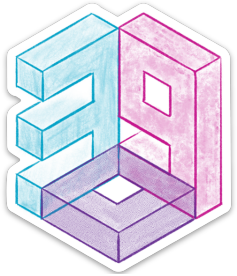


https://introml.mit.edu/

# 6.390 Intro to Machine Learning

## Lecture 10: Non-parametric Models

Shen Shen

Nov 6, 2025

11am, Room 10-250

Interactive Slides and Lecture Recording

# Outline

- Non-parametric models overview
- Supervised learning non-parametric
  - $k$-nearest neighbor
  - Decision Tree
- Unsupervised learning non-parametric
  - $k$-means clustering

Neural networks reason over billions of parameters, have incredible predictive power

Yet in many situations, we may not want to immediately fully commit to deep learning


How some election officials are trying to verify the vote more easily

Oct 29, 2020 6:20 PM EST

Rich Caruana
Microsoft Research
rcaruana@microsoft.com

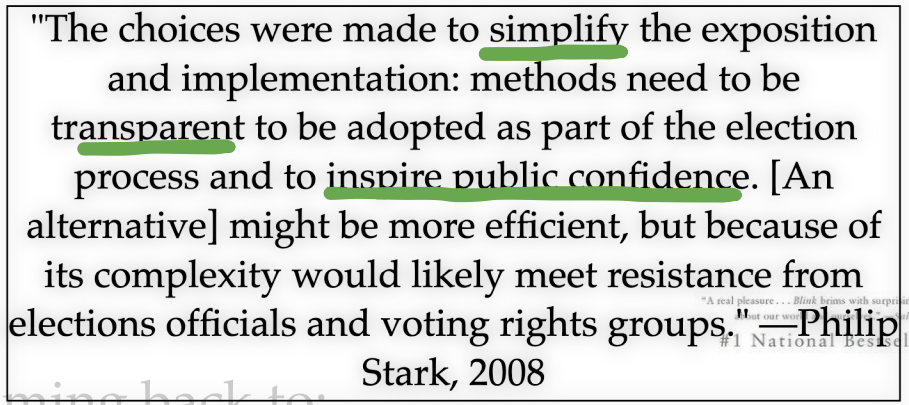
Yin Lou
LinkedIn Corporation
ylou@linkedin.com

Johannes Gehrke
Microsoft
johannes@microsoft.com

Paul Koch
Microsoft Research
paulkoch@microsoft.com

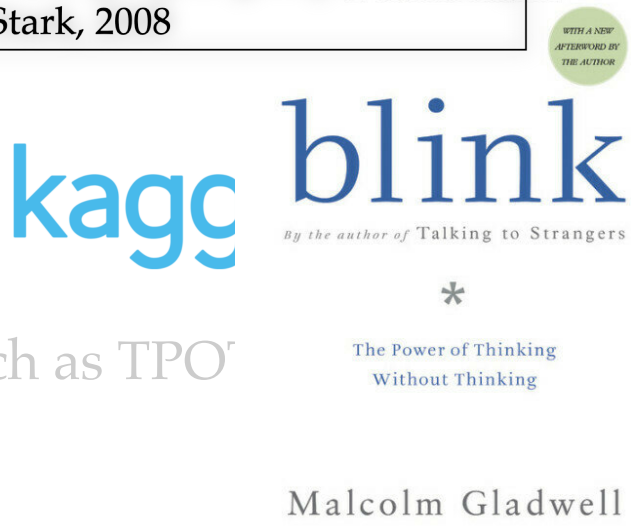
Marc Sturm
NewYork-Presbyterian Hospital
mas9161@nyp.org

Noémie Elhadad
Columbia University
noemie.elhadad@columbia.edu


"The choices were made to simplify the exposition and implementation: methods need to be transparent to be adopted as part of the election process and to inspire public confidence. [An alternative] might be more efficient, but because of its complexity would likely meet resistance from elections officials and voting rights groups." —Philip Stark, 2008

A look at the tools I keeps coming back to:

- Packages: scikit learn, pandas, numpy
- Frameworks: Keras, Tensorflow, Pytorch and Fastai
- Algorithms: lightgbm, xgboost, catboost
- AutoML tools: Prevision.io, h2o and other open sources such as TPO
- Cloud services: Google colab and kaggle kernels




[source]

There’s more to machine learning than predictive power

- Interpretability — understanding why a model makes a decision.
- Adaptivity — updating to new data without retraining from scratch.
- Transparency & trust — explaining results to humans and stakeholders.

Even if all we care about is vanilla predictive power, we

- Data exploration — learning structure before committing to a model.
- Efficiency — using the right model for the right data scale.
- Insight generation — discovering mechanisms, not just correlations.

All ML models *learn* parameters — but not all are parametric.

Non-parametric doesn't mean no parameters

Instead, it means the model doesn't assumes a fixed functional form for $h$

- the model complexity grows with data
- usually fast to implement and train, and often effective as a baseline

"Having parameters" ≠ "Being parametric,"
just as "Having mechanisms" ≠ "Being mechanical":

| | |
|---|---|
| Electronic | Transistors, logic gates, amplifiers |
| Chemical | Catalytic converters, enzyme reactions |
| Biological | Vision, photosynthesis, DNA repair |
| Computational | Algorithms, logic circuits, neural networks |
| Social or economic | Market price formation, voting dynamics |

# Outline

- Non-parametric models overview
- Supervised learning non-parametric
  - $k$-nearest neighbor
  - Decision Tree
- Unsupervised learning non-parametric
  - $k$-means clustering

When we can't fit an equation, we just remember examples.

- Training: None (just memorize the entire training data)
- Predicting (inferencing, testing):
  - For a new data point $x_{\text{new}}$, take majority (class) or mean (regression) label of its $k$ nearest neighbors.

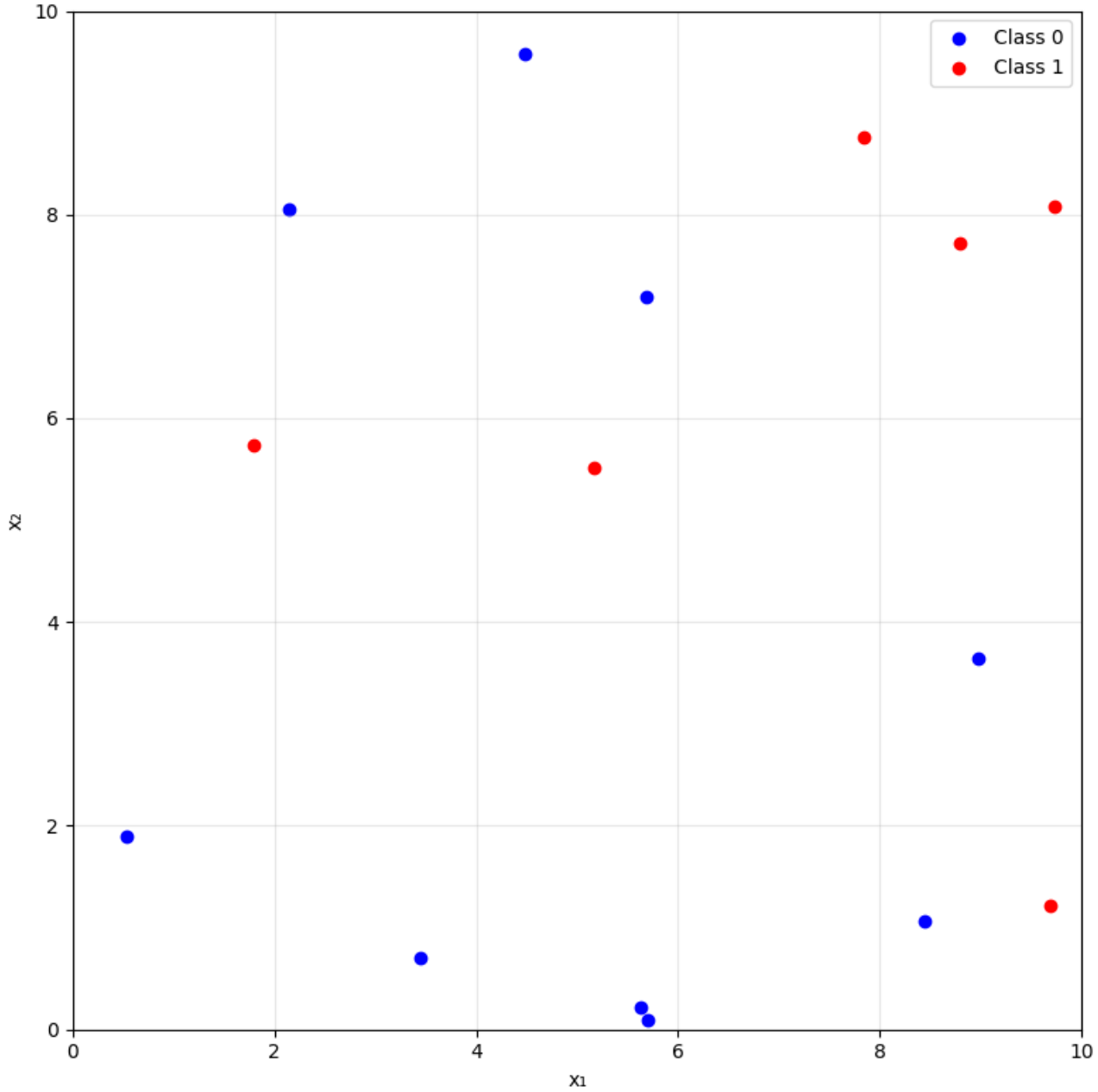
parameters learned:

the entire training dataset's features and labels

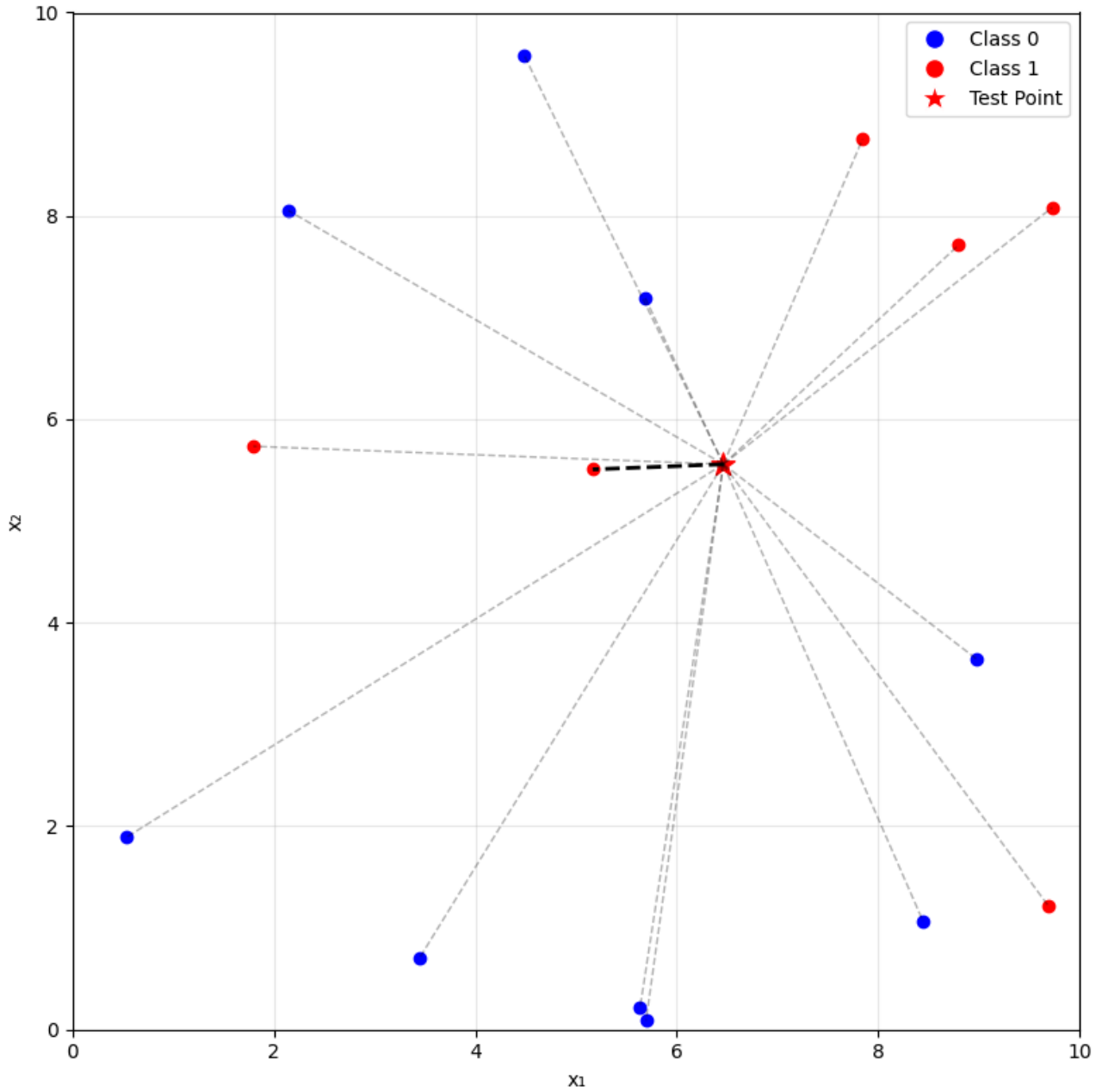
hyper-parameter:

- $k$ : number of neighbors considered
- distance metric (typically Euclidean or Manhattan distance)
- tie-breaking scheme (typically at random)

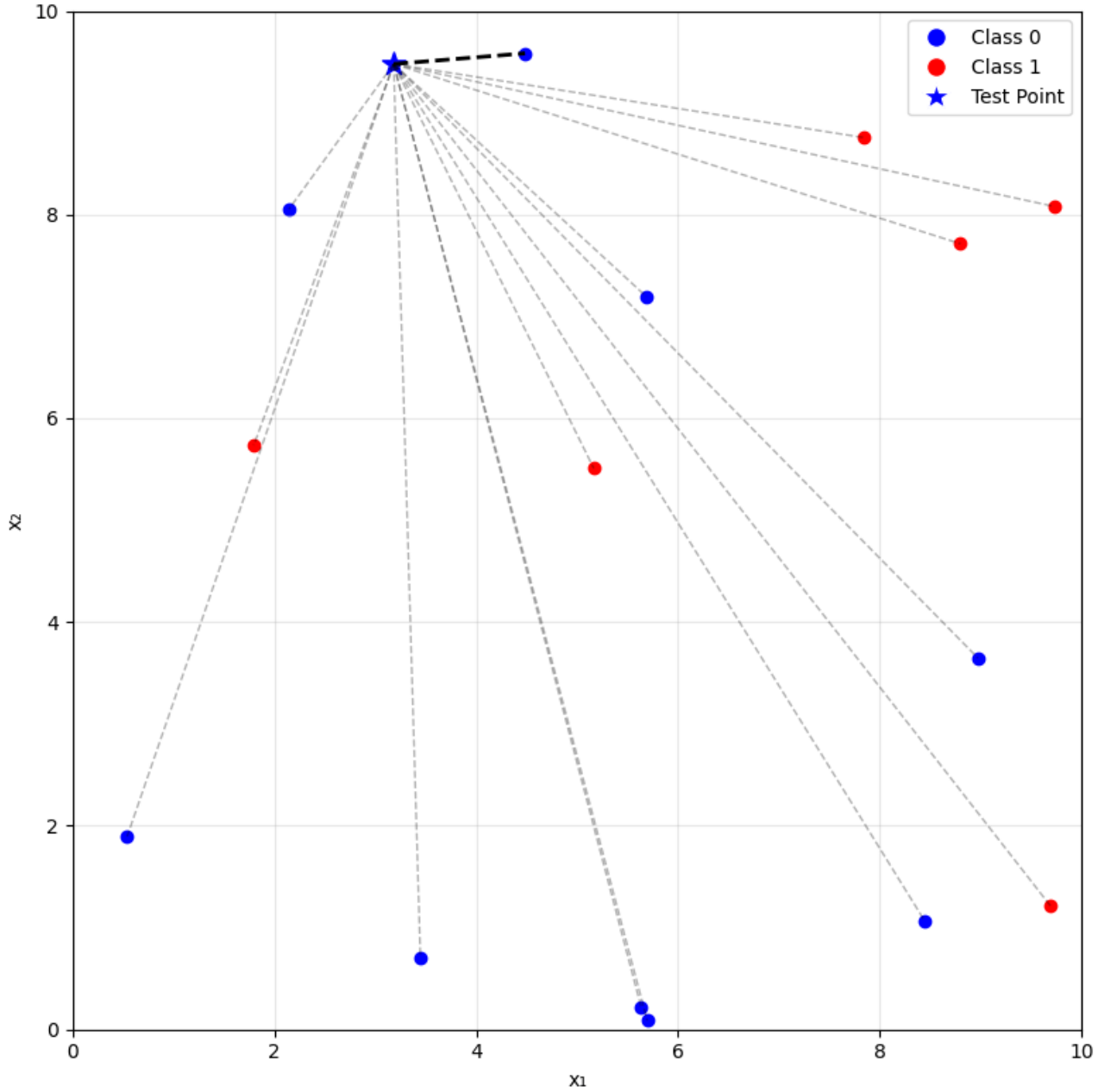
training data



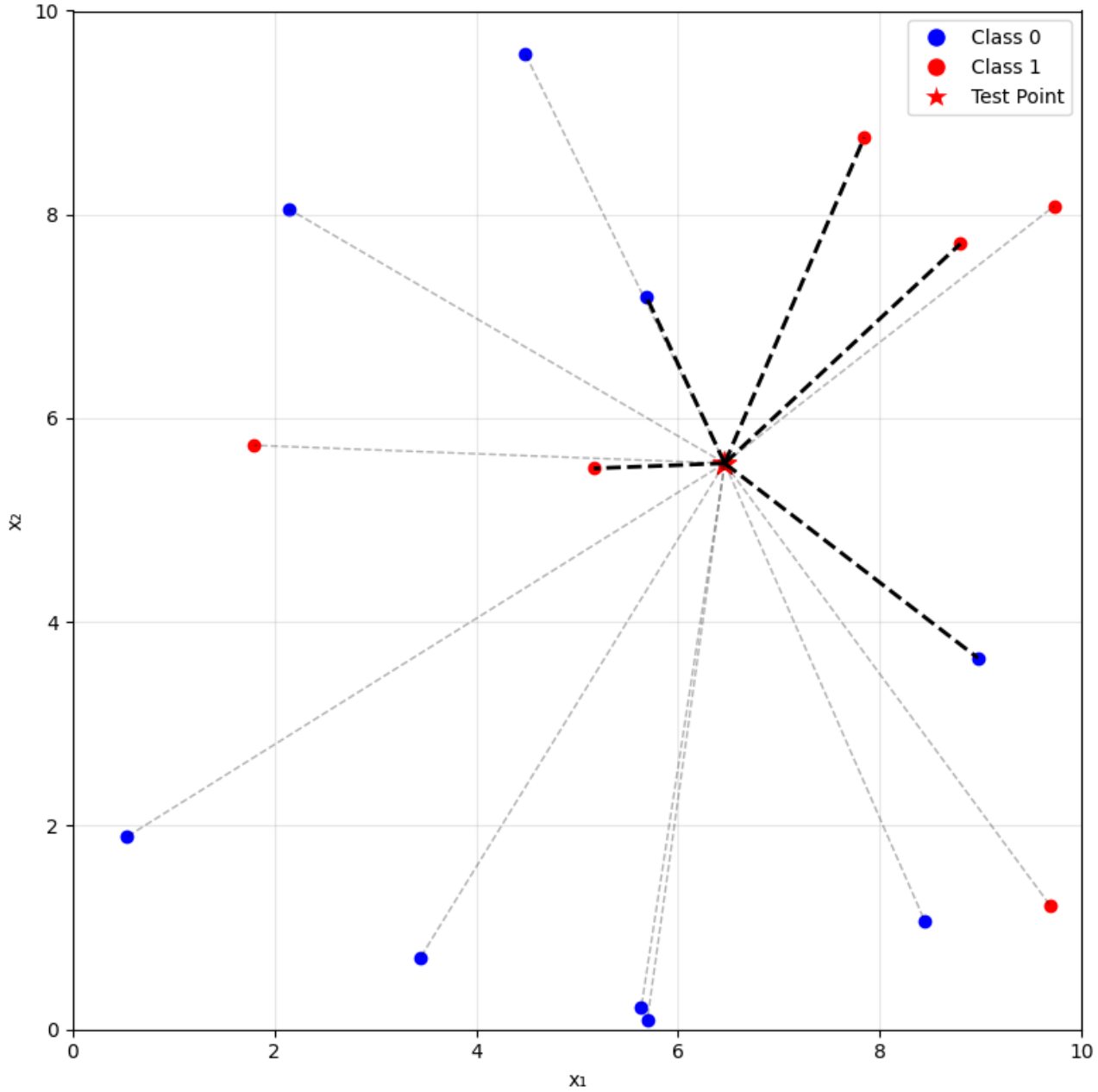
$k = 1$

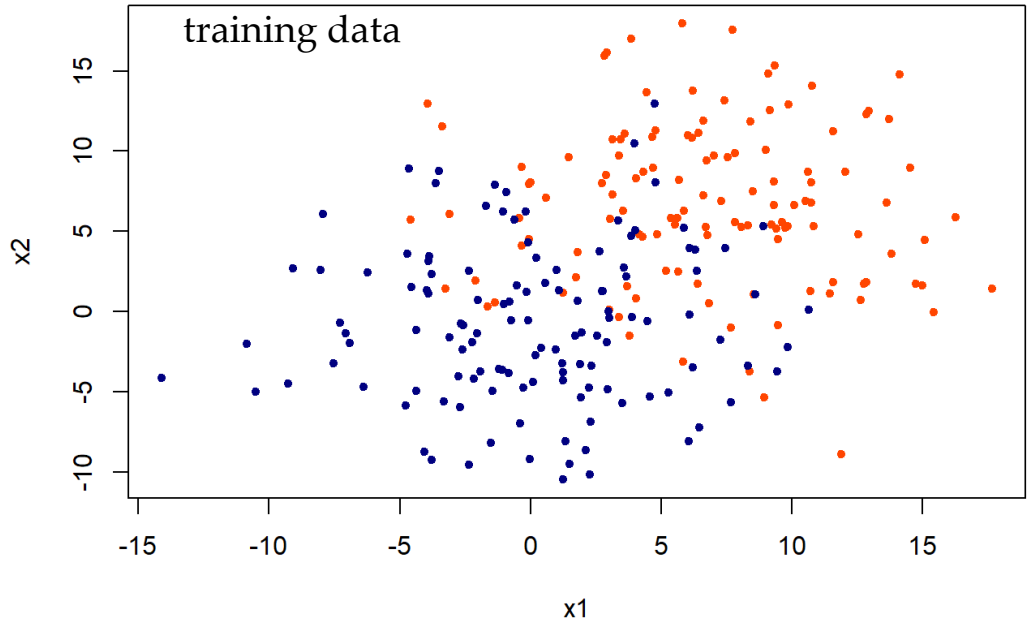


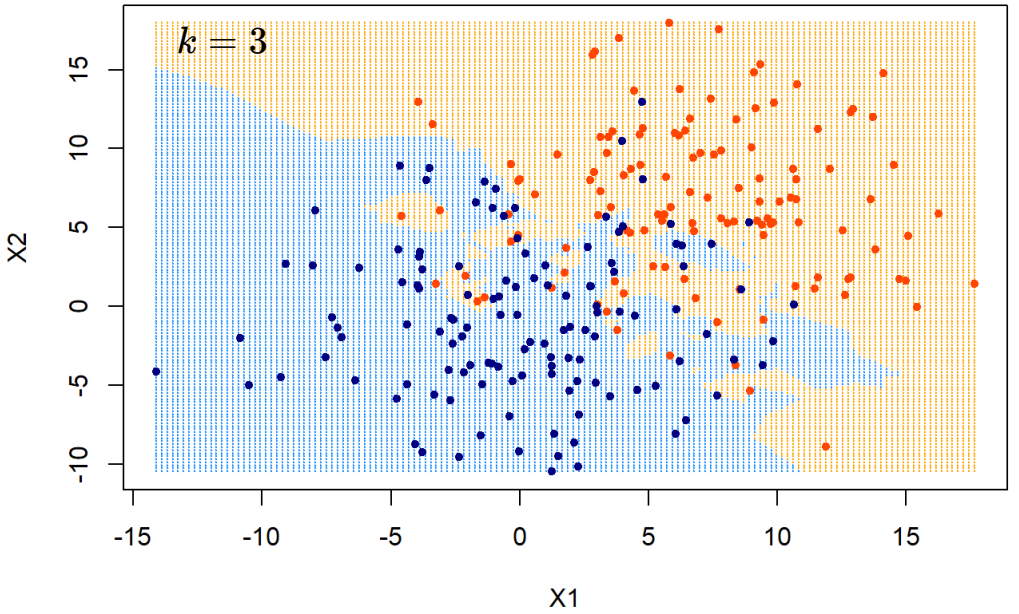
$k = 1$

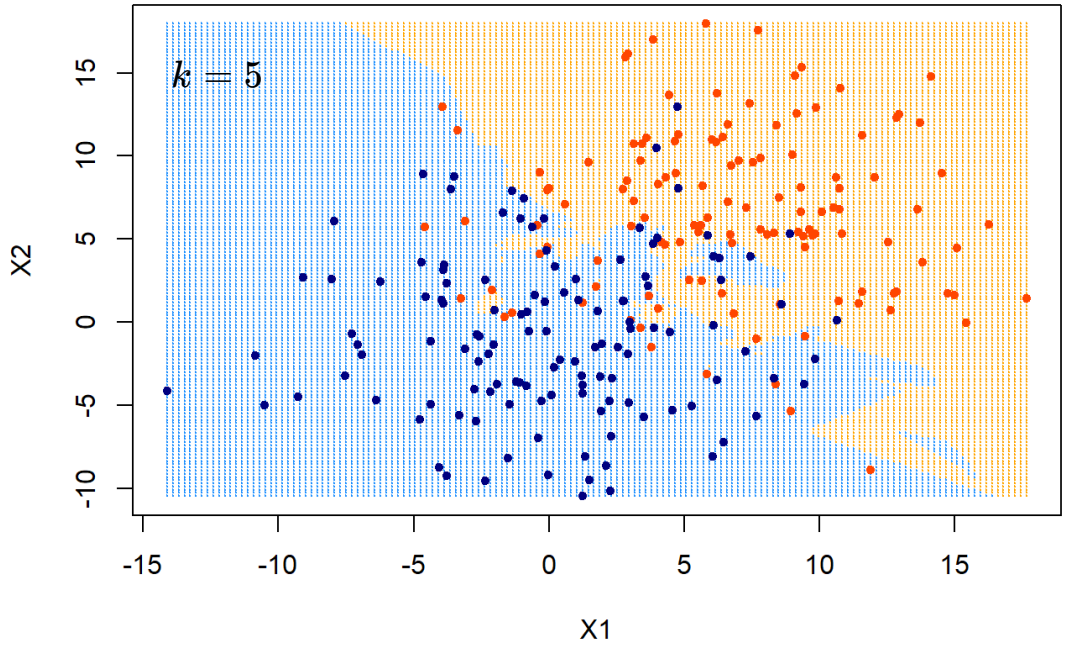


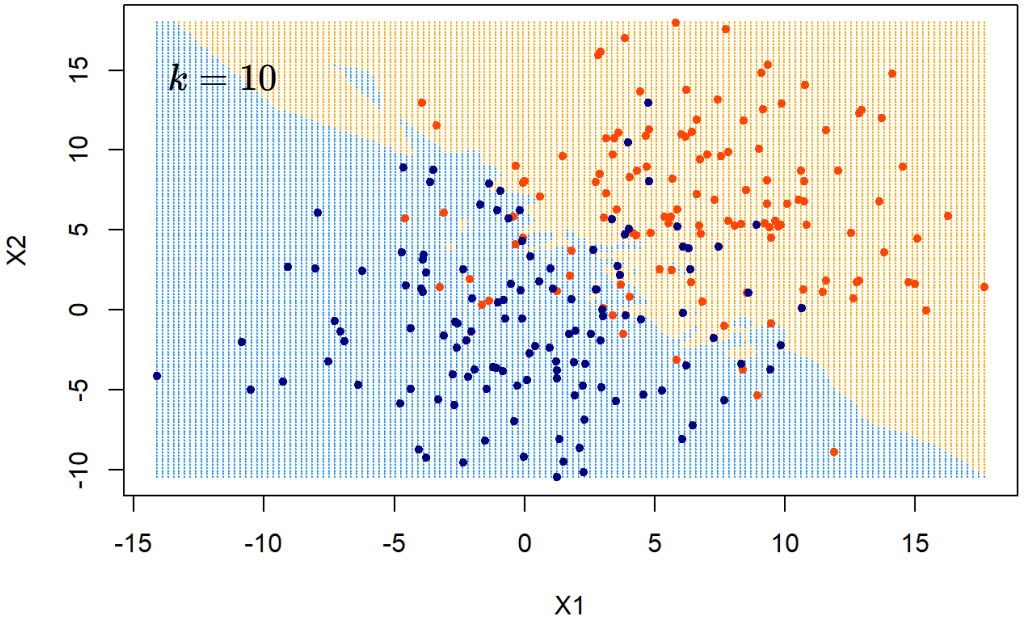
$k = 5$




training data
x2
x1
15
10
5
0
-5
-10
-15
-10
-5
0
5
10
15


$k = 3$
X2
X1
15
10
5
0
-5
-10
-15
-10
-5
0
5
10
15


$k = 5$
X2
X1
15
10
5
0
-5
-10
-15
-10
-5
0
5
10
15

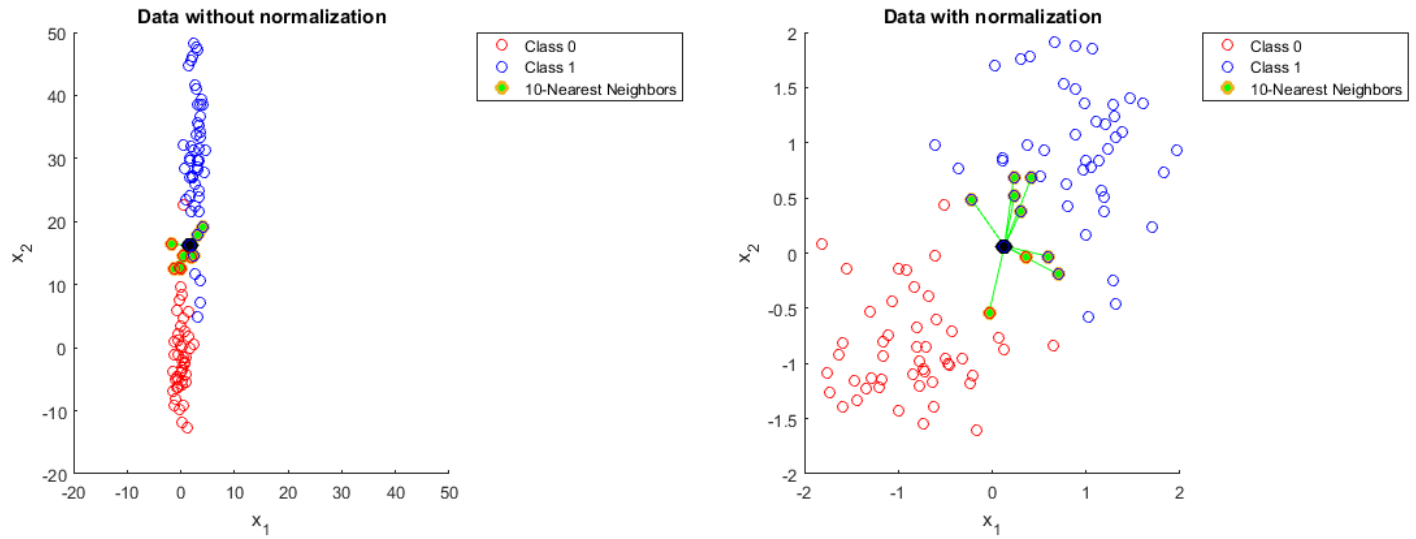

$k = 10$
X2
X1
15
10
5
0
-5
-10
-15
-10
-5
0
5
10
15

$k$-NN is a good choice when:

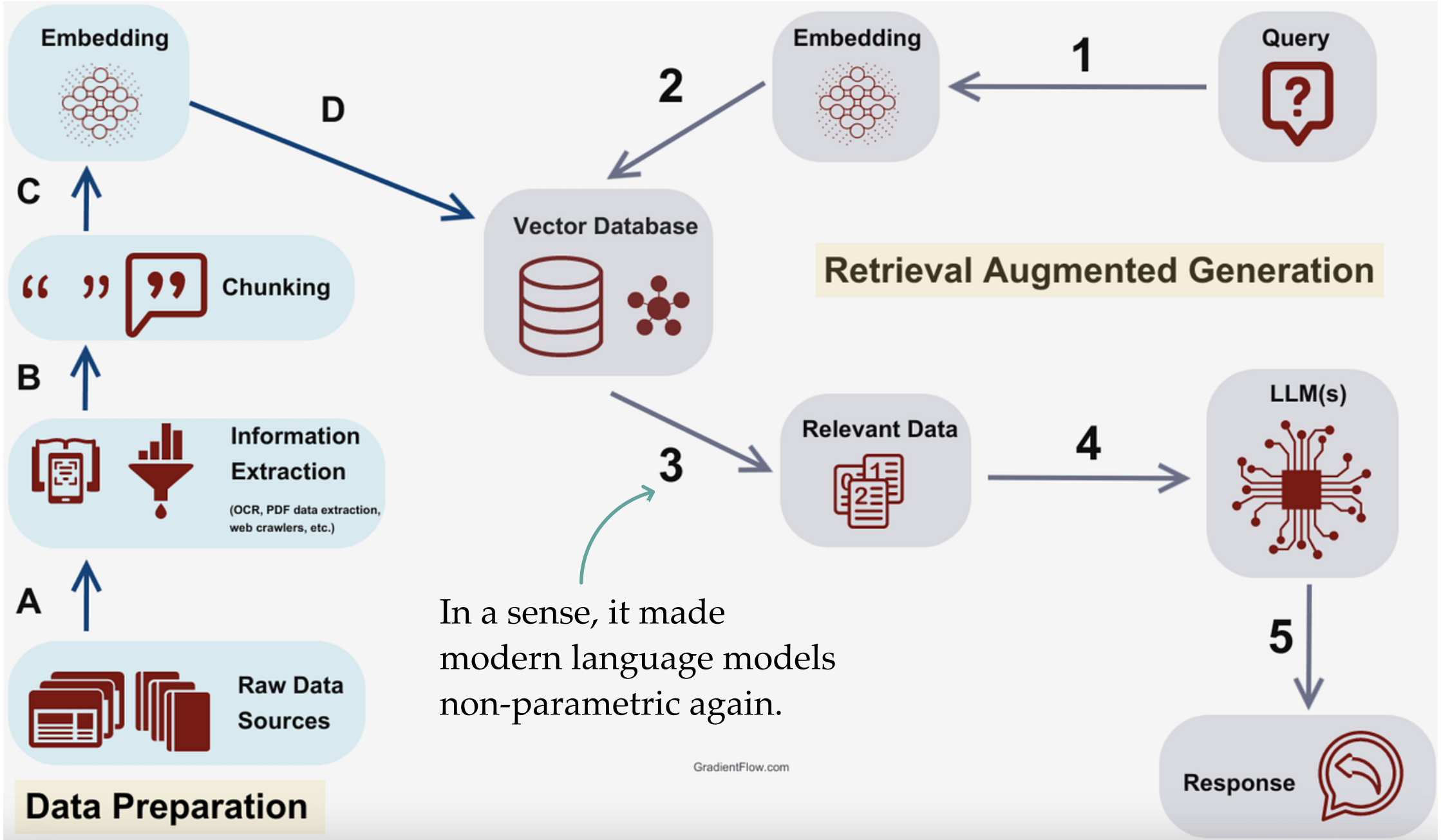
- Decision boundary is irregular / hard to parametrize.
- Dataset is small-to-medium (memory fits; latency acceptable).

Be cautious when:

- Very large $n$ or $d$ (naively needs all pair-wise distance calc, curse of dimensionality).
- Many irrelevant/noisy features (distance metric degrades).
- Features on different scales (must scale/normalize first).




[source]

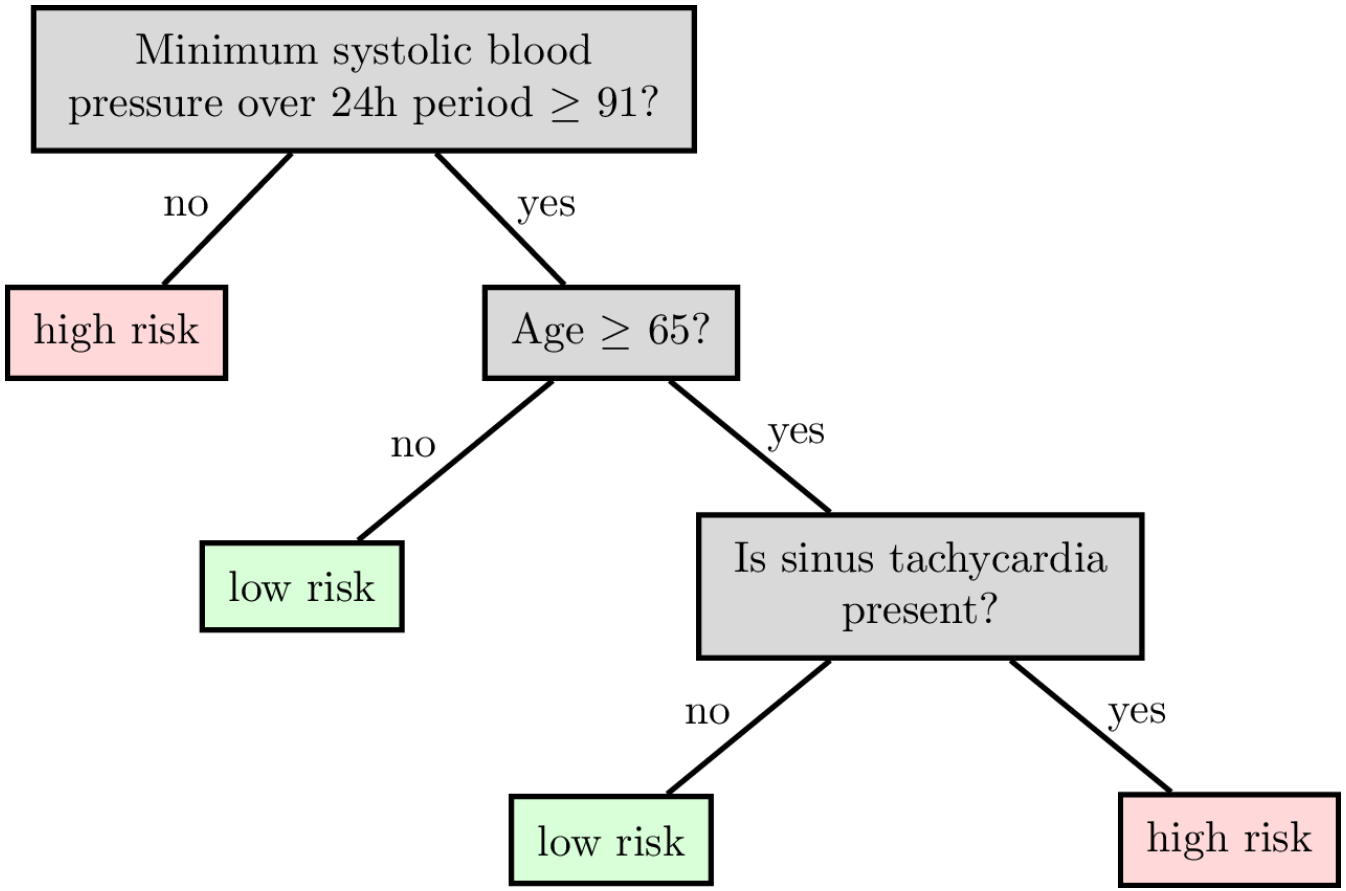



In a sense, it made modern language models non-parametric again.

# Outline

- Non-parametric models overview
- Supervised learning non-parametric
  - $k$-nearest neighbor
  - Decision Tree
- Unsupervised learning non-parametric
  - $k$-means clustering

# Reading a classification decision tree




features:

$x_1$ : date

$x_2$ : age

$x_3$ : height

$x_4$ : weight

$x_5$ : sinus tachycardia?

$x_6$ : min systolic bp, 24h

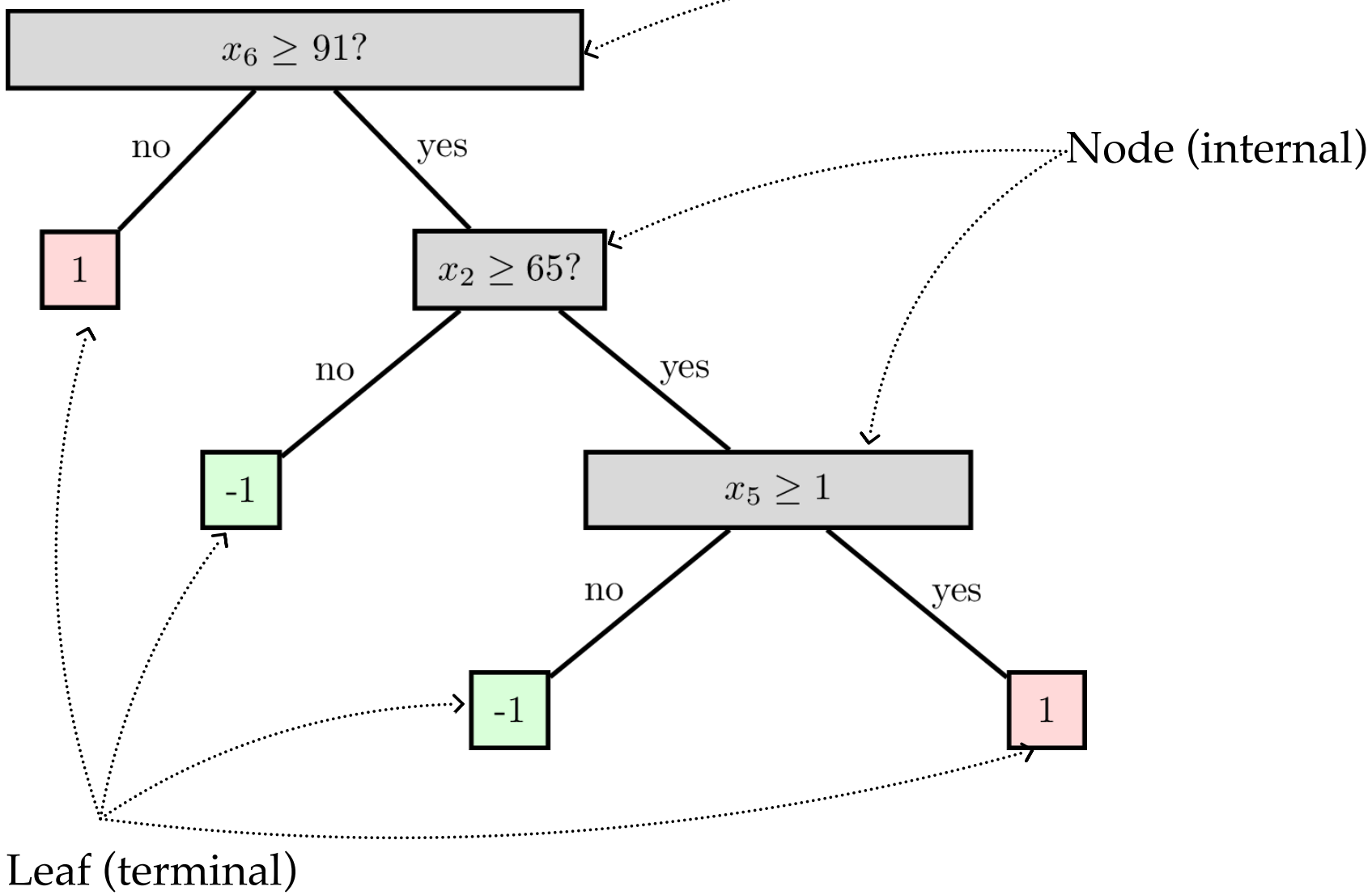
$x_7$ : latest diastolic bp

labels $y$ :

1: high risk

-1: low risk

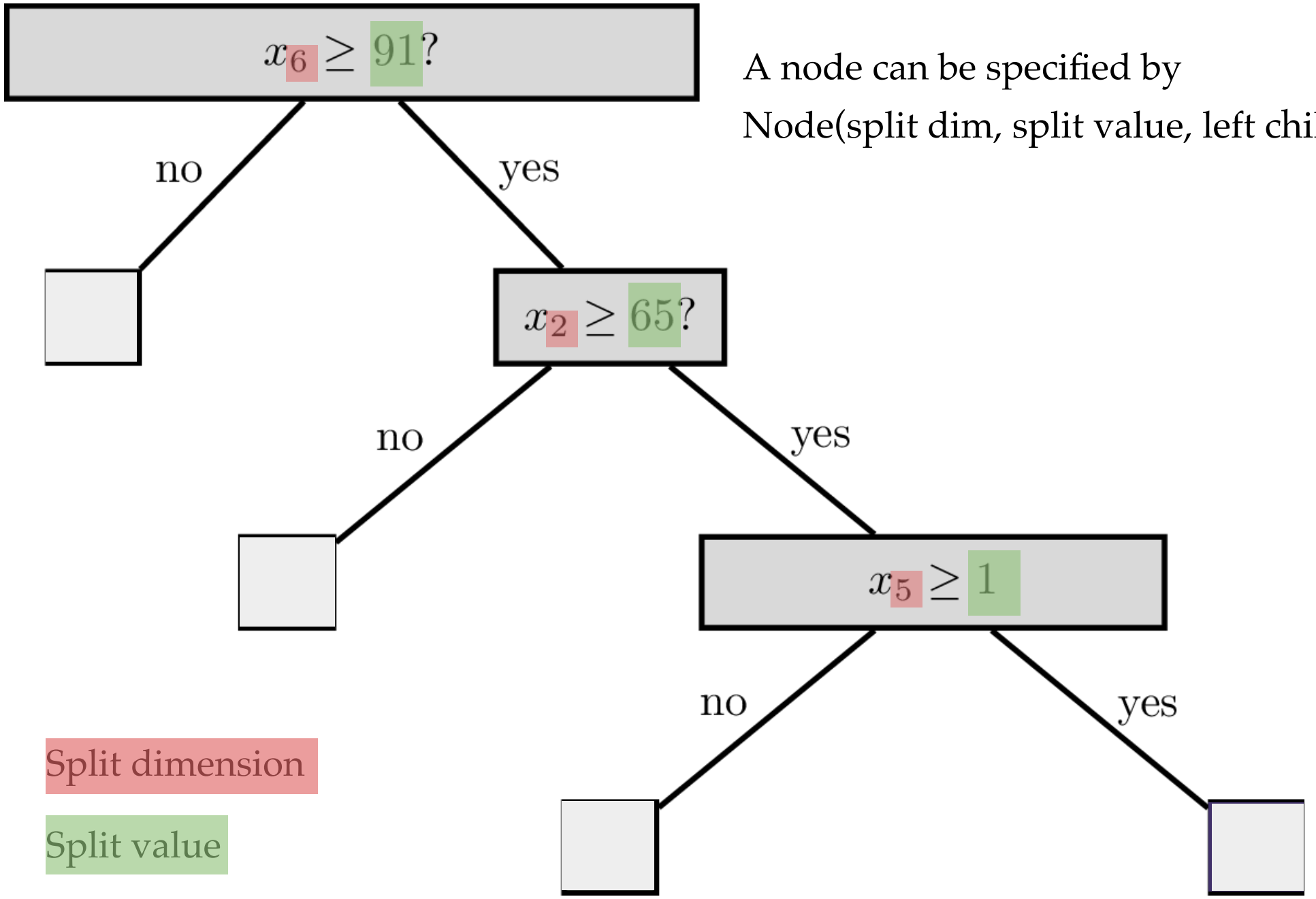
Decision tree terminologies




Node (root)

Node (internal)

Leaf (terminal)

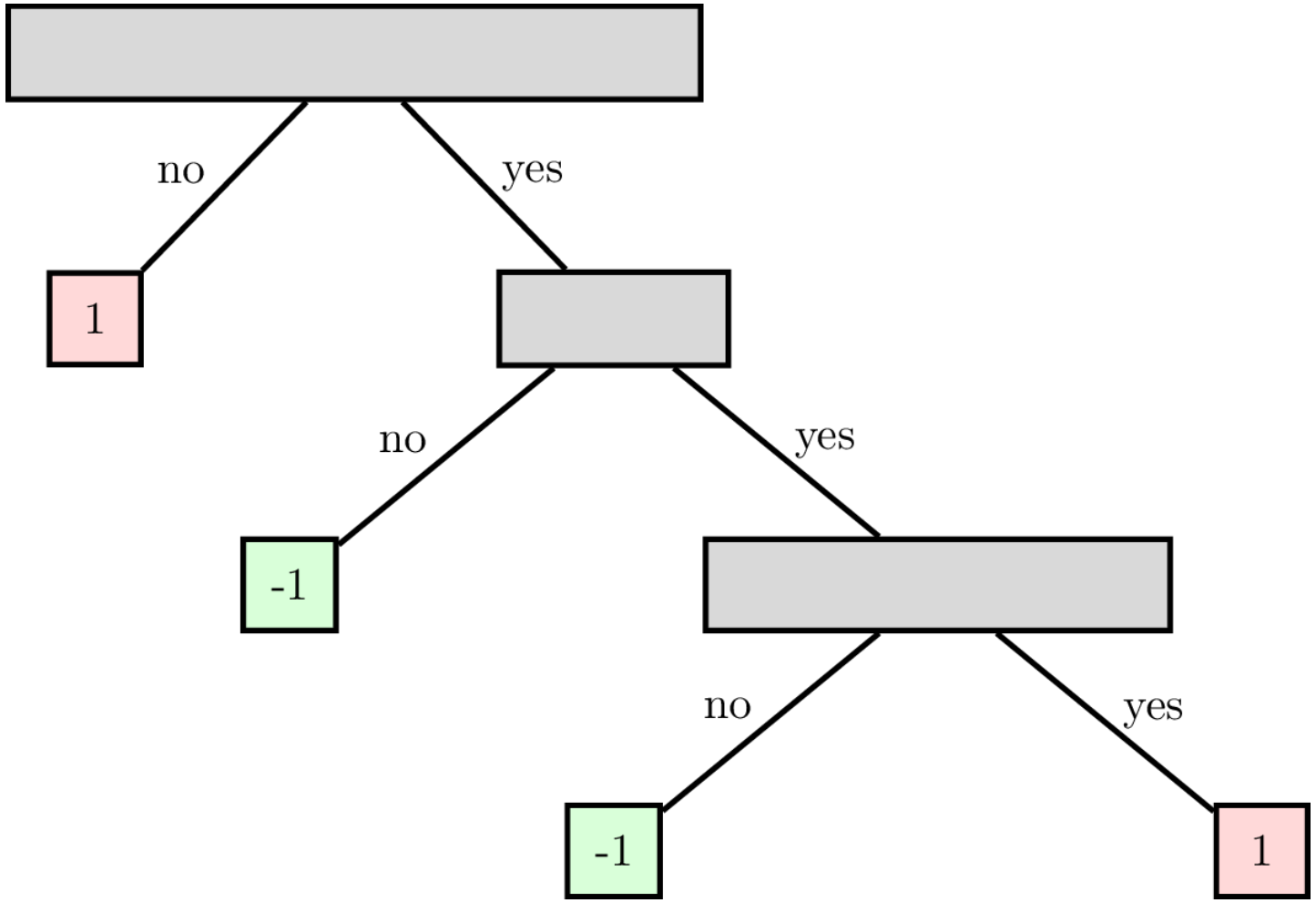



A node can be specified by

Node(split dim, split value, left child, right child)

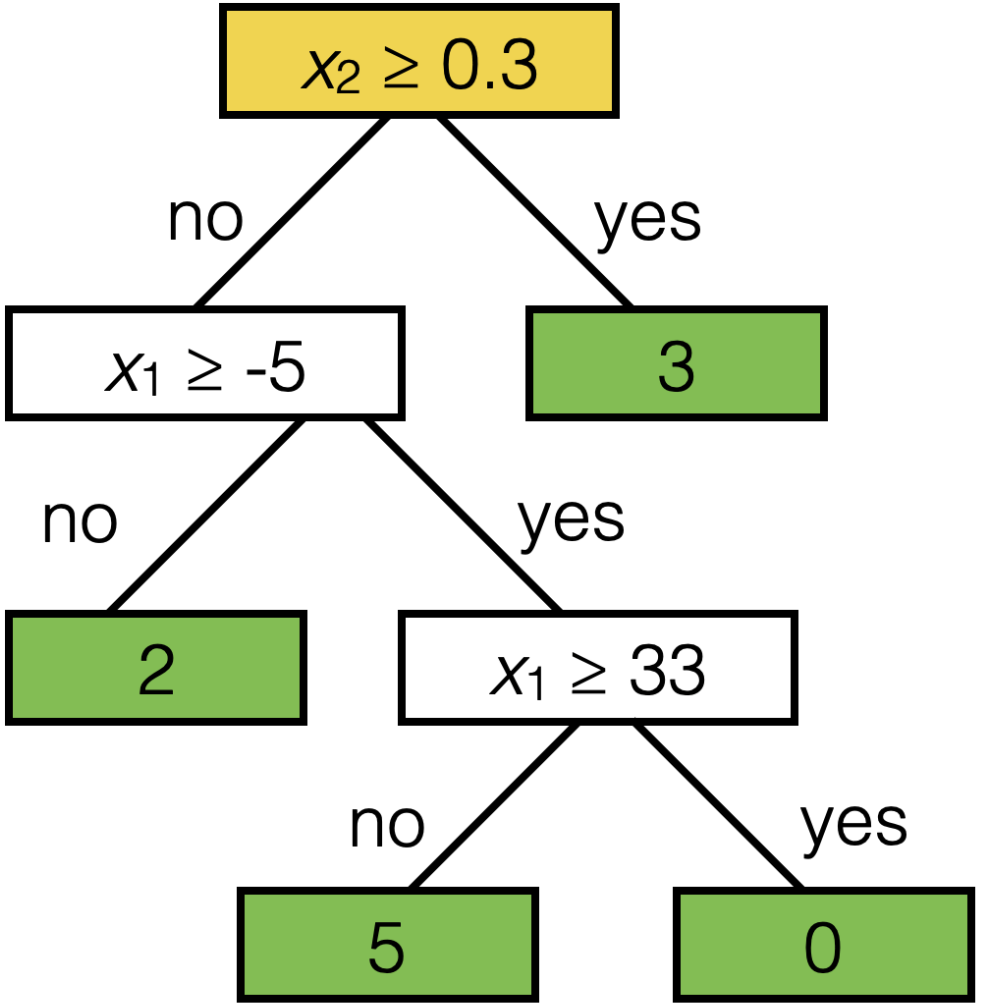
Split dimension

Split value




A leaf is specified by `Leaf(leaf_value)

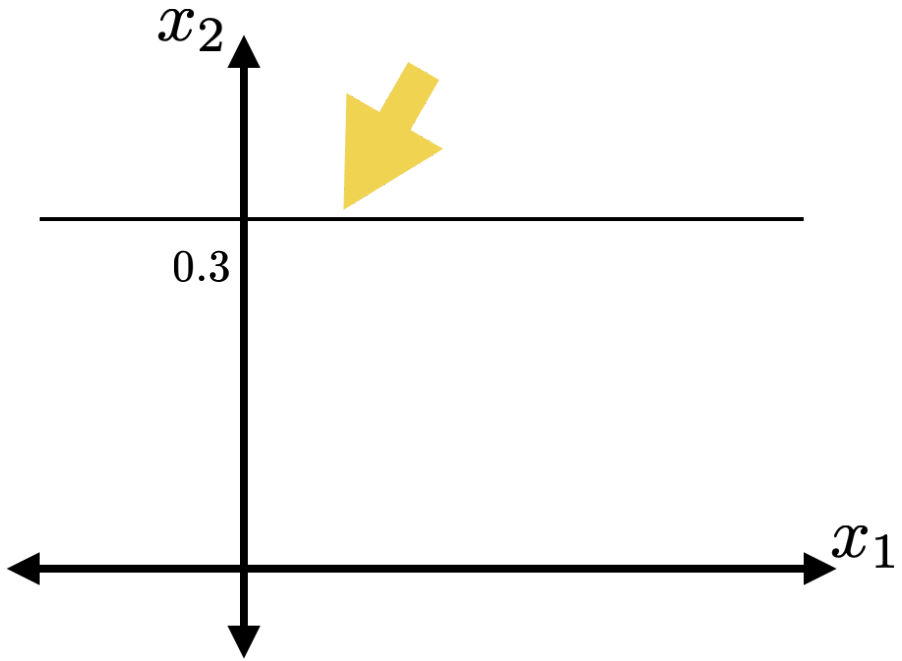
# Reading a regression decision tree

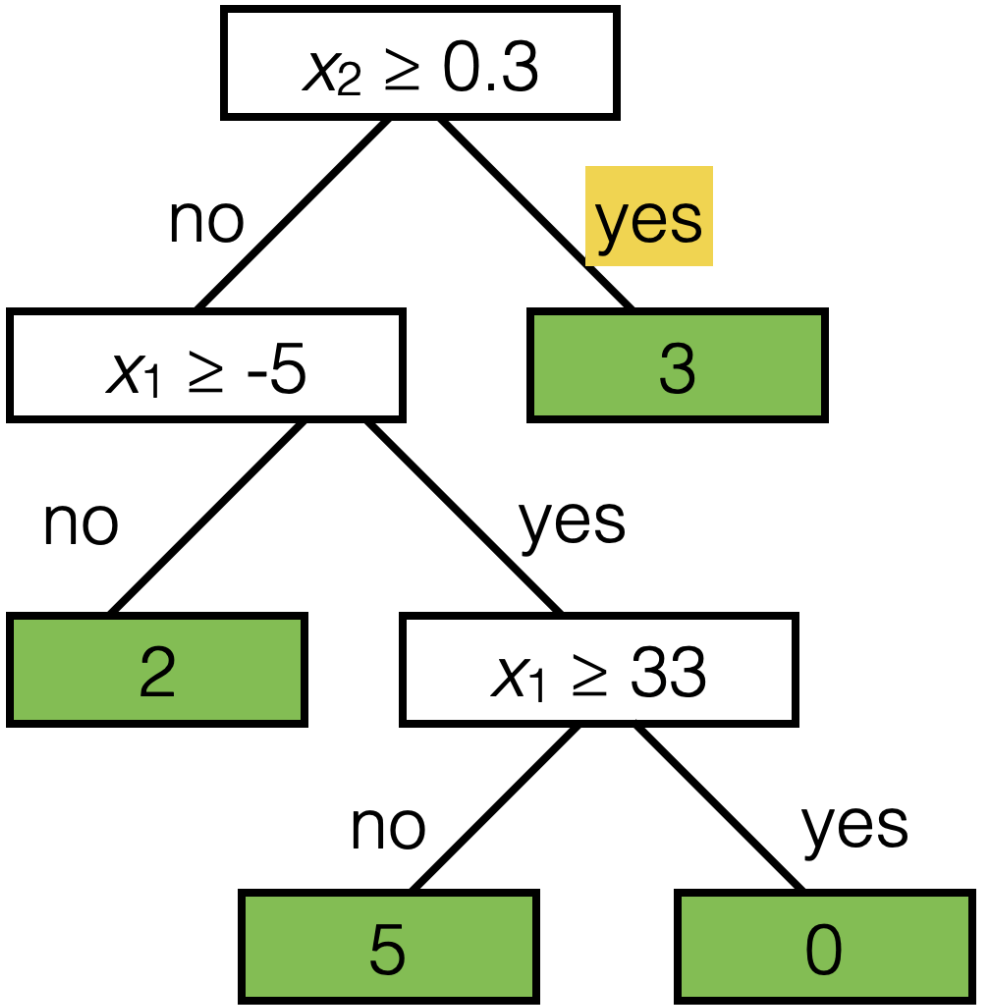


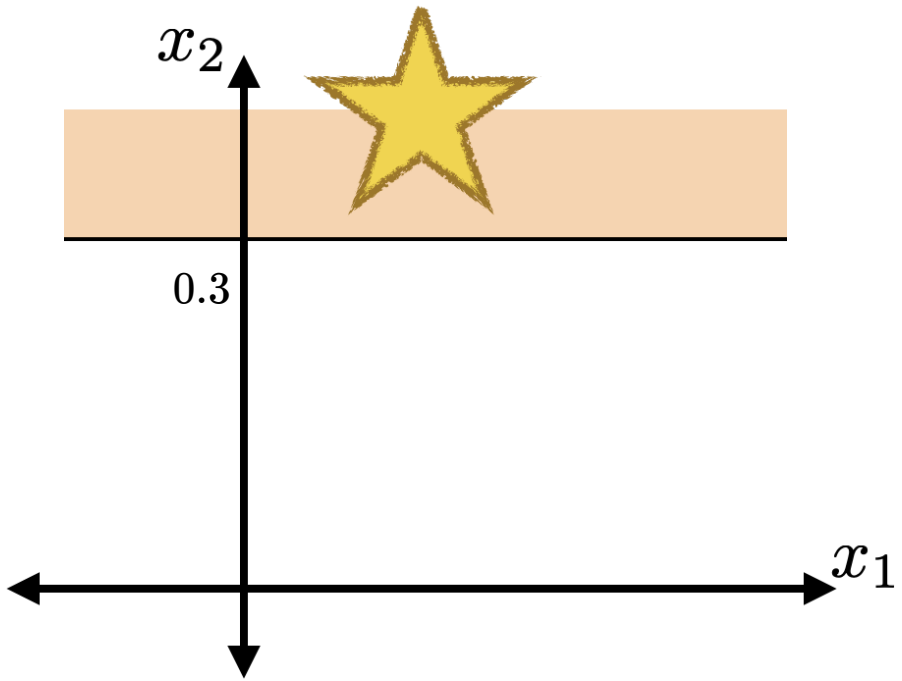

features:

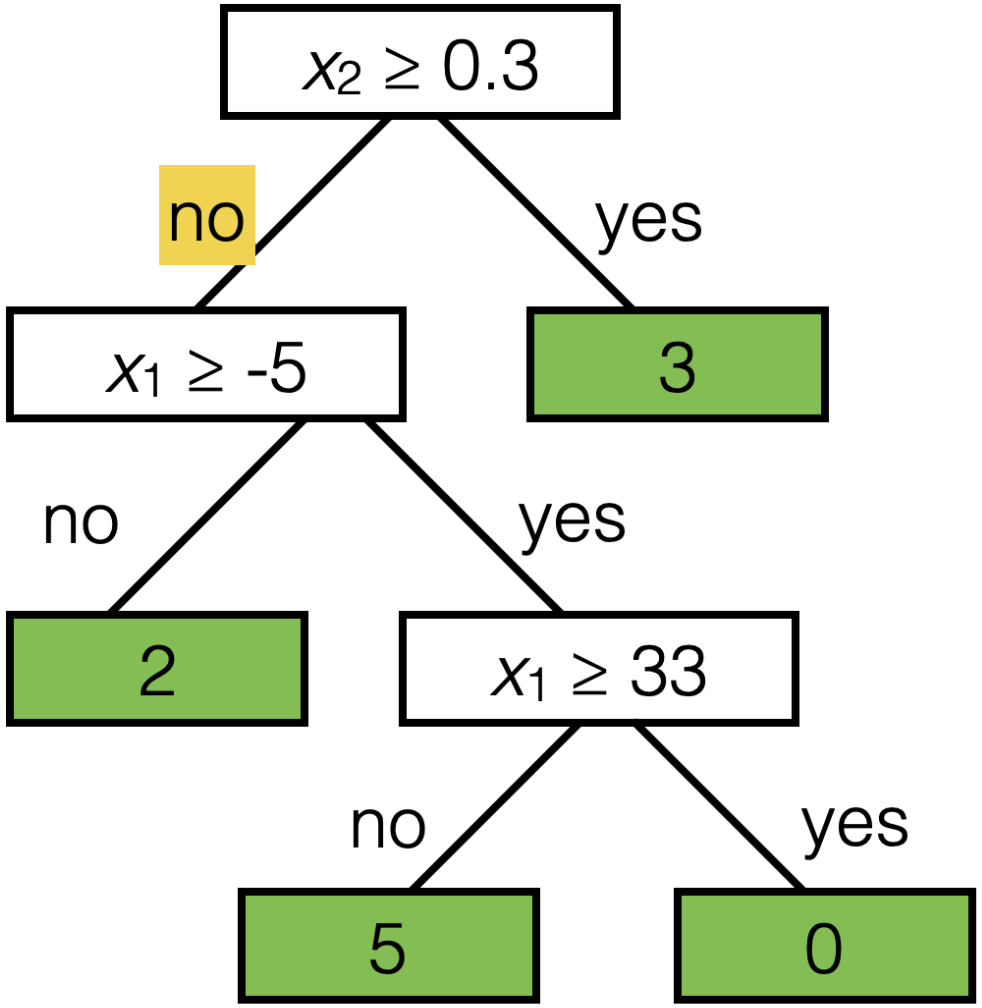
- $x_1$: temperature (deg C)
- $x_2$: precipitation (cm/hr)

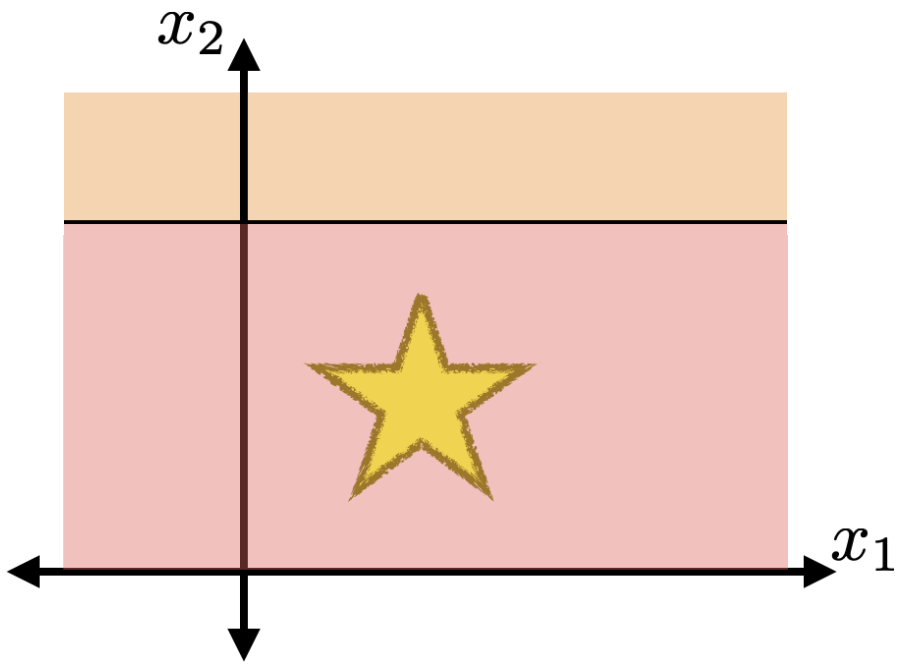
label: km run

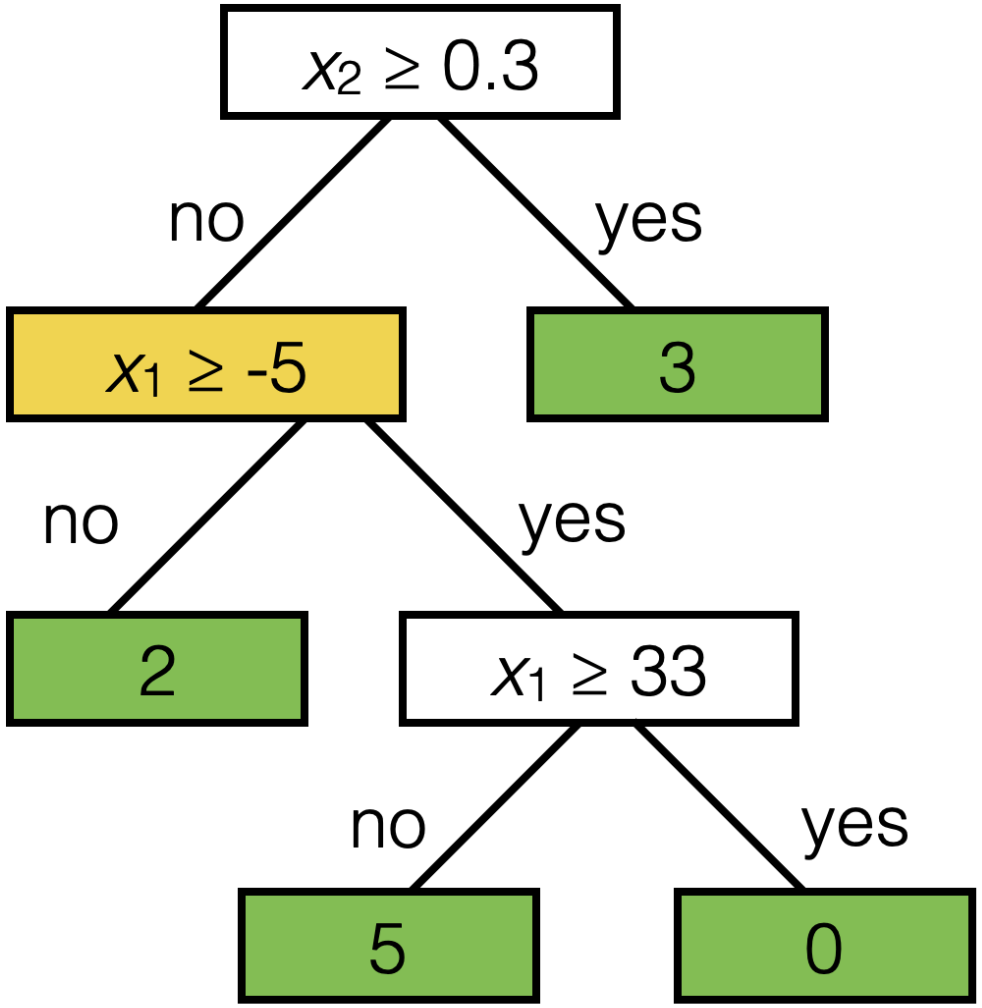


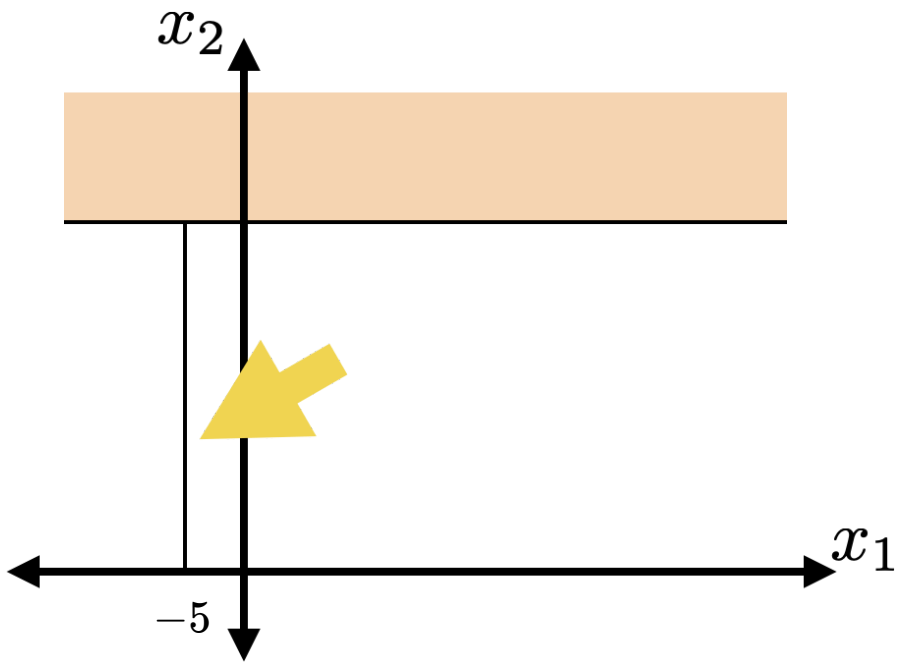

$x_2 \geq 0.3$
no
yes
$x_1 \geq -5$
3
no
yes
2
$x_1 \geq 33$
no
yes
5
0

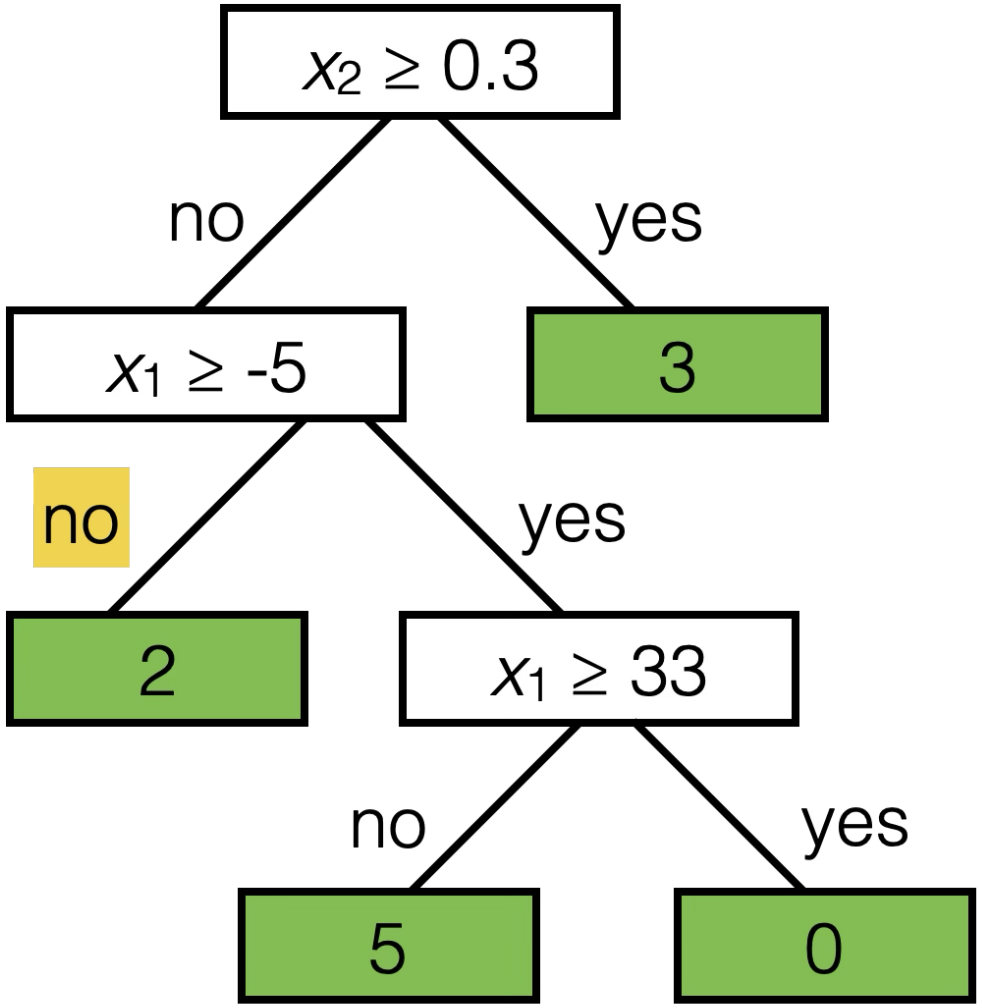

$x_2$
0.3
$x_1$

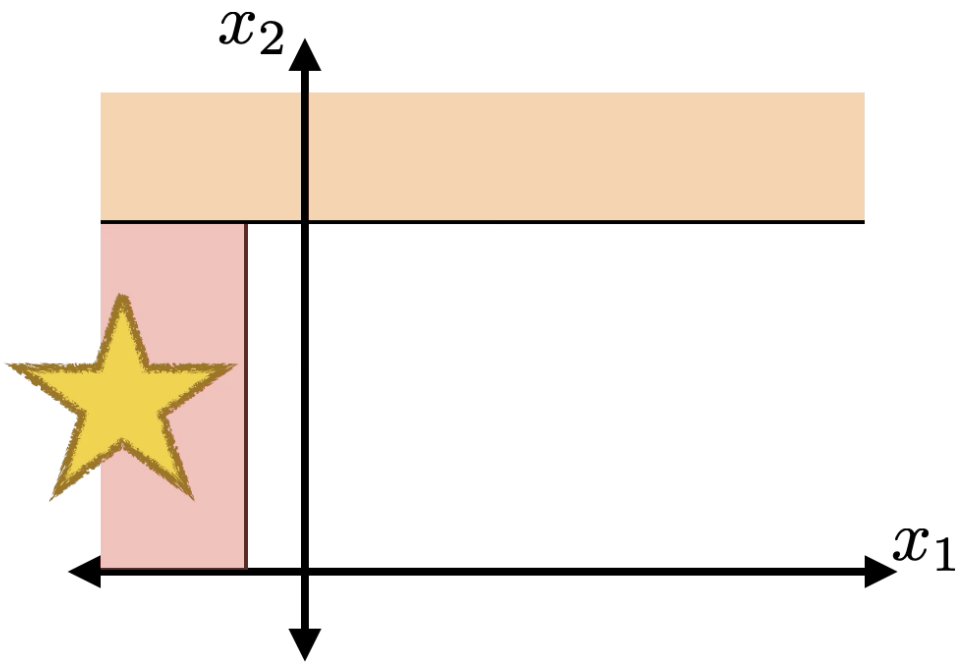

$x_2 \geq 0.3$
no
yes
$x_1 \geq -5$
3
no
yes
2
$x_1 \geq 33$
no
yes
5
0

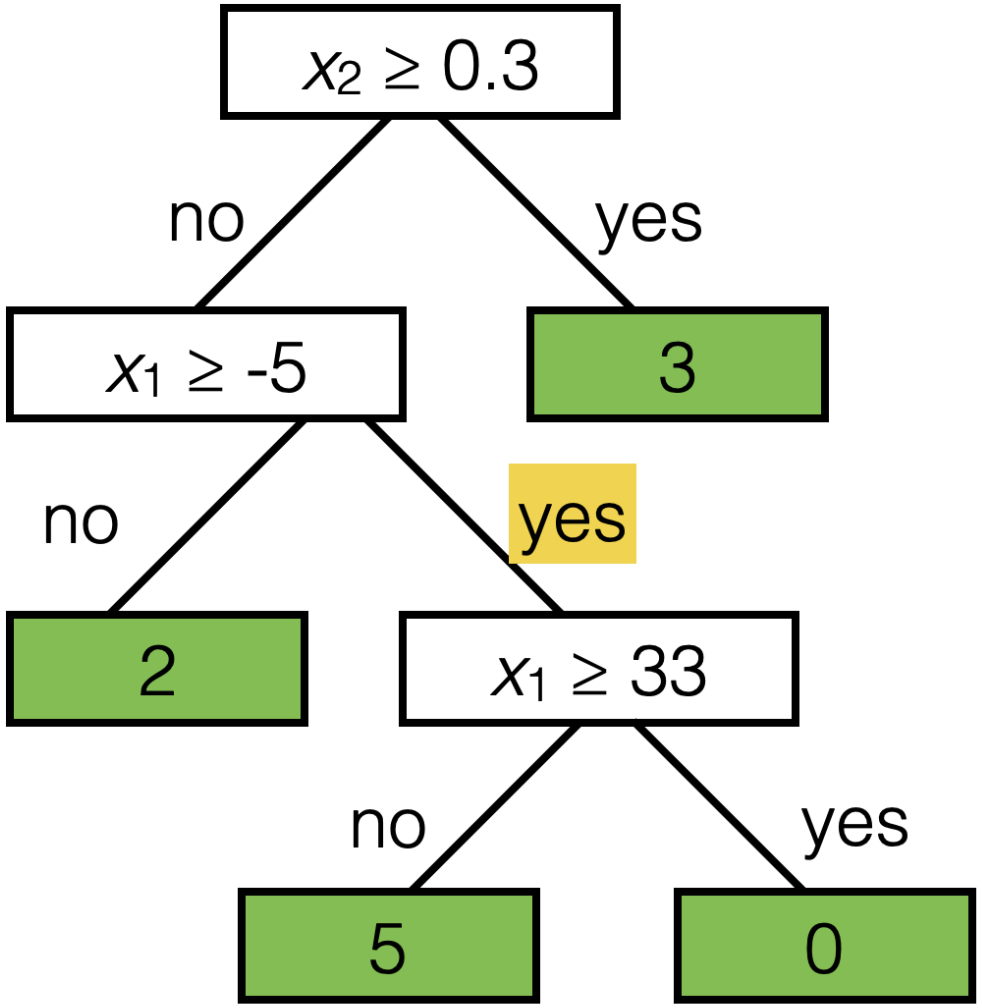

$x_2$
$x_1$

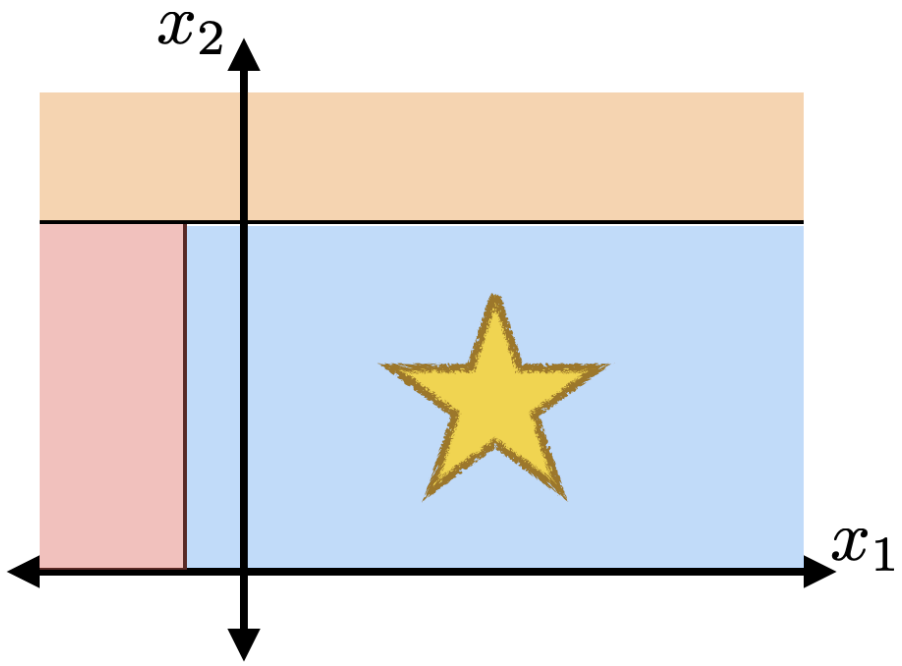

$x_2 \geq 0.3$
no
yes
$x_1 \geq -5$
3
no
yes
2
$x_1 \geq 33$
no
yes
5
0

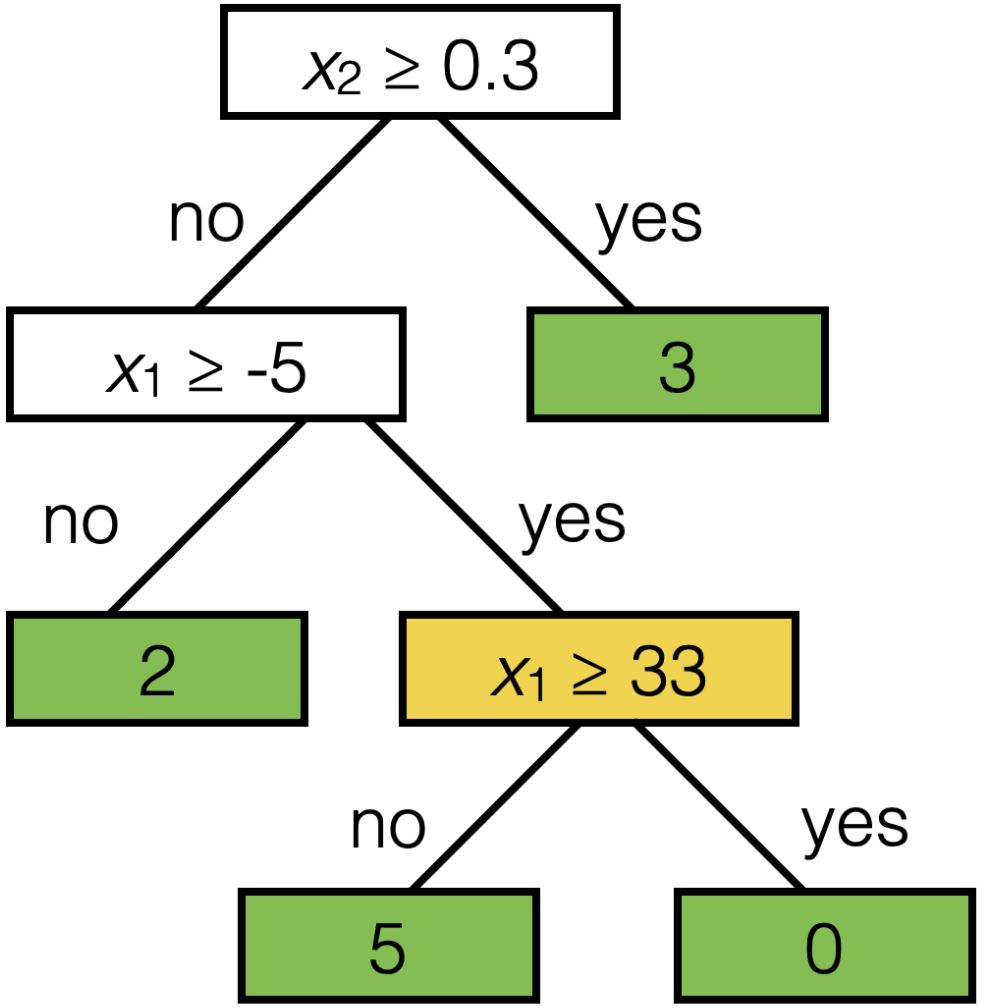

$x_2$
$x_1$
$-5$

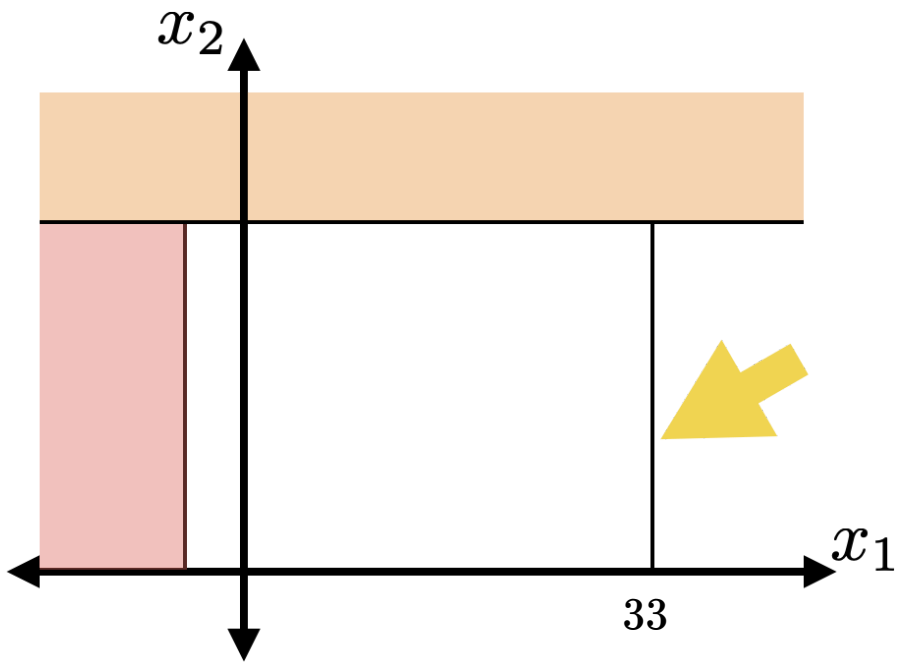

$x_2 \geq 0.3$
no
yes
$x_1 \geq -5$
3
no
yes
2
$x_1 \geq 33$
no
yes
5
0

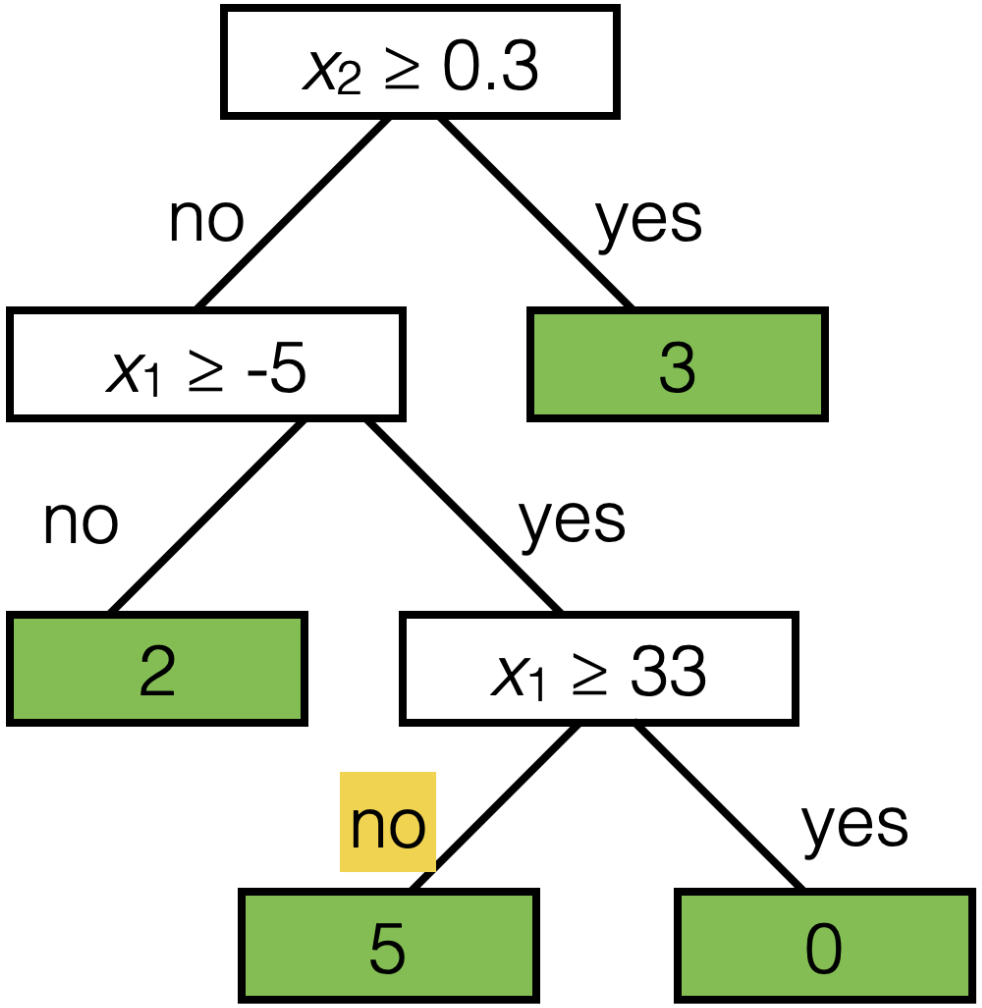

$x_2$
$x_1$

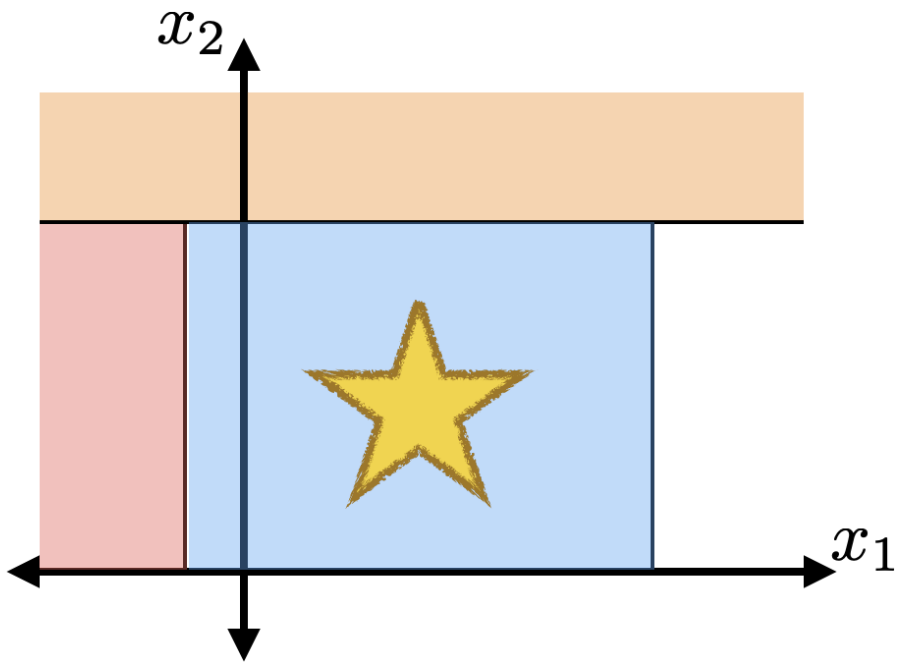

$x_2 \geq 0.3$
no
yes
$x_1 \geq -5$
3
no
yes
2
$x_1 \geq 33$
no
yes
5
0

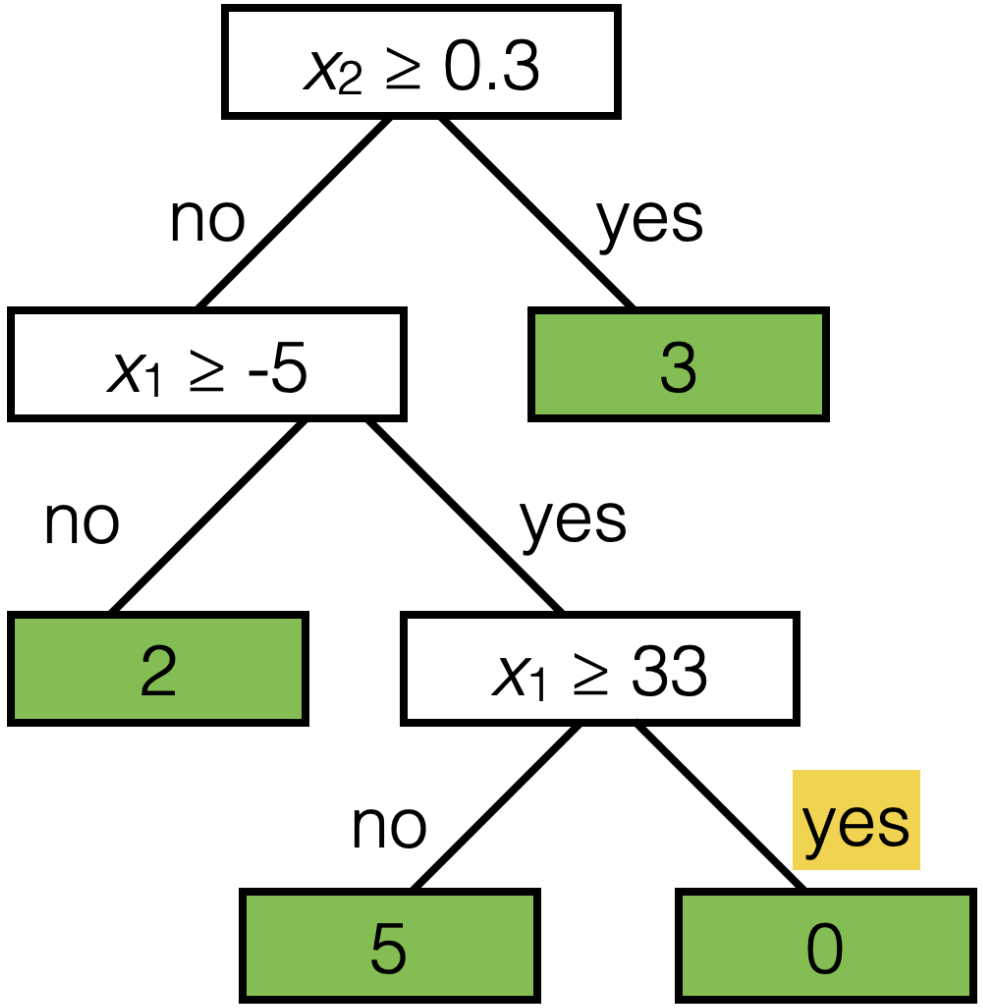

$x_2$
$x_1$

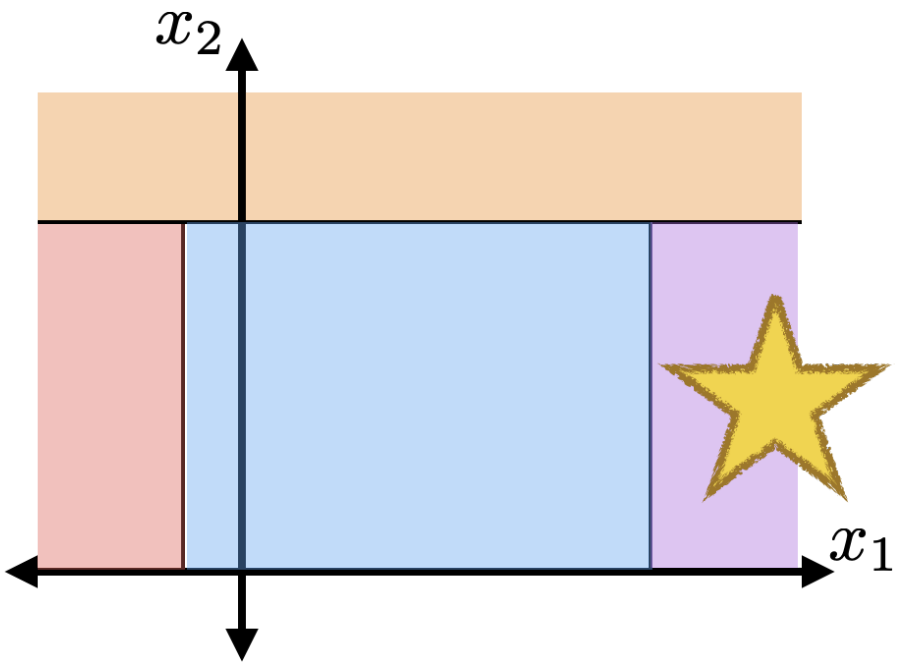

$x_2 \geq 0.3$
no
yes
$x_1 \geq -5$
3
no
yes
2
$x_1 \geq 33$
no
yes
5
0

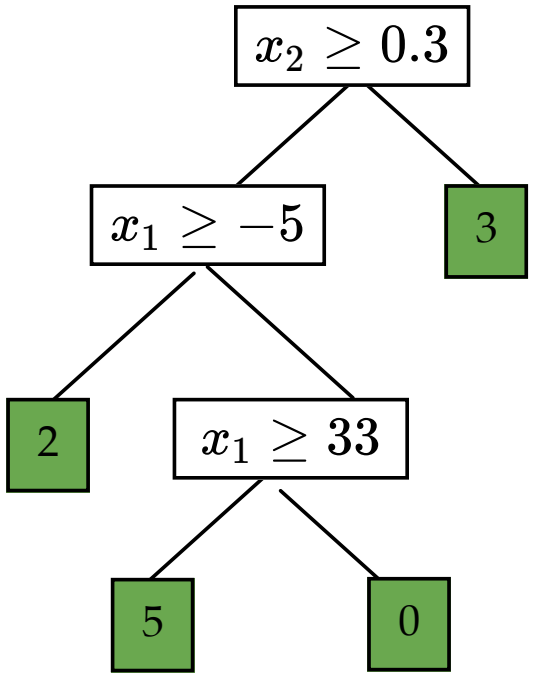

$x_2$
$x_1$
33


$x_2 \geq 0.3$
no
yes
$x_1 \geq -5$
3
no
yes
2
$x_1 \geq 33$
no
yes
5
0

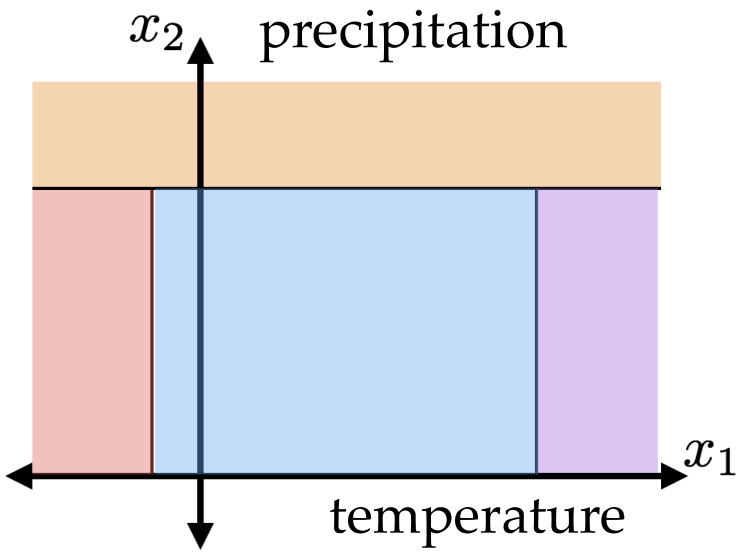

$x_2$
$x_1$


$x_2 \geq 0.3$
no
yes
$x_1 \geq -5$
3
no
yes
2
$x_1 \geq 33$
no
yes
5
0

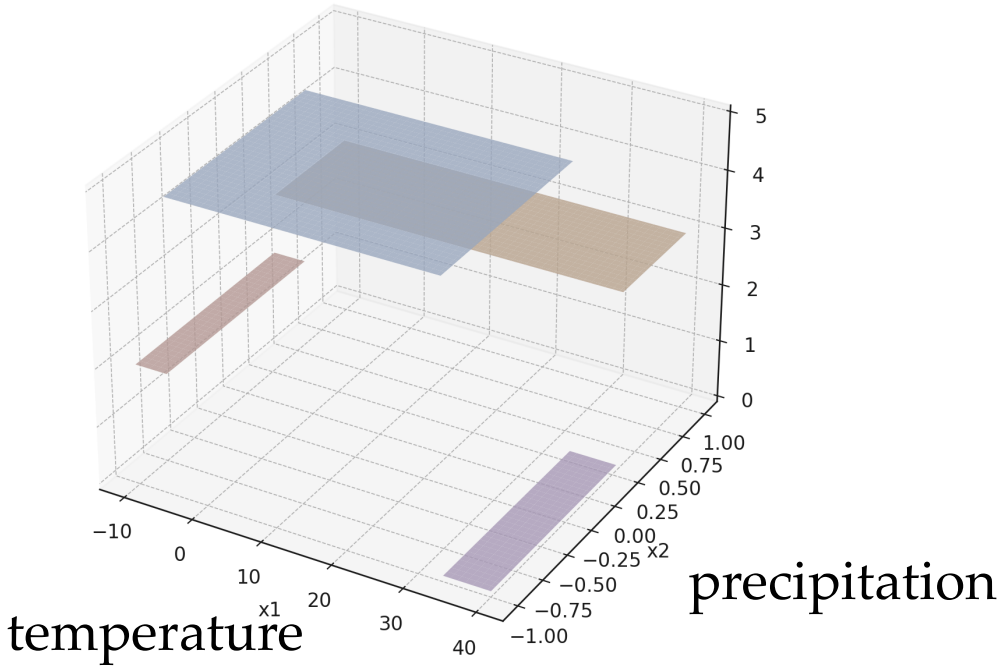

$x_2$
$x_1$




Each axis-aligned box in feature space shares one prediction




label: km run




precipitation

temperature

## 🌳 How to Grow a Regression Tree

The big idea:

1. Grow — start with all data at the root.
   Try candidate splits along each feature.
2. Split — pick the split that best reduces prediction error
   (lowest weighted variance of target values).
3. Stop — if a region is small or already consistent
   (data within ≤ leaf size, or variance below a threshold), make it a leaf.

recursive partitioning: each split divides data into regions where the target values are more similar, and each leaf predicts by the average of its data points.

set of indices

hyper-parameter: largest leaf size

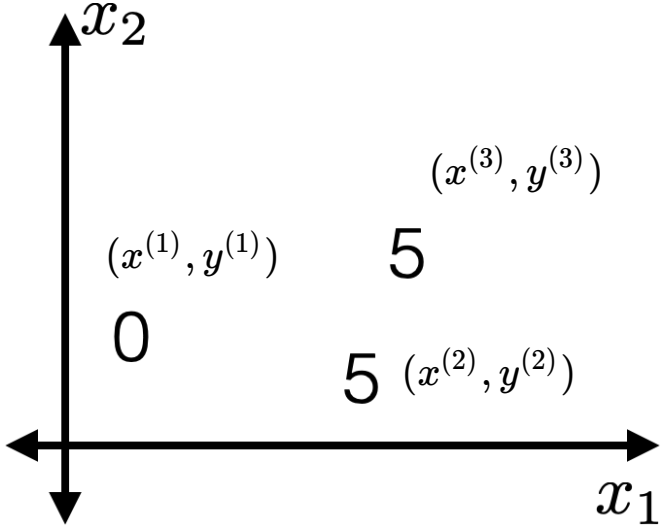
maximum number of training samples allowed in a leaf

$\text{BuildTree}(I, k, \mathcal{D})$

1. **if** $|I| > k$
2. **for** each split dim $j$ and split value $s$
3. Set $I_{j,s}^{+} = \left\{ i \in I \mid x_j^{(i)} \geq s \right\}$
4. Set $I_{j,s}^{-} = \left\{ i \in I \mid x_j^{(i)} < s \right\}$
5. Set $\hat{y}_{j,s}^{+} = \text{average}_{\ i \in I_{j,s}^{+}}\ y^{(i)}$
6. Set $\hat{y}_{j,s}^{-} = \text{average}_{\ i \in I_{j,s}^{-}}\ y^{(i)}$
7. Set $E_{j,s} = \sum_{i \in I_{j,s}^{+}} \left( y^{(i)} - \hat{y}_{j,s}^{+} \right)^2 + \sum_{i \in I_{j,s}^{-}} \left( y^{(i)} - \hat{y}_{j,s}^{-} \right)^2$
8. Set $(j^*, s^*) = \arg\min_{j,s} E_{j,s}$
9. **else**
10. Set $\hat{y} = \text{average}_{\ i \in I}\ y^{(i)}$
11. **return** Leaf(leave_value=$\hat{y}$)
12. **return** $\text{Node}\left(j^*, s^*, \text{BuildTree}\left(I_{j^*,s^*}^{-}, k, \mathcal{D}\right), \text{BuildTree}\left(I_{j^*,s^*}^{+}, k, \mathcal{D}\right)\right)$

BuildTree$(I, k, \mathcal{D})$

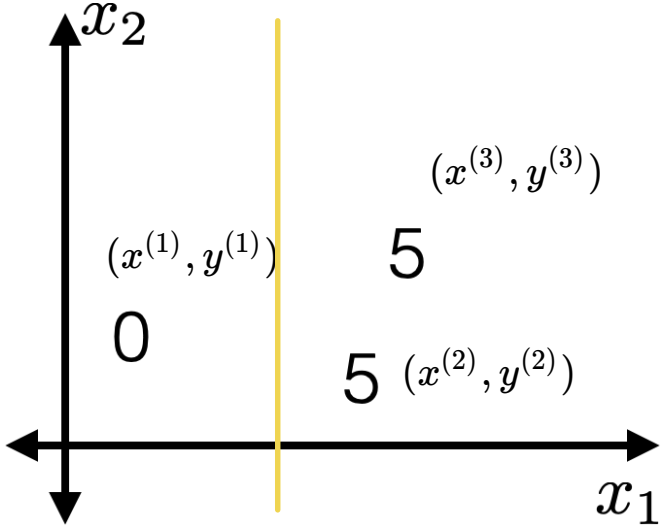
1. **if** $|I| > k$
2. **for** each split dim $j$ and split value $s$
3. Set $I_{j,s}^{+} = \left\{ i \in I \mid x_j^{(i)} \geq s \right\}$
4. Set $I_{j,s}^{-} = \left\{ i \in I \mid x_j^{(i)} < s \right\}$
5. Set $\hat{y}_{j,s}^{+} = \text{average}_{\, i \in I_{j,s}^{+}} \, y^{(i)}$
6. Set $\hat{y}_{j,s}^{-} = \text{average}_{\, i \in I_{j,s}^{-}} \, y^{(i)}$
7. Set $E_{j,s} = \sum_{i \in I_{j,s}^{+}} \left( y^{(i)} - \hat{y}_{j,s}^{+} \right)^2 + \sum_{i \in I_{j,s}^{-}} \left( y^{(i)} - \hat{y}_{j,s}^{-} \right)^2$
8. Set $(j^*, s^*) = \arg\min_{j,s} E_{j,s}$
9. **else**
10. Set $\hat{y} = \text{average}_{\, i \in I} \, y^{(i)}$
11. **return** Leaf(leave_value=$\hat{y}$)
12. **return** Node $\left( j^*, s^*, \text{BuildTree}\left( I_{j^*,s^*}^{-}, k \right), \text{BuildTree}\left( I_{j^*,s^*}^{+}, k \right) \right)$




- Choose $k = 2$
- BuildTree$(\{1, 2, 3\}; 2)$
- Line 1 true

BuildTree$(I, k, \mathcal{D})$

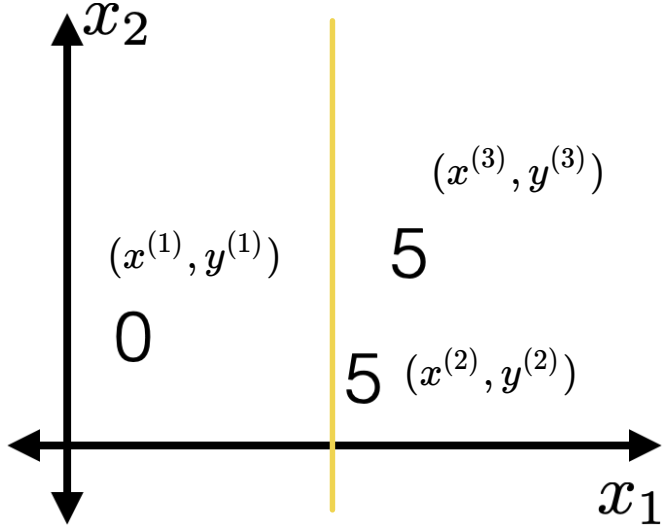
1. **if** $|I| > k$
2. **for** each split dim $j$ and split value $s$
3. Set $I_{j,s}^{+} = \left\{ i \in I \mid x_j^{(i)} \geq s \right\}$
4. Set $I_{j,s}^{-} = \left\{ i \in I \mid x_j^{(i)} < s \right\}$
5. Set $\hat{y}_{j,s}^{+} = \text{average}_{\, i \in I_{j,s}^{+}} \, y^{(i)}$
6. Set $\hat{y}_{j,s}^{-} = \text{average}_{\, i \in I_{j,s}^{-}} \, y^{(i)}$
7. Set $E_{j,s} = \sum_{i \in I_{j,s}^{+}} \left( y^{(i)} - \hat{y}_{j,s}^{+} \right)^2 + \sum_{i \in I_{j,s}^{-}} \left( y^{(i)} - \hat{y}_{j,s}^{-} \right)^2$
8. Set $(j^*, s^*) = \arg\min_{j,s} E_{j,s}$
9. **else**
10. Set $\hat{y} = \text{average}_{\, i \in I} \, y^{(i)}$
11. **return** Leaf(leave_value=$\hat{y}$)
12. **return** Node $\left( j^*, s^*, \text{BuildTree}\left( I_{j^*,s^*}^{-}, k \right), \text{BuildTree}\left( I_{j^*,s^*}^{+}, k \right) \right)$




- For this fixed $(j, s)$
  - $I_{j,s}^{+} = \{2, 3\}$
  - $I_{j,s}^{-} = \{1\}$
  - $\hat{y}_{j,s}^{+} = 5$
  - $\hat{y}_{j,s}^{-} = 0$
  - $E_{j,s} = 0$

BuildTree$(I, k, \mathcal{D})$

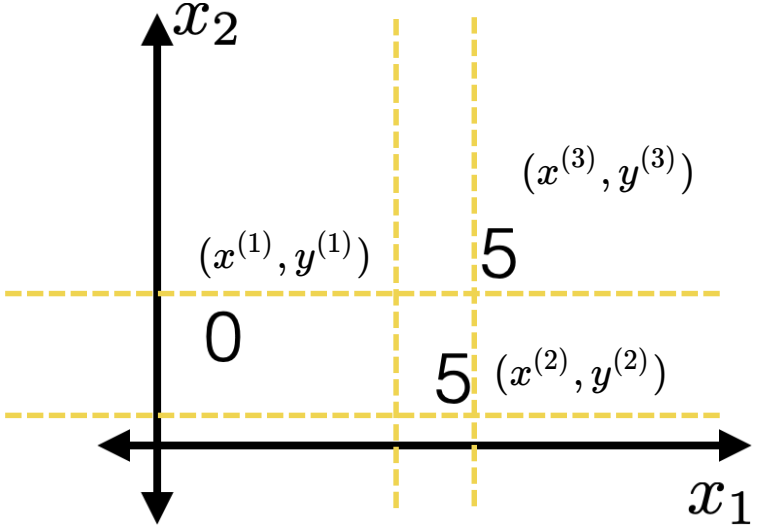
1. **if** $|I| > k$
2. **for** each split dim $j$ and split value $s$
3. Set $I_{j,s}^{+} = \left\{ i \in I \mid x_j^{(i)} \geq s \right\}$
4. Set $I_{j,s}^{-} = \left\{ i \in I \mid x_j^{(i)} < s \right\}$
5. Set $\hat{y}_{j,s}^{+} = \text{average}_{\ i \in I_{j,s}^{+}}\ y^{(i)}$
6. Set $\hat{y}_{j,s}^{-} = \text{average}_{\ i \in I_{j,s}^{-}}\ y^{(i)}$
7. Set $E_{j,s} = \sum_{i \in I_{j,s}^{+}} \left( y^{(i)} - \hat{y}_{j,s}^{+} \right)^2 + \sum_{i \in I_{j,s}^{-}} \left( y^{(i)} - \hat{y}_{j,s}^{-} \right)^2$
8. Set $(j^*, s^*) = \arg\min_{j,s} E_{j,s}$
9. **else**
10. Set $\hat{y} = \text{average}_{\ i \in I}\ y^{(i)}$
11. **return** Leaf(leave_value=$\hat{y}$)
12. **return** Node $\left( j^*, s^*, \text{BuildTree}\left( I_{j^*,s^*}^{-}, k \right), \text{BuildTree}\left( I_{j^*,s^*}^{+}, k \right) \right)$




- For this fixed $(j, s)$
  - $I_{j,s}^{+} = \{2, 3\}$
  - $I_{j,s}^{-} = \{1\}$
  - $\hat{y}_{j,s}^{+} = 5$
  - $\hat{y}_{j,s}^{-} = 0$
  - $E_{j,s} = 0$

BuildTree$(I, k, \mathcal{D})$

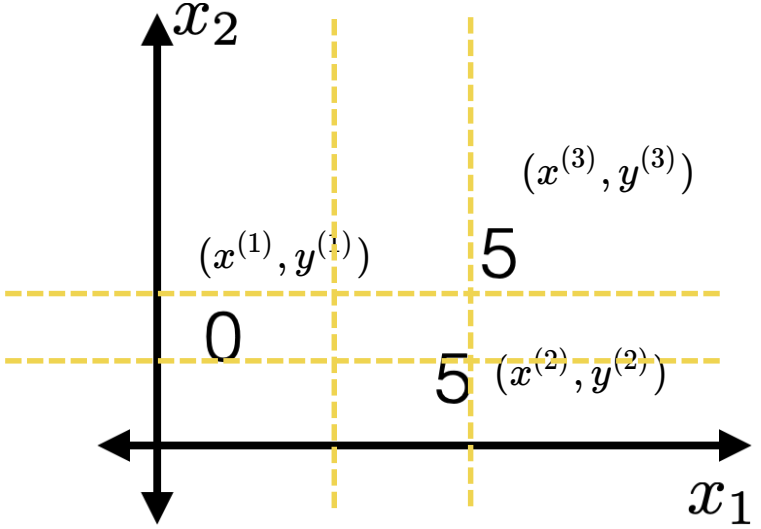
1. **if** $|I| > k$
2. **for** each split dim $j$ and split value $s$
3. Set $I^+_{j,s} = \left\{ i \in I \mid x^{(i)}_j \geq s \right\}$
4. Set $I^-_{j,s} = \left\{ i \in I \mid x^{(i)}_j < s \right\}$
5. Set $\hat{y}^+_{j,s} = \text{average}_{\ i \in I^+_{j,s}}\ y^{(i)}$
6. Set $\hat{y}^-_{j,s} = \text{average}_{\ i \in I^-_{j,s}}\ y^{(i)}$
7. Set $E_{j,s} = \sum_{i \in I^+_{j,s}} \left( y^{(i)} - \hat{y}^+_{j,s} \right)^2 + \sum_{i \in I^-_{j,s}} \left( y^{(i)} - \hat{y}^-_{j,s} \right)^2$
8. Set $(j^*, s^*) = \arg\min_{j,s} E_{j,s}$
9. **else**
10. Set $\hat{y} = \text{average}_{\ i \in I}\ y^{(i)}$
11. **return** Leaf(leave_value=$\hat{y}$)
12. **return** Node $\left(j^*, s^*, \text{BuildTree}\left(I^-_{j^*,s^*}, k\right), \text{BuildTree}\left(I^+_{j^*,s^*}, k\right)\right)$




- Line 2: It suffices to consider a finite number of $(j, s)$ combo suffices (those that split in-between data points)

BuildTree$(I, k, \mathcal{D})$

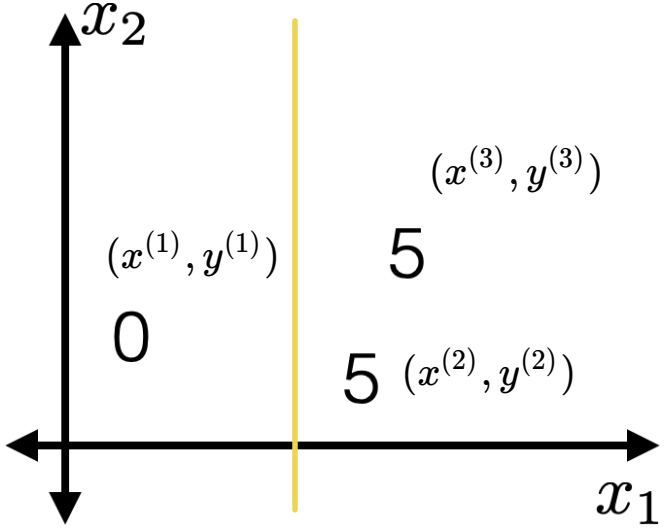
1. **if** $|I| > k$
2. **for** each split dim $j$ and split value $s$
3. Set $I^+_{j,s} = \left\{ i \in I \mid x^{(i)}_j \geq s \right\}$
4. Set $I^-_{j,s} = \left\{ i \in I \mid x^{(i)}_j < s \right\}$
5. Set $\hat{y}^+_{j,s} = \text{average}_{\ i \in I^+_{j,s}}\ y^{(i)}$
6. Set $\hat{y}^-_{j,s} = \text{average}_{\ i \in I^-_{j,s}}\ y^{(i)}$
7. Set $E_{j,s} = \sum_{i \in I^+_{j,s}} \left( y^{(i)} - \hat{y}^+_{j,s} \right)^2 + \sum_{i \in I^-_{j,s}} \left( y^{(i)} - \hat{y}^-_{j,s} \right)^2$
8. Set $(j^*, s^*) = \arg\min_{j,s} E_{j,s}$
9. **else**
10. Set $\hat{y} = \text{average}_{\ i \in I}\ y^{(i)}$
11. **return** Leaf(leave_value=$\hat{y}$)
12. **return** Node $\left(j^*, s^*, \text{BuildTree}\left(I^-_{j^*,s^*}, k\right), \text{BuildTree}\left(I^+_{j^*,s^*}, k\right)\right)$




- Line 8: picks the "best" among these finite choices of $(j, s)$ combos (random tie-breaking).

BuildTree$(I, k, \mathcal{D})$

1. **if** $|I| > k$
2. **for** each split dim $j$ and split value $s$
3. Set $I^+_{j,s} = \left\{ i \in I \mid x^{(i)}_j \geq s \right\}$
4. Set $I^-_{j,s} = \left\{ i \in I \mid x^{(i)}_j < s \right\}$
5. Set $\hat{y}^+_{j,s} = \text{average}_{\ i \in I^+_{j,s}}\ y^{(i)}$
6. Set $\hat{y}^-_{j,s} = \text{average}_{\ i \in I^-_{j,s}}\ y^{(i)}$
7. Set $E_{j,s} = \sum_{i \in I^+_{j,s}} \left( y^{(i)} - \hat{y}^+_{j,s} \right)^2 + \sum_{i \in I^-_{j,s}} \left( y^{(i)} - \hat{y}^-_{j,s} \right)^2$
8. Set $(j^*, s^*) = \arg\min_{j,s} E_{j,s}$
9. **else**
10. Set $\hat{y} = \text{average}_{\ i \in I}\ y^{(i)}$
11. **return** Leaf(leave_value=$\hat{y}$)
12. **return** Node $\left(j^*, s^*, \text{BuildTree}\left(I^-_{j^*,s^*}, k\right), \text{BuildTree}\left(I^+_{j^*,s^*}, k\right)\right)$

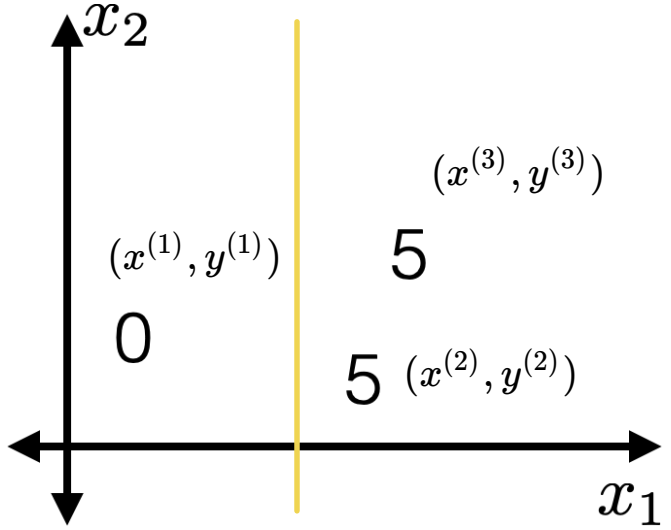



Suppose line 8 sets this $(j^*, s^*)$,

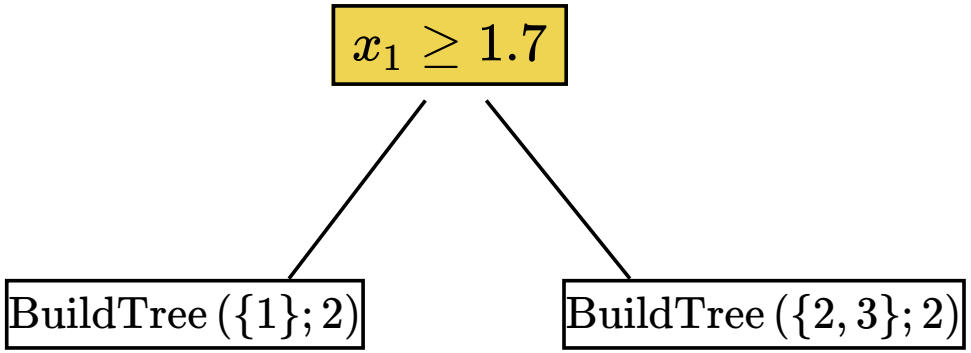
say $= (j^*, s^*) = (1, 1.7)$

BuildTree$(I, k, \mathcal{D})$

1. **if** $|I| > k$
2. **for** each split dim $j$ and split value $s$
3. Set $I^+_{j,s} = \left\{ i \in I \mid x^{(i)}_j \geq s \right\}$
4. Set $I^-_{j,s} = \left\{ i \in I \mid x^{(i)}_j < s \right\}$
5. Set $\hat{y}^+_{j,s} = \text{average}_{\ i \in I^+_{j,s}}\ y^{(i)}$
6. Set $\hat{y}^-_{j,s} = \text{average}_{\ i \in I^-_{j,s}}\ y^{(i)}$
7. Set $E_{j,s} = \sum_{i \in I^+_{j,s}} \left( y^{(i)} - \hat{y}^+_{j,s} \right)^2 + \sum_{i \in I^-_{j,s}} \left( y^{(i)} - \hat{y}^-_{j,s} \right)^2$
8. Set $(j^*, s^*) = \arg\min_{j,s} E_{j,s}$
9. **else**
10. Set $\hat{y} = \text{average}_{\ i \in I}\ y^{(i)}$
11. **return** Leaf(leave_value=$\hat{y}$)
12. **return** Node $\left(j^*, s^*, \text{BuildTree}\left(I^-_{j^*,s^*}, k\right), \text{BuildTree}\left(I^+_{j^*,s^*}, k\right)\right)$

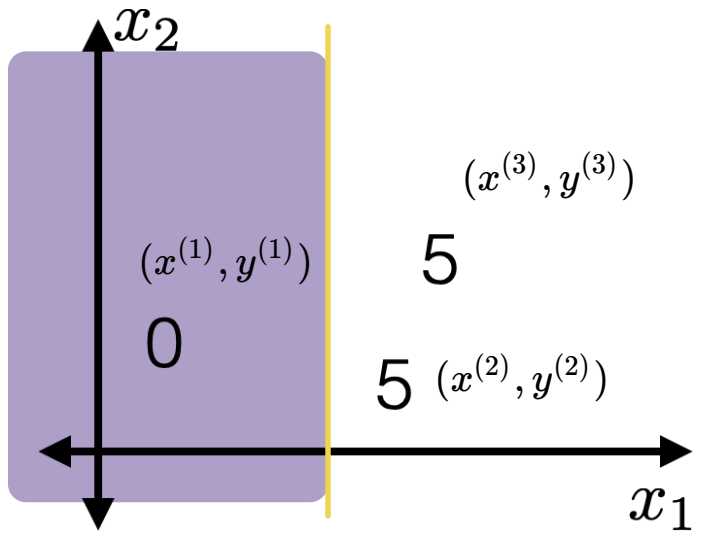


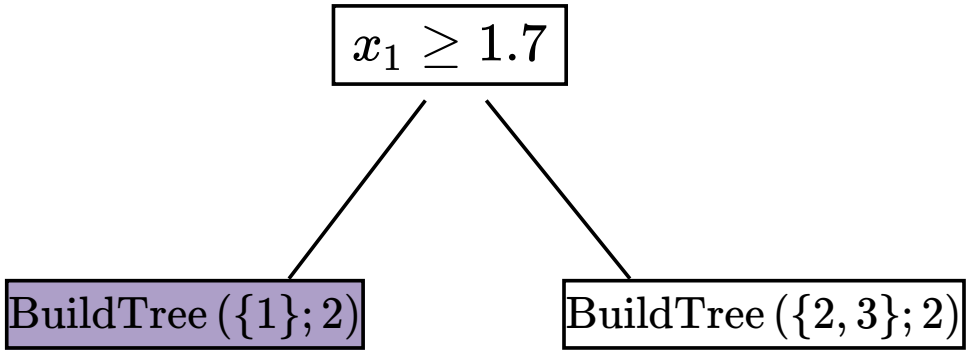

Line 12 recursion



BuildTree$(I, k, \mathcal{D})$

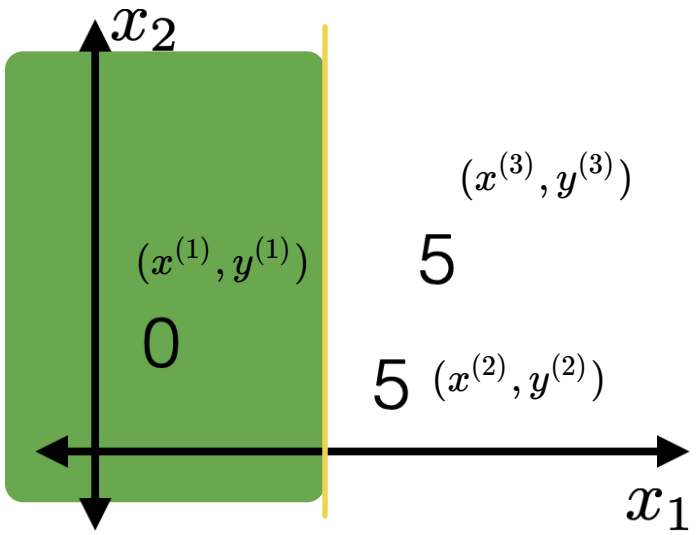
1. **if** $|I| > k$
2. **for** each split dim $j$ and split value $s$
3. Set $I^+_{j,s} = \left\{ i \in I \mid x^{(i)}_j \geq s \right\}$
4. Set $I^-_{j,s} = \left\{ i \in I \mid x^{(i)}_j < s \right\}$
5. Set $\hat{y}^+_{j,s} = \text{average}_{\ i \in I^+_{j,s}}\ y^{(i)}$
6. Set $\hat{y}^-_{j,s} = \text{average}_{\ i \in I^-_{j,s}}\ y^{(i)}$
7. Set $E_{j,s} = \sum_{i \in I^+_{j,s}} \left( y^{(i)} - \hat{y}^+_{j,s} \right)^2 + \sum_{i \in I^-_{j,s}} \left( y^{(i)} - \hat{y}^-_{j,s} \right)^2$
8. Set $(j^*, s^*) = \arg\min_{j,s} E_{j,s}$
9. **else**
10. Set $\hat{y} = \text{average}_{\ i \in I}\ y^{(i)}$
11. **return** Leaf(leave_value=$\hat{y}$)
12. **return** Node $\left(j^*, s^*, \text{BuildTree}\left(I^-_{j^*,s^*}, k\right), \text{BuildTree}\left(I^+_{j^*,s^*}, k\right)\right)$

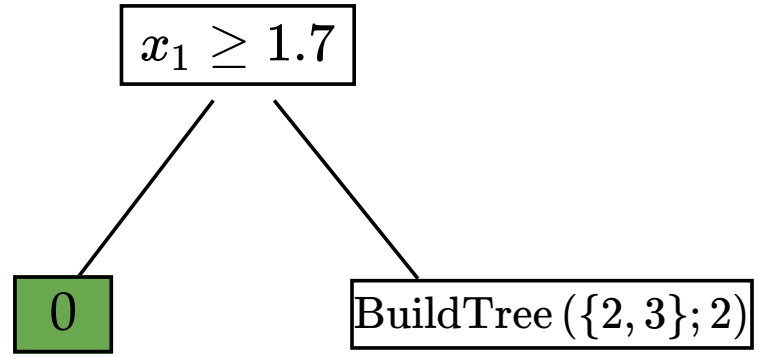


BuildTree$(I, k, \mathcal{D})$

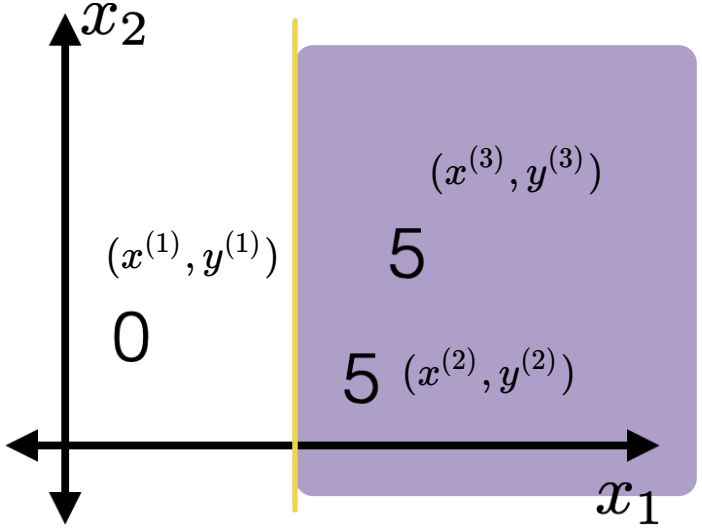
1. **if** $|I| > k$
2. **for** each split dim $j$ and split value $s$
3. Set $I^+_{j,s} = \left\{ i \in I \mid x^{(i)}_j \geq s \right\}$
4. Set $I^-_{j,s} = \left\{ i \in I \mid x^{(i)}_j < s \right\}$
5. Set $\hat{y}^+_{j,s} = \text{average}_{\ i \in I^+_{j,s}}\ y^{(i)}$
6. Set $\hat{y}^-_{j,s} = \text{average}_{\ i \in I^-_{j,s}}\ y^{(i)}$
7. Set $E_{j,s} = \sum_{i \in I^+_{j,s}} \left( y^{(i)} - \hat{y}^+_{j,s} \right)^2 + \sum_{i \in I^-_{j,s}} \left( y^{(i)} - \hat{y}^-_{j,s} \right)^2$
8. Set $(j^*, s^*) = \arg\min_{j,s} E_{j,s}$
9. **else**
10. Set $\hat{y} = \text{average}_{\ i \in I}\ y^{(i)}$
11. **return** Leaf(leave_value=$\hat{y}$)
12. **return** Node $\left(j^*, s^*, \text{BuildTree}\left(I^-_{j^*,s^*}, k\right), \text{BuildTree}\left(I^+_{j^*,s^*}, k\right)\right)$

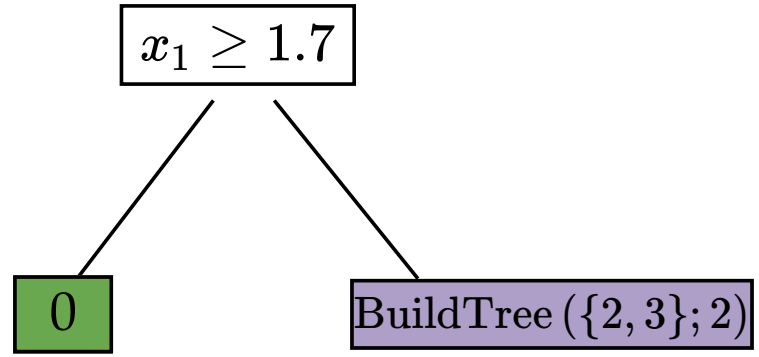


BuildTree$(I, k, \mathcal{D})$

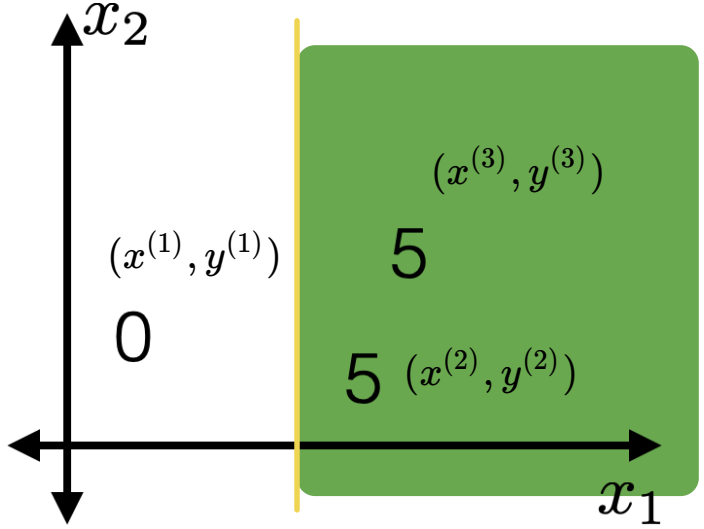
1. **if** $|I| > k$
2. **for** each split dim $j$ and split value $s$
3. Set $I^+_{j,s} = \left\{ i \in I \mid x^{(i)}_j \geq s \right\}$
4. Set $I^-_{j,s} = \left\{ i \in I \mid x^{(i)}_j < s \right\}$
5. Set $\hat{y}^+_{j,s} = \text{average}_{\ i \in I^+_{j,s}}\ y^{(i)}$
6. Set $\hat{y}^-_{j,s} = \text{average}_{\ i \in I^-_{j,s}}\ y^{(i)}$
7. Set $E_{j,s} = \sum_{i \in I^+_{j,s}} \left( y^{(i)} - \hat{y}^+_{j,s} \right)^2 + \sum_{i \in I^-_{j,s}} \left( y^{(i)} - \hat{y}^-_{j,s} \right)^2$
8. Set $(j^*, s^*) = \arg\min_{j,s} E_{j,s}$
9. **else**
10. Set $\hat{y} = \text{average}_{\ i \in I}\ y^{(i)}$
11. **return** Leaf(leave_value=$\hat{y}$)
12. **return** Node $\left(j^*, s^*, \text{BuildTree}\left(I^-_{j^*,s^*}, k\right), \text{BuildTree}\left(I^+_{j^*,s^*}, k\right)\right)$

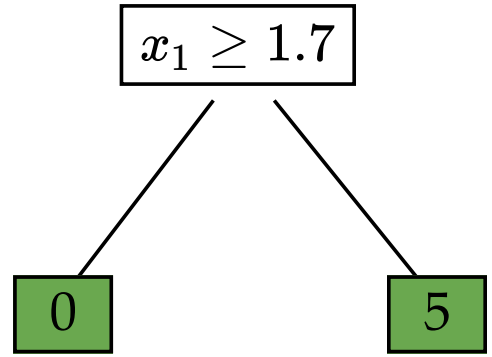


BuildTree$(I, k, \mathcal{D})$

1. **if** $|I| > k$
2. **for** each split dim $j$ and split value $s$
3. Set $I^+_{j,s} = \left\{ i \in I \mid x^{(i)}_j \geq s \right\}$
4. Set $I^-_{j,s} = \left\{ i \in I \mid x^{(i)}_j < s \right\}$
5. Set $\hat{y}^+_{j,s} = \text{average}_{\ i \in I^+_{j,s}}\ y^{(i)}$
6. Set $\hat{y}^-_{j,s} = \text{average}_{\ i \in I^-_{j,s}}\ y^{(i)}$
7. Set $E_{j,s} = \sum_{i \in I^+_{j,s}} \left( y^{(i)} - \hat{y}^+_{j,s} \right)^2 + \sum_{i \in I^-_{j,s}} \left( y^{(i)} - \hat{y}^-_{j,s} \right)^2$
8. Set $(j^*, s^*) = \arg\min_{j,s} E_{j,s}$
9. **else**
10. Set $\hat{y} = \text{average}_{\ i \in I}\ y^{(i)}$
11. **return** Leaf(leave_value=$\hat{y}$)
12. **return** Node $\left(j^*, s^*, \text{BuildTree}\left(I^-_{j^*,s^*}, k\right), \text{BuildTree}\left(I^+_{j^*,s^*}, k\right)\right)$



# 🌳 How to Grow a Classification Tree

The big idea:

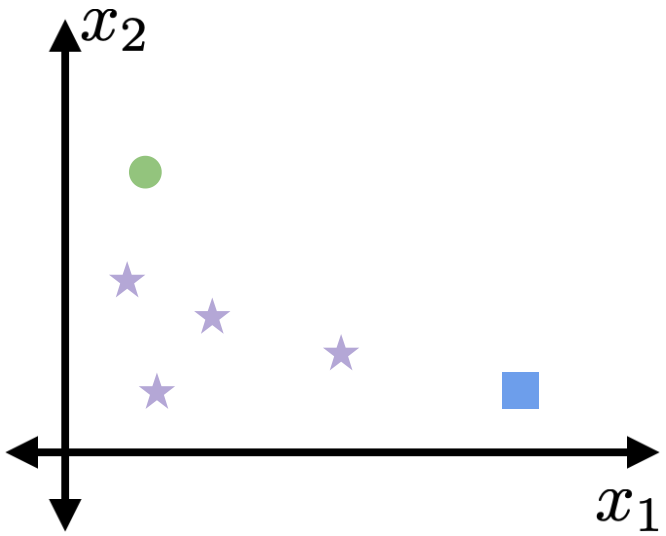
1. Grow — start with all data at the root.
   Try candidate splits along each feature.
2. Split — pick the split that best separates the classes
   (lowest weighted entropy or highest information gain).
3. Stop — if a region is small or already pure
   (data within ≤ leaf size, or most data share the same label), make it a leaf.

Recursive partitioning: each split divides data into regions that are more homogeneous in class labels, and each leaf predicts by the majority label of its data points.

entropy $H := -\sum_{\text{class } c} \hat{P}_c(\log_2 \hat{P}_c)$

empirical probability

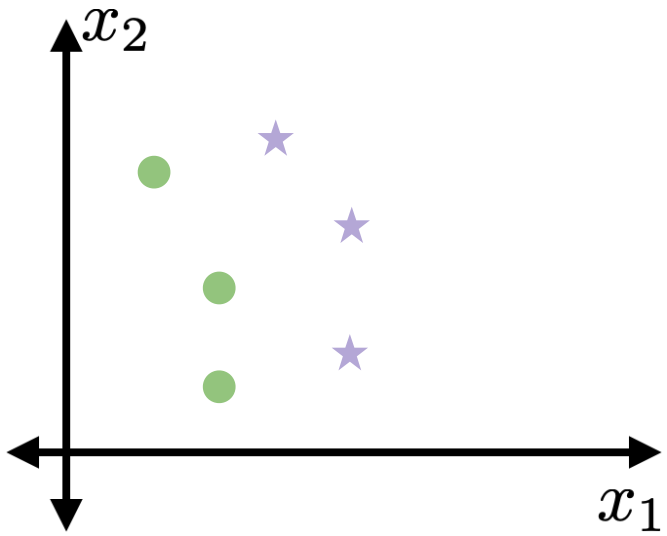
for example: $c$ iterates over 3 classes




$\hat{P}_c$ : $\frac{4}{6}$ $\frac{1}{6}$ $\frac{1}{6}$

$H = -[\frac{4}{6}\log_2(\frac{4}{6}) + \frac{1}{6}\log_2(\frac{1}{6}) + \frac{1}{6}\log_2(\frac{1}{6})]$

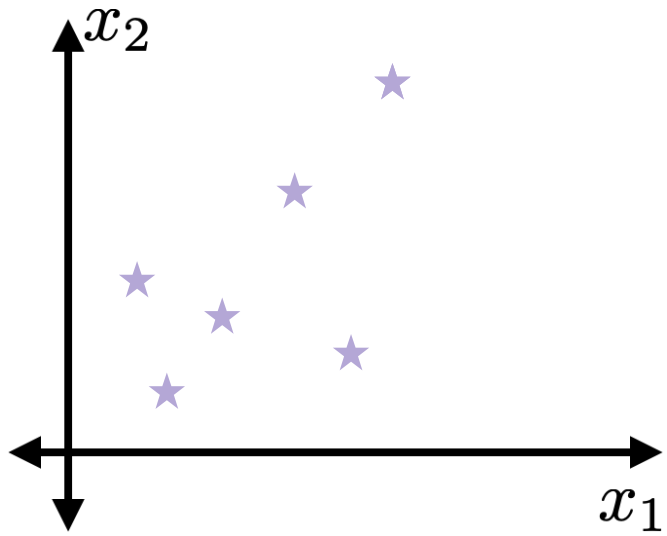
(about 1.252)




$\hat{P}_c$ : $\frac{3}{6}$ $\frac{3}{6}$ $0$

$H = -[\frac{3}{6}\log_2(\frac{3}{6}) + \frac{3}{6}\log_2(\frac{3}{6}) + 0]$

(about 1.1)




$\hat{P}_c$ : $\frac{6}{6}$ $0$ $0$

$H = -[\frac{6}{6}\log_2(\frac{6}{6}) + 0 + 0]$

(= 0)

For classification

majority vote
as regional prediction

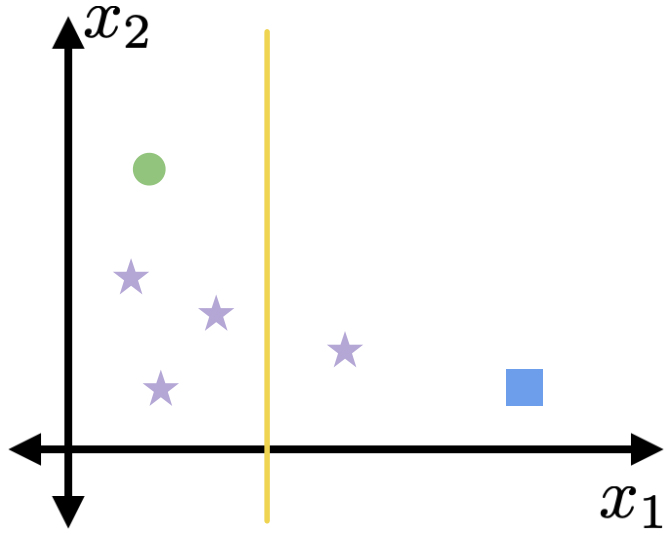
BuildTree$(I, k, \mathcal{D})$

1. **if** $|I| > k$
2. **for** each split dim $j$ and split value $s$
3. Set $I_{j,s}^{+} = \left\{ i \in I \mid x_j^{(i)} \geq s \right\}$
4. Set $I_{j,s}^{-} = \left\{ i \in I \mid x_j^{(i)} < s \right\}$
5. Set $\hat{y}_{j,s}^{+} = \text{majority}_{\ i \in I_{j,s}^{+}}\ y^{(i)}$
6. Set $\hat{y}_{j,s}^{-} = \text{majority}_{\ i \in I_{j,s}^{-}}\ y^{(i)}$
7. Set $E_{j,s} = \frac{|I_{j,s}^{-}|}{|I|} \cdot H\left(I_{j,s}^{-}\right) + \frac{|I_{j,s}^{+}|}{|I|} \cdot H\left(I_{j,s}^{+}\right)$
8. Set $(j^*, s^*) = \arg\min_{j,s} E_{j,s}$
9. **else**
10. Set $\hat{y} = \text{majority}_{\ i \in I}\ y^{(i)}$
11. **return** Leaf(leave_value=$\hat{y}$)
12. **return** Node $\left(j^*, s^*, \text{BuildTree}\left(I_{j^*,s^*}^{-}, k\right), \text{BuildTree}\left(I_{j^*,s^*}^{+}, k\right)\right)$

weighted average entropy (WAE)
as performance metric

BuildTree$(I, k, \mathcal{D})$

1. **if** $|I| > k$
2. **for** each split dim $j$ and split value $s$
3. Set $I_{j,s}^{+} = \left\{ i \in I \mid x_j^{(i)} \geq s \right\}$
4. Set $I_{j,s}^{-} = \left\{ i \in I \mid x_j^{(i)} < s \right\}$
5. Set $\hat{y}_{j,s}^{+} = \text{majority}_{\ i \in I_{j,s}^{+}}\ y^{(i)}$
6. Set $\hat{y}_{j,s}^{-} = \text{majority}_{\ i \in I_{j,s}^{-}}\ y^{(i)}$
7. Set $E_{j,s} = \frac{|I_{j,s}^{-}|}{|I|} \cdot H\left(I_{j,s}^{-}\right) + \frac{|I_{j,s}^{+}|}{|I|} \cdot H\left(I_{j,s}^{+}\right)$
8. Set $(j^*, s^*) = \arg\min_{j,s} E_{j,s}$
9. **else**
10. Set $\hat{y} = \text{majority}_{\ i \in I}\ y^{(i)}$
11. **return** Leaf(leave_value=$\hat{y}$)
12. **return** Node $\left(j^*, s^*, \text{BuildTree}\left(I_{j^*,s^*}^{-}, k\right), \text{BuildTree}\left(I_{j^*,s^*}^{+}, k\right)\right)$



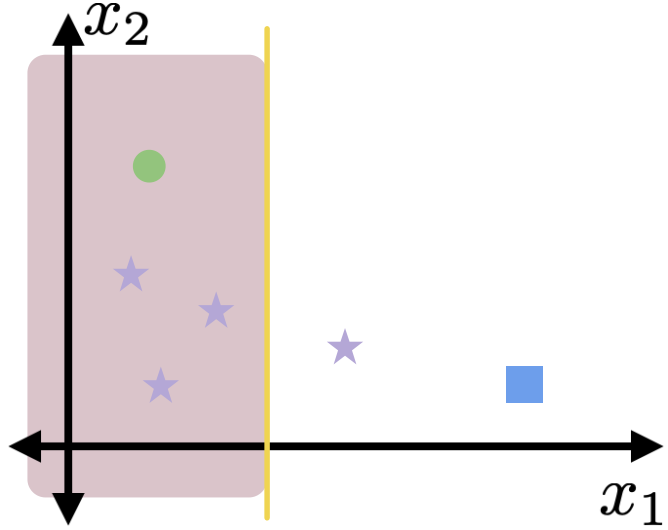

$$E_{j,s} = \frac{4}{6} \cdot H\left(I_{j,s}^{-}\right) + \frac{2}{6} \cdot H\left(I_{j,s}^{+}\right)$$

fraction of points to the left of the split

fraction of points to the right of the split

BuildTree$(I, k, \mathcal{D})$

1. **if** $|I| > k$
2. **for** each split dim $j$ and split value $s$
3. Set $I_{j,s}^{+} = \left\{ i \in I \mid x_j^{(i)} \geq s \right\}$
4. Set $I_{j,s}^{-} = \left\{ i \in I \mid x_j^{(i)} < s \right\}$
5. Set $\hat{y}_{j,s}^{+} = \text{majority}_{\ i \in I_{j,s}^{+}}\ y^{(i)}$
6. Set $\hat{y}_{j,s}^{-} = \text{majority}_{\ i \in I_{j,s}^{-}}\ y^{(i)}$
7. Set $E_{j,s} = \frac{|I_{j,s}^{-}|}{|I|} \cdot H\left(I_{j,s}^{-}\right) + \frac{|I_{j,s}^{+}|}{|I|} \cdot H\left(I_{j,s}^{+}\right)$
8. Set $(j^*, s^*) = \arg\min_{j,s} E_{j,s}$
9. **else**
10. Set $\hat{y} = \text{majority}_{\ i \in I}\ y^{(i)}$
11. **return** Leaf(leave_value=$\hat{y}$)
12. **return** Node $\left(j^*, s^*, \text{BuildTree}\left(I_{j^*,s^*}^{-}, k\right), \text{BuildTree}\left(I_{j^*,s^*}^{+}, k\right)\right)$



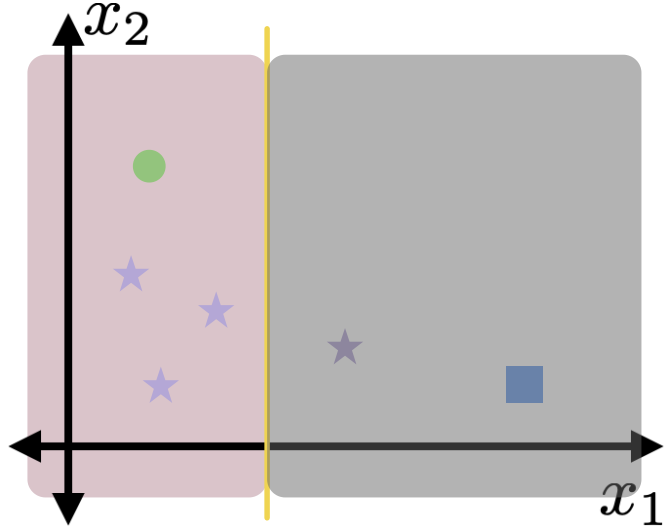

$$\frac{4}{6} \cdot H\left(I_{j,s}^{-}\right) + \frac{2}{6} \cdot H\left(I_{j,s}^{+}\right)$$

$\hat{P}_c$ : ★ $\frac{3}{4}$ ● $\frac{1}{4}$ ■ $\frac{0}{4}$

$$-\left[\tfrac{3}{4}\log_2\left(\tfrac{3}{4}\right) + \tfrac{1}{4}\log_2\left(\tfrac{1}{4}\right) + 0\right] \approx 0.811$$

BuildTree$(I, k, \mathcal{D})$

1. **if** $|I| > k$
2. **for** each split dim $j$ and split value $s$
3. Set $I_{j,s}^{+} = \left\{ i \in I \mid x_j^{(i)} \geq s \right\}$
4. Set $I_{j,s}^{-} = \left\{ i \in I \mid x_j^{(i)} < s \right\}$
5. Set $\hat{y}_{j,s}^{+} = \text{majority}_{\ i \in I_{j,s}^{+}}\ y^{(i)}$
6. Set $\hat{y}_{j,s}^{-} = \text{majority}_{\ i \in I_{j,s}^{-}}\ y^{(i)}$
7. Set $E_{j,s} = \frac{|I_{j,s}^{-}|}{|I|} \cdot H\left(I_{j,s}^{-}\right) + \frac{|I_{j,s}^{+}|}{|I|} \cdot H\left(I_{j,s}^{+}\right)$
8. Set $(j^*, s^*) = \arg\min_{j,s} E_{j,s}$
9. **else**
10. Set $\hat{y} = \text{majority}_{\ i \in I}\ y^{(i)}$
11. **return** Leaf(leave_value=$\hat{y}$)
12. **return** Node $\left(j^*, s^*, \text{BuildTree}\left(I_{j^*,s^*}^{-}, k\right), \text{BuildTree}\left(I_{j^*,s^*}^{+}, k\right)\right)$




$\approx 0.811$

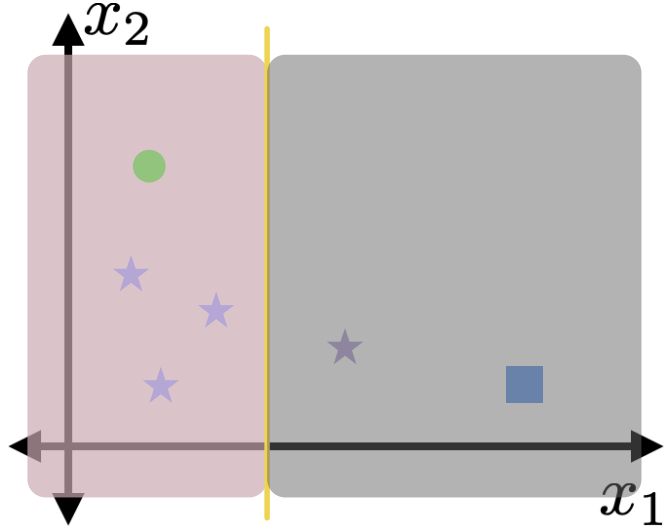
$$\frac{4}{6} \cdot H\left(I_{j,s}^{-}\right) + \frac{2}{6} \cdot H\left(I_{j,s}^{+}\right)$$

$$\hat{P}_c : \bigstar \ \frac{1}{2} \quad \bullet \ \frac{0}{2} \quad \blacksquare \ \frac{1}{2}$$

$$-\left[\tfrac{1}{2}\log_2\left(\tfrac{1}{2}\right) + \tfrac{1}{2}\log_2\left(\tfrac{1}{2}\right) + 0\right] = 1$$

BuildTree$(I, k, \mathcal{D})$

1. **if** $|I| > k$
2. **for** each split dim $j$ and split value $s$
3. Set $I_{j,s}^{+} = \left\{ i \in I \mid x_j^{(i)} \geq s \right\}$
4. Set $I_{j,s}^{-} = \left\{ i \in I \mid x_j^{(i)} < s \right\}$
5. Set $\hat{y}_{j,s}^{+} = \text{majority}_{\ i \in I_{j,s}^{+}}\ y^{(i)}$
6. Set $\hat{y}_{j,s}^{-} = \text{majority}_{\ i \in I_{j,s}^{-}}\ y^{(i)}$
7. Set $E_{j,s} = \frac{|I_{j,s}^{-}|}{|I|} \cdot H\left(I_{j,s}^{-}\right) + \frac{|I_{j,s}^{+}|}{|I|} \cdot H\left(I_{j,s}^{+}\right)$
8. Set $(j^*, s^*) = \arg\min_{j,s} E_{j,s}$
9. **else**
10. Set $\hat{y} = \text{majority}_{\ i \in I}\ y^{(i)}$
11. **return** Leaf(leave_value=$\hat{y}$)
12. **return** Node $\left(j^*, s^*, \text{BuildTree}\left(I_{j^*,s^*}^{-}, k\right), \text{BuildTree}\left(I_{j^*,s^*}^{+}, k\right)\right)$




$\approx 0.811$ $= 1$

$$\frac{4}{6} \cdot H\left(I_{j,s}^{-}\right) + \frac{2}{6} \cdot H\left(I_{j,s}^{+}\right)$$

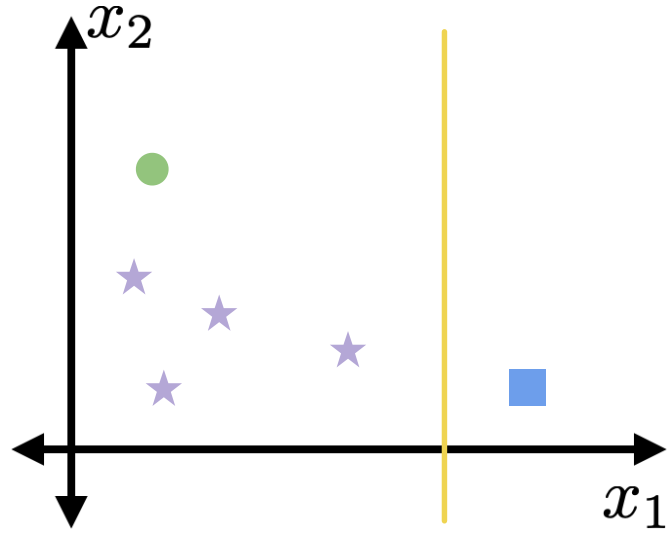
$$\approx \frac{4}{6} \cdot 0.811 + \frac{2}{6} \cdot 1$$

$$= 0.874$$

(for this split choice, line 7 $E_{j,s} \approx 0.874$)

BuildTree$(I, k, \mathcal{D})$

1. **if** $|I| > k$
2. **for** each split dim $j$ and split value $s$
3. Set $I_{j,s}^{+} = \left\{ i \in I \mid x_j^{(i)} \geq s \right\}$
4. Set $I_{j,s}^{-} = \left\{ i \in I \mid x_j^{(i)} < s \right\}$
5. Set $\hat{y}_{j,s}^{+} = \text{majority}_{\ i \in I_{j,s}^{+}}\ y^{(i)}$
6. Set $\hat{y}_{j,s}^{-} = \text{majority}_{\ i \in I_{j,s}^{-}}\ y^{(i)}$
7. Set $E_{j,s} = \frac{|I_{j,s}^{-}|}{|I|} \cdot H\left(I_{j,s}^{-}\right) + \frac{|I_{j,s}^{+}|}{|I|} \cdot H\left(I_{j,s}^{+}\right)$
8. Set $(j^*, s^*) = \arg\min_{j,s} E_{j,s}$
9. **else**
10. Set $\hat{y} = \text{majority}_{\ i \in I}\ y^{(i)}$
11. **return** Leaf(leave_value=$\hat{y}$)
12. **return** Node $\left(j^*, s^*, \text{BuildTree}\left(I_{j^*,s^*}^{-}, k\right), \text{BuildTree}\left(I_{j^*,s^*}^{+}, k\right)\right)$




$$-\left[\tfrac{4}{5}\log_2\left(\tfrac{4}{5}\right) + \tfrac{1}{5}\log_2\left(\tfrac{1}{5}\right) + 0\right] \approx 0.722$$

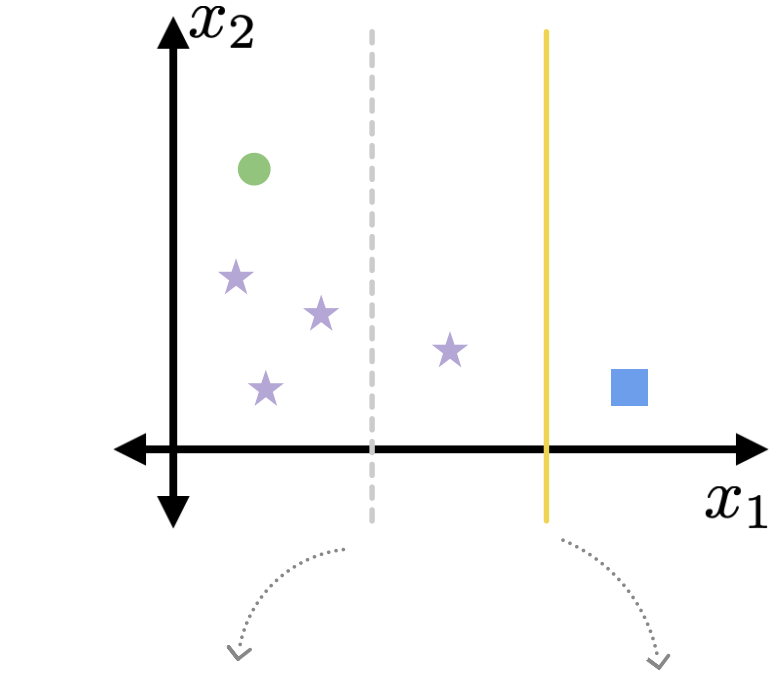
$$\tfrac{5}{6} \cdot H\left(I_{j,s}^{-}\right) + \tfrac{1}{6} \cdot H\left(I_{j,s}^{+}\right)$$

$$-\left[1\log_2(1) + 0 + 0\right] = 0$$

(line 7, overall $E_{j,s} \approx 0.602$)

BuildTree$(I, k, \mathcal{D})$

1. **if** $|I| > k$
2. **for** each split dim $j$ and split value $s$
3. Set $I_{j,s}^{+} = \left\{ i \in I \mid x_j^{(i)} \geq s \right\}$
4. Set $I_{j,s}^{-} = \left\{ i \in I \mid x_j^{(i)} < s \right\}$
5. Set $\hat{y}_{j,s}^{+} = \text{majority}_{\ i \in I_{j,s}^{+}}\ y^{(i)}$
6. Set $\hat{y}_{j,s}^{-} = \text{majority}_{\ i \in I_{j,s}^{-}}\ y^{(i)}$
7. Set $E_{j,s} = \frac{|I_{j,s}^{-}|}{|I|} \cdot H\left(I_{j,s}^{-}\right) + \frac{|I_{j,s}^{+}|}{|I|} \cdot H\left(I_{j,s}^{+}\right)$
8. Set $(j^*, s^*) = \arg\min_{j,s} E_{j,s}$
9. **else**
10. Set $\hat{y} = \text{majority}_{\ i \in I}\ y^{(i)}$
11. **return** Leaf(leave_value=$\hat{y}$)
12. **return** Node $\left(j^*, s^*, \text{BuildTree}\left(I_{j^*,s^*}^{-}, k\right), \text{BuildTree}\left(I_{j^*,s^*}^{+}, k\right)\right)$




line 8, set the better $(j, s)$

Key takeaway:

Entropy measures impurity → the best split minimizes the weighted entropy.

*(Weighted = each branch's entropy weighted by how many samples fall there.)*

🪓 In classification decision trees, we keep "cutting" where uncertainty drops the most.

Ensemble

WIRED SUBSCRIBE

ELIOT VAN BUSKIRK BUSINESS 09.22.2009 11:19 AM

# How the Netflix Prize Was Won

**Like BellKor's Pragmatic Chaos, the winner of the Netflix Prize, second-place The Ensemble was an amalgam of teams which had been competing individually for the million-dollar prize. But it wasn't until leaders joined forces with also-rans that real progress was made in the contest's goal to improve the Netflix movie recommendation algorithm by 10 percent. [...]**

International Journal of Forecasting
ELSEVIER Volume 36, Issue 1, January–March 2020, Pages 54-74

# The M4 Competition: 100,000 time series and 61 forecasting methods

Spyros Makridakis [a], Evangelos Spiliotis [b], Vassilios Assimakopoulos [b]

Coronavirus Disease MENU >

CASES, DATA & SURVEILLANCE

## Forecasts of COVID-19 Deaths

Updated Nov. 12, 2020

Observed and forecasted new and total reported COVID-19 deaths as of November 9, 2020.

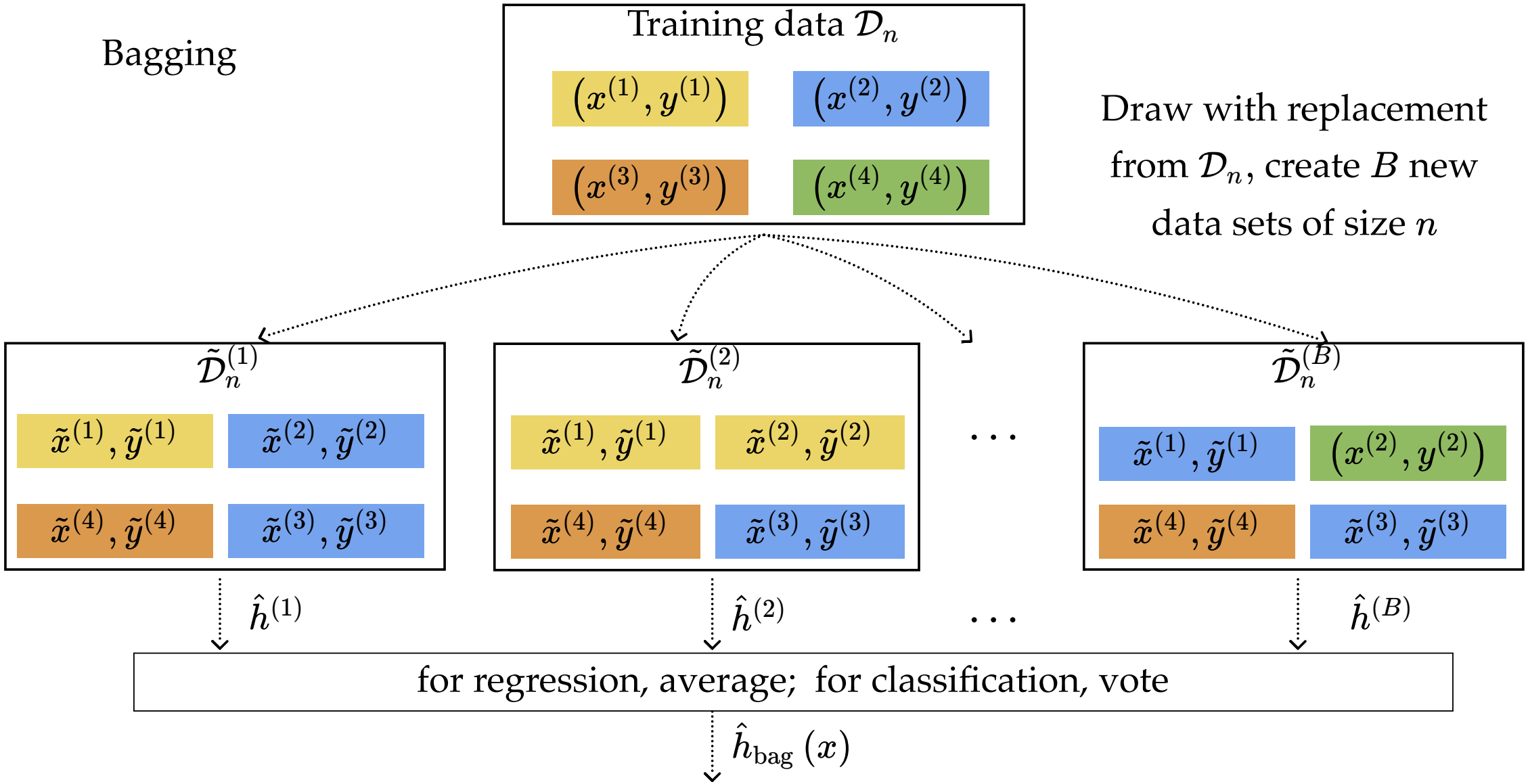
## Interpretation of Forecasts of New and Total Deaths

- This week CDC received forecasts of COVID-19 deaths over the next 4 weeks from 36 modeling groups that were included in the ensemble forecast. Of the 36 groups, 33 provided forecasts for both new and total deaths, two groups forecasted total deaths only, and one forecasted new death only.

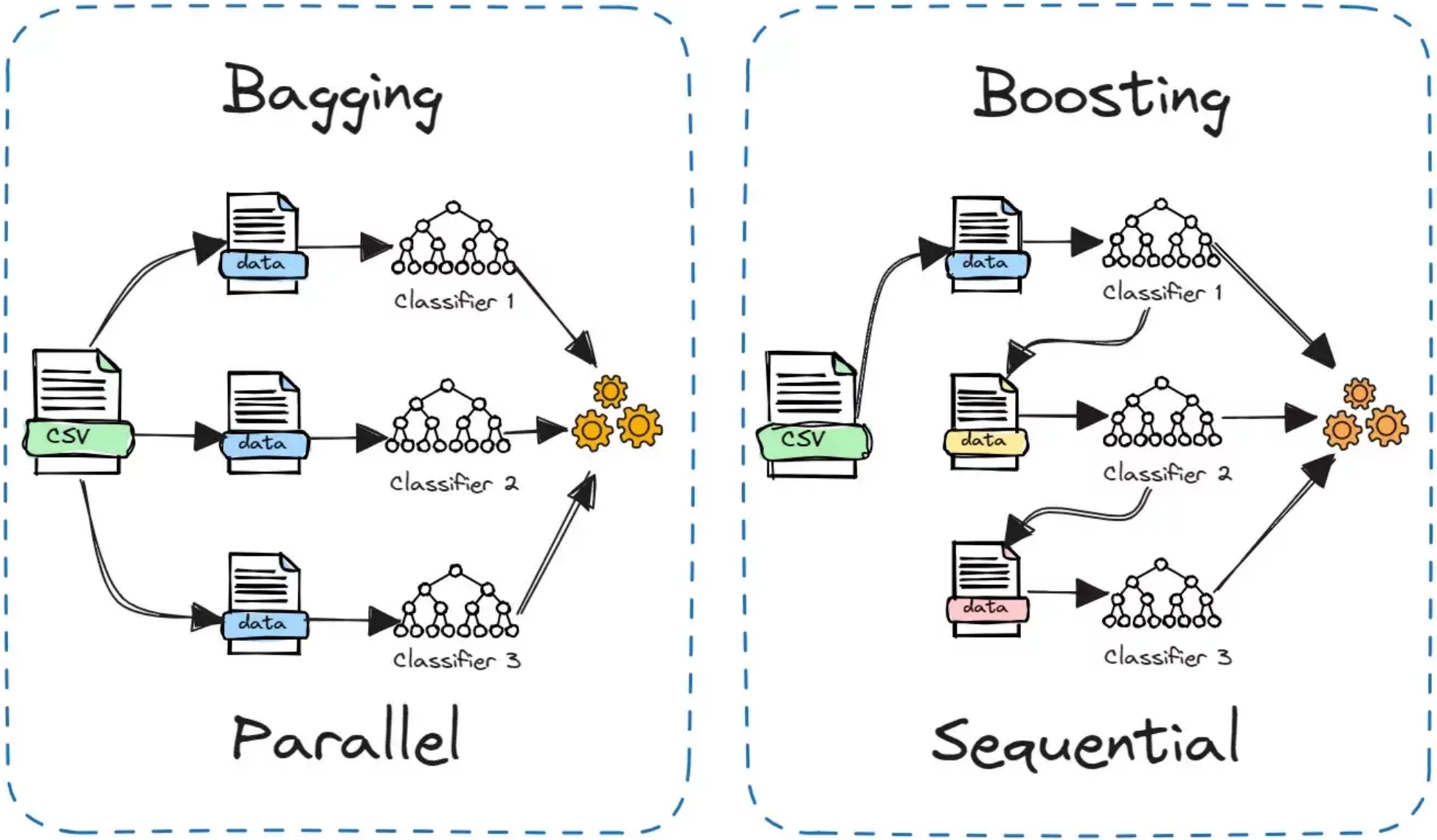
Bagging

- One of multiple ways to make and use an ensemble
- Bagging = **B**ootstrap **agg**regat**ing**

# Bagging


Training data $\mathcal{D}_n$
$(x^{(1)}, y^{(1)})$
$(x^{(2)}, y^{(2)})$
$(x^{(3)}, y^{(3)})$
$(x^{(4)}, y^{(4)})$
Draw with replacement from $\mathcal{D}_n$, create $B$ new data sets of size $n$
$\tilde{\mathcal{D}}_n^{(1)}$
$\tilde{x}^{(1)}, \tilde{y}^{(1)}$
$\tilde{x}^{(2)}, \tilde{y}^{(2)}$
$\tilde{x}^{(4)}, \tilde{y}^{(4)}$
$\tilde{x}^{(3)}, \tilde{y}^{(3)}$
$\tilde{\mathcal{D}}_n^{(2)}$
$\tilde{x}^{(1)}, \tilde{y}^{(1)}$
$\tilde{x}^{(2)}, \tilde{y}^{(2)}$
$\tilde{x}^{(4)}, \tilde{y}^{(4)}$
$\tilde{x}^{(3)}, \tilde{y}^{(3)}$
$\cdots$
$\tilde{\mathcal{D}}_n^{(B)}$
$\tilde{x}^{(1)}, \tilde{y}^{(1)}$
$(x^{(2)}, y^{(2)})$
$\tilde{x}^{(4)}, \tilde{y}^{(4)}$
$\tilde{x}^{(3)}, \tilde{y}^{(3)}$
$\hat{h}^{(1)}$
$\hat{h}^{(2)}$
$\cdots$
$\hat{h}^{(B)}$
for regression, average; for classification, vote
$\hat{h}_{\text{bag}}(x)$

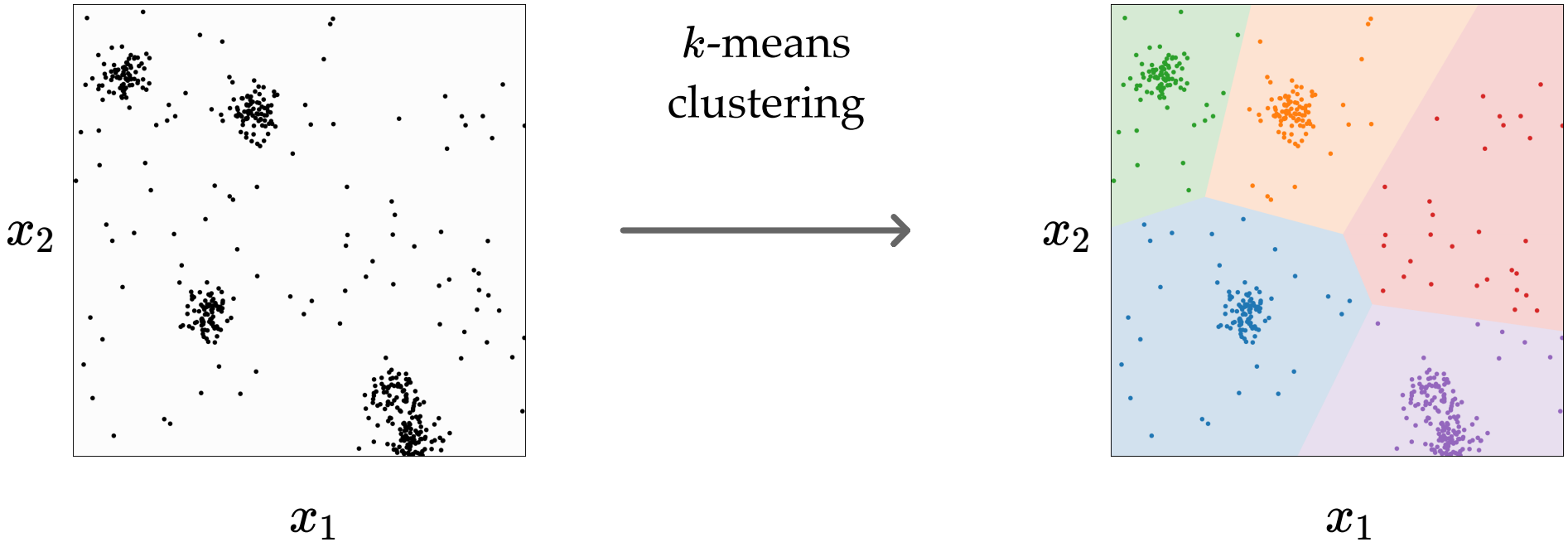
(boosting)


Bagging
data
Classifier 1
CSV
data
Classifier 2
data
Classifier 3
Parallel
Boosting
data
Classifier 1
CSV
data
Classifier 2
data
Classifier 3
Sequential

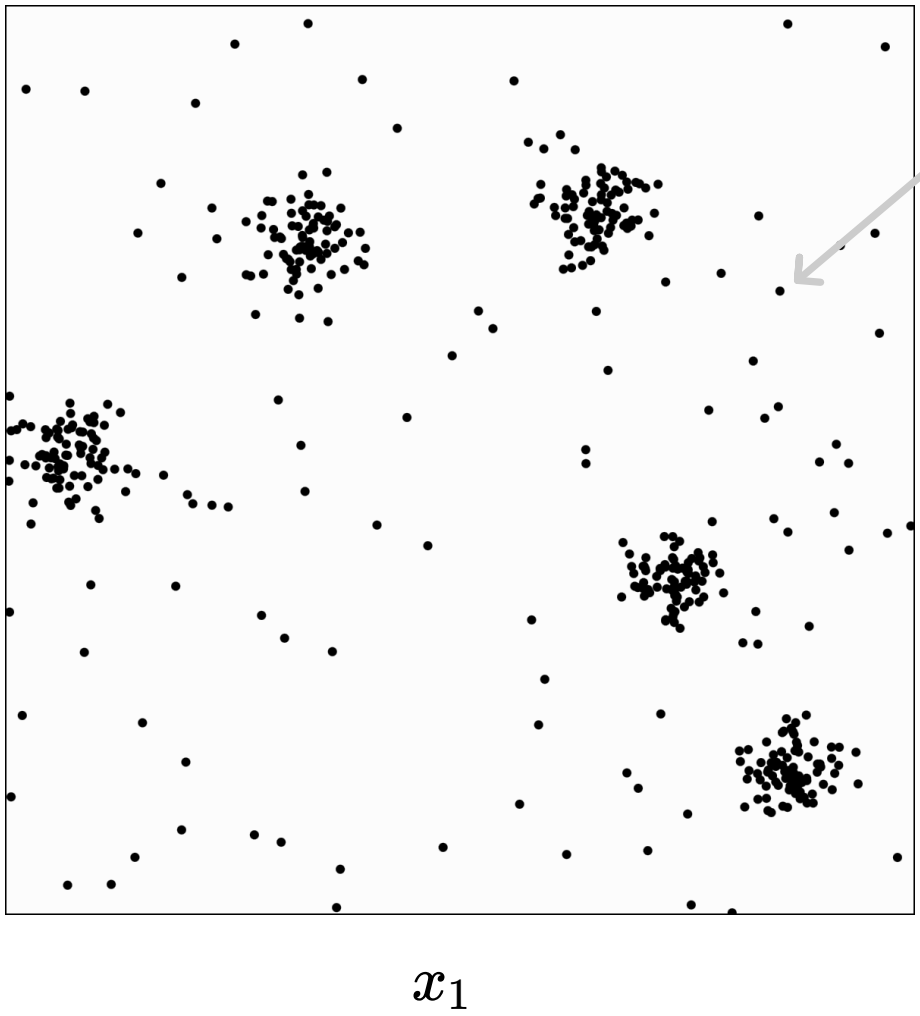
# Outline

- Non-parametric models overview
- Supervised learning non-parametric
  - $k$-nearest neighbor
  - Decision Tree (read a tree, BuildTree, Bagging)
- Unsupervised learning non-parametric
  - $k$-means clustering

structure discovery without labels

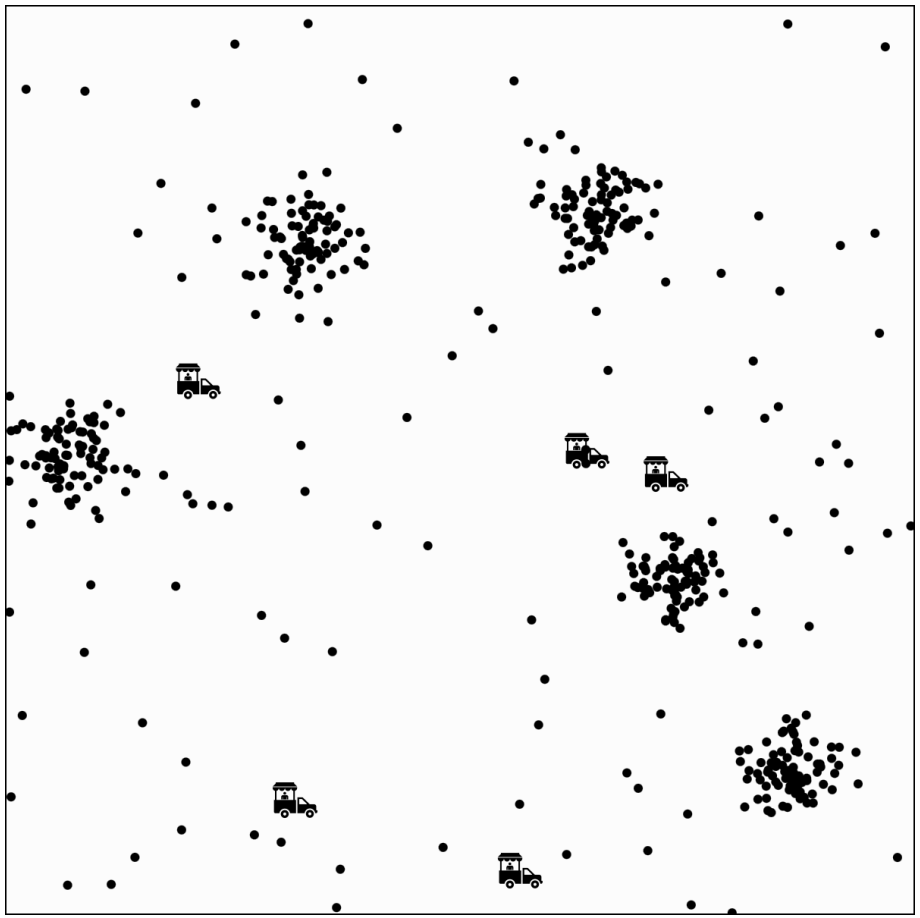


# Food-truck placement



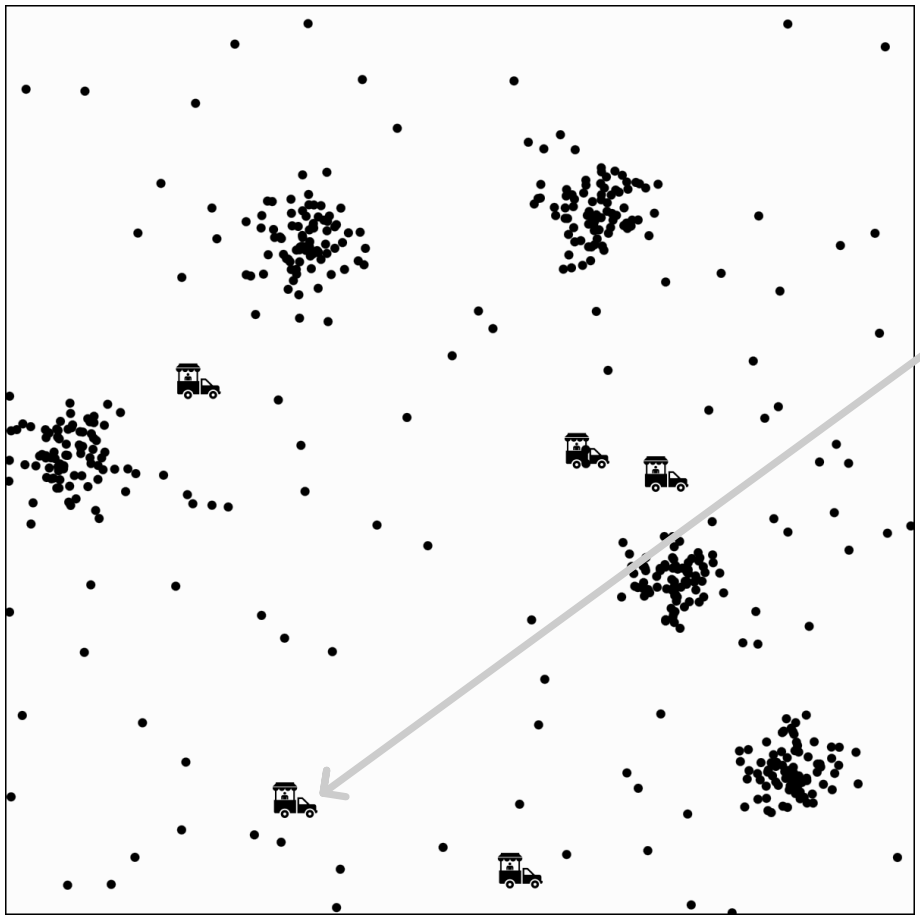

- $x_1$: longitude, $x_2$: latitude
- $n$ person, person $i$ location $x^{(i)}$

# Food-truck placement



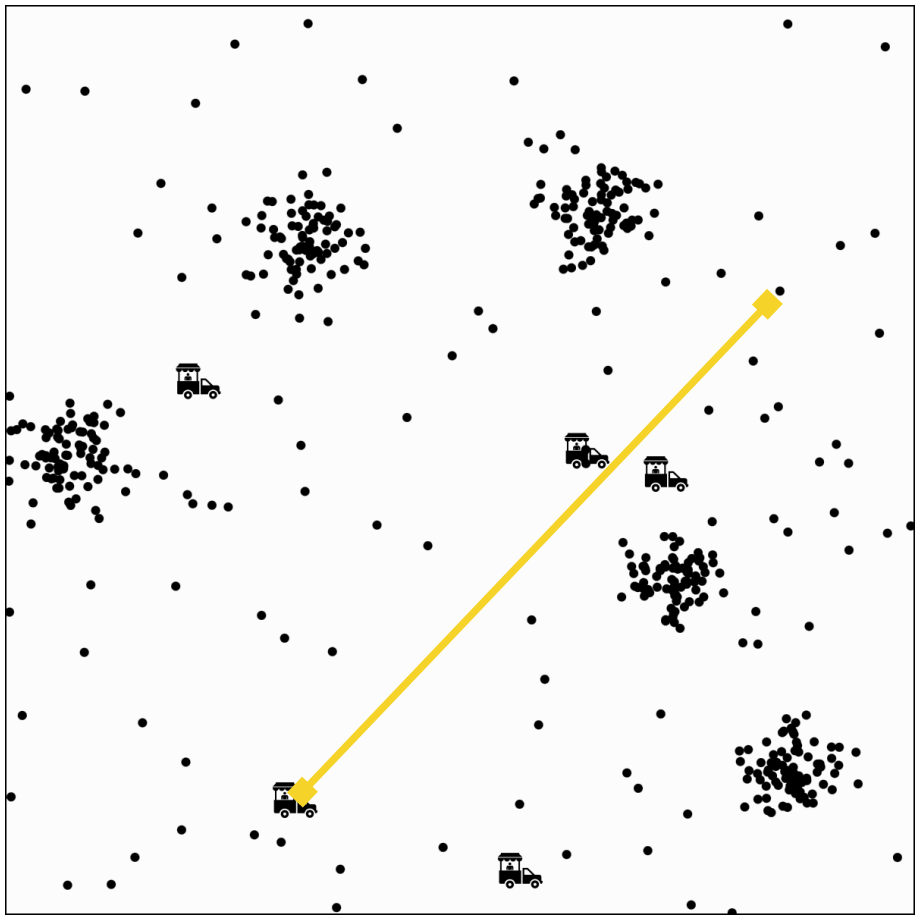

- $x_1$: longitude, $x_2$: latitude
- $n$ person, person $i$ location $x^{(i)}$
- $k$ food trucks, say $k = 5$

# Food-truck placement



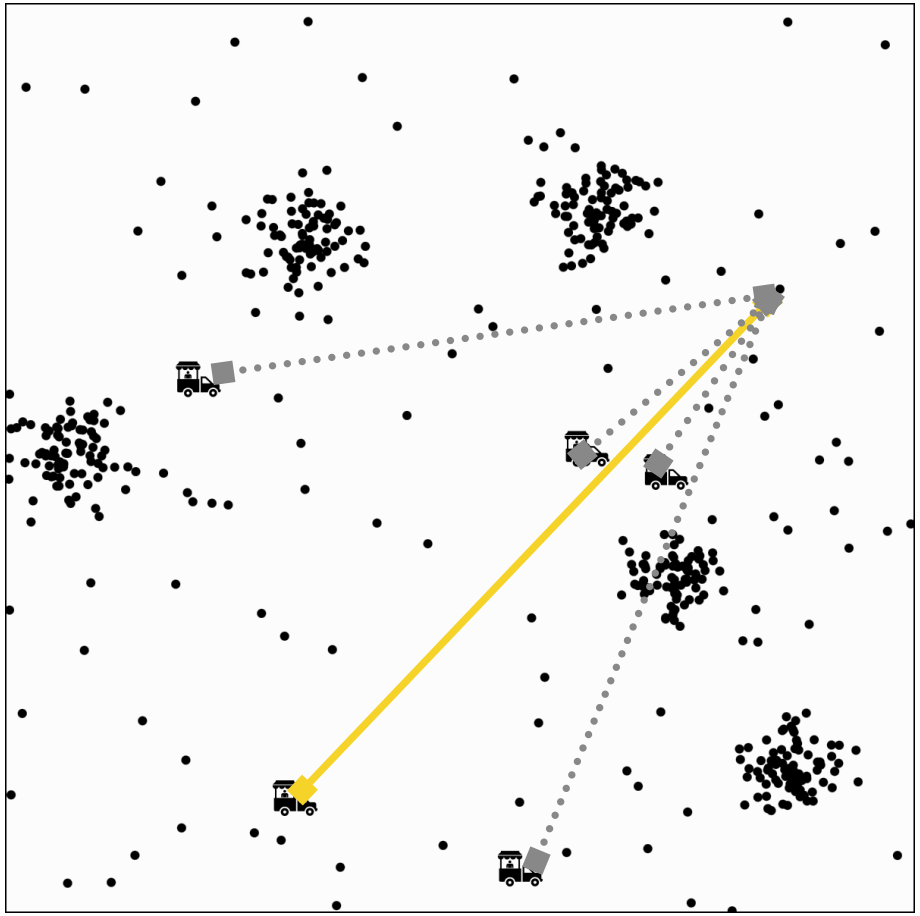

- $x_1$: longitude, $x_2$: latitude
- $n$ person, person $i$ location $x^{(i)}$
- $k$ food trucks, say $k = 5$
- Food truck $j$ location $\mu^{(j)}$

# Food-truck placement




- $x_1$: longitude, $x_2$: latitude
- $n$ person, person $i$ location $x^{(i)}$
- $k$ food trucks, say $k = 5$
- Food truck $j$ location $\mu^{(j)}$
- Loss for $i$ to walk to truck $j$ : $\left\| x^{(i)} - \mu^{(j)} \right\|^2$

## Food-truck placement




- $x_1$: longitude, $x_2$: latitude
- $n$ person, person $i$ location $x^{(i)}$
- $k$ food trucks, say $k = 5$
- Food truck $j$ location $\mu^{(j)}$
- Loss for $i$ to walk to truck $j$ : $\left\|x^{(i)} - \mu^{(j)}\right\|^2$
- Index of the truck to which $i$ walks: $y^{(i)}$
- Person $i$ overall loss:

$$\sum_{j=1}^{k} \mathbf{1}\left\{y^{(i)} = j\right\} \left\|x^{(i)} - \mu^{(j)}\right\|_2^2$$

indicator function

1 if person $i$ is assigned to truck $j$, 0 o/w.

$k$-means objective

what we learn

clustering membership

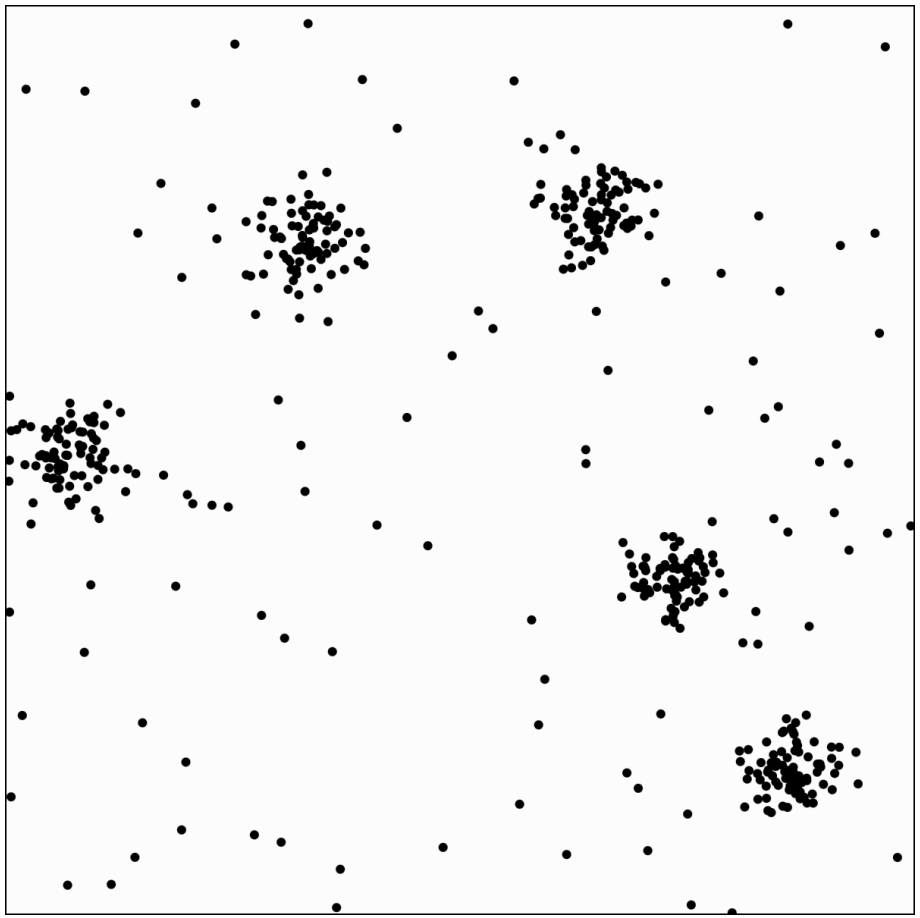
clustering centroid location

$$\sum_{i=1}^{n} \sum_{j=1}^{k} \mathbf{1}\left\{y^{(i)} = j\right\} \left\|x^{(i)} - \mu^{(j)}\right\|^2$$

sum over data points

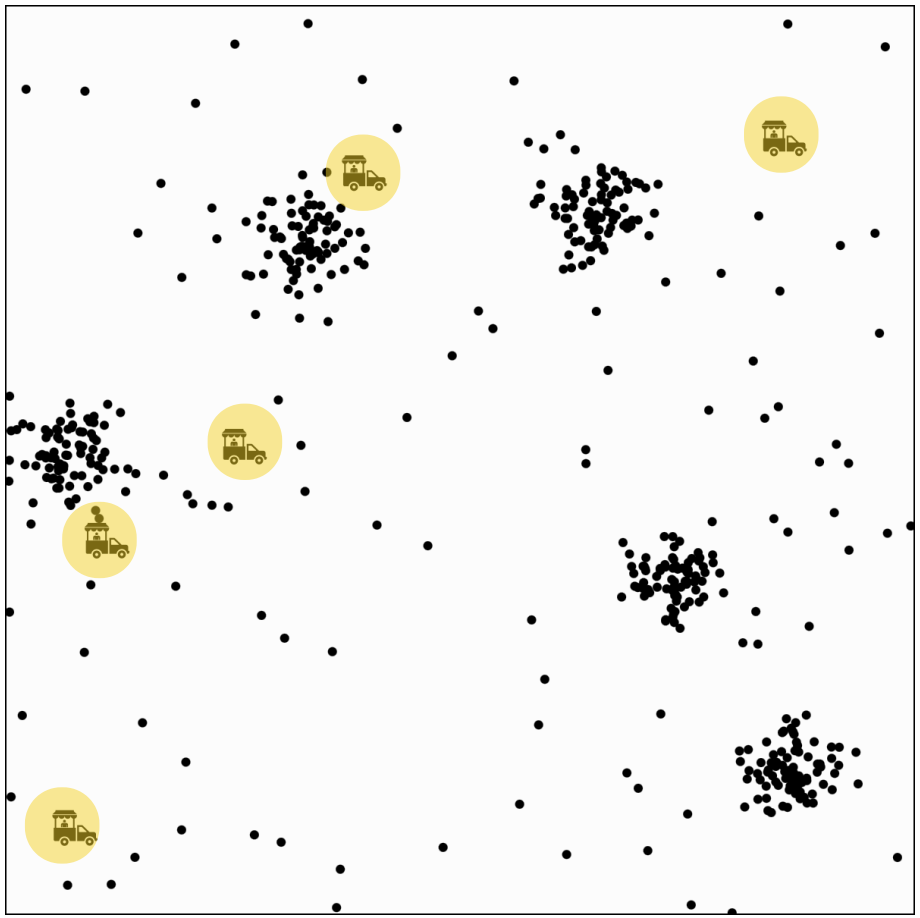
sum over cluster

can switch the order $= \sum_{j=1}^{k} \sum_{i=1}^{n} \mathbf{1}\left\{y^{(i)} = j\right\} \left\|x^{(i)} - \mu^{(j)}\right\|_2^2$




K-MEANS$(k, \tau, \{x^{(i)}\}_{i=1}^{n})$

$\mu, y =$ random initialization
**for** $t = 1$ to $\tau$
    $y_{\text{old}} = y$
    **for** $i = 1$ to $n$
        $y^{(i)} = \arg\min_j \left\| x^{(i)} - \mu^{(j)} \right\|^2$
    **for** $j = 1$ to $k$
        $\mu^{(j)} = \frac{1}{N_j} \sum_{i=1}^{n} \mathbf{1}\left(y^{(i)} = \mathrm{j}\right) x^{(i)}$
    **if** $y == y_{\text{old}}$
        break
return $\mu, y$




K-MEANS$(k, \tau, \{x^{(i)}\}_{i=1}^{n})$

$\mu, y =$ random initialization

**for** $t = 1$ to $\tau$

  $y_{\text{old}} = y$

  **for** $i = 1$ to $n$

    $y^{(i)} = \arg\min_j \left\| x^{(i)} - \mu^{(j)} \right\|^2$

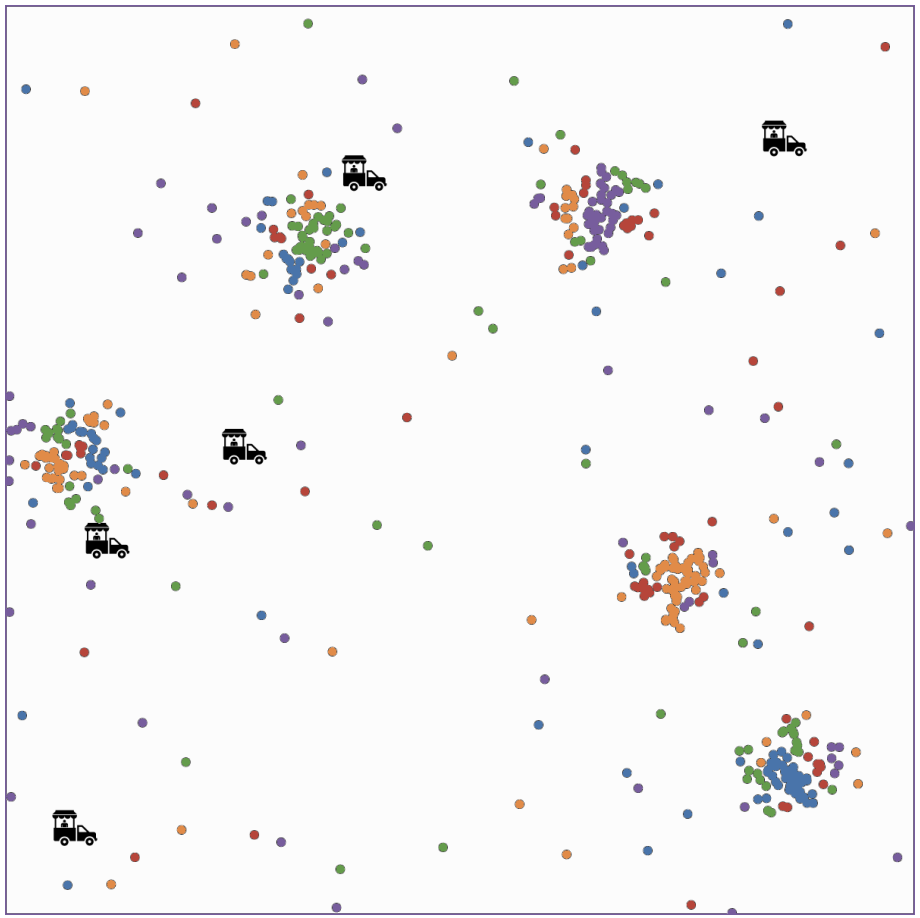
  **for** $j = 1$ to $k$

    $\mu^{(j)} = \frac{1}{N_j} \sum_{i=1}^{n} \mathbf{1}\left(y^{(i)} = \mathrm{j}\right) x^{(i)}$

  **if** $y == y_{\text{old}}$

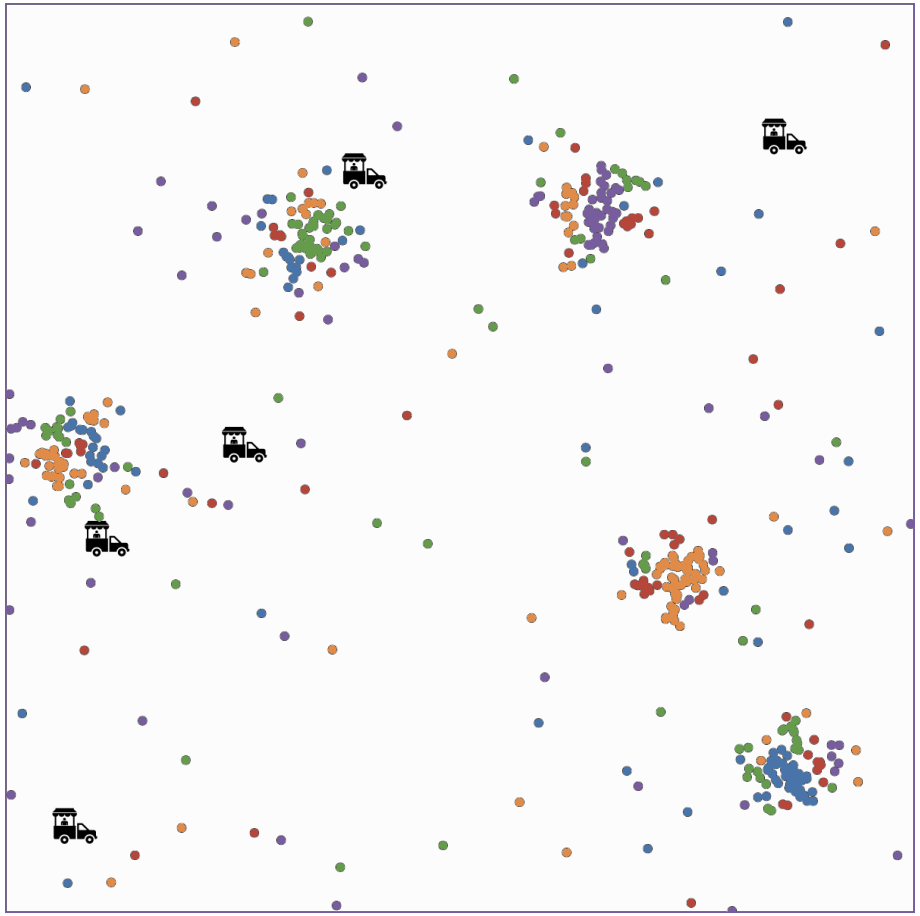
    break

return $\mu, y$




K-MEANS$(k, \tau, \{x^{(i)}\}_{i=1}^{n})$

1 $\mu, y$ = random initialization
2 **for** $t = 1$ to $\tau$
3 $\quad y_{\text{old}} = y$
4 $\quad$ **for** $i = 1$ to $n$
5 $\quad\quad y^{(i)} = \arg\min_j \left\| x^{(i)} - \mu^{(j)} \right\|^2$
6 $\quad$ **for** $j = 1$ to $k$
7 $\quad\quad \mu^{(j)} = \frac{1}{N_j} \sum_{i=1}^{n} \mathbf{1}\left(y^{(i)} = \mathrm{j}\right) x^{(i)}$
8 $\quad$ **if** $y == y_{\text{old}}$
9 $\quad\quad$ break
10 return $\mu, y$




K-MEANS$(k, \tau, \{x^{(i)}\}_{i=1}^{n})$

$\mu, y =$ random initialization

**for** $t = 1$ to $\tau$

  $y_{\text{old}} = y$

  **for** $i = 1$ to $n$

    $y^{(i)} = \arg\min_j \left\| x^{(i)} - \mu^{(j)} \right\|^2$

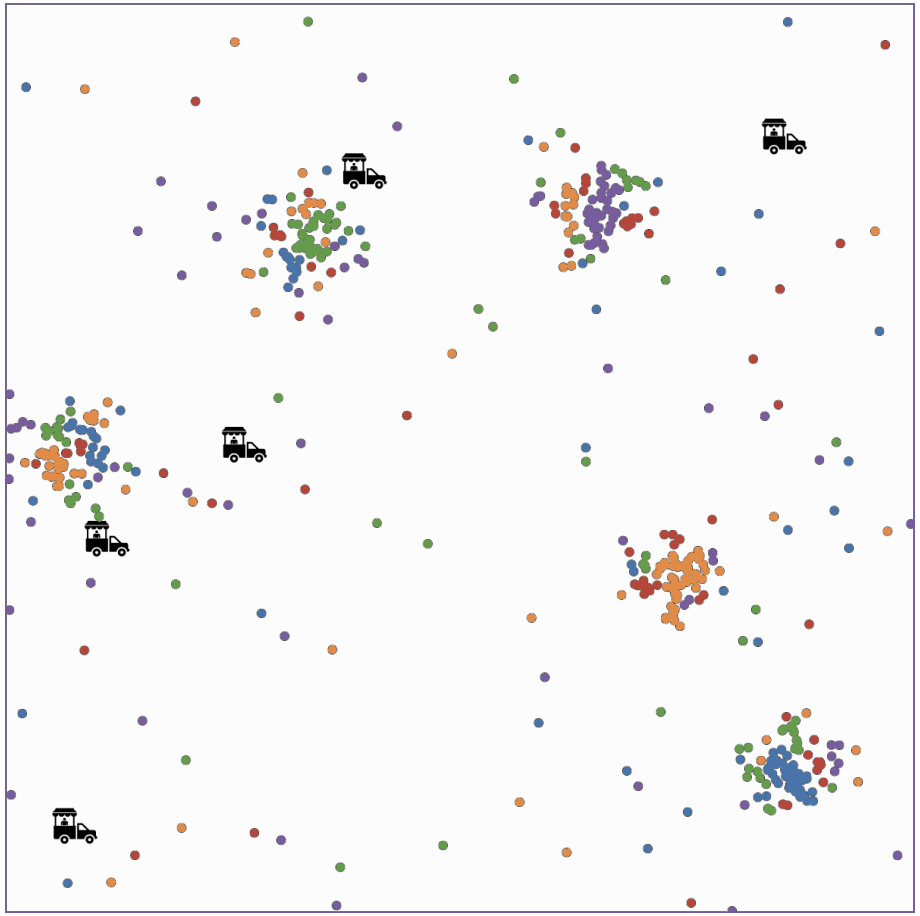
  **for** $j = 1$ to $k$

    $\mu^{(j)} = \frac{1}{N_j} \sum_{i=1}^{n} \mathbf{1}\left(y^{(i)} = \mathrm{j}\right) x^{(i)}$

  **if** $y == y_{\text{old}}$

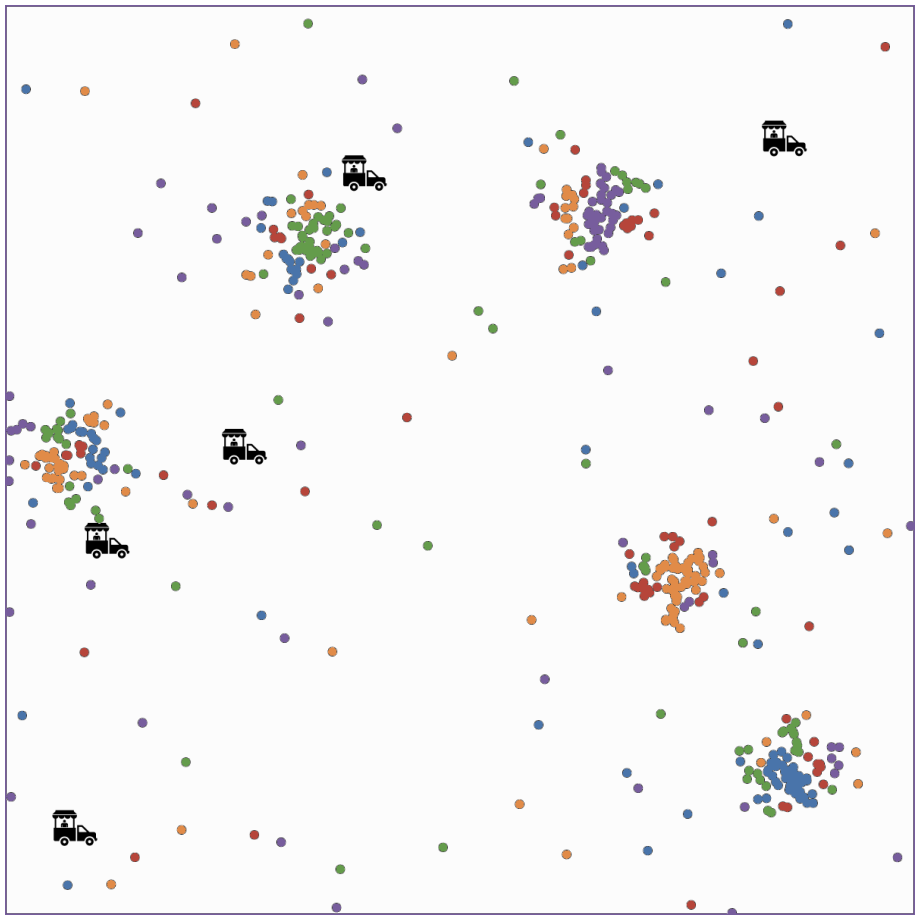
    break

return $\mu, y$




K-MEANS$(k, \tau, \{x^{(i)}\}_{i=1}^{n})$

1 $\mu, y =$ random initialization
2 **for** $t = 1$ to $\tau$
3 $\quad y_{\text{old}} = y$
4 $\quad$ **for** $i = 1$ to $n$
5 $\quad\quad y^{(i)} = \arg\min_j \left\| x^{(i)} - \mu^{(j)} \right\|^2$
6 $\quad$ **for** $j = 1$ to $k$
7 $\quad\quad \mu^{(j)} = \frac{1}{N_j} \sum_{i=1}^{n} \mathbf{1}\left(y^{(i)} = \mathrm{j}\right) x^{(i)}$
8 $\quad$ **if** $y == y_{\text{old}}$
9 $\quad\quad$ break
10 return $\mu, y$




K-MEANS$(k, \tau, \{x^{(i)}\}_{i=1}^{n})$

1 $\mu, y =$ random initialization

2 **for** $t = 1$ to $\tau$

3 $\quad y_{\text{old}} = y$

4 $\quad$ **for** $i = 1$ to $n$

5 $\quad\quad y^{(i)} = \arg\min_j \left\| x^{(i)} - \mu^{(j)} \right\|^2$

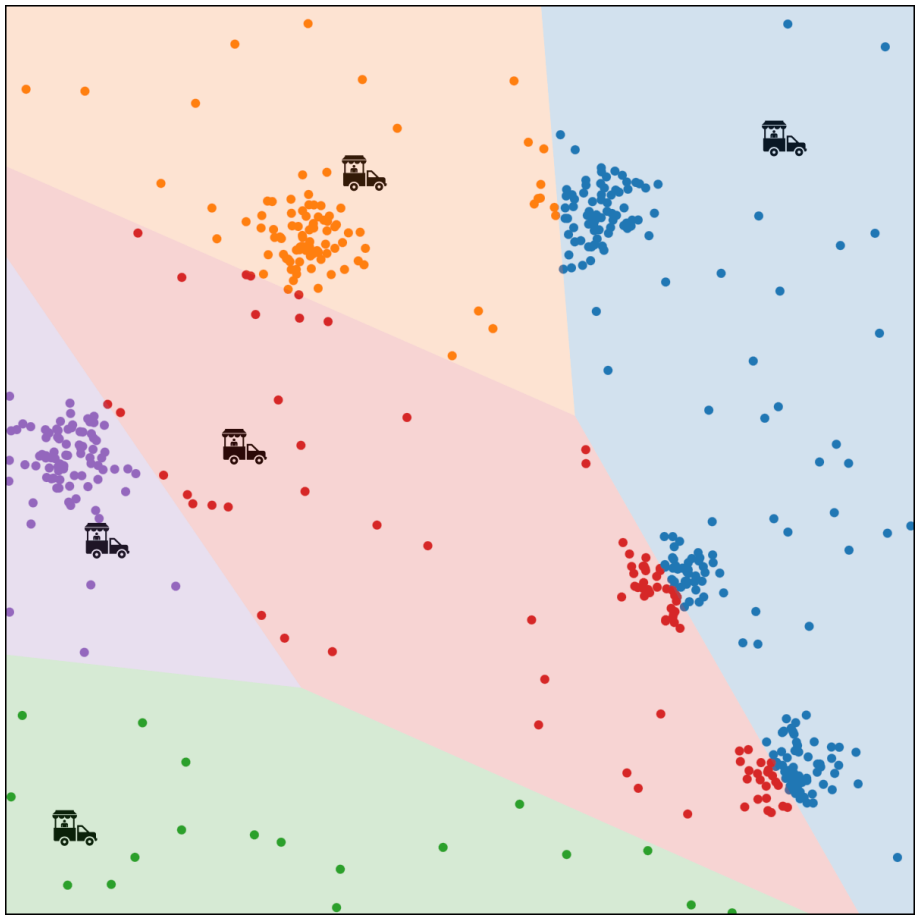
6 $\quad$ **for** $j = 1$ to $k$

7 $\quad\quad \mu^{(j)} = \frac{1}{N_j} \sum_{i=1}^{n} \mathbf{1}\left(y^{(i)} = \mathrm{j}\right) x^{(i)}$

8 $\quad$ **if** $y == y_{\text{old}}$

9 $\quad\quad$ break

10 return $\mu, y$




K-MEANS$(k, \tau, \{x^{(i)}\}_{i=1}^{n})$

1 $\mu, y =$ random initialization

2 **for** $t = 1$ to $\tau$

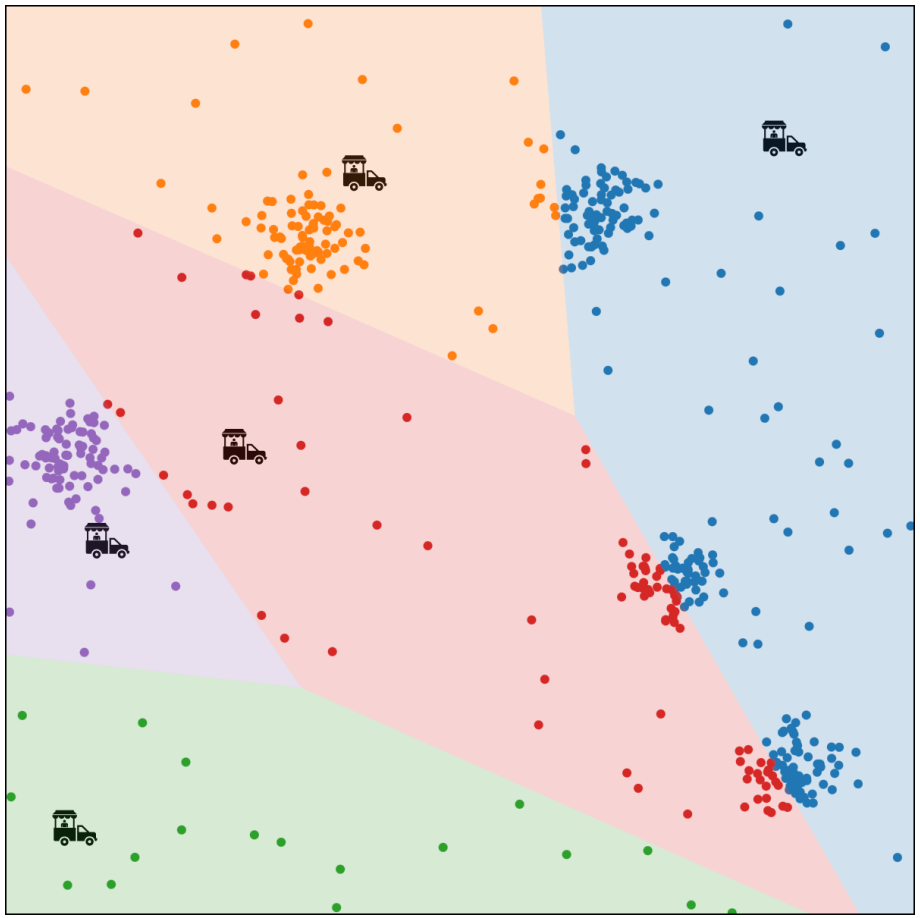
3 $\quad y_{\text{old}} = y$

4 $\quad$ **for** $i = 1$ to $n$

5 $\quad\quad y^{(i)} = \arg\min_j \left|\left|x^{(i)} - \mu^{(j)}\right|\right|^2$

$\quad\quad \ldots$

person $i$ assigned to food truck $j$,
feature space partition color-coded.




K-MEANS$(k, \tau, \{x^{(i)}\}_{i=1}^{n})$

1 $\mu, y =$ random initialization

2 **for** $t = 1$ to $\tau$

3 $\quad y_{\text{old}} = y$

4 $\quad$ **for** $i = 1$ to $n$

5 $\quad\quad y^{(i)} = \arg\min_j \left\| x^{(i)} - \mu^{(j)} \right\|^2$

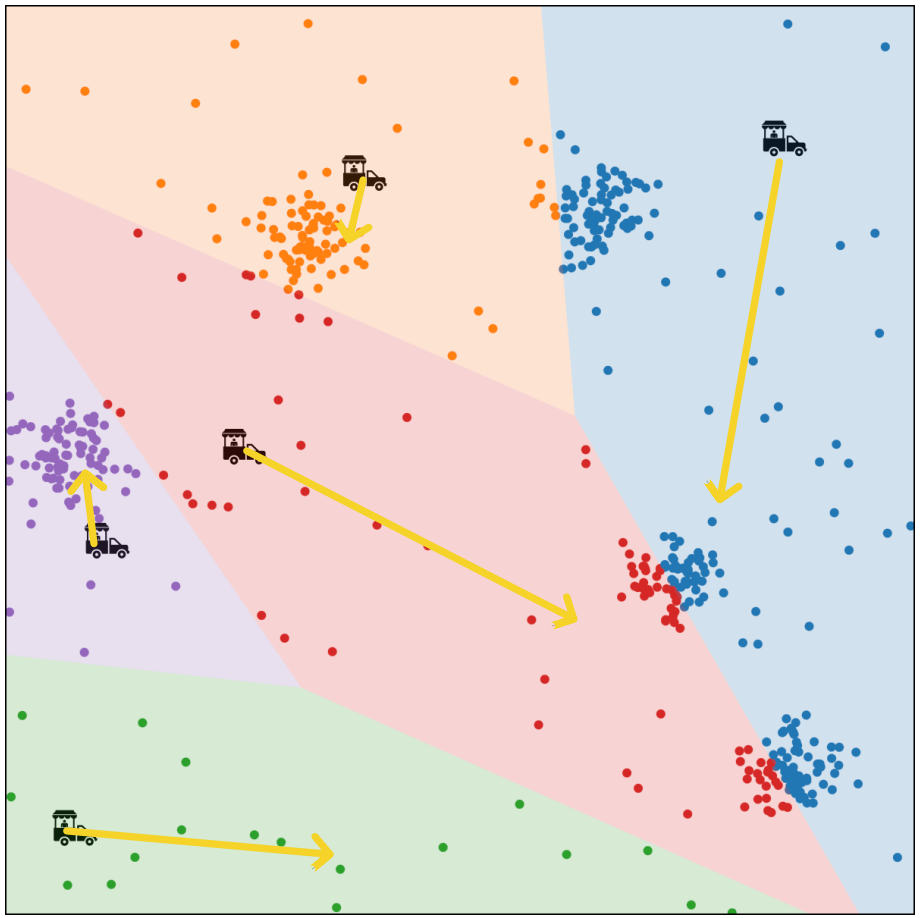
6 $\quad$ **for** $j = 1$ to $k$

7 $\quad\quad \mu^{(j)} = \frac{1}{N_j} \sum_{i=1}^{n} \mathbf{1}\left(y^{(i)} = \mathrm{j}\right) x^{(i)}$

8 $\quad$ **if** $y == y_{\text{old}}$

9 $\quad\quad$ break

10 return $\mu, y$




K-MEANS$(k, \tau, \{x^{(i)}\}_{i=1}^{n})$

1 $\mu, y =$ random initialization

2 **for** $t = 1$ to $\tau$

3 $\quad y_{\text{old}} = y$

4 $\quad$ **for** $i = 1$ to $n$

5 $\quad\quad y^{(i)} = \arg\min_j \left\| x^{(i)} - \mu^{(j)} \right\|^2$

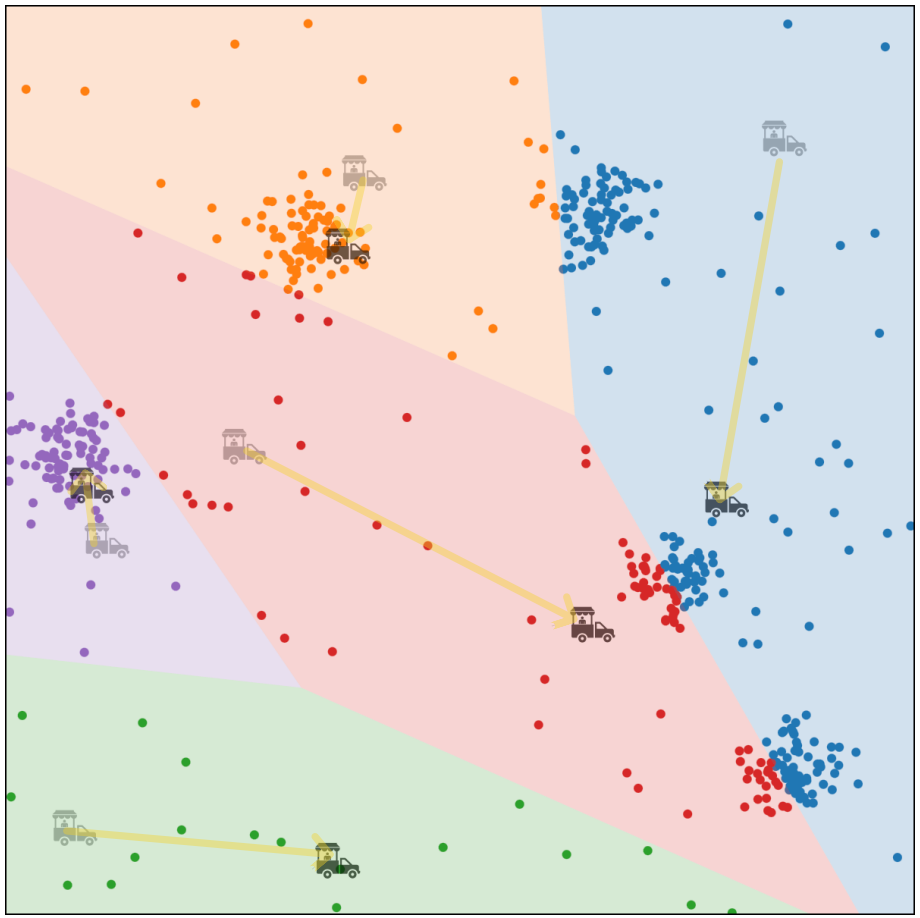
6 $\quad$ **for** $j = 1$ to $k$

7 $\quad\quad \mu^{(j)} = \frac{1}{N_j} \sum_{i=1}^{n} \mathbf{1}\left(y^{(i)} = \mathrm{j}\right) x^{(i)}$

...

$N_j = \sum_{i=1}^{n} \mathbf{1}\left\{y^{(i)} = j\right\}$

move food truck $j$ to the central location of all people assigned to it.




K-MEANS$(k, \tau, \{x^{(i)}\}_{i=1}^{n})$

$\mu, y =$ random initialization

**for** $t = 1$ to $\tau$

  $y_{\text{old}} = y$

  **for** $i = 1$ to $n$

    $y^{(i)} = \arg\min_j \left\| x^{(i)} - \mu^{(j)} \right\|^2$

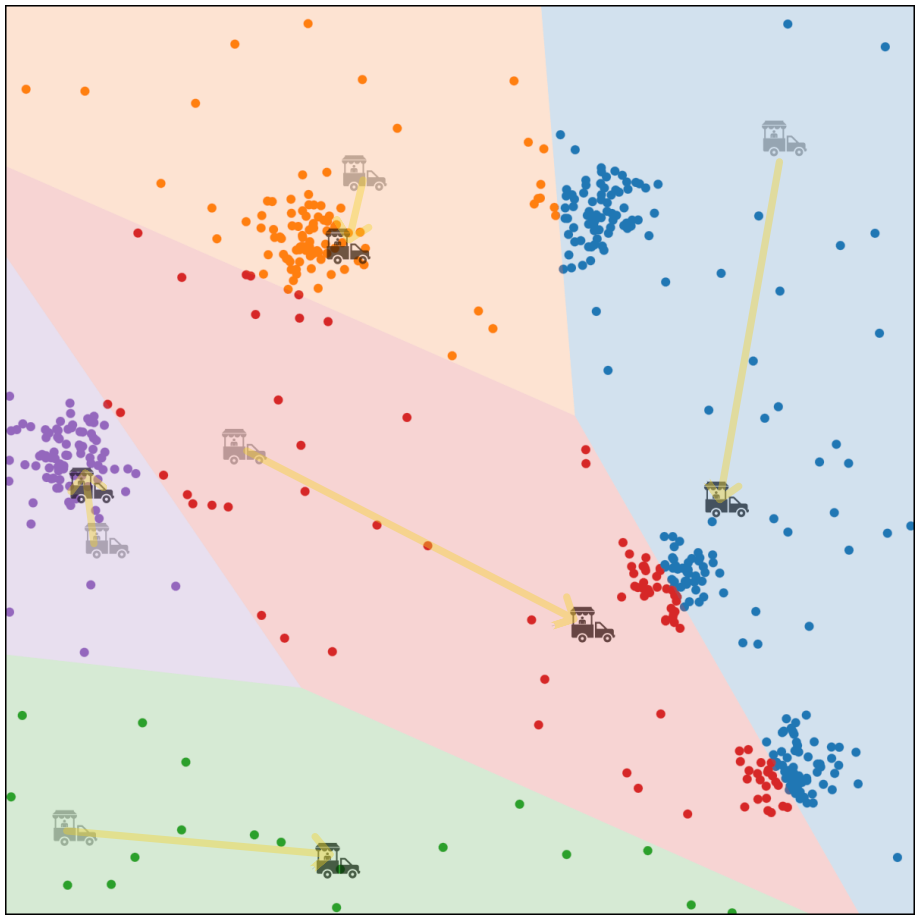
  **for** $j = 1$ to $k$

    $\mu^{(j)} = \frac{1}{N_j} \sum_{i=1}^{n} \mathbf{1}\left(y^{(i)} = \mathrm{j}\right) x^{(i)}$

  **if** $y == y_{\text{old}}$

    break

return $\mu, y$




K-MEANS$(k, \tau, \{x^{(i)}\}_{i=1}^{n})$

$\mu, y =$ random initialization

**for** $t = 1$ to $\tau$

    $y_{\text{old}} = y$

    **for** $i = 1$ to $n$

        $y^{(i)} = \arg\min_j \left\| x^{(i)} - \mu^{(j)} \right\|^2$

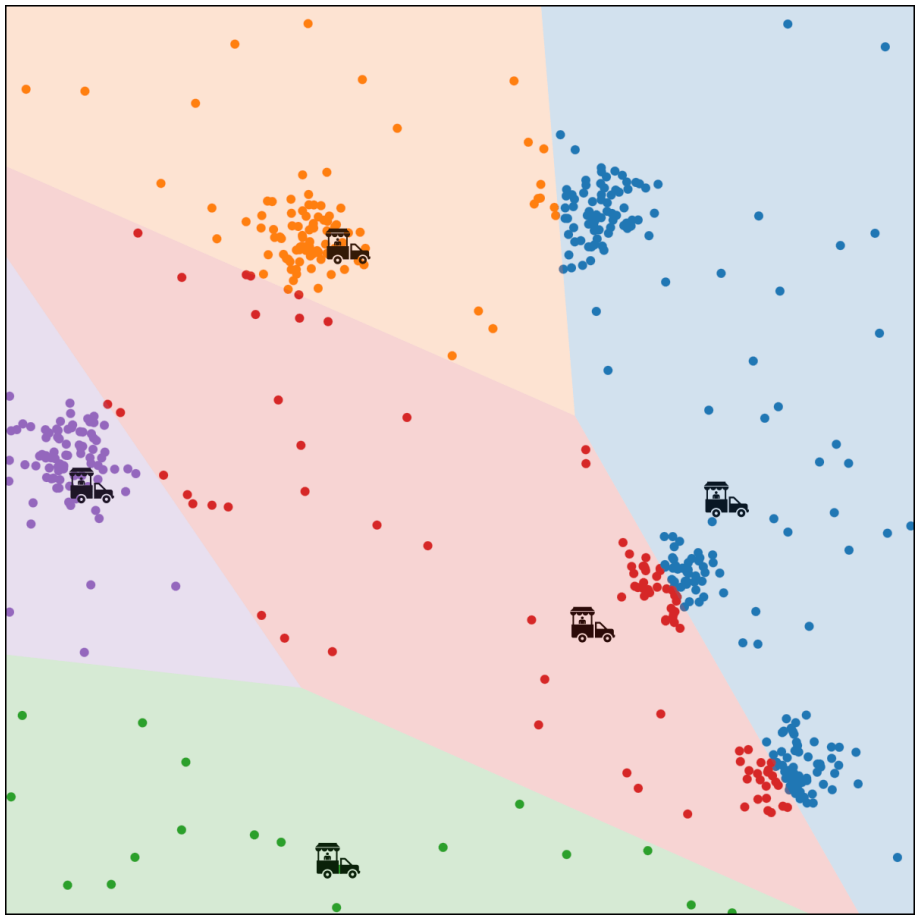
    **for** $j = 1$ to $k$

        $\mu^{(j)} = \frac{1}{N_j} \sum_{i=1}^{n} \mathbf{1}\left(y^{(i)} = \mathrm{j}\right) x^{(i)}$

    **if** $y == y_{\text{old}}$

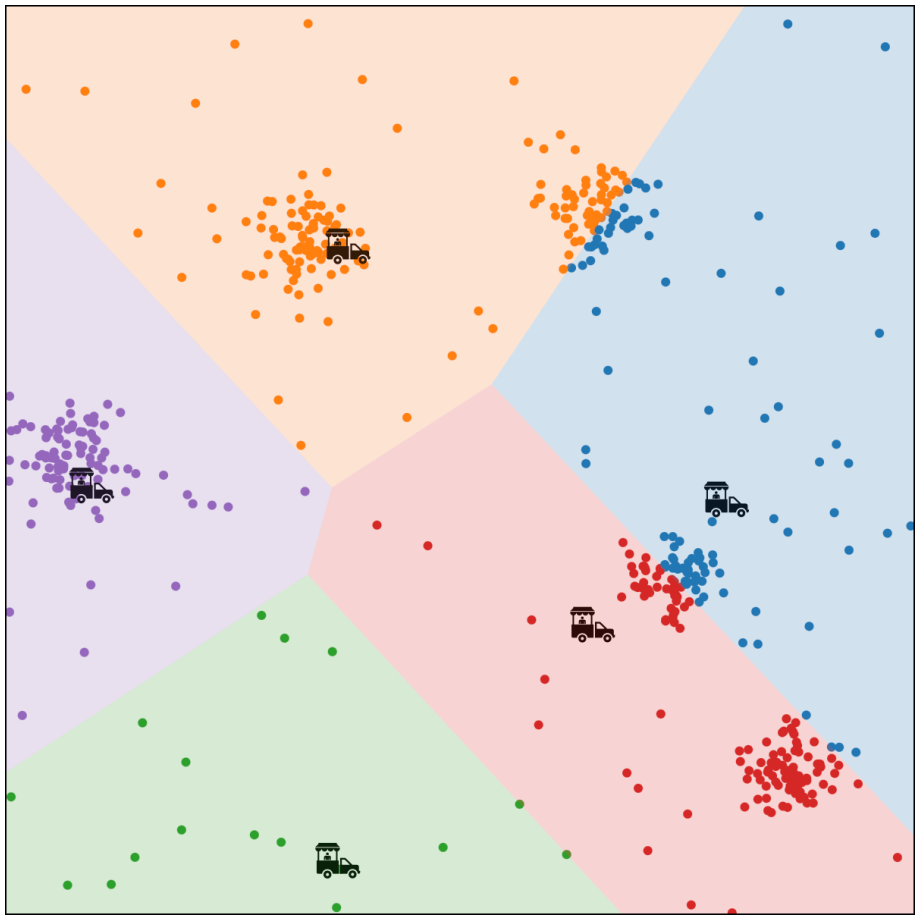
        break

return $\mu, y$




K-MEANS$(k, \tau, \{x^{(i)}\}_{i=1}^{n})$

$\mu, y =$ random initialization
**for** $t = 1$ to $\tau$
    $y_{\text{old}} = y$
    **for** $i = 1$ to $n$
        $y^{(i)} = \arg\min_j \left\| x^{(i)} - \mu^{(j)} \right\|^2$
    **for** $j = 1$ to $k$
        $\mu^{(j)} = \frac{1}{N_j} \sum_{i=1}^{n} \mathbf{1}\left(y^{(i)} = \mathrm{j}\right) x^{(i)}$
    **if** $y == y_{\text{old}}$
        break
return $\mu, y$




K-MEANS$(k, \tau, \{x^{(i)}\}_{i=1}^{n})$

1 $\mu, y =$ random initialization

2 **for** $t = 1$ to $\tau$

3 $\quad y_{\text{old}} = y$

4 $\quad$ **for** $i = 1$ to $n$

5 $\quad\quad y^{(i)} = \arg\min_j \left\| x^{(i)} - \mu^{(j)} \right\|^2$

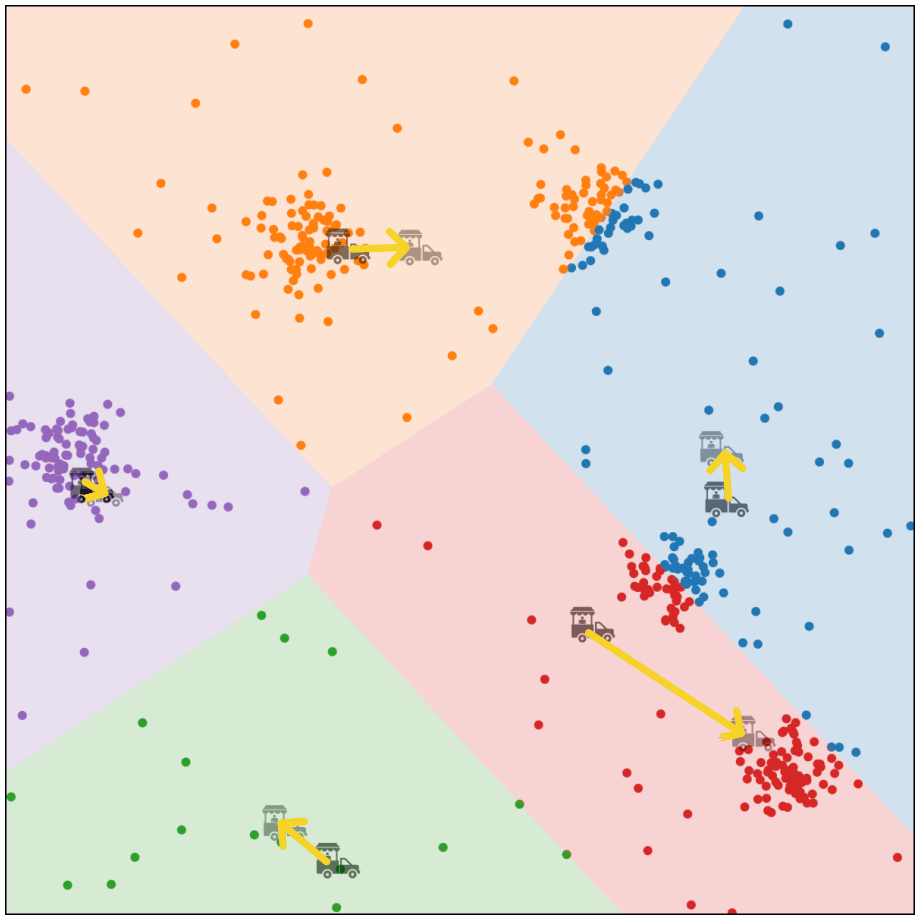
6 $\quad$ **for** $j = 1$ to $k$

7 $\quad\quad \mu^{(j)} = \frac{1}{N_j} \sum_{i=1}^{n} \mathbf{1}\left(y^{(i)} = \mathrm{j}\right) x^{(i)}$

8 $\quad$ **if** $y == y_{\text{old}}$

9 $\quad\quad$ break

10 return $\mu, y$




K-MEANS$(k, \tau, \{x^{(i)}\}_{i=1}^{n})$

1 $\mu, y =$ random initialization

2 **for** $t = 1$ to $\tau$

3 $\quad y_{\text{old}} = y$

4 $\quad$ **for** $i = 1$ to $n$

5 $\quad\quad y^{(i)} = \arg\min_j \left\| x^{(i)} - \mu^{(j)} \right\|^2$

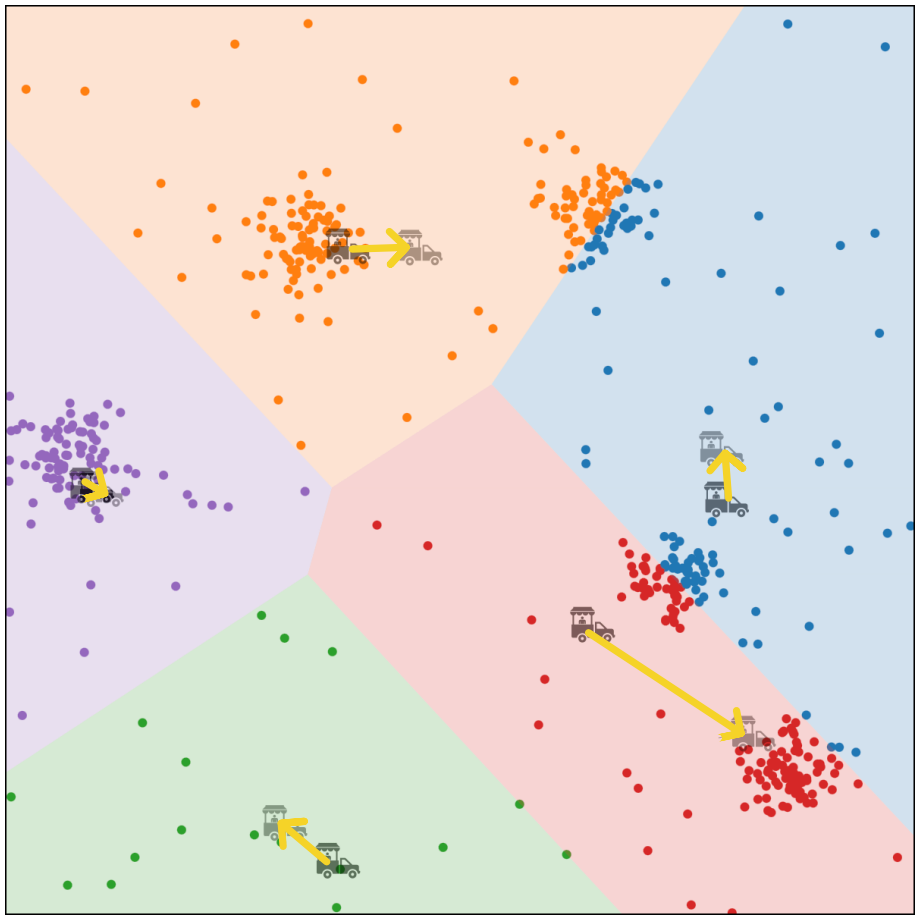
6 $\quad$ **for** $j = 1$ to $k$

7 $\quad\quad \mu^{(j)} = \frac{1}{N_j} \sum_{i=1}^{n} \mathbf{1}\left(y^{(i)} = \mathrm{j}\right) x^{(i)}$

8 $\quad$ **if** $y == y_{\text{old}}$

9 $\quad\quad$ break

10 return $\mu, y$




K-MEANS$(k, \tau, \{x^{(i)}\}_{i=1}^{n})$

$\mu, y =$ random initialization

**for** $t = 1$ to $\tau$

  $y_{\text{old}} = y$

  **for** $i = 1$ to $n$

    $y^{(i)} = \arg\min_j \left\| x^{(i)} - \mu^{(j)} \right\|^2$

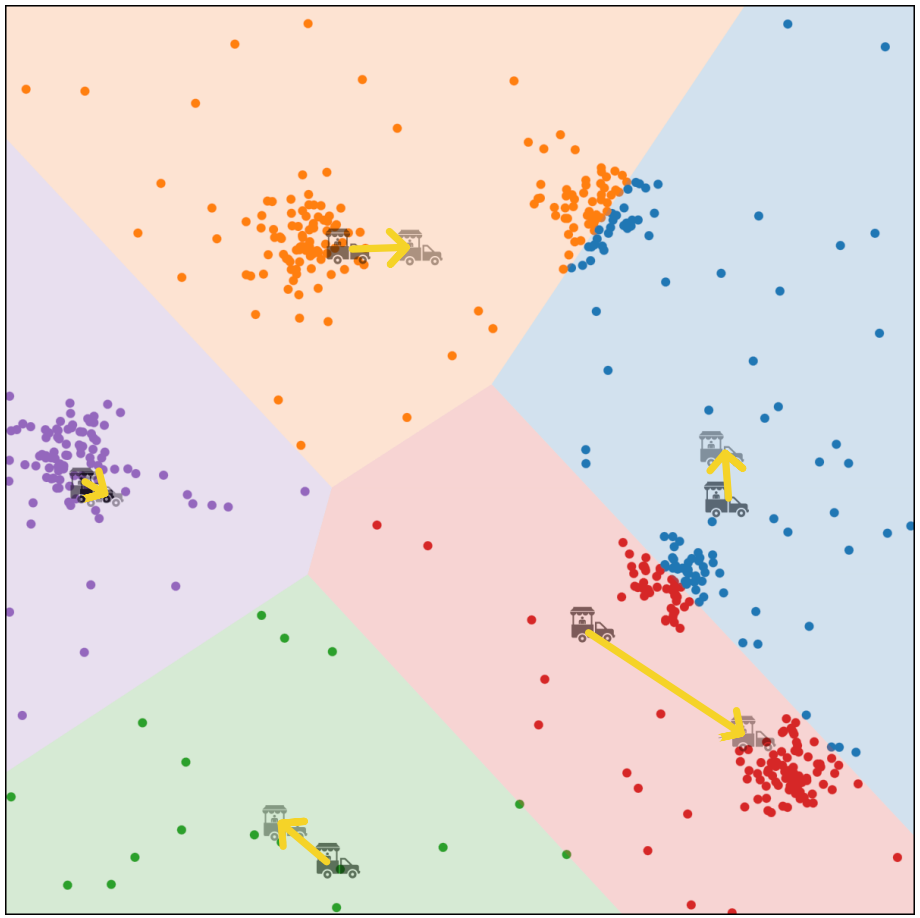
  **for** $j = 1$ to $k$

    $\mu^{(j)} = \frac{1}{N_j} \sum_{i=1}^{n} \mathbf{1}\left(y^{(i)} = \mathrm{j}\right) x^{(i)}$

  **if** $y == y_{\text{old}}$

    break

return $\mu, y$




K-MEANS$(k, \tau, \{x^{(i)}\}_{i=1}^{n})$

$\mu, y =$ random initialization

**for** $t = 1$ to $\tau$

  $y_{\text{old}} = y$

  **for** $i = 1$ to $n$

    $y^{(i)} = \arg\min_j \left\| x^{(i)} - \mu^{(j)} \right\|^2$

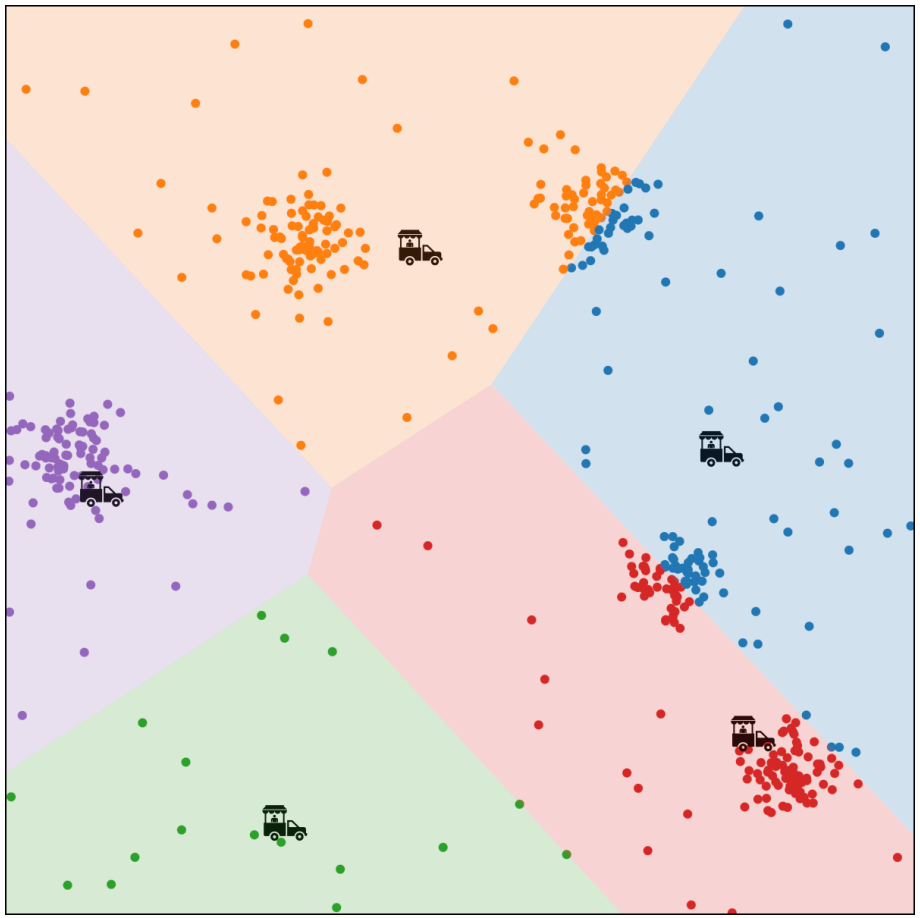
  **for** $j = 1$ to $k$

    $\mu^{(j)} = \frac{1}{N_j} \sum_{i=1}^{n} \mathbf{1}\left(y^{(i)} = \mathrm{j}\right) x^{(i)}$

  **if** $y == y_{\text{old}}$

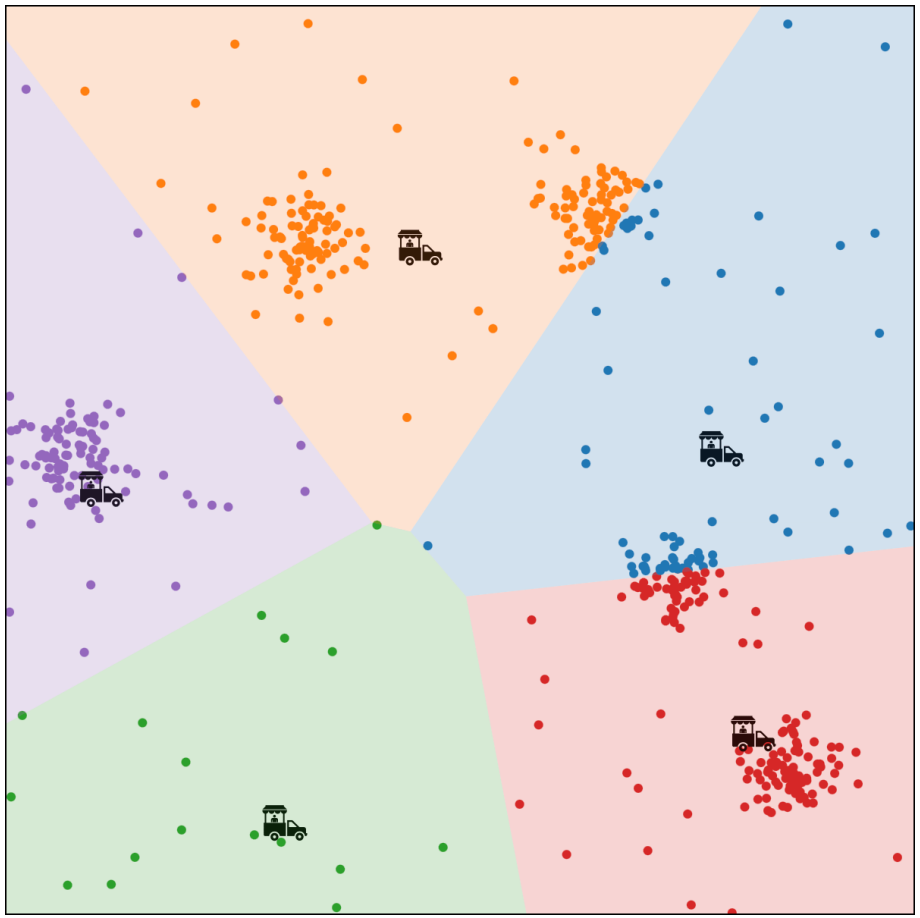
    break

return $\mu, y$




K-MEANS$(k, \tau, \{x^{(i)}\}_{i=1}^{n})$

$\mu, y =$ random initialization
**for** $t = 1$ to $\tau$
    $y_{\text{old}} = y$
    **for** $i = 1$ to $n$
        $y^{(i)} = \arg\min_j \left\| x^{(i)} - \mu^{(j)} \right\|^2$
    **for** $j = 1$ to $k$
        $\mu^{(j)} = \frac{1}{N_j} \sum_{i=1}^{n} \mathbf{1}\left(y^{(i)} = \mathrm{j}\right) x^{(i)}$
    **if** $y == y_{\text{old}}$
        break
return $\mu, y$




K-MEANS$(k, \tau, \{x^{(i)}\}_{i=1}^{n})$

1 $\mu, y =$ random initialization

2 **for** $t = 1$ to $\tau$

3 $\quad y_{\text{old}} = y$

4 $\quad$ **for** $i = 1$ to $n$

5 $\quad\quad y^{(i)} = \arg\min_j \left\| x^{(i)} - \mu^{(j)} \right\|^2$

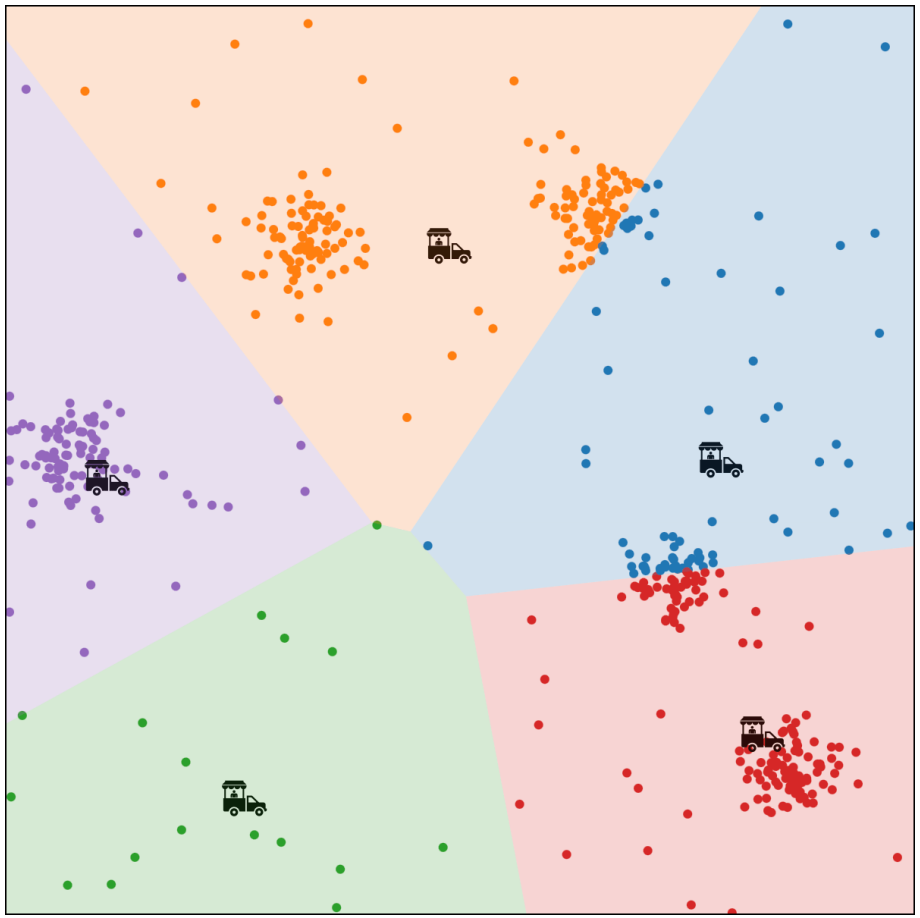
6 $\quad$ **for** $j = 1$ to $k$

7 $\quad\quad \mu^{(j)} = \frac{1}{N_j} \sum_{i=1}^{n} \mathbf{1}\left(y^{(i)} = \mathrm{j}\right) x^{(i)}$

8 $\quad$ **if** $y == y_{\text{old}}$

9 $\quad\quad$ break

10 return $\mu, y$




K-MEANS$(k, \tau, \{x^{(i)}\}_{i=1}^{n})$

1 $\mu, y =$ random initialization

2 **for** $t = 1$ to $\tau$

3 $\quad y_{\text{old}} = y$

4 $\quad$ **for** $i = 1$ to $n$

5 $\quad\quad y^{(i)} = \arg\min_j \left\| x^{(i)} - \mu^{(j)} \right\|^2$

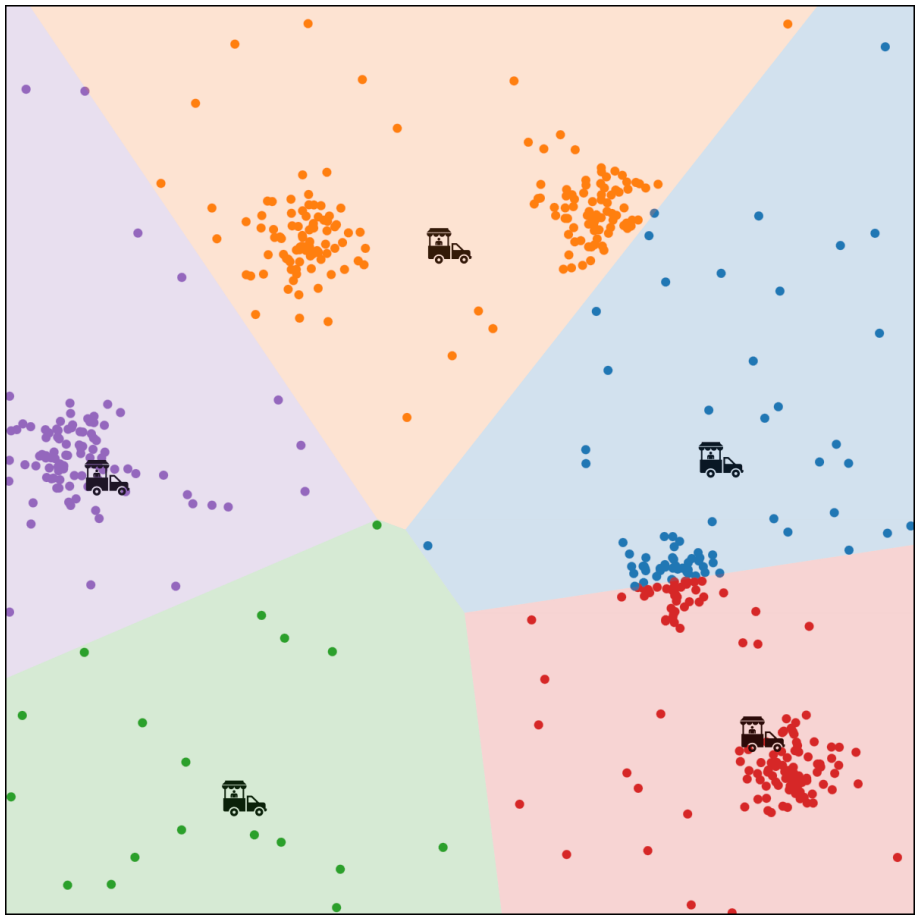
6 $\quad$ **for** $j = 1$ to $k$

7 $\quad\quad \mu^{(j)} = \frac{1}{N_j} \sum_{i=1}^{n} \mathbf{1}\left(y^{(i)} = \mathrm{j}\right) x^{(i)}$

8 $\quad$ **if** $y == y_{\text{old}}$

9 $\quad\quad$ break

10 return $\mu, y$




K-MEANS$(k, \tau, \{x^{(i)}\}_{i=1}^{n})$

1 $\mu, y =$ random initialization

2 **for** $t = 1$ to $\tau$

3 $\quad y_{\text{old}} = y$

4 $\quad$ **for** $i = 1$ to $n$

5 $\quad\quad y^{(i)} = \arg\min_j \left\| x^{(i)} - \mu^{(j)} \right\|^2$

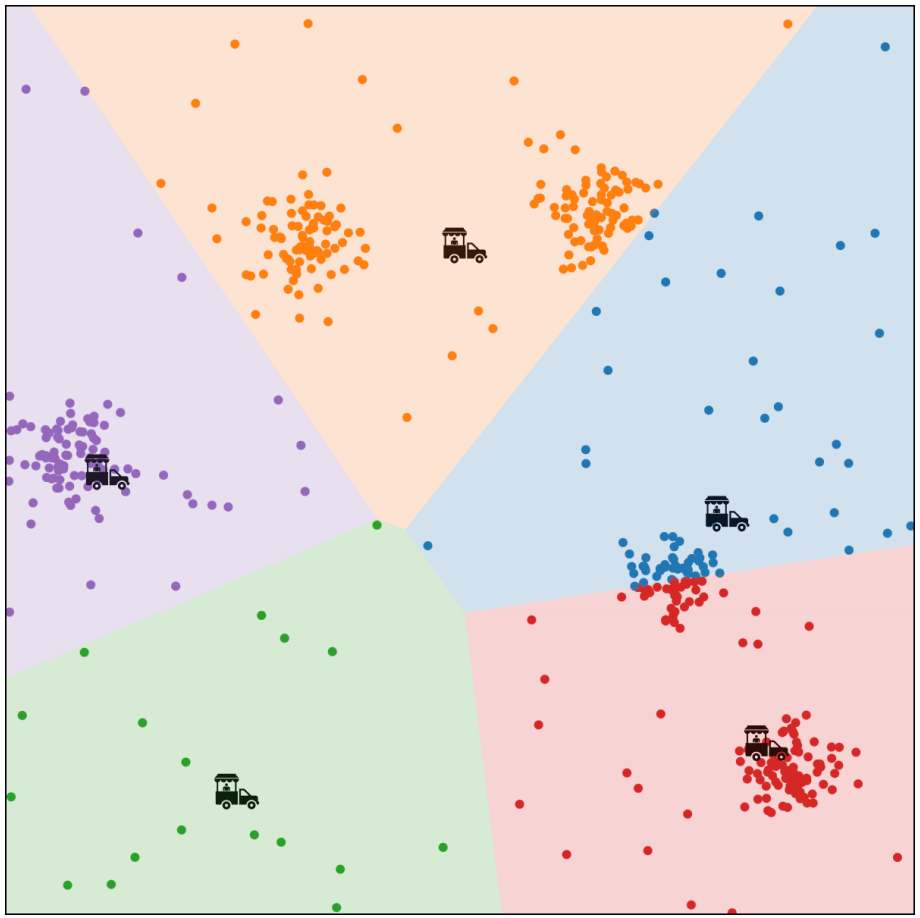
6 $\quad$ **for** $j = 1$ to $k$

7 $\quad\quad \mu^{(j)} = \frac{1}{N_j} \sum_{i=1}^{n} \mathbf{1}\left(y^{(i)} = \mathrm{j}\right) x^{(i)}$

8 $\quad$ **if** $y == y_{\text{old}}$

9 $\quad\quad$ break

10 return $\mu, y$




K-MEANS$(k, \tau, \{x^{(i)}\}_{i=1}^{n})$

1 $\mu, y =$ random initialization

2 **for** $t = 1$ to $\tau$

3 $\quad y_{\text{old}} = y$

4 $\quad$ **for** $i = 1$ to $n$

5 $\quad\quad y^{(i)} = \arg\min_j \left\| x^{(i)} - \mu^{(j)} \right\|^2$

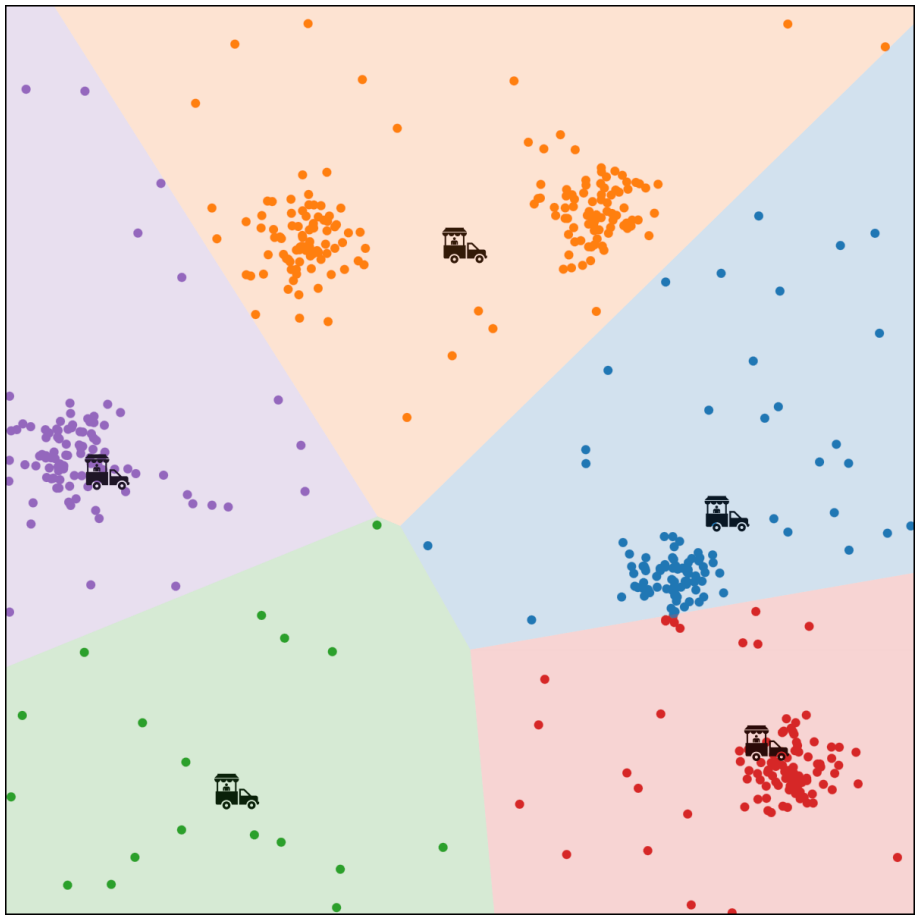
6 $\quad$ **for** $j = 1$ to $k$

7 $\quad\quad \mu^{(j)} = \frac{1}{N_j} \sum_{i=1}^{n} \mathbf{1}\left(y^{(i)} = \mathrm{j}\right) x^{(i)}$

8 $\quad$ **if** $y == y_{\text{old}}$

9 $\quad\quad$ break

10 return $\mu, y$




K-MEANS$(k, \tau, \{x^{(i)}\}_{i=1}^{n})$

1 $\mu, y =$ random initialization

2 **for** $t = 1$ to $\tau$

3 $\quad y_{\text{old}} = y$

4 $\quad$ **for** $i = 1$ to $n$

5 $\quad\quad y^{(i)} = \arg\min_j \left\| x^{(i)} - \mu^{(j)} \right\|^2$

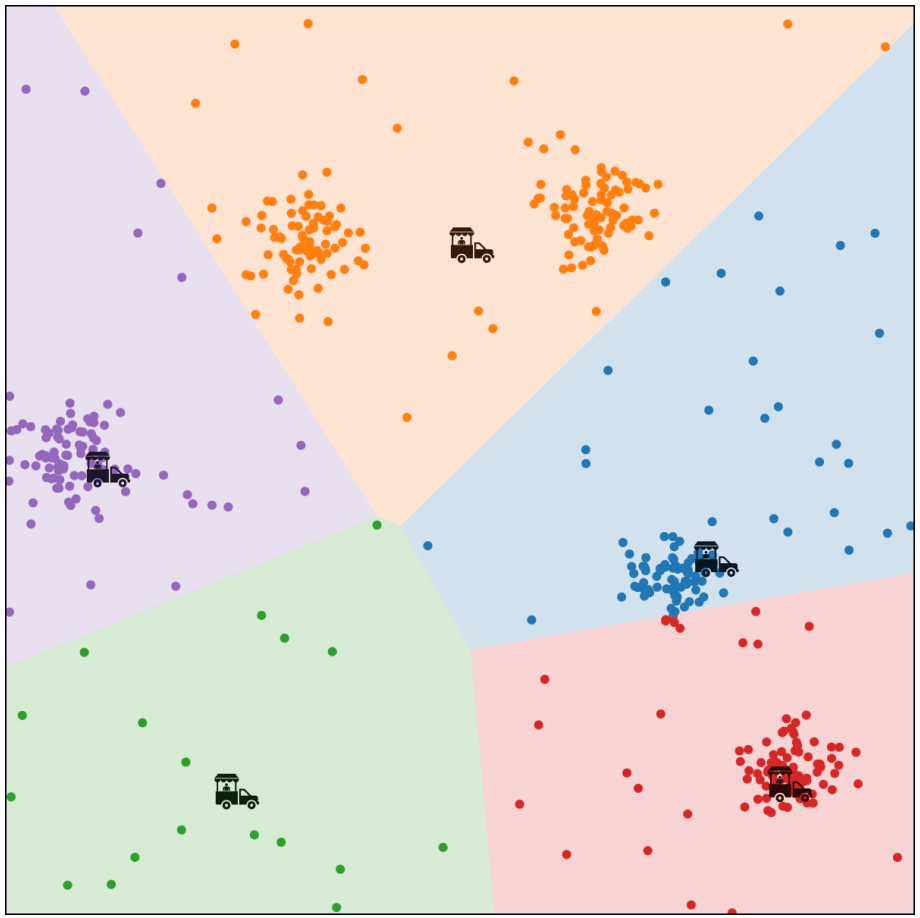
6 $\quad$ **for** $j = 1$ to $k$

7 $\quad\quad \mu^{(j)} = \frac{1}{N_j} \sum_{i=1}^{n} \mathbf{1}\left(y^{(i)} = \mathrm{j}\right) x^{(i)}$

8 $\quad$ **if** $y == y_{\text{old}}$

9 $\quad\quad$ break

10 return $\mu, y$




K-MEANS$(k, \tau, \{x^{(i)}\}_{i=1}^{n})$

1 $\mu, y =$ random initialization

2 **for** $t = 1$ to $\tau$

3 $\quad y_{\text{old}} = y$

4 $\quad$ **for** $i = 1$ to $n$

5 $\quad\quad y^{(i)} = \arg\min_j \left\| x^{(i)} - \mu^{(j)} \right\|^2$

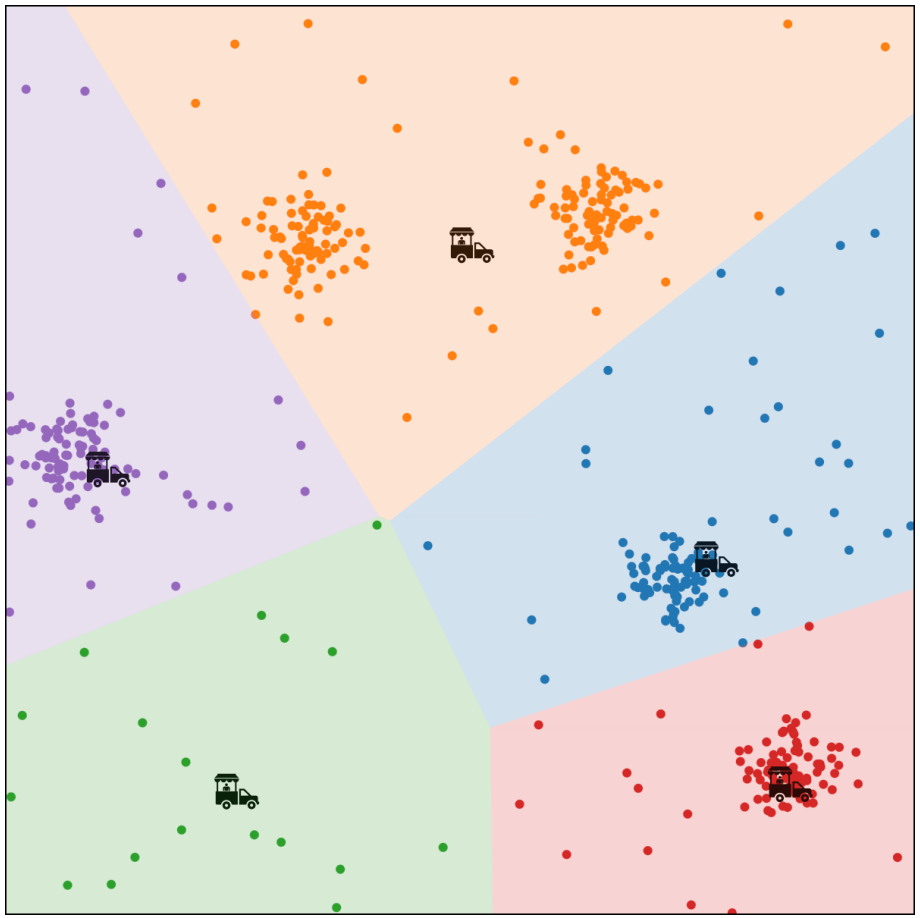
6 $\quad$ **for** $j = 1$ to $k$

7 $\quad\quad \mu^{(j)} = \frac{1}{N_j} \sum_{i=1}^{n} \mathbf{1}\left(y^{(i)} = \mathrm{j}\right) x^{(i)}$

8 $\quad$ **if** $y == y_{\text{old}}$

9 $\quad\quad$ break

10 return $\mu, y$




K-MEANS$(k, \tau, \{x^{(i)}\}_{i=1}^{n})$

1 $\mu, y =$ random initialization

2 **for** $t = 1$ to $\tau$

3 $\quad y_{\text{old}} = y$

4 $\quad$ **for** $i = 1$ to $n$

5 $\quad\quad y^{(i)} = \arg\min_j \left\| x^{(i)} - \mu^{(j)} \right\|^2$

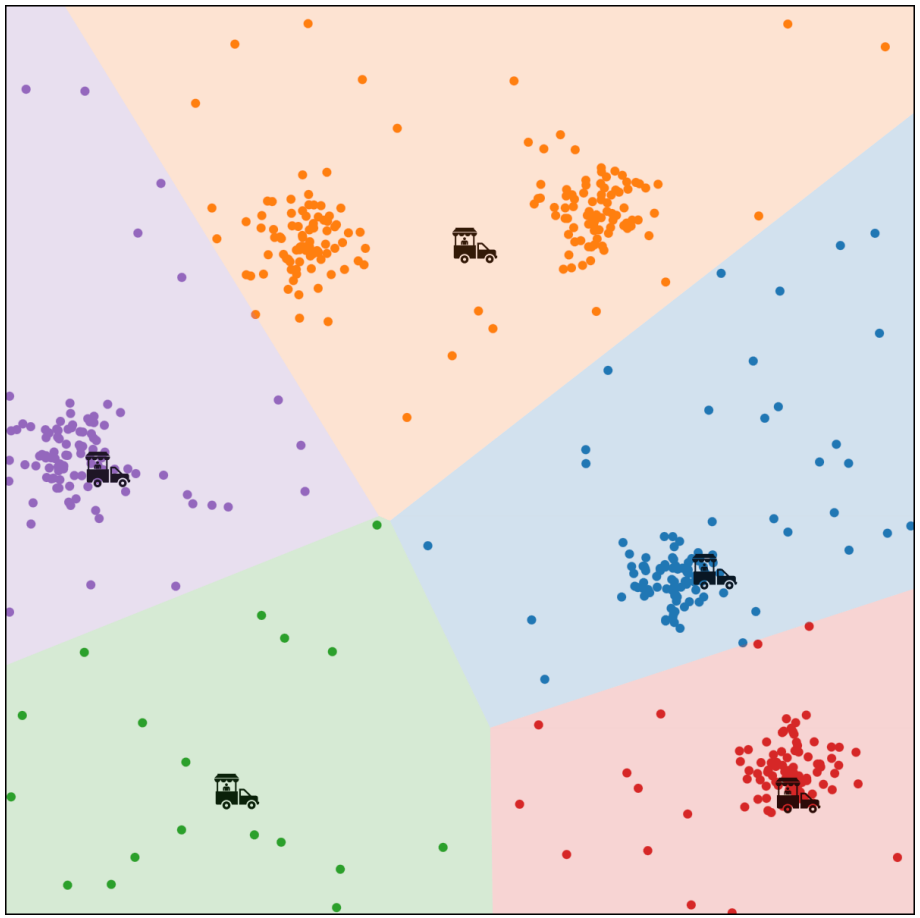
6 $\quad$ **for** $j = 1$ to $k$

7 $\quad\quad \mu^{(j)} = \frac{1}{N_j} \sum_{i=1}^{n} \mathbf{1}\left(y^{(i)} = \mathrm{j}\right) x^{(i)}$

8 $\quad$ **if** $y == y_{\text{old}}$

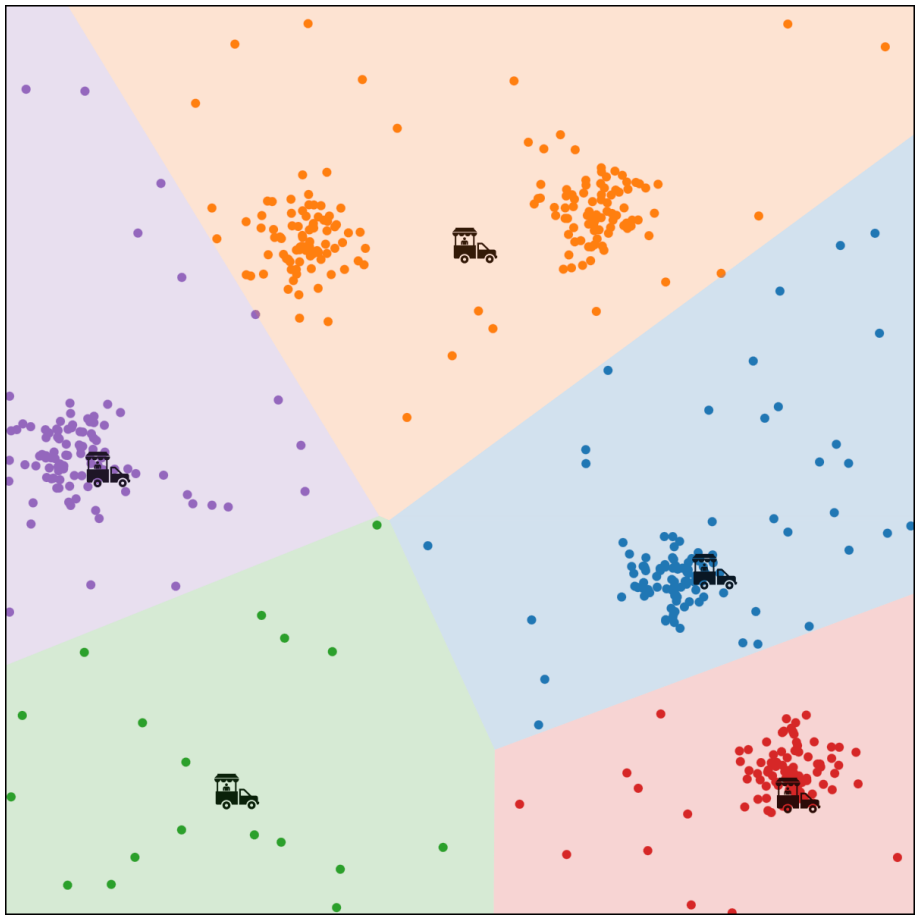
9 $\quad\quad$ break

10 return $\mu, y$




K-MEANS$(k, \tau, \{x^{(i)}\}_{i=1}^{n})$

1 $\mu, y =$ random initialization
2 **for** $t = 1$ to $\tau$
3 $\quad y_{\text{old}} = y$
4 $\quad$ **for** $i = 1$ to $n$
5 $\quad\quad y^{(i)} = \arg\min_j \left\| x^{(i)} - \mu^{(j)} \right\|^2$
6 $\quad$ **for** $j = 1$ to $k$
7 $\quad\quad \mu^{(j)} = \frac{1}{N_j} \sum_{i=1}^{n} \mathbf{1}\left(y^{(i)} = \mathrm{j}\right) x^{(i)}$
8 $\quad$ **if** $y == y_{\text{old}}$
9 $\quad\quad$ break
10 return $\mu, y$




K-MEANS$(k, \tau, \{x^{(i)}\}_{i=1}^{n})$

1 $\mu, y =$ random initialization

2 **for** $t = 1$ to $\tau$

3 $\quad y_{\text{old}} = y$

4 $\quad$ **for** $i = 1$ to $n$

5 $\quad\quad y^{(i)} = \arg\min_j \left\| x^{(i)} - \mu^{(j)} \right\|^2$

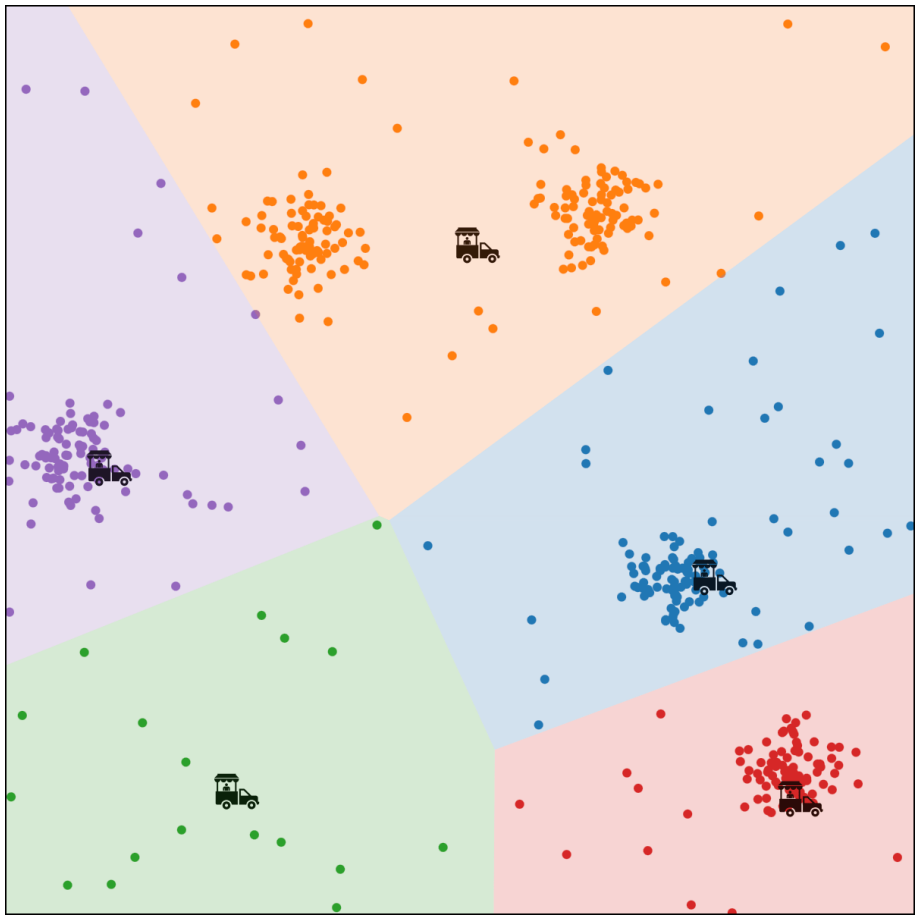
6 $\quad$ **for** $j = 1$ to $k$

7 $\quad\quad \mu^{(j)} = \frac{1}{N_j} \sum_{i=1}^{n} \mathbf{1}\left(y^{(i)} = \mathrm{j}\right) x^{(i)}$

8 $\quad$ **if** $y == y_{\text{old}}$

9 $\quad\quad$ break

10 return $\mu, y$




K-MEANS$(k, \tau, \{x^{(i)}\}_{i=1}^{n})$

1 $\mu, y =$ random initialization

2 **for** $t = 1$ to $\tau$

3 $\quad y_{\text{old}} = y$

4 $\quad$ **for** $i = 1$ to $n$

5 $\quad\quad y^{(i)} = \arg\min_j \left\| x^{(i)} - \mu^{(j)} \right\|^2$

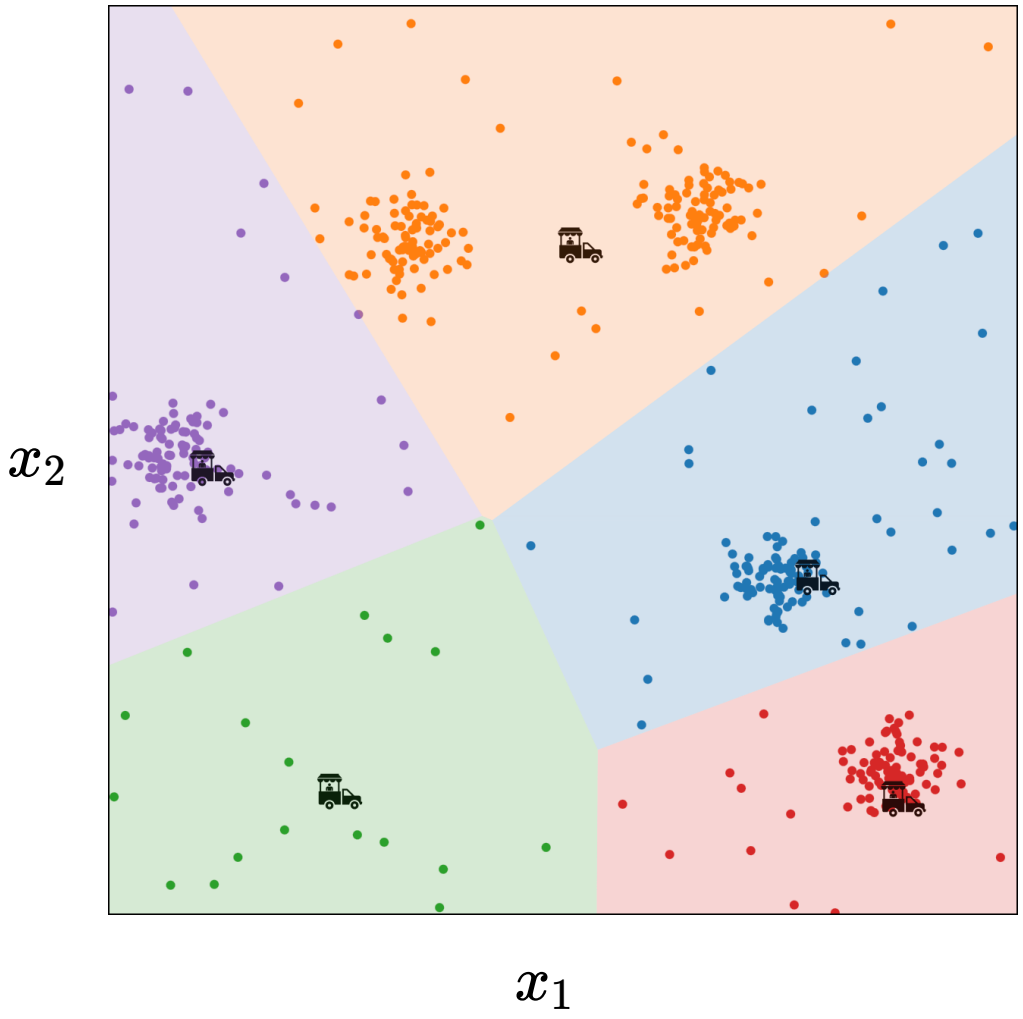
6 $\quad$ **for** $j = 1$ to $k$

7 $\quad\quad \mu^{(j)} = \frac{1}{N_j} \sum_{i=1}^{n} \mathbf{1}\left(y^{(i)} = \mathrm{j}\right) x^{(i)}$

8 $\quad$ **if** $y == y_{\text{old}}$

9 $\quad\quad$ break

10 return $\mu, y$




K-MEANS$(k, \tau, \{x^{(i)}\}_{i=1}^{n})$

$\mu, y =$ random initialization

**for** $t = 1$ to $\tau$

  $y_{\text{old}} = y$

  **for** $i = 1$ to $n$

    $y^{(i)} = \arg\min_j \left\| x^{(i)} - \mu^{(j)} \right\|^2$

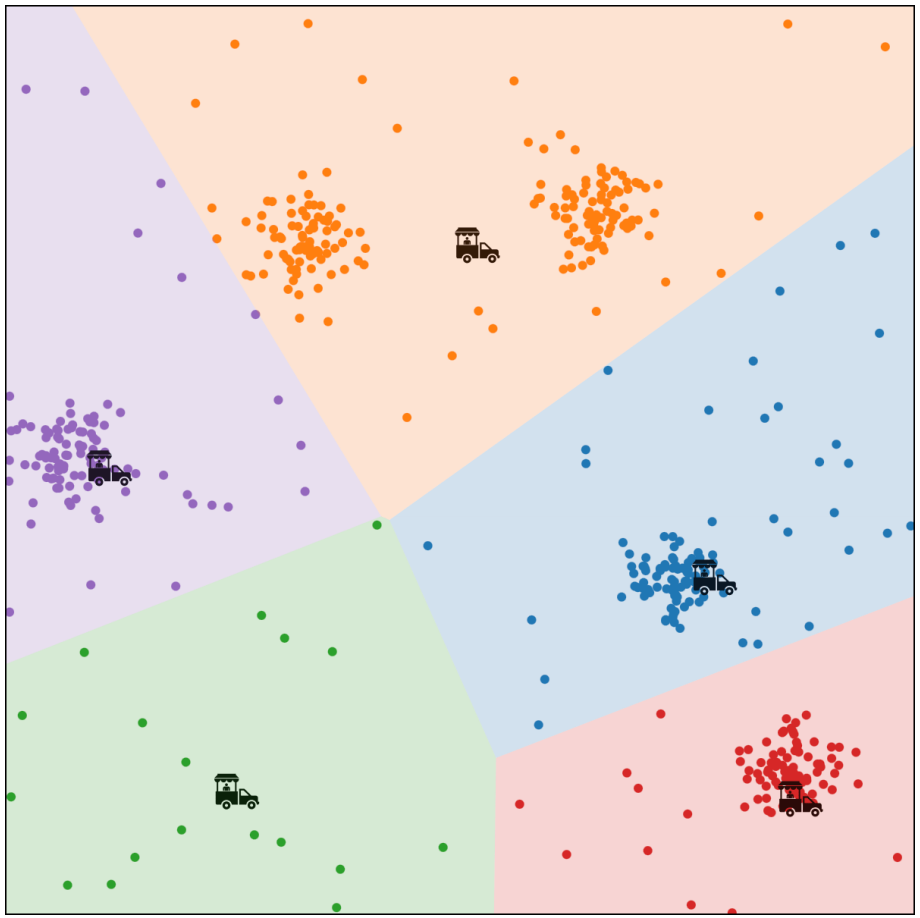
  **for** $j = 1$ to $k$

    $\mu^{(j)} = \frac{1}{N_j} \sum_{i=1}^{n} \mathbf{1}\left(y^{(i)} = \mathrm{j}\right) x^{(i)}$

  **if** $y == y_{\text{old}}$

    break

return $\mu, y$




K-MEANS$(k, \tau, \{x^{(i)}\}_{i=1}^{n})$

1 $\mu, y =$ random initialization

2 **for** $t = 1$ to $\tau$

3 $\quad y_{\text{old}} = y$

4 $\quad$ **for** $i = 1$ to $n$

5 $\quad\quad y^{(i)} = \arg\min_j \left\| x^{(i)} - \mu^{(j)} \right\|^2$

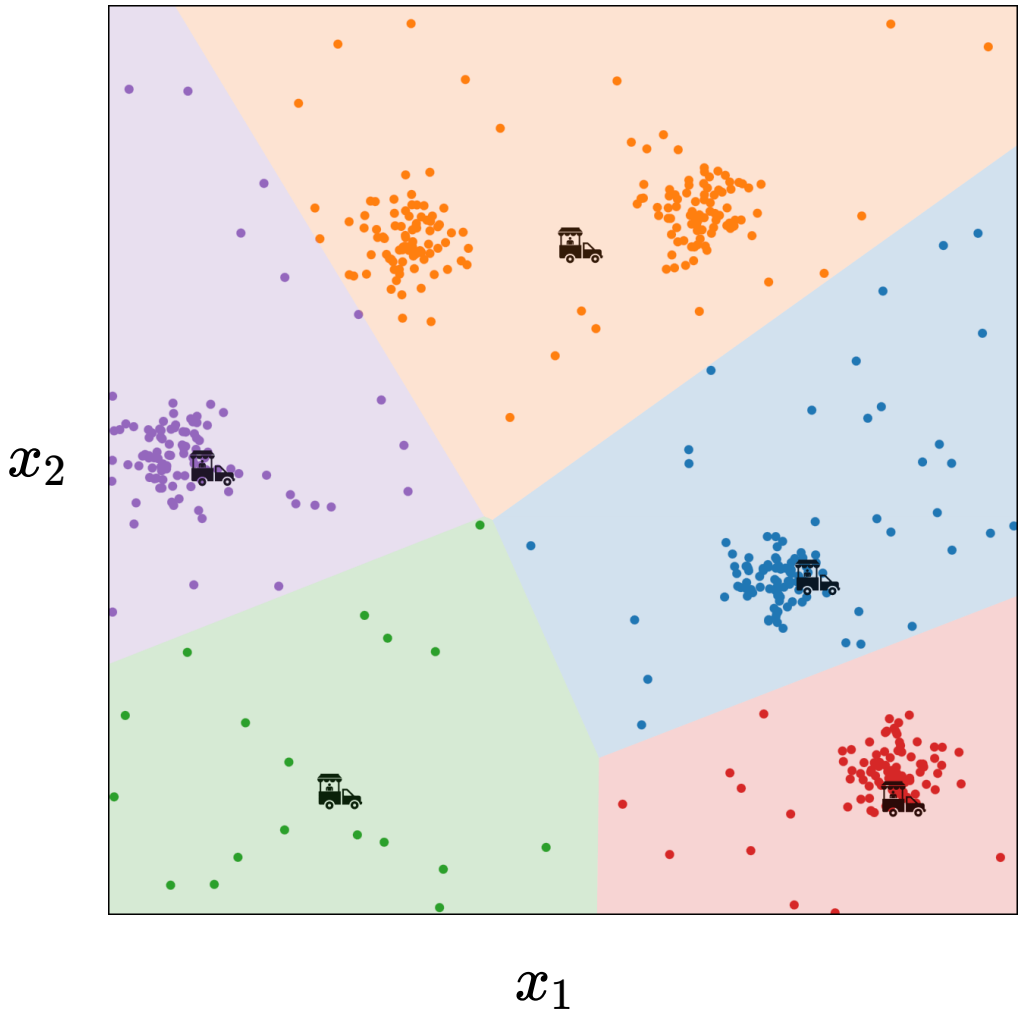
6 $\quad$ **for** $j = 1$ to $k$

7 $\quad\quad \mu^{(j)} = \frac{1}{N_j} \sum_{i=1}^{n} \mathbf{1}\left(y^{(i)} = \mathrm{j}\right) x^{(i)}$

8 $\quad$ **if** $y == y_{\text{old}}$

9 $\quad\quad$ break

10 return $\mu, y$




K-MEANS$(k, \tau, \{x^{(i)}\}_{i=1}^{n})$

1 $\mu, y =$ random initialization

2 **for** $t = 1$ to $\tau$

3 $\quad y_{\text{old}} = y$

4 $\quad$ **for** $i = 1$ to $n$

5 $\quad\quad y^{(i)} = \arg\min_j \left\| x^{(i)} - \mu^{(j)} \right\|^2$

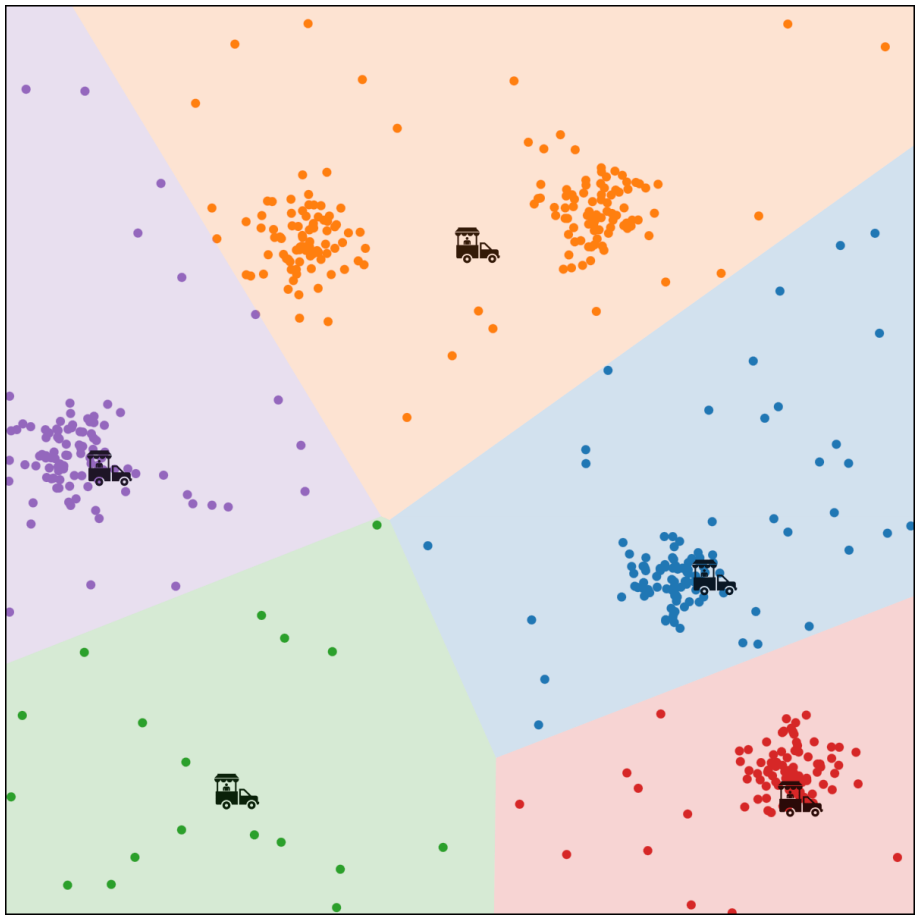
6 $\quad$ **for** $j = 1$ to $k$

7 $\quad\quad \mu^{(j)} = \frac{1}{N_j} \sum_{i=1}^{n} \mathbf{1}\left(y^{(i)} = \mathrm{j}\right) x^{(i)}$

8 $\quad$ **if** $y == y_{\text{old}}$

9 $\quad\quad$ break

10 return $\mu, y$




K-MEANS$(k, \tau, \{x^{(i)}\}_{i=1}^{n})$

1 $\mu, y =$ random initialization

2 **for** $t = 1$ to $\tau$

3 $\quad y_{\text{old}} = y$

4 $\quad$ **for** $i = 1$ to $n$

5 $\quad\quad y^{(i)} = \arg\min_j \left\| x^{(i)} - \mu^{(j)} \right\|^2$

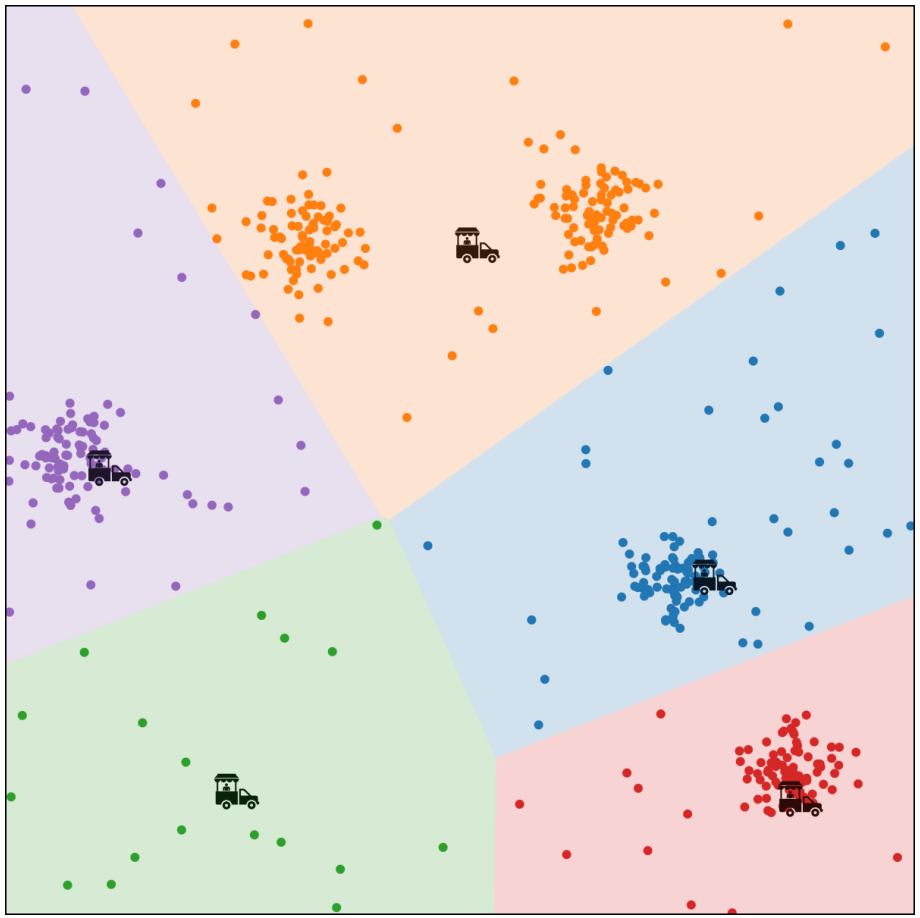
6 $\quad$ **for** $j = 1$ to $k$

7 $\quad\quad \mu^{(j)} = \frac{1}{N_j} \sum_{i=1}^{n} \mathbf{1}\left(y^{(i)} = \mathrm{j}\right) x^{(i)}$

8 $\quad$ **if** $y == y_{\text{old}}$

9 $\quad\quad$ break

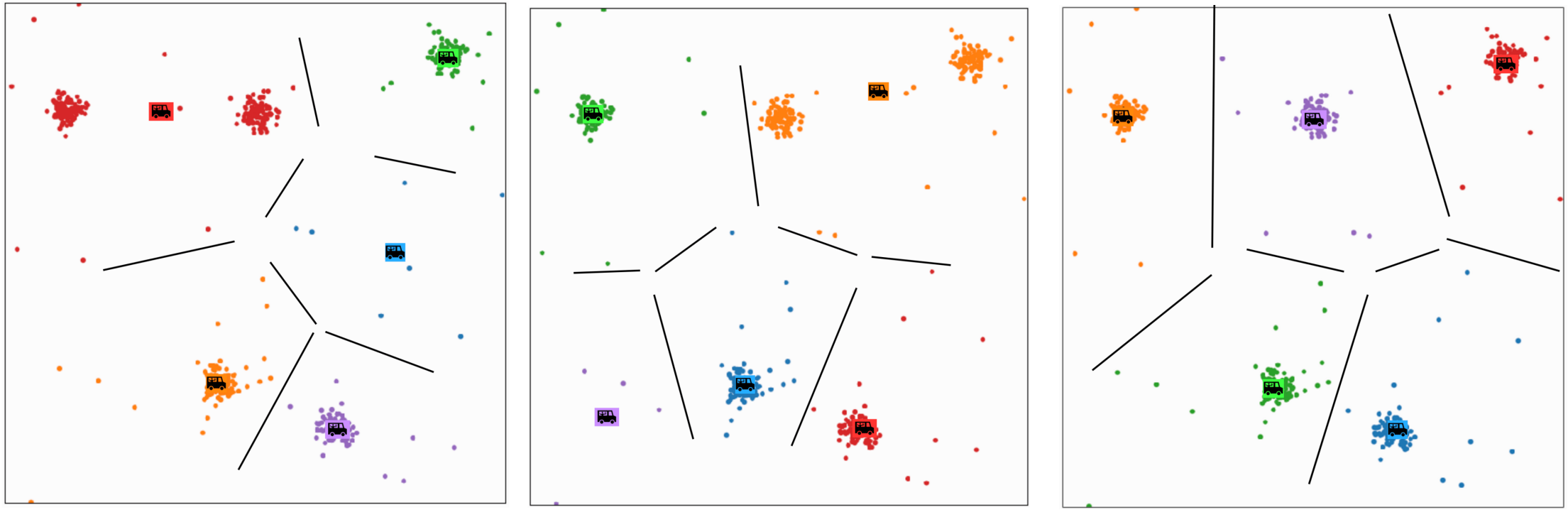
10 return $\mu, y$



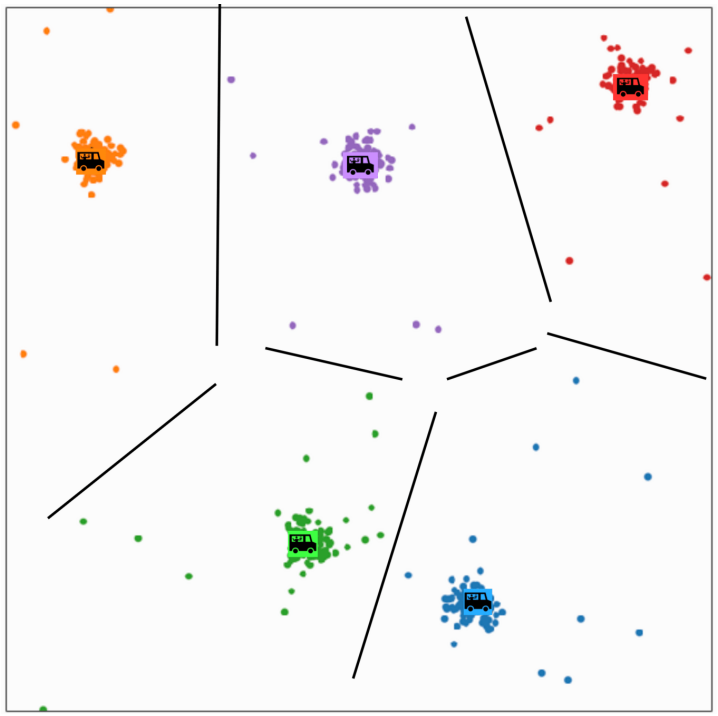

K-MEANS$(k, \tau, \{x^{(i)}\}_{i=1}^{n})$

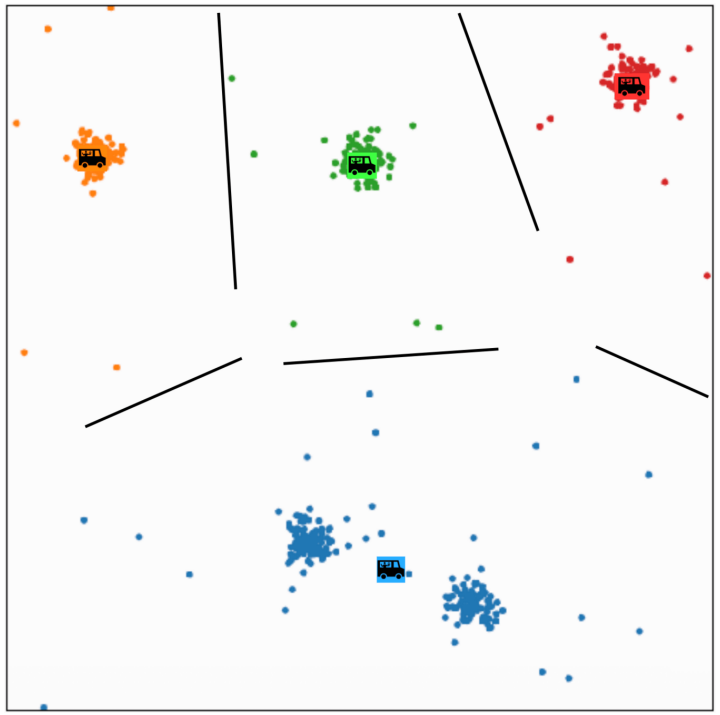
$\mu, y =$ random initialization
**for** $t = 1$ to $\tau$
    $y_{\text{old}} = y$
    **for** $i = 1$ to $n$
        $y^{(i)} = \arg\min_j \left\| x^{(i)} - \mu^{(j)} \right\|^2$
    **for** $j = 1$ to $k$
        $\mu^{(j)} = \frac{1}{N_j} \sum_{i=1}^{n} \mathbf{1}\left(y^{(i)} = \mathrm{j}\right) x^{(i)}$
    **if** $y == y_{\text{old}}$
        break
return $\mu, y$

# $k$-means algorithm is sensitive to initialization

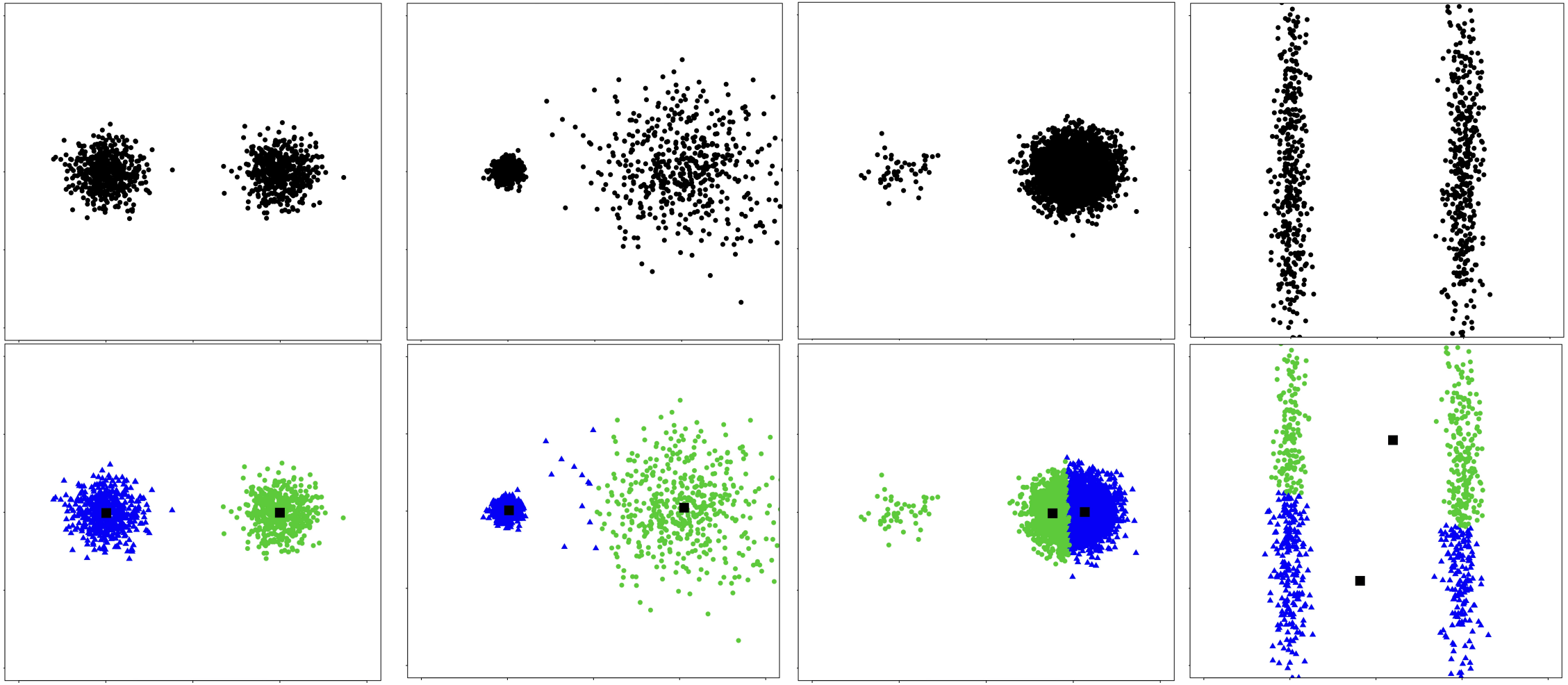


# $k$-means algorithm is sensitive to $k$





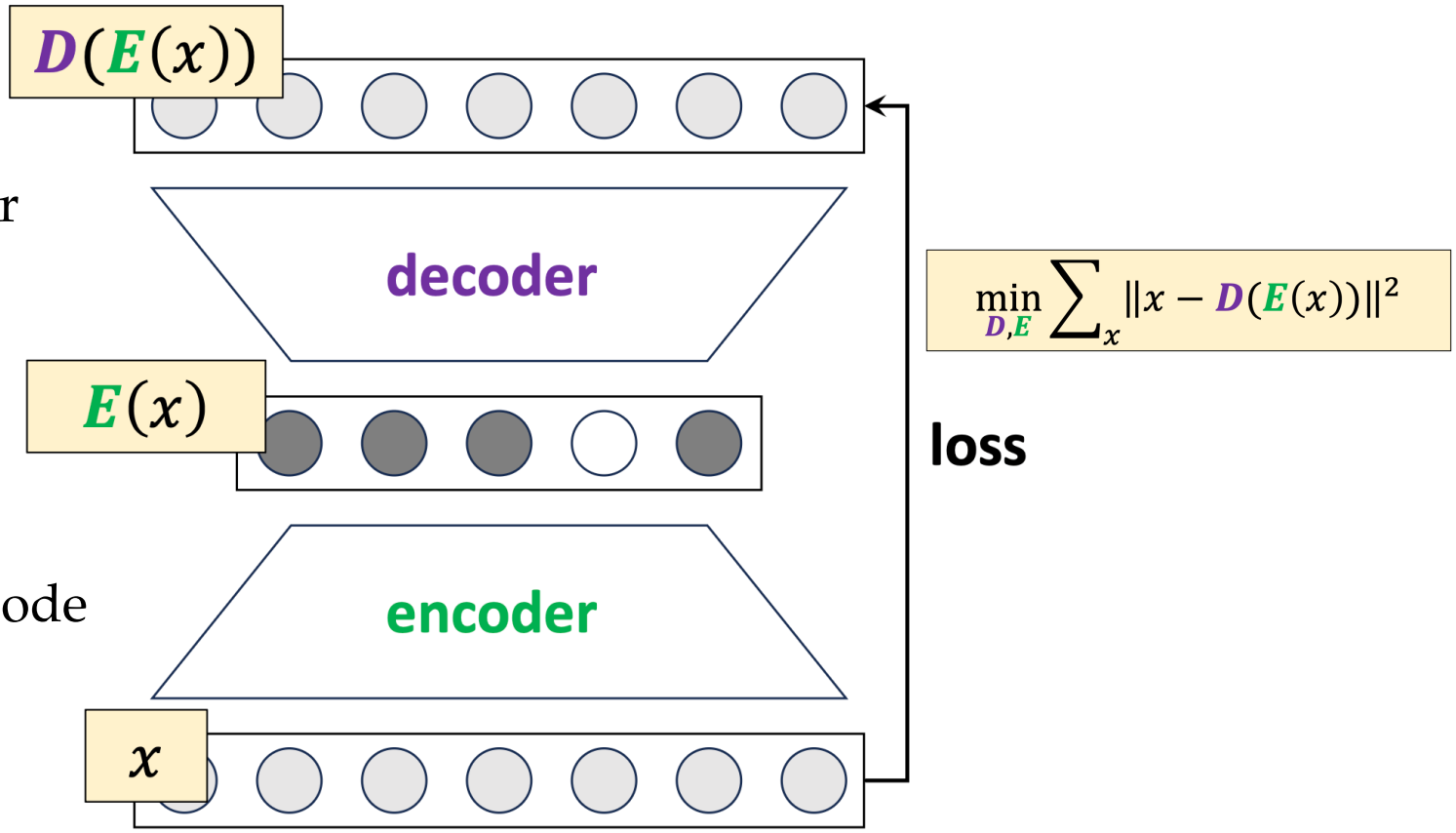
($k$-means works well for well-separated circular clusters of the same size)



($k$-means can be thought of as performing special auto-encodering)

decoder $\boldsymbol{D}$: map a code to a center

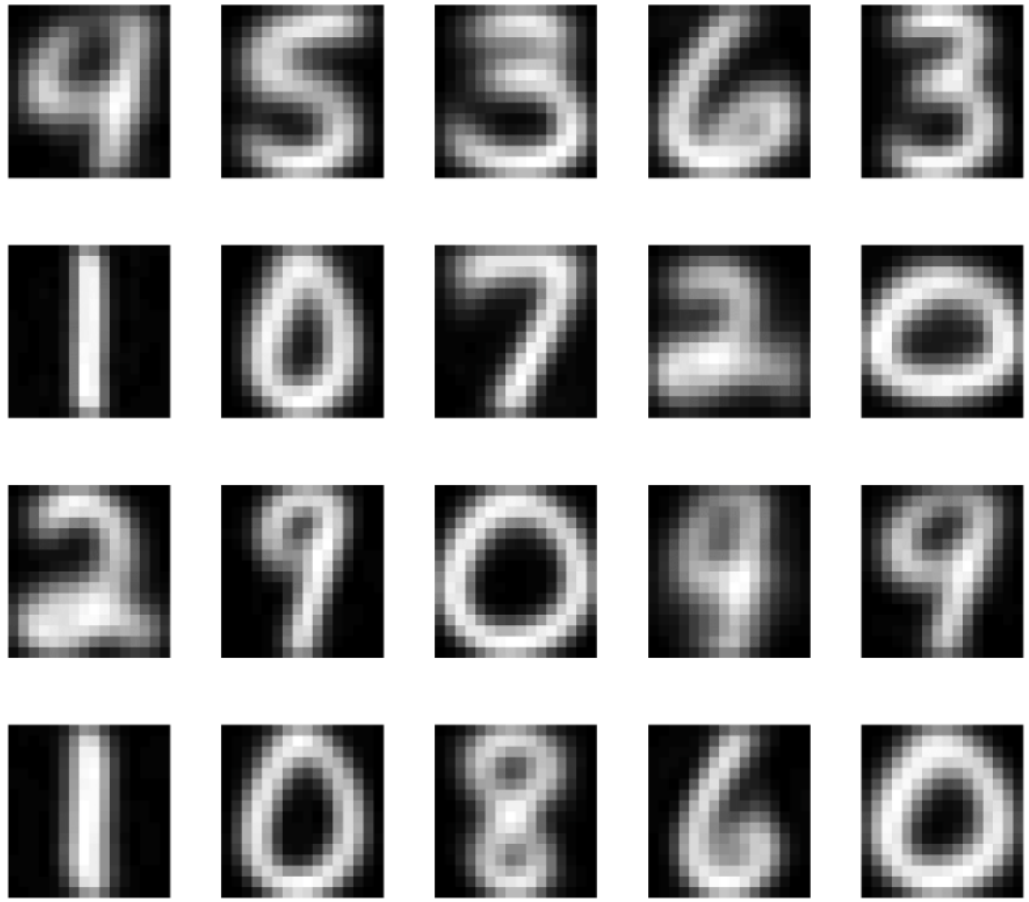
encoder $\boldsymbol{E}$: map $x$ to a (one-hot) code



(cluster center are prototypes)

not actual data points MNIST
However, they clearly show that
MNIST has strong cluster structure
defined by the different digit types.

learned representations



# Summary

- Non-parametric models let data define structure — balancing flexibility, interpretability, and intuition. Complexity grows with data; make few assumptions about functional form.
- $k$-Nearest Neighbors (kNN): Predict by looking at nearby training points; depends on chosen distance metric. Works well for small, low-dim data but costly in high dimensions.
- Decision Trees: Learn interpretable, flow-chart-like rules by recursively splitting data. Regularize by growing deep trees, then pruning to simplify. Ensembles: Combine many simple models so they can vote → stronger, more stable predictors.
- $k$-Means Clustering: Unsupervised grouping of similar data using a distance measure. The k-means algorithm is sensitive to initialization and the choice of $k$.
- kNN ↔ local averaging, Trees ↔ adaptive partition, Ensembles ↔ aggregation, k-Means ↔ unsupervised structure.

https://forms.gle/DefAyvq8KA9kg37X8

We'd love to hear
your thoughts.

# Thanks!