luggage", "standing", "holding", "large", "woman", "yellow", "suitcase"], "captions": [{ "text": "people waiting at a train station", "confidence": 0.833099365 }] }
Tags	[{ "name": "train", "confidence": 0.9975446 }, { "name": "platform", "confidence": 0.995543063 }, { "name": "station", "confidence": 0.9798007 }, { "name": "indoor", "confidence": 0.927719653 }, { "name": "subway", "confidence": 0.838939846 }, { "name": "pulling", "confidence": 0.431715637 }]
Image format	"Jpeg"

Today’s Topics

- Computer Vision
- Ensemble Learning
- Unsupervised Learning
- Lab

Idea: How Many Predictors to Use?

More than 1: Ensemble

Why Choose Ensemble Instead of an Algorithm?

- Reduces probability for making a wrong prediction, assuming:
  - Classifiers are independent (not true in practice!)
- Suppose:
  - n classifiers for binary classification task
  - Each classifier has same error rate ε
  - Probability mass function indicates the probability of error from an ensemble:

$$P(y \geq k) = \sum_{k}^{n} \left\langle \begin{matrix} n \\ k \end{matrix} \right\rangle \varepsilon^{k} (1-\varepsilon)^{n-k} = \varepsilon_{ensemble}$$

  - e.g., n = 11, ε = 0.25; k = 6: probability of error is ~0.034 which is much lower than probability of error from a single algorithm (0.25)

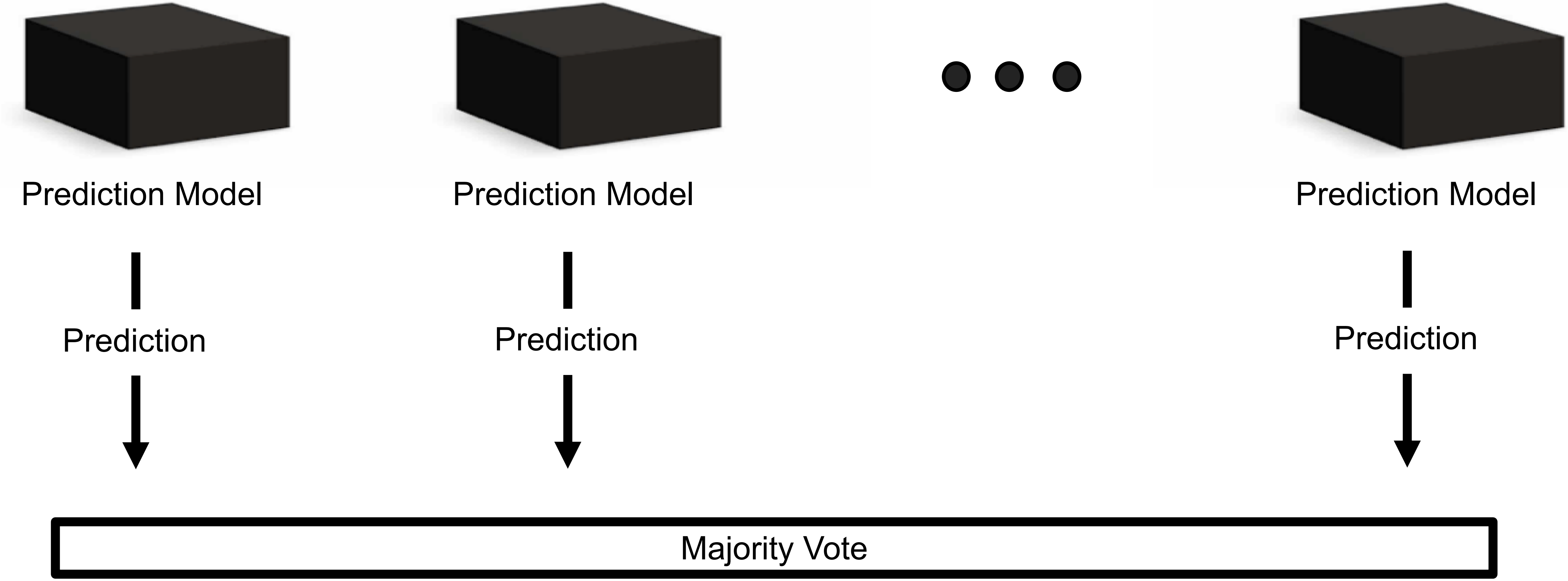
How to Predict with an Ensemble?

- Majority Voting
  - Return most popular prediction from multiple prediction algorithms
- Bootstrap Aggregation, aka Bagging
  - Resample data to train algorithm on different random subsets
- Boosting
  - Reweight data to train algorithms to specialize on different “hard” examples
- Stacking
  - Train a model that learns how to aggregate classifiers’ predictions

How to Predict with an Ensemble of Algorithms?

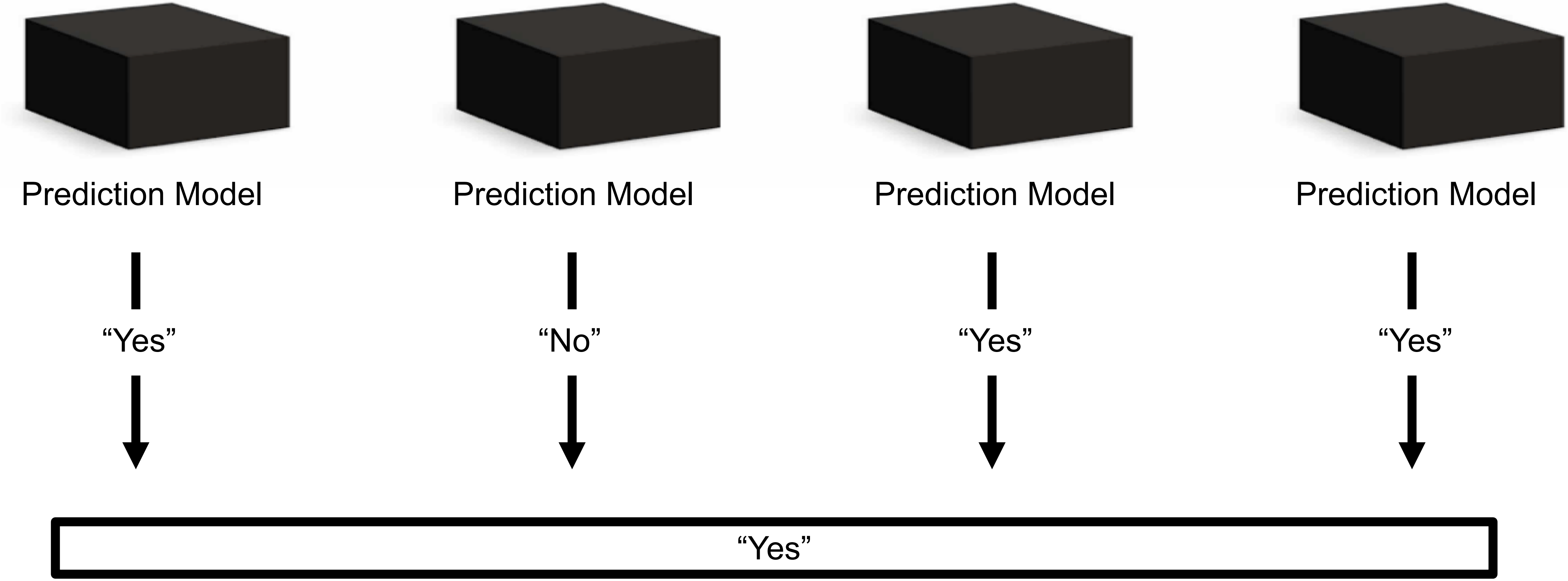
- Majority Voting
  - Return most popular prediction from multiple prediction algorithms
- Bootstrap Aggregation, aka Bagging
  - Train algorithm repeatedly on different random subsets of the training set
- Boosting
  - Train algorithms that each specialize on different “hard” training examples
- Stacking
  - Train a model that learns how to aggregate classifiers’ predictions

Majority Voting



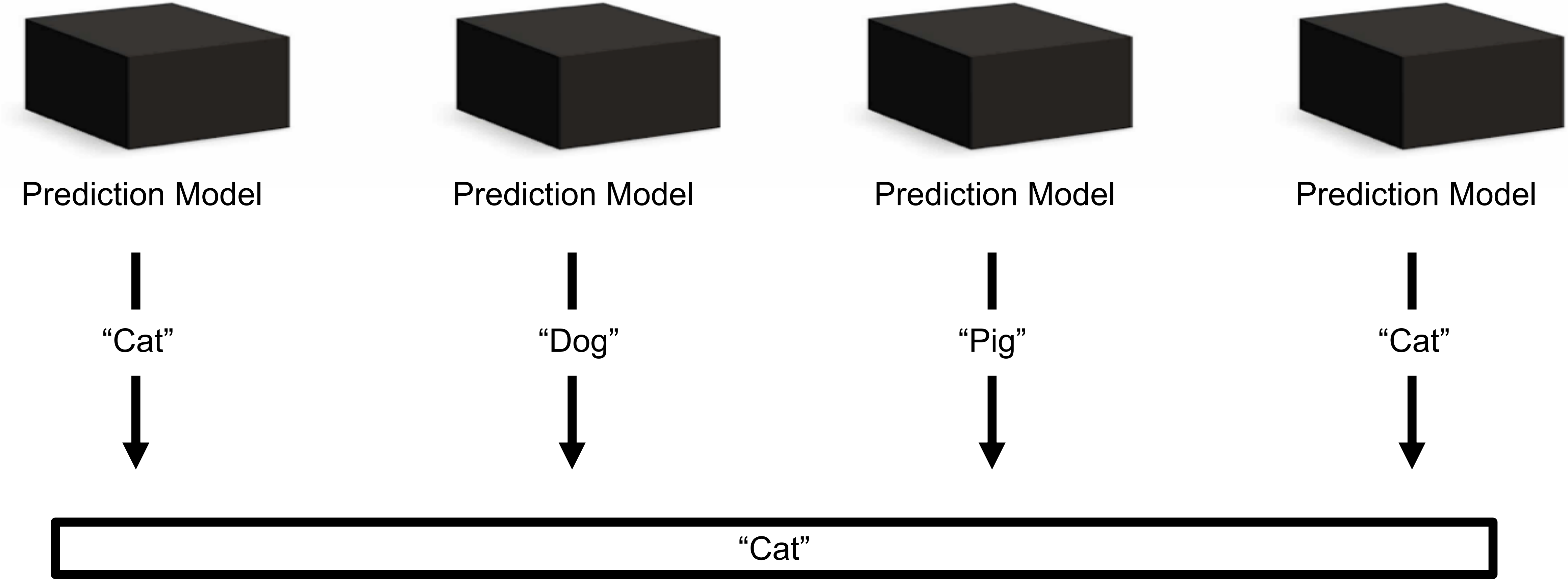
Majority Voting: Binary Task

e.g., “Is it sunny today?”

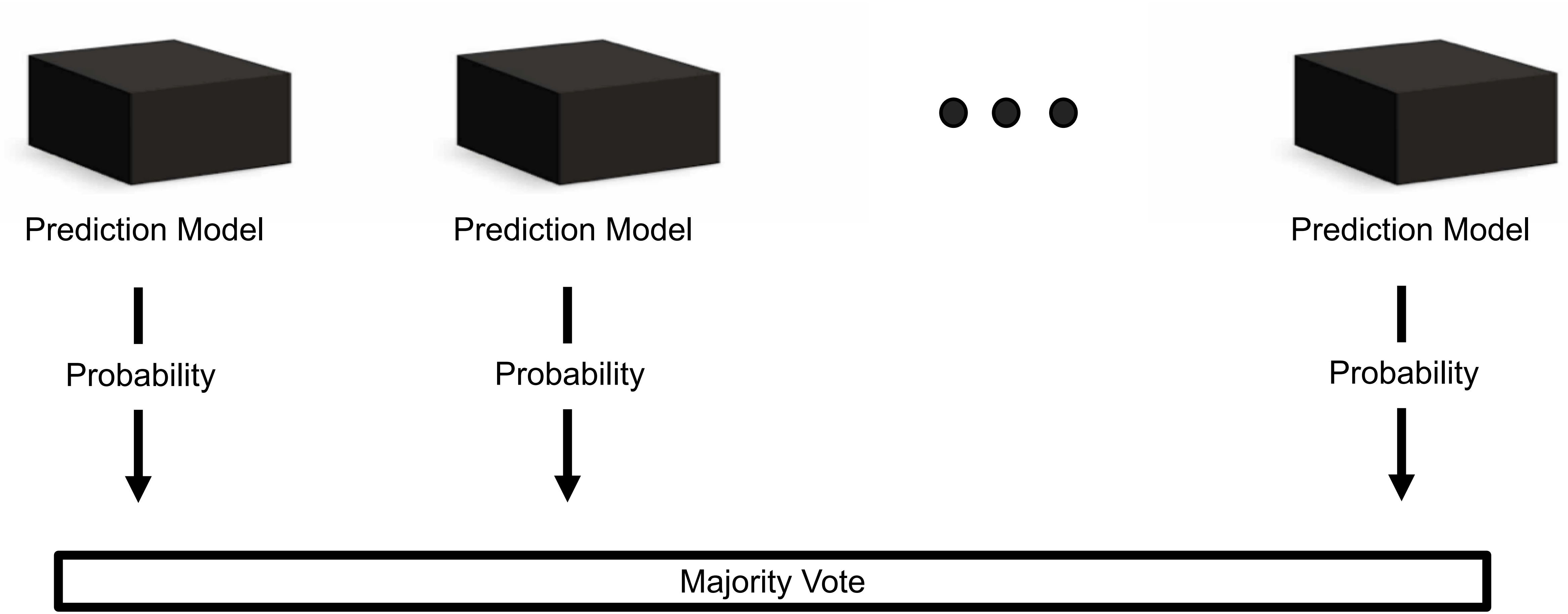


Majority Voting: Non-Binary Task

e.g., “What object is in the image?”



Majority Voting: “Soft” (not “Hard”)



Majority Voting: Soft Voting on Binary Task

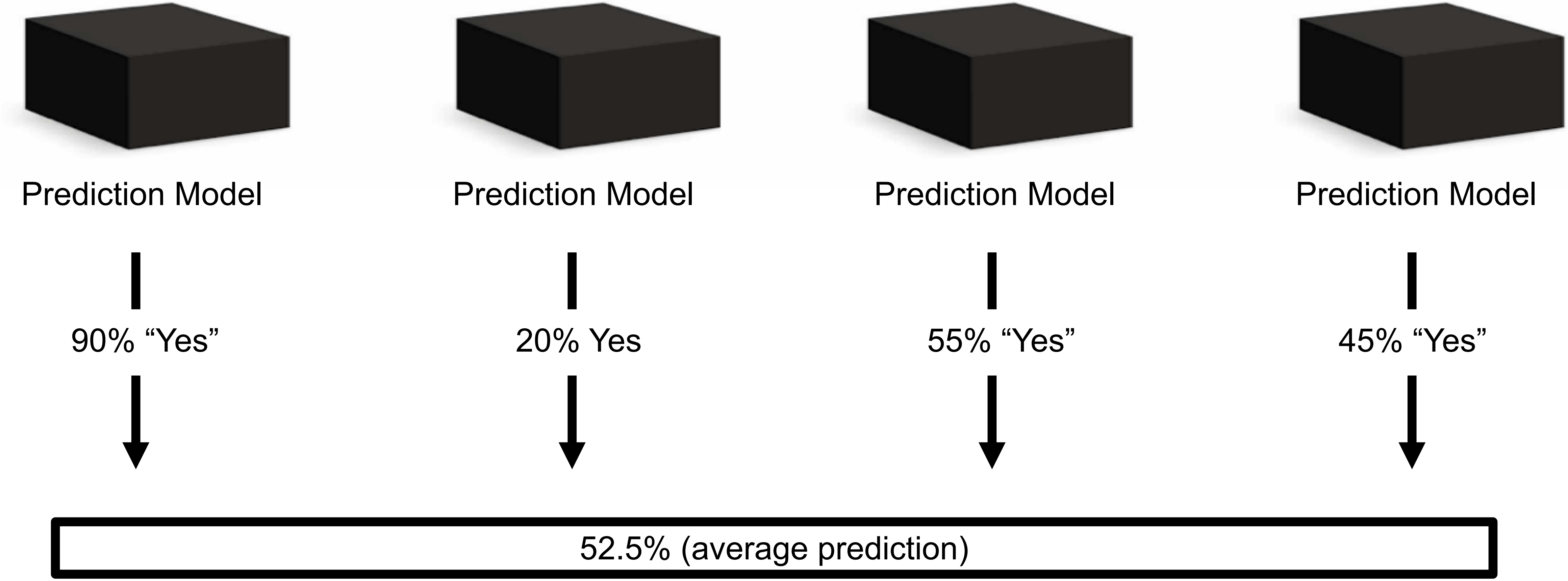
e.g., “Is it sunny today?”

Prediction Model	Prediction Model	Prediction Model	Prediction Model
90% “Yes”	20% Yes	55% “Yes”	45% “Yes”

“Yes” (210/4 = 52.5% Yes)

Majority Voting: Regression

e.g., “Is it sunny today?”



How to Predict with an Ensemble of Algorithms?

- Majority Voting
  - Return most popular prediction from multiple prediction algorithms
- Bootstrap Aggregation, aka Bagging
  - Train algorithm repeatedly on different random subsets of the training set
- Boosting
  - Train algorithms that each specialize on different “hard” training examples
- Stacking
  - Train a model that learns how to aggregate classifiers’ predictions

Bagging: Training

- Build ensemble from "bootstrap samples" drawn with replacement
- e.g.,

Duplicates can occur

Some examples missing from training data; e.g., round 1

Sample indices	Bagging round 1	Bagging round 2	...
1	2	7	...
2	2	3	...
3	1	2	...
4	3	1	...
5	7	1	...
6	2	7	...
7	4	7	...

C_1 C_2 C_m

Each classifier trained on different subset of data

"Random forest"

- Fit decision trees by also selecting random feature subsets

Breiman, Bagging Predictors, 1994.

Ho, Random Decision Forests, 1995.

Raschka & Mirjalili, Python Machine Learning.

Bagging: Predicting

Prediction Model

Prediction Model

Prediction Model

Prediction Model

- Predict as done for “majority voting”
  - e.g., “hard” voting
  - e.g., “soft” voting
  - e.g., averaging values for regression

How to Predict with an Ensemble of Algorithms?

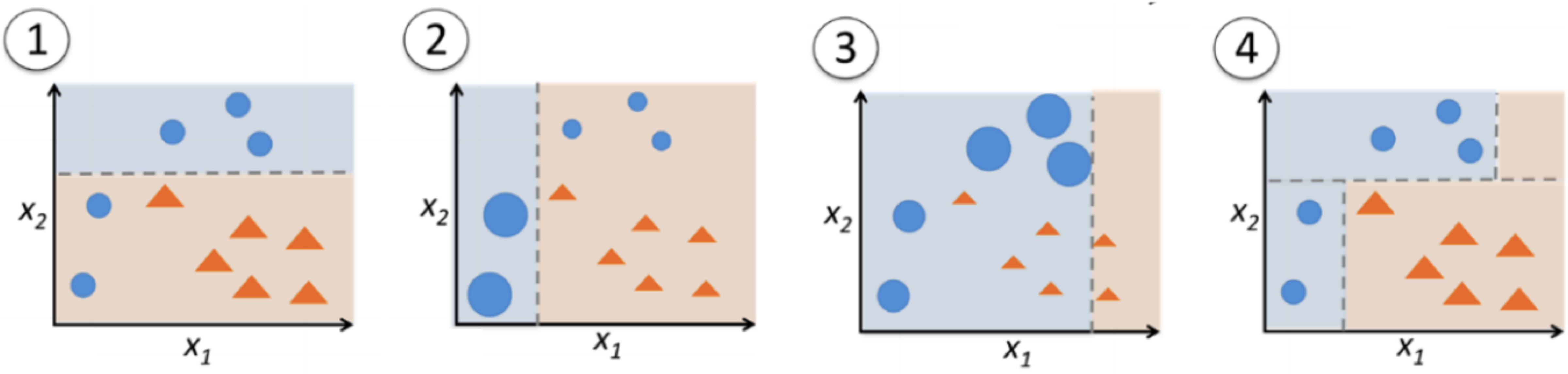
- Majority Voting
  - Return most popular prediction from multiple prediction algorithms
- Bootstrap Aggregation, aka Bagging
  - Train algorithm repeatedly on different random subsets of the training set
- Boosting
  - Train algorithms that each specialize on different “hard” training examples
- Stacking
  - Train a model that learns how to aggregate classifiers’ predictions

Boosting

- Key idea: sequentially train predictors that each try to correctly predict examples that were hard for previous predictors

- Original Algorithm:
  - Train classifier 1: use random subset of examples without replacement
  - Train classifier 2: use a second random subset of examples without replacement and add 50% of examples misclassified by classifier 1
  - Train classifier 3: use examples that classifiers 1 and 2 disagree on
  - Predict using majority vote from 3 classifiers

Boosting – Adaboost (**Ada**ptive **Boost**ing)



Assign equal weights to all examples

- Assign larger weights to previous misclassifications
- Assign smaller weights to previous correct classifications

- Assign larger weights to previous misclassifications
- Assign smaller weights to previous correct classifications

Predict with weighted majority vote

Freund and Schapire, Experiments with a New Boosting Algorithm, 1996.

Raschka and Mirjalili; Python Machine Learning

Boosting – Adaboost (**Ada**ptive **Boost**ing)

e.g., 1d dataset

Sample indices	x	y	Weights	$\hat{y}$(x <= 3.0)?	Correct?	Updated weights
1	1.0	1	0.1	1	Yes	0.072
2	2.0	1	0.1	1	Yes	0.072
3	3.0	1	0.1	1	Yes	0.072
4	4.0	-1	0.1	-1	Yes	0.072
5	5.0	-1	0.1	-1	Yes	0.072
6	6.0	-1	0.1	-1	Yes	0.072
7	7.0	1	0.1	-1	No	0.167
8	8.0	1	0.1	-1	No	0.167
9	9.0	1	0.1	-1	No	0.167
10	10.0	-1	0.1	-1	Yes	0.072

Round 2: update weights

Round 1: training data, weights, predictions

Raschka and Mirjalili; Python Machine Learning

Boosting – Adaboost (**Ada**ptive **Boost**ing)

e.g., 1d dataset

1. Compute error rate (sum misclassified examples' weights):

$$\varepsilon = 0.1\times 0+0.1\times 0+0.1\times 0+0.1\times 0+0.1\times 0+0.1\times 0+0.1\times 1+0.1\times 1 +0.1\times 1+0.1\times 0 = \frac{3}{10} = 0.3$$

2. Compute coefficient used to update weights and make majority vote prediction:

$$\alpha_j = 0.5\log\left(\frac{1-\varepsilon}{\varepsilon}\right) \approx 0.424$$

3. Update weight vector:

$$\boldsymbol{w} := \boldsymbol{w}\times\exp\left(-\alpha_j \times \hat{\boldsymbol{y}} \times \boldsymbol{y}\right)$$

- Correct predictions will decrease weight and vice versa

$$0.1\times\exp(-0.424\times 1\times 1)\approx 0.065 \quad 0.1\times\exp(-0.424\times(-1)\times(1))\approx 0.153$$

4. Normalize weights to sum to 1:

$$\boldsymbol{w} := \frac{\boldsymbol{w}}{\sum_i w_i}$$

$$\sum_i w_i = 7\times 0.065+3\times 0.153 = 0.914$$

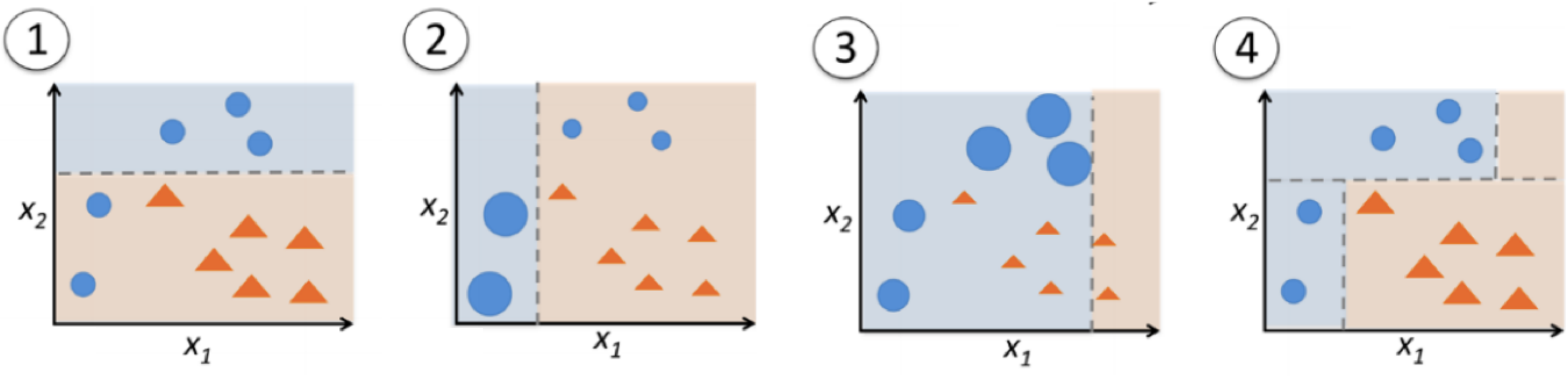
Correct?	Updated weights
Yes	0.072
Yes	0.072
Yes	0.072
Yes	0.072
Yes	0.072
Yes	0.072
No	0.167
No	0.167
No	0.167
Yes	0.072

$0.065 / 0.914$

$0.153 / 0.914$

Raschka and Mirjalili; Python Machine Learning

Boosting – Adaboost (**Ada**ptive **Boost**ing)



To predict, use α calculated for each classifier as its weight when voting with all trained classifiers.

Idea: value the prediction of each classifier based on the accuracies they had on the training dataset.

Raschka and Mirjalili; Python Machine Learning

How to Predict with an Ensemble of Algorithms?

- Majority Voting
  - Return most popular prediction from multiple prediction algorithms
- Bootstrap Aggregation, aka Bagging
  - Train algorithm repeatedly on different random subsets of the training set
- Boosting
  - Train algorithms that each specialize on different “hard” training examples
- Stacking
  - Train a model that learns how to aggregate classifiers’ predictions

Stacked Generalization, aka Stacking

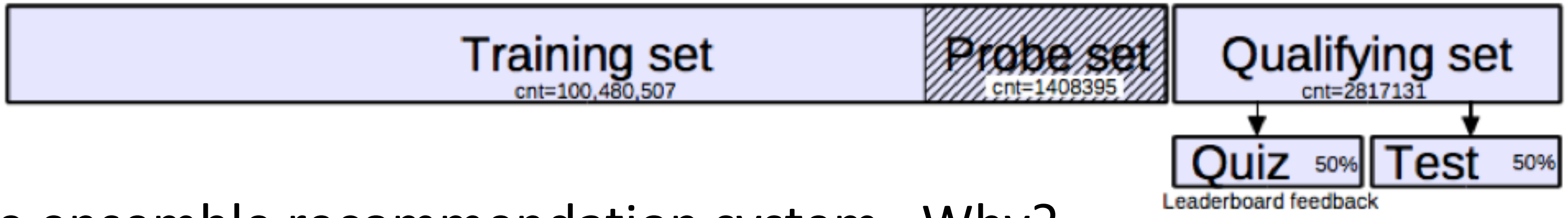
- Train meta-learner to learn the optimal weighting of each classifiers' predictions for making the final prediction
- Algorithm:
  1. Split dataset into three disjoint sets.
  2. Train several base learners on the first partition.
  3. Test the base learners on the second partition and third partition.
  4. Train meta-learner on second partition using classifiers' predictions as features
  5. Evaluate meta-learner on third prediction using classifiers' predictions as features

David, H. Wolpert, Stacked Generalization, 1992.

Tutorial: http://blog.kaggle.com/2017/06/15/stacking-made-easy-an-introduction-to-stacknet-by-competitions-grandmaster-marios-michailidis-kazanova/

Ensemble Learner Won Netflix Prize "Challenge"

- In 2009 challenge, winning team won $1 million using ensemble approach:
  - https://www.netflixprize.com/assets/GrandPrize2009_BPC_BigChaos.pdf
  - Dataset: 5-star ratings on 17770 movies from 480189 "anonymous" users collected by Netflix over ~7 years. In total, the number of ratings is 100,480,507.



- Netflix did not use ensemble recommendation system. Why?
  - "We evaluated some of the new methods offline but the additional accuracy gains that we measured did not seem to justify the engineering effort needed to bring them into a production environment" - https://medium.com/netflix-techblog/netflix-recommendations-beyond-the-5-stars-part-1-55838468f429
  - Computationally slow and complex from using "sequential" training of learners

Today’s Topics

- Computer Vision
- Ensemble Learning
- Unsupervised Learning
- Lab

Unsupervised Learning

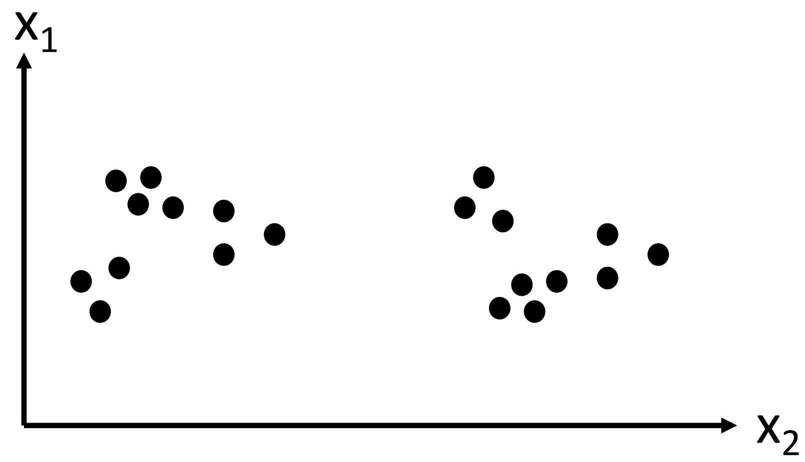
No known output.

e.g.,

1. Transformations (PCA)
2. $\boxed{\text{Clustering}}$

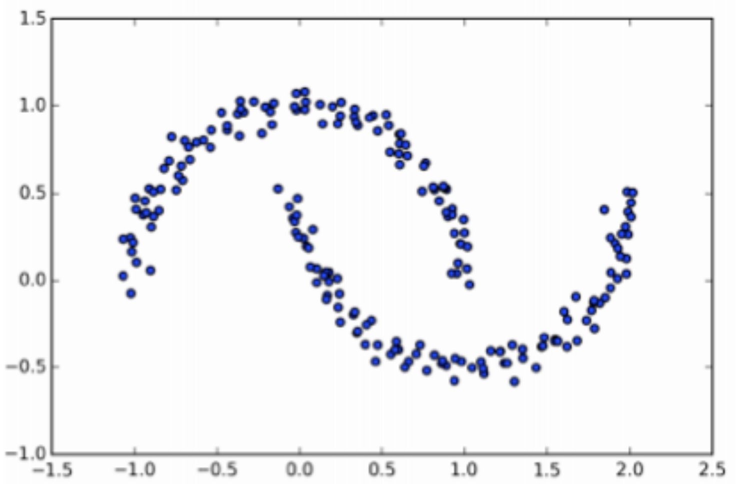
Clustering

A.



B.

C.



Find groupings such that entities in a group will be similar to each another and different from the entities in other groups.

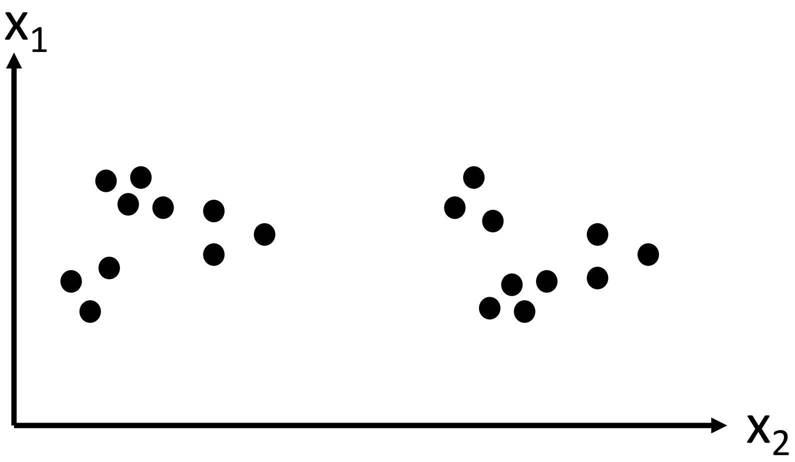
Material from Raschka and Mirjalili; Python Machine Learning

Clustering

What are applications use clustering?

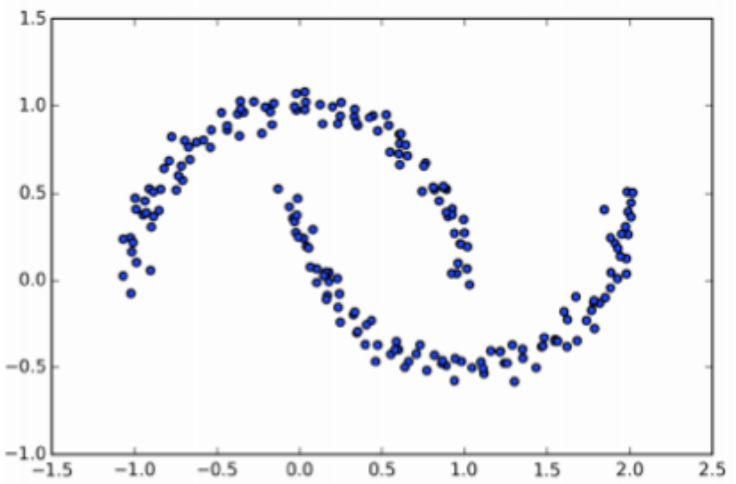
Class Discussion: Clustering

A.



B.

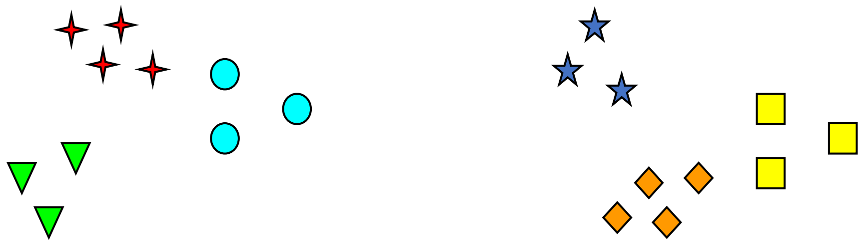
C.



- How many data clusters would you create?
- What “algorithm” would you use to partition the data?
- What “features” would you use to partition the data?

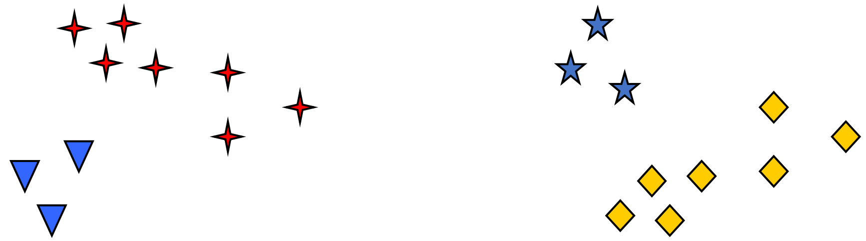
Figure C from Raschka and Mirjalili; Python Machine Learning

How Many Clusters?



Six Clusters

Two Clusters



Four Clusters

Number of clusters can be ambiguous.

Slide adapted from: https://www-users.cs.umn.edu/~kumar001/dmbook/slides/chap7_basic_cluster_analysis.pdf

Types of Clustering

- Partitional Clustering
  - A division of data objects into non-overlapping subsets (clusters) such that each data object is in exactly one subset
- Hierarchical clustering
  - A set of nested clusters organized as a hierarchical tree

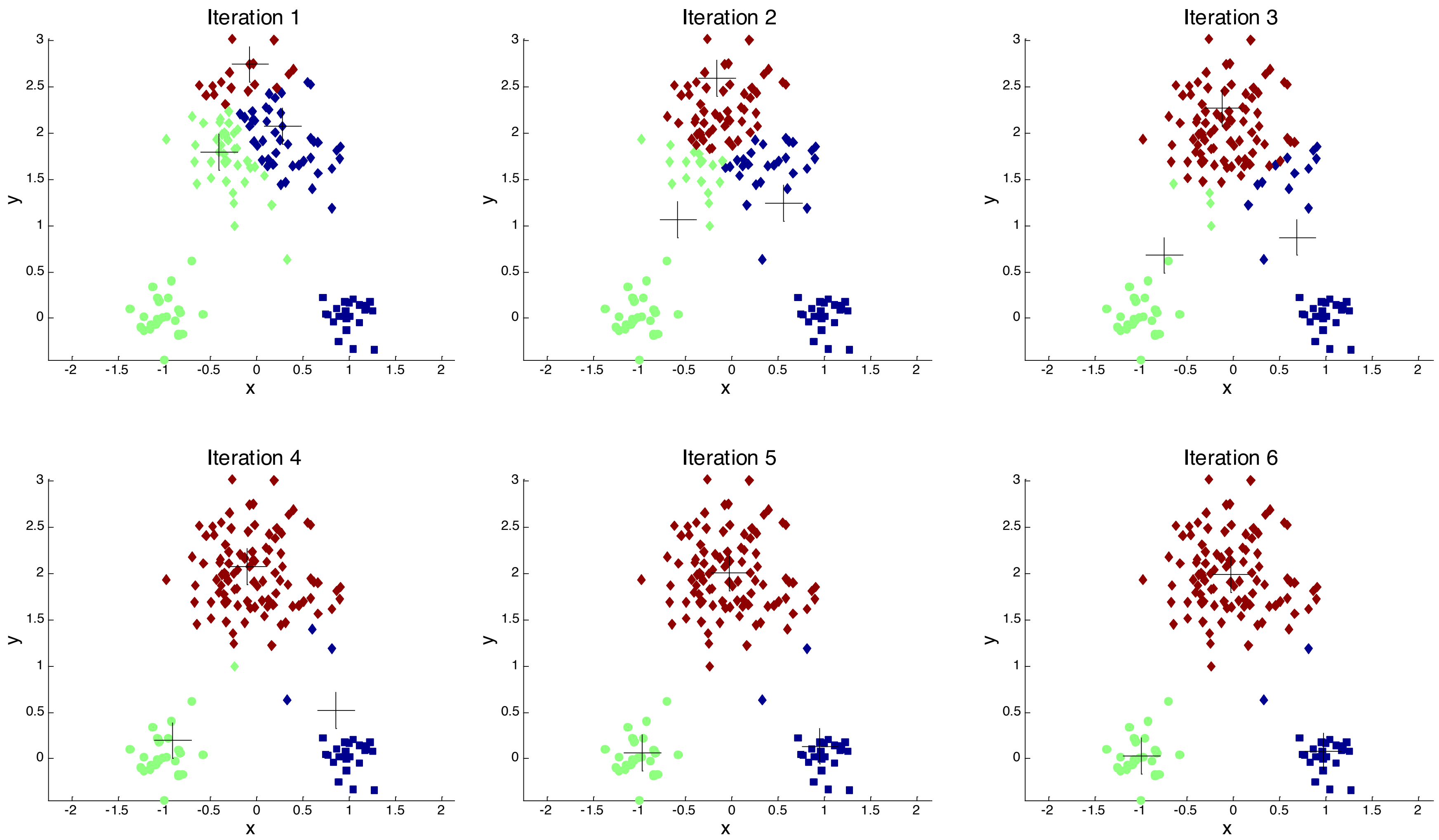
Slide adapted from: https://www-users.cs.umn.edu/~kumar001/dmbook/slides/chap7_basic_cluster_analysis.pdf

K-Means Clustering

Select K points as the initial centroids.
repeat
    Form K clusters by assigning all points to the closest centroid.
    Recompute the centroid of each cluster.
until The centroids don't change

Slide adapted from: https://www-users.cs.umn.edu/~kumar001/dmbook/slides/chap7_basic_cluster_analysis.pdf

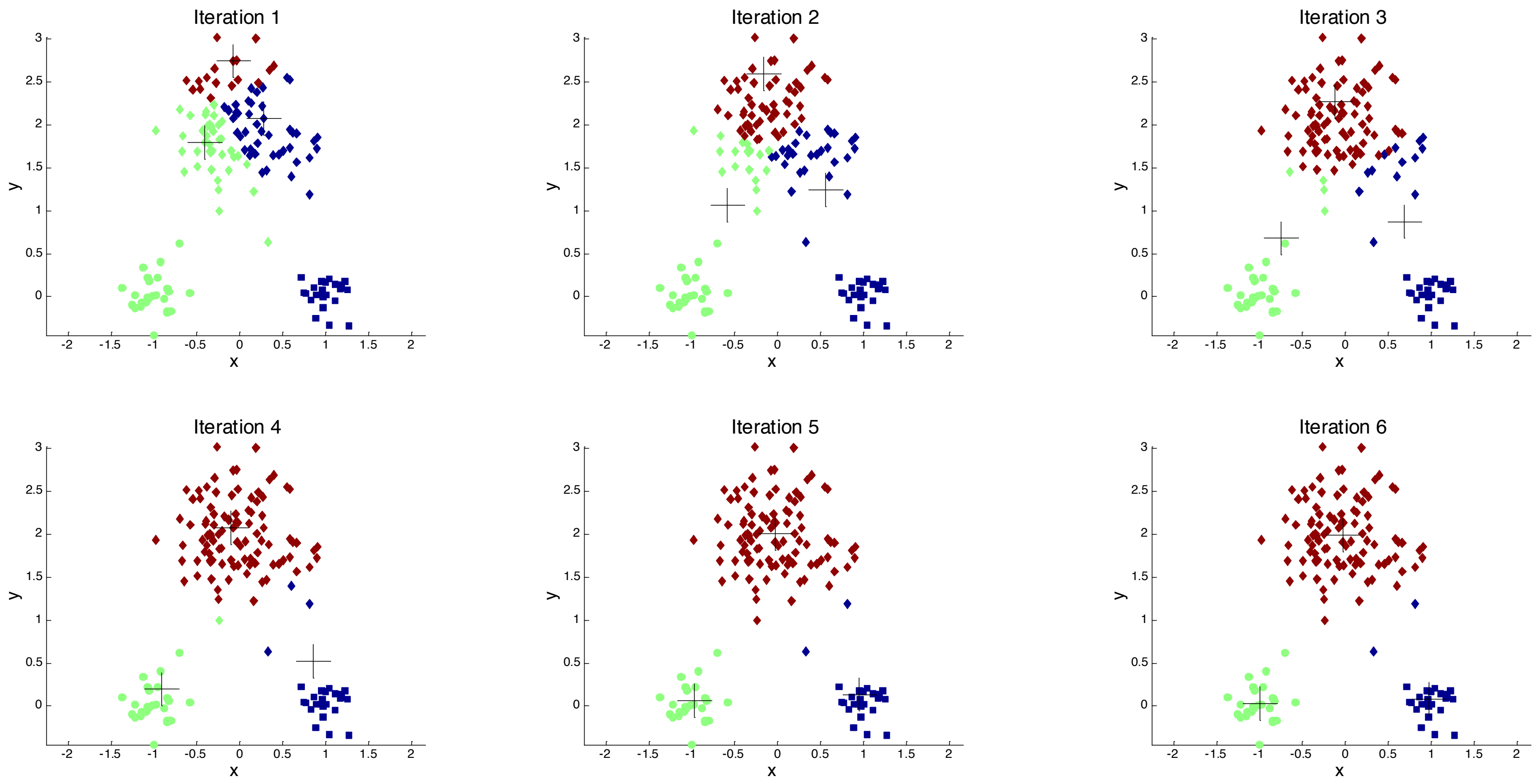
K-Means Clustering



Slide adapted from: https://www-users.cs.umn.edu/~kumar001/dmbook/slides/chap7_basic_cluster_analysis.pdf

K-Means Clustering: Weaknesses?

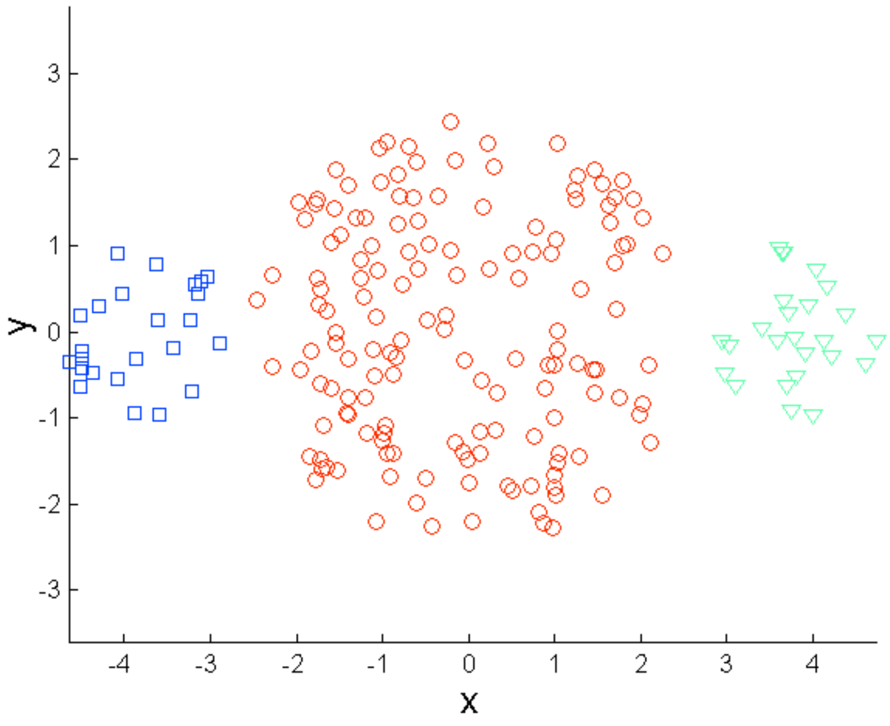
- Sensitive to initial centroids: different outcomes for same data



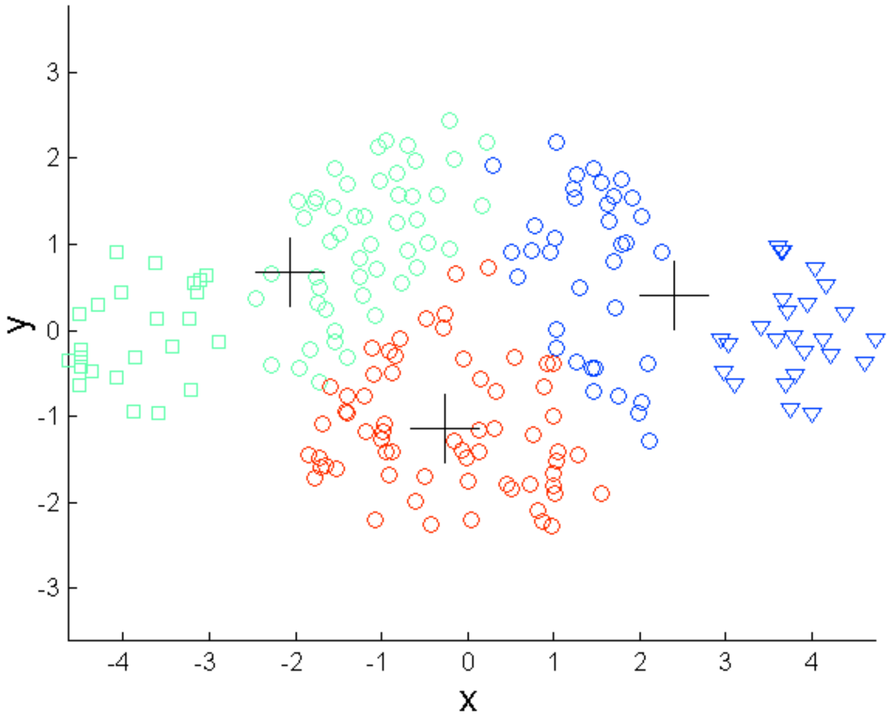
Slide adapted from: https://www-users.cs.umn.edu/~kumar001/dmbook/slides/chap7_basic_cluster_analysis.pdf

K-Means Clustering: Weaknesses?

- Not robust when clusters have different sizes:



Original Points

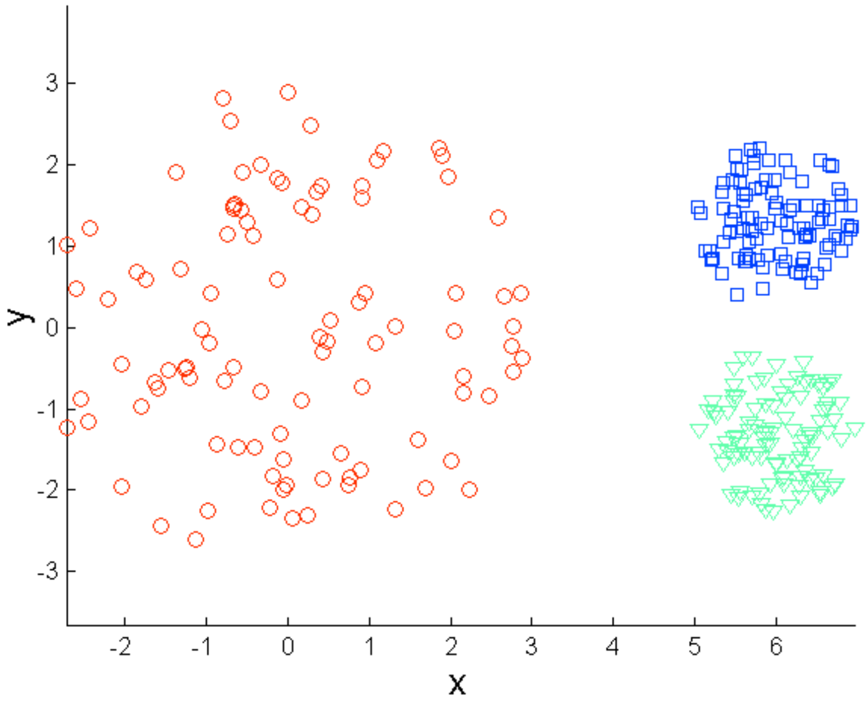


K-means (3 Clusters)

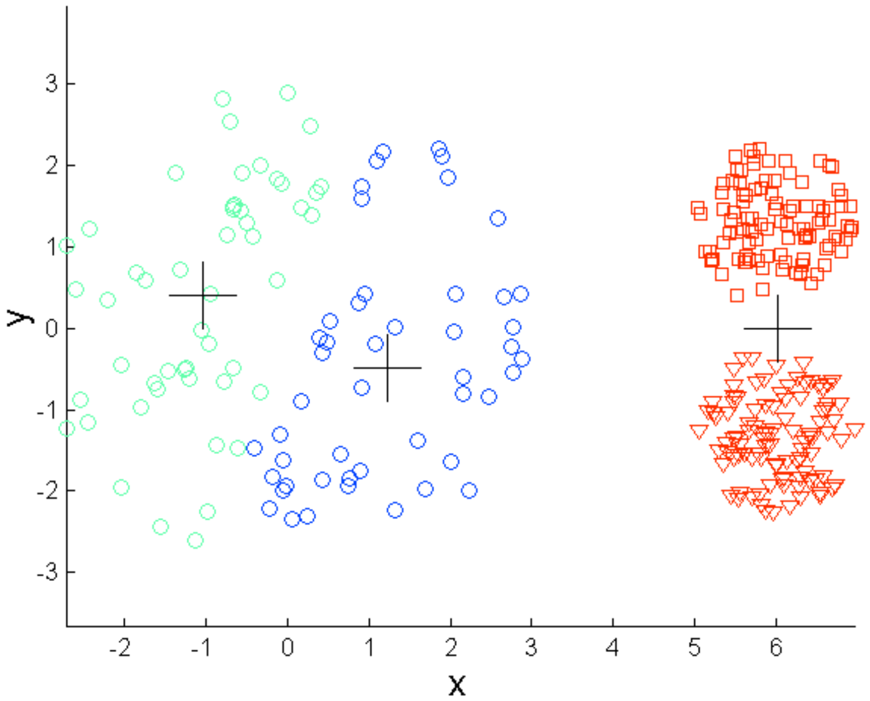
Slide adapted from: https://www-users.cs.umn.edu/~kumar001/dmbook/slides/chap7_basic_cluster_analysis.pdf

K-Means Clustering: Weaknesses?

- Not robust when clusters have different densities:



Original Points

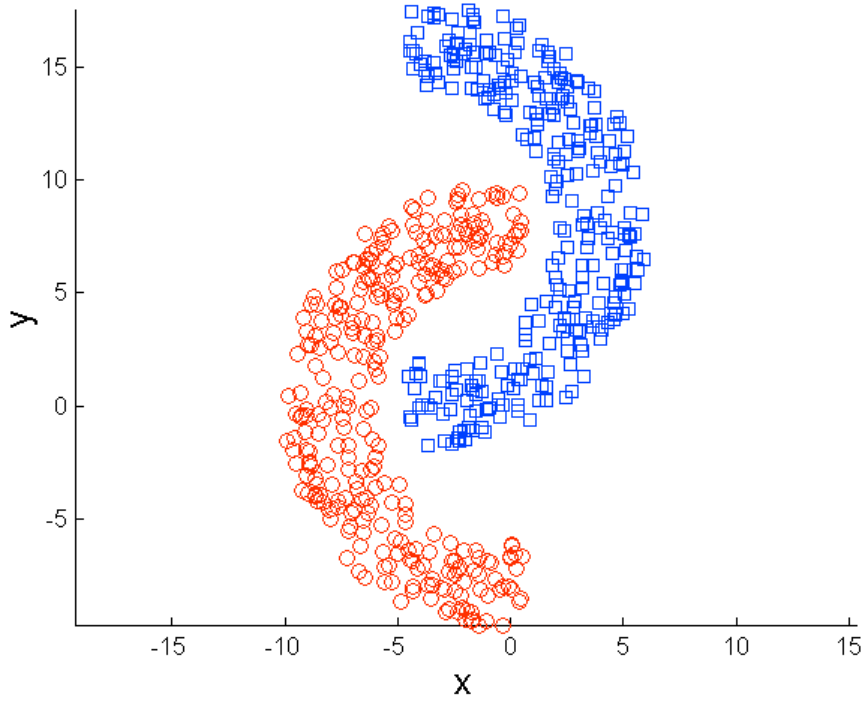


K-means (3 Clusters)

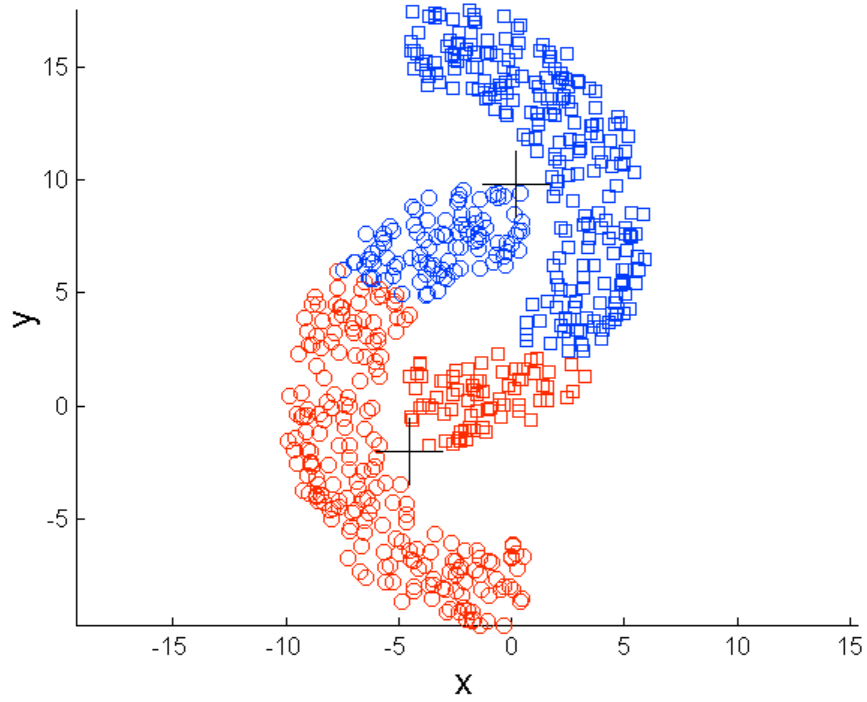
Slide adapted from: https://www-users.cs.umn.edu/~kumar001/dmbook/slides/chap7_basic_cluster_analysis.pdf

K-Means Clustering: Weaknesses?

- Not robust when clusters have different globular shapes:



Original Points

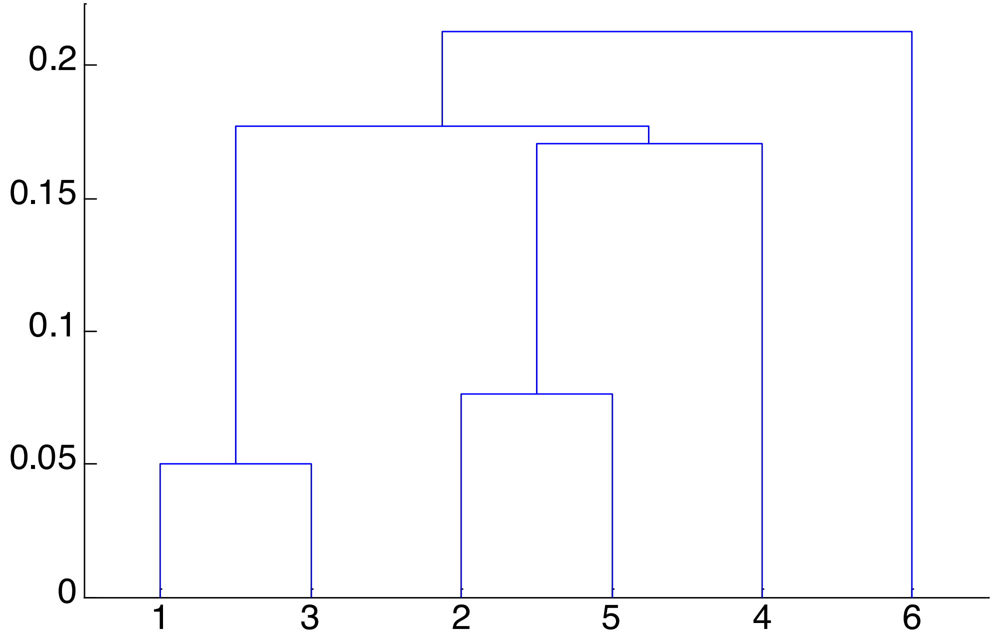


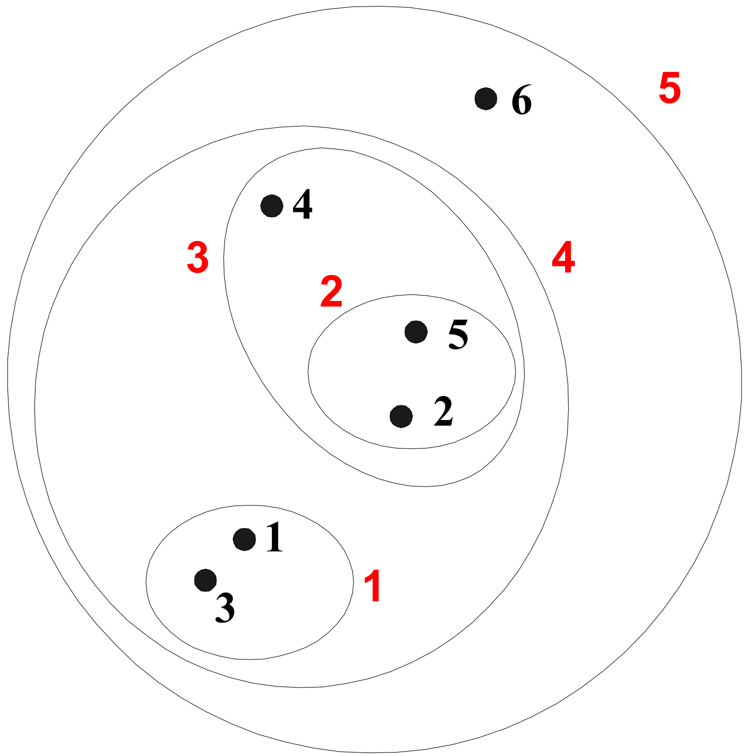
K-means (3 Clusters)

Slide adapted from: https://www-users.cs.umn.edu/~kumar001/dmbook/slides/chap7_basic_cluster_analysis.pdf

Hierarchical Clustering

- Set of nested clusters organized in hierarchical tree by merging/splitting
- Dendrogram visualization: shows sequence of merges/splits





Slide adapted from: https://www-users.cs.umn.edu/~kumar001/dmbook/slides/chap7_basic_cluster_analysis.pdf

Hierarchical Clustering: Two Main Approaches

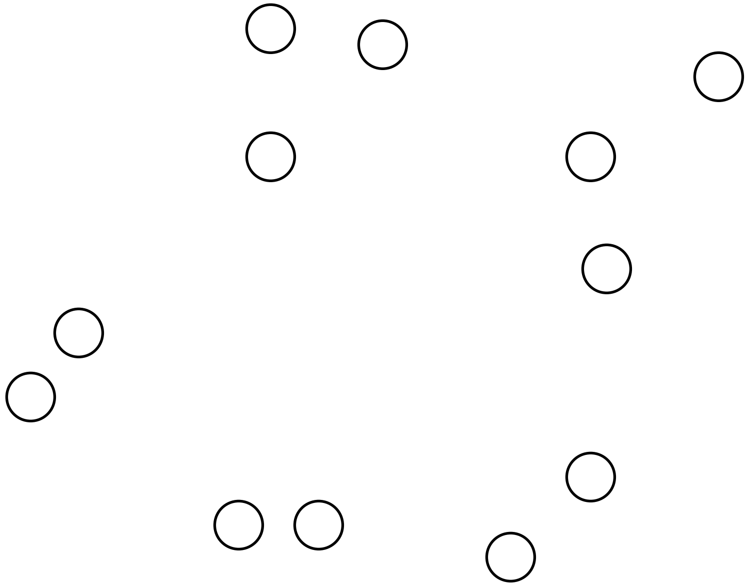
- Agglomerative:
  - Start with points as individual clusters
  - At each step, merge closest pair of clusters until only one cluster (or k clusters) left

- Divisive:
  - Start with one, all-inclusive cluster
  - At each step, split a cluster until each cluster contains an individual point (or there are k clusters)

Slide adapted from: https://www-users.cs.umn.edu/~kumar001/dmbook/slides/chap7_basic_cluster_analysis.pdf

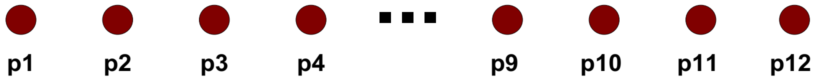
Agglomerative Clustering: First Step

- Start with clusters of individual points and a proximity matrix



	p1	p2	p3	p4	p5	. . .
p1						
p2						
p3						
p4						
p5						
.						
.						
.						

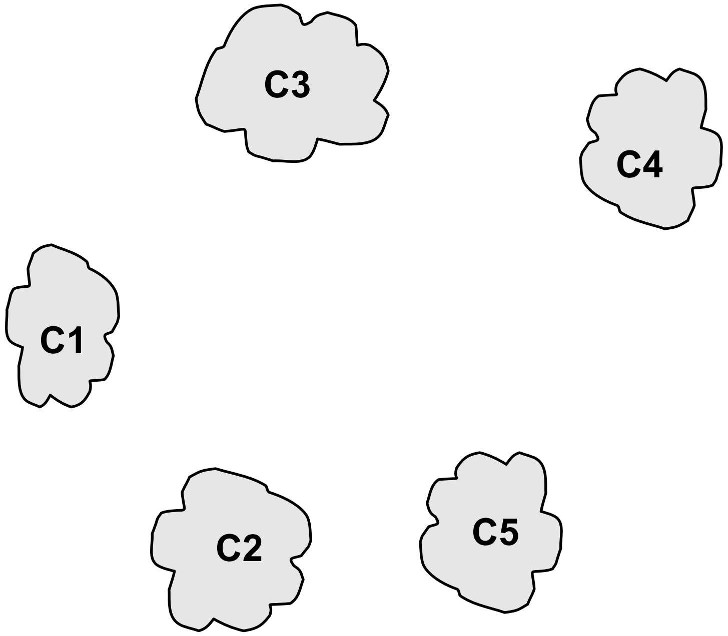
Proximity Matrix



Slide adapted from: https://www-users.cs.umn.edu/~kumar001/dmbook/slides/chap7_basic_cluster_analysis.pdf

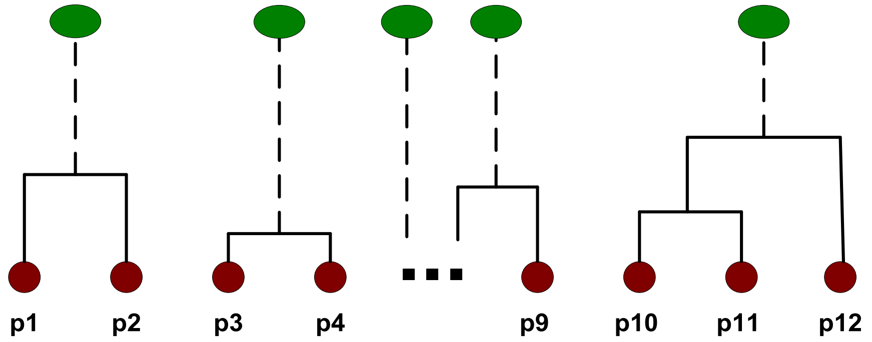
Agglomerative Clustering: Intermediate Step

- Start with clusters of individual points and a proximity matrix



	C1	C2	C3	C4	C5
C1					
C2					
C3					
C4					
C5					

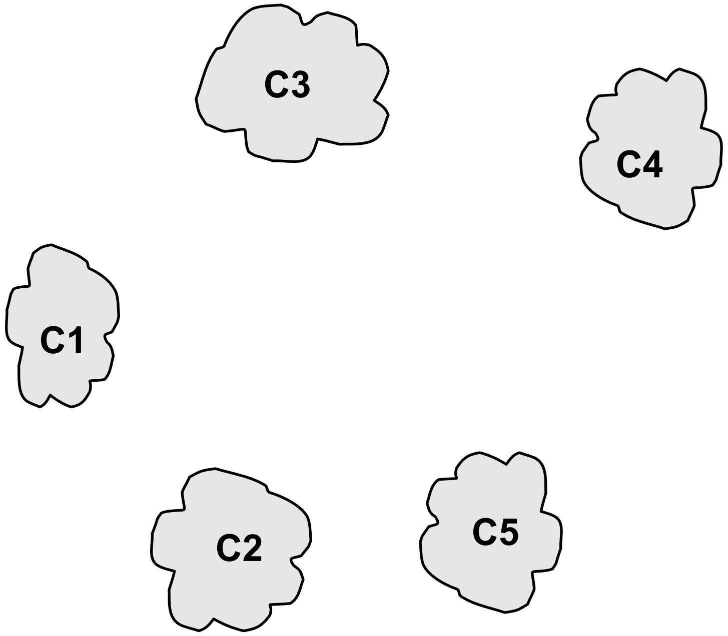
Proximity Matrix



Slide adapted from: https://www-users.cs.umn.edu/~kumar001/dmbook/slides/chap7_basic_cluster_analysis.pdf

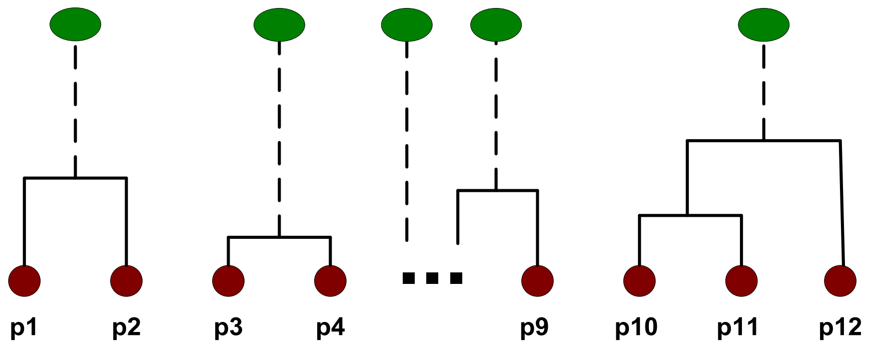
Agglomerative Clustering: Intermediate Step

- After several merging steps, we have some clusters



	C1	C2	C3	C4	C5
C1					
C2					
C3					
C4					
C5					

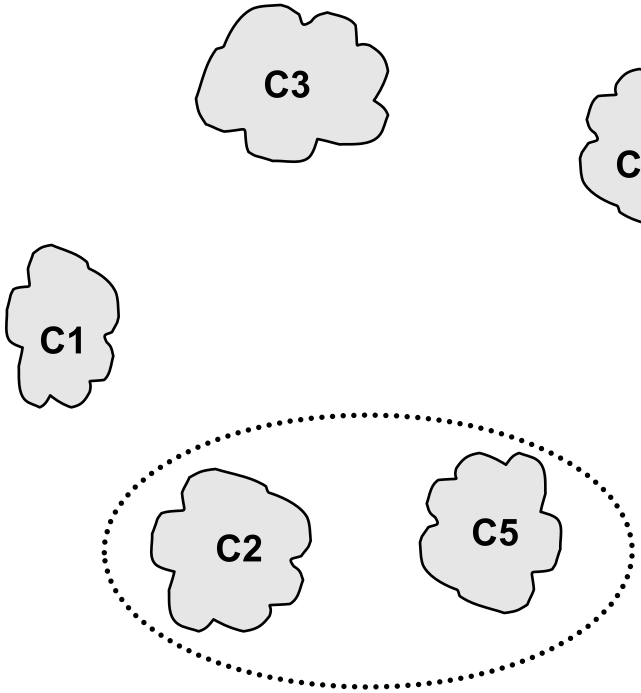
Proximity Matrix

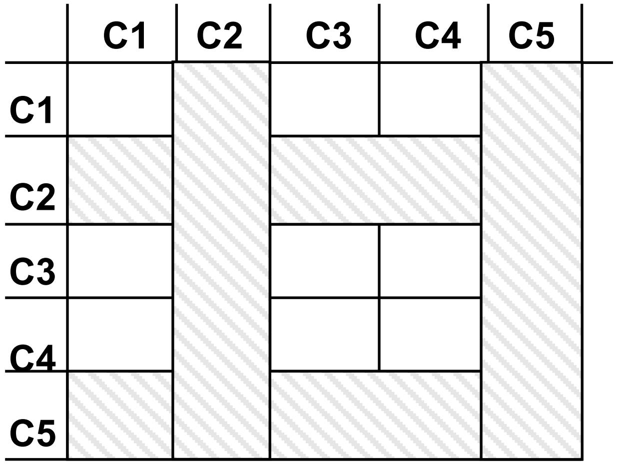


Slide adapted from: https://www-users.cs.umn.edu/~kumar001/dmbook/slides/chap7_basic_cluster_analysis.pdf

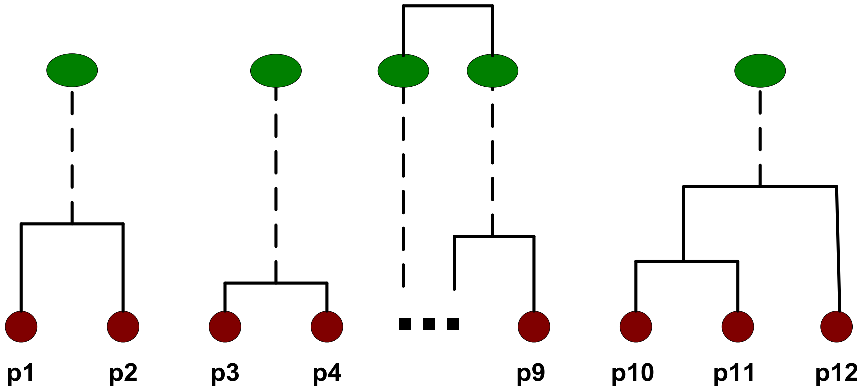
Agglomerative Clustering: Intermediate Step

- Merge two closest clusters (C2 and C5) and update proximity matrix.



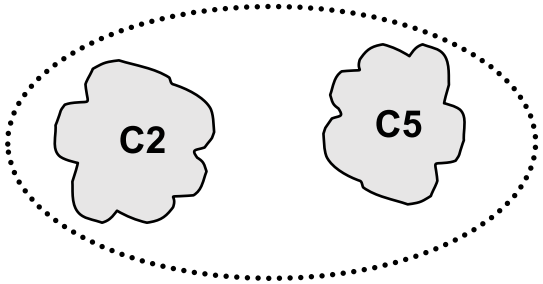


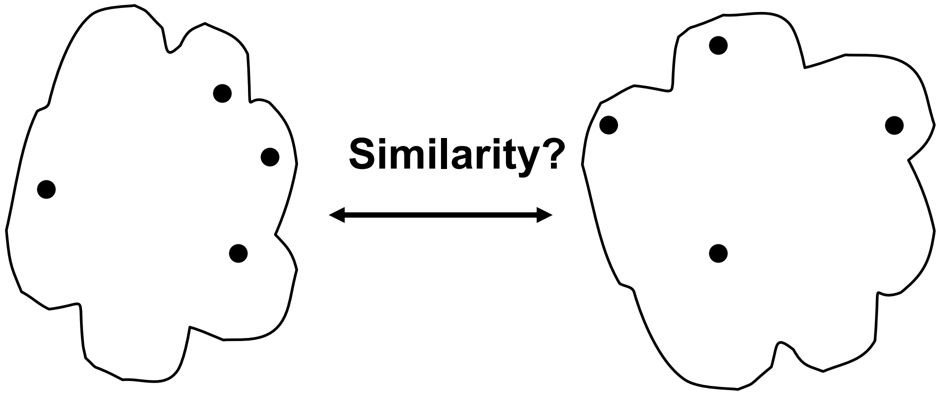
Proximity Matrix



Slide adapted from: https://www-users.cs.umn.edu/~kumar001/dmbook/slides/chap7_basic_cluster_analysis.pdf

How to Measure Inter-Cluster Distance?

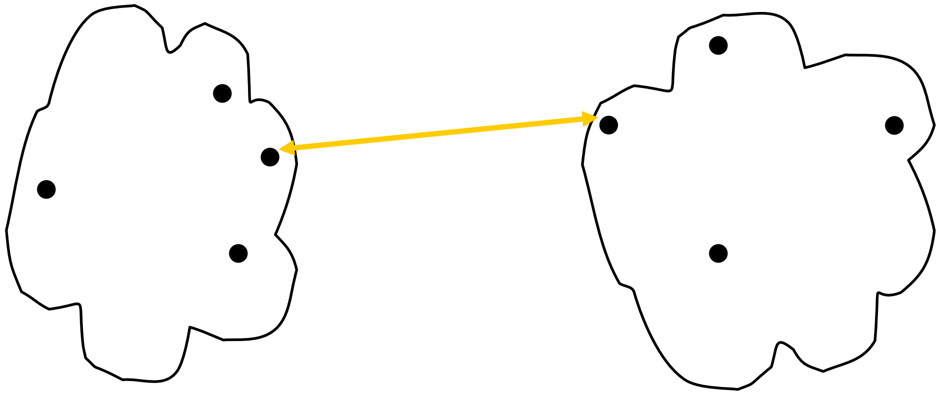




Slide adapted from: https://www-users.cs.umn.edu/~kumar001/dmbook/slides/chap7_basic_cluster_analysis.pdf

How to Measure Inter-Cluster Distance?

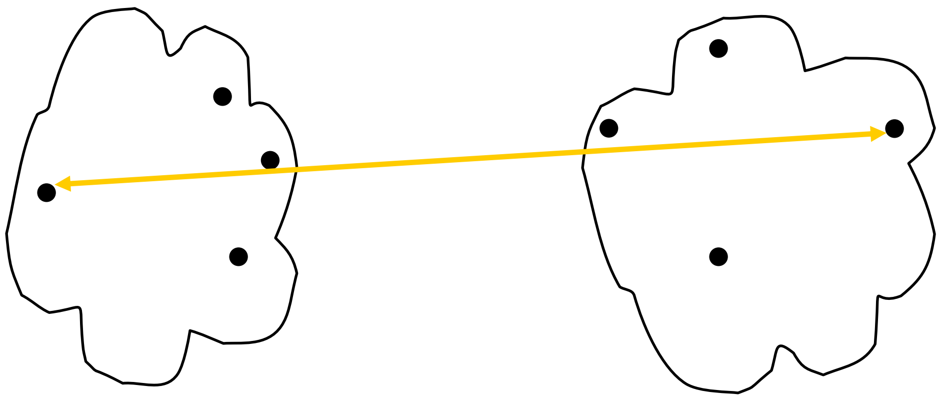
- Minimum distance



Slide adapted from: https://www-users.cs.umn.edu/~kumar001/dmbook/slides/chap7_basic_cluster_analysis.pdf

How to Measure Inter-Cluster Distance?

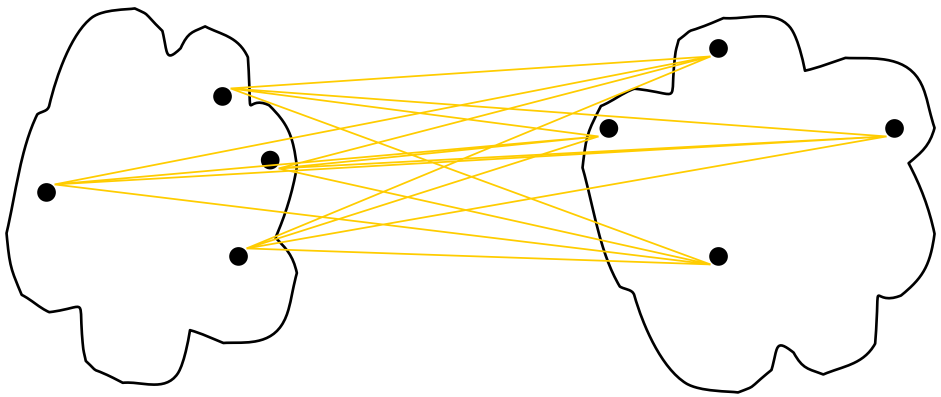
- Minimum distance
- Maximum distance



Slide adapted from: https://www-users.cs.umn.edu/~kumar001/dmbook/slides/chap7_basic_cluster_analysis.pdf

How to Measure Inter-Cluster Distance?

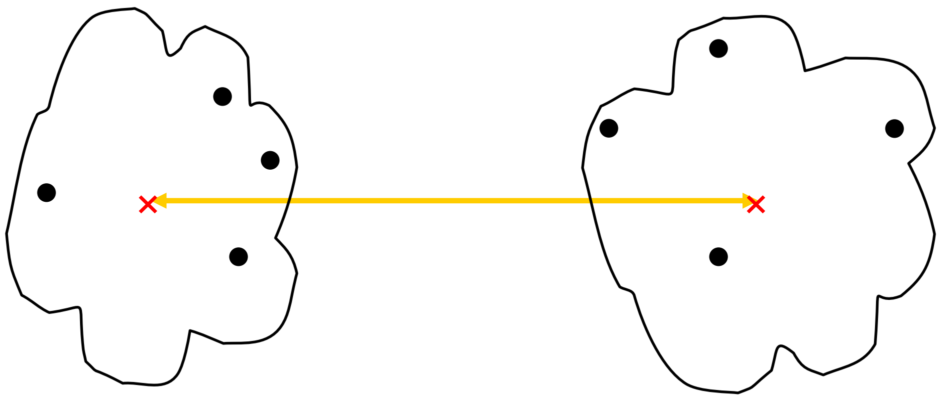
- Minimum distance
- Maximum distance
- Group average



Slide adapted from: https://www-users.cs.umn.edu/~kumar001/dmbook/slides/chap7_basic_cluster_analysis.pdf

How to Measure Inter-Cluster Distance?

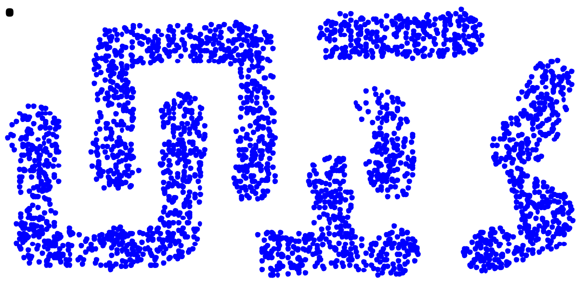
- Minimum distance
- Maximum distance
- Group average
- Distance Between Centroids



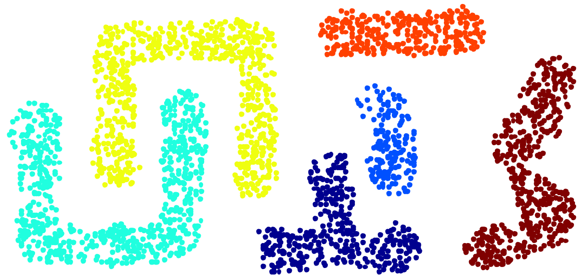
Slide adapted from: https://www-users.cs.umn.edu/~kumar001/dmbook/slides/chap7_basic_cluster_analysis.pdf

Minimum Distance: Strengths/Weaknesses?

- Can handle non-elliptical shapes:

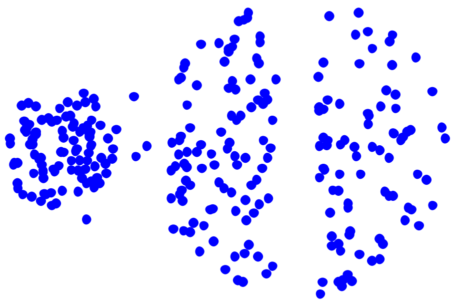


Original Points

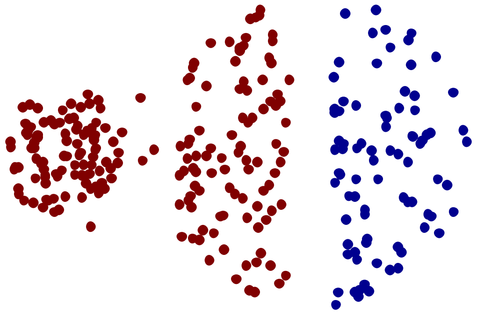


Six Clusters

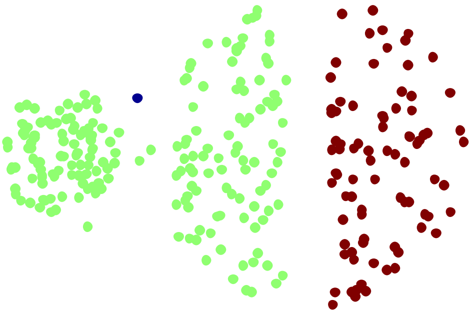
- Sensitive to noise and outliers:



Original Points



Two Clusters

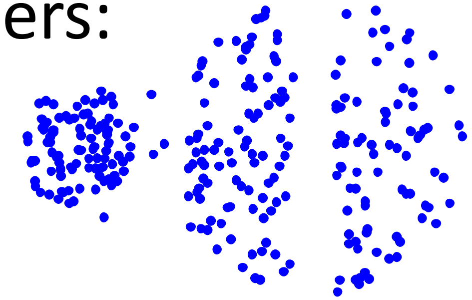


Three Clusters

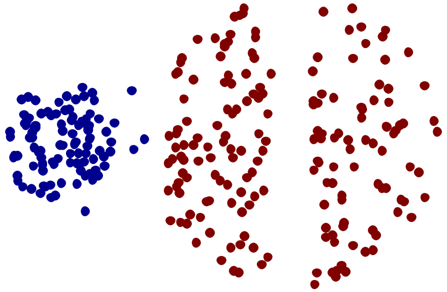
Slide adapted from: https://www-users.cs.umn.edu/~kumar001/dmbook/slides/chap7_basic_cluster_analysis.pdf

Maximum Distance: Strengths/Weaknesses?

- Less susceptible to noise and outliers:

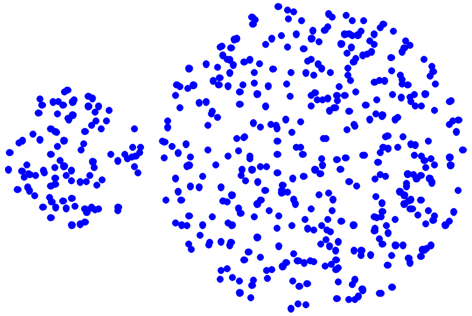


Original Points

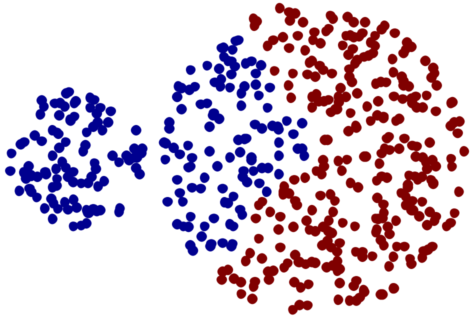


Two Clusters

- Tends to break large clusters:



Original Points



Two Clusters

Slide adapted from: https://www-users.cs.umn.edu/~kumar001/dmbook/slides/chap7_basic_cluster_analysis.pdf

Hierarchical Clustering: Strengths?

- Any number of clusters can be obtained by ‘cutting’ the dendrogram at the proper level
- They may correspond to meaningful taxonomies
  - Example in biological sciences (e.g., animal kingdom, phylogeny reconstruction, …)

Slide adapted from: https://www-users.cs.umn.edu/~kumar001/dmbook/slides/chap7_basic_cluster_analysis.pdf

Today’s Topics

- Computer Vision
- Ensemble Learning
- Unsupervised Learning
- Lab