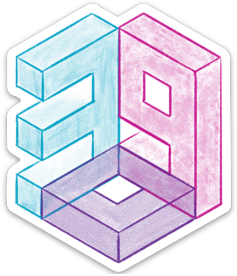


https://introml.mit.edu/

6.390 Intro to Machine Learning

Lecture 12: Non-parametric Models

Shen Shen

May 2, 2025

11am, Room 10-250

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

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

Predictive performance and beyond

There's more to machine learning than predictive performance

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

Oct 29, 2020 6:20 PM EST

"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 [illegible] of its complexity wo[illegible]n elections officials[illegible]

Intelligible Models for HealthCare: Predicting Pneumonia Risk and Hospital 30-day Readmission

Rich Caruana
Microsoft Research
rcaruana@microsoft.com

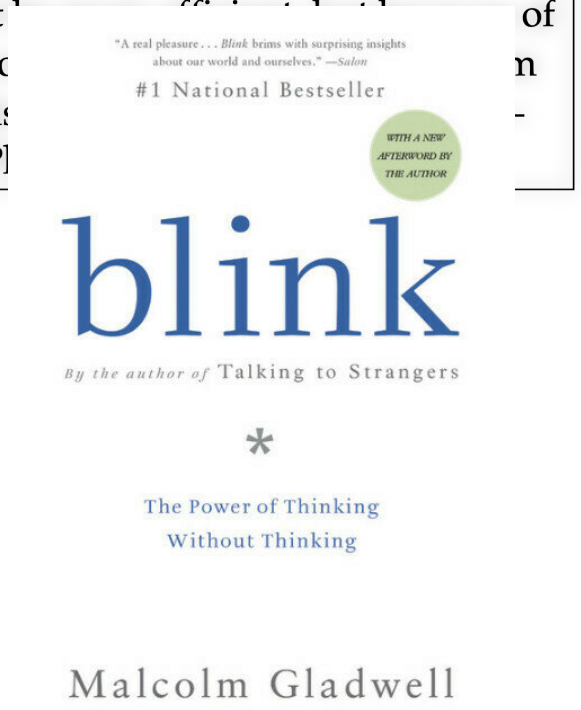
Yin Lou
LinkedIn Corporation
ylou@linkedin.com

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



Predictive performance and beyond

Even if all we care about is vanilla predicative performance, we should care about non-parametric models

A look at Mathurin's toolkit, which he 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 TPOT, auto sklearn
- **Cloud services:** Google colab and kaggle kernels

Non-parametric models

- there are still parameters to be learned to build a hypothesis/model.
- just that, the model/hypothesis does not have a fixed parameterization (e.g. even the number of parameters can change.)

- they are usually fast to implement / train and often very effective.
- often a good baseline (especially when the data doesn't involve complex structures like image or languages)

Outline

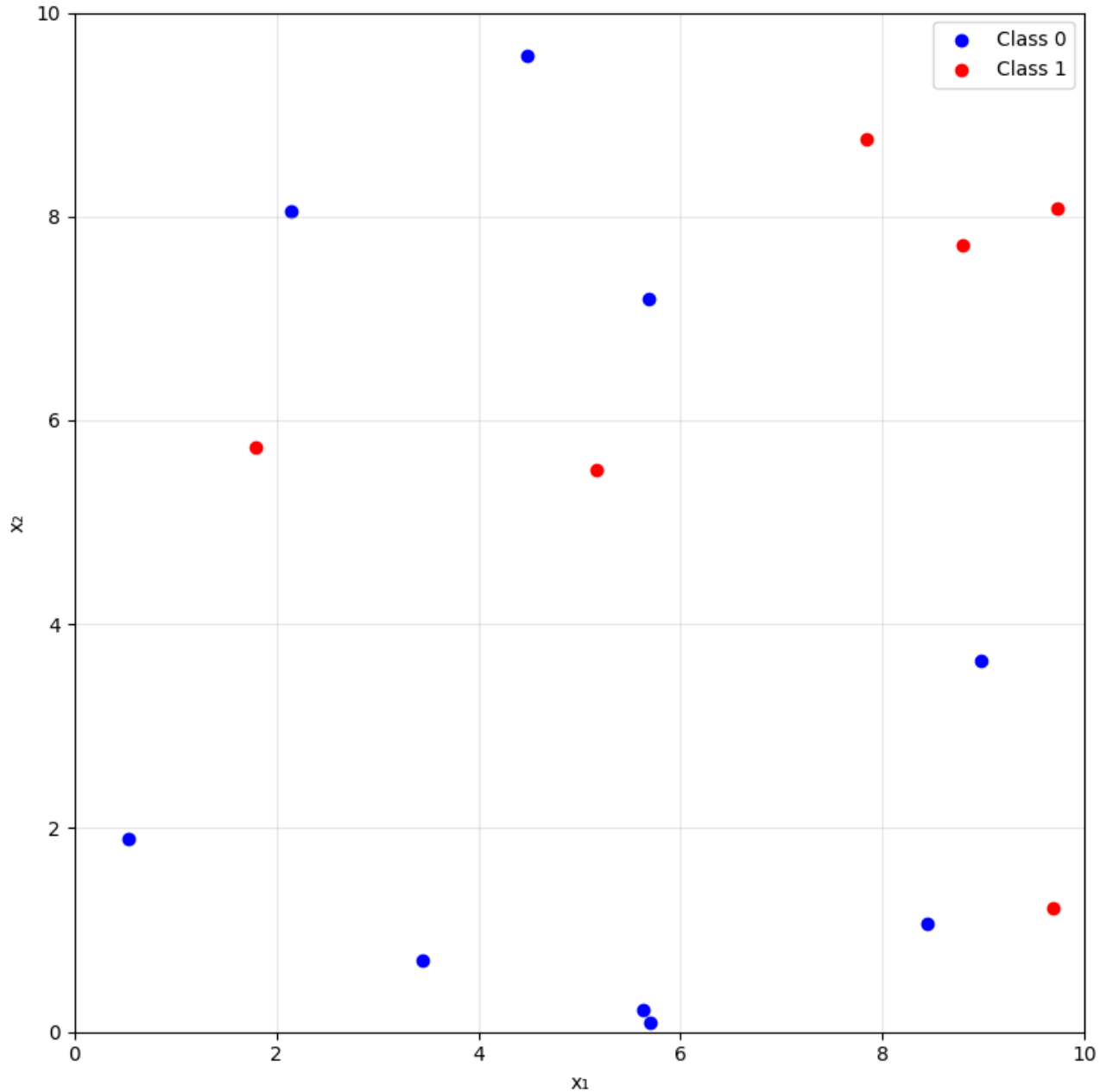
- Non-parametric models overview
- Supervised learning non-parametric
  - k-nearest neighbor
  - Decision Tree (read a tree, BuildTree, Bagging)
- Unsupervised learning non-parametric
  - k-means clustering

- Training: None (or rather: memorize the entire training data)
- Predicting (inferencing, testing):
  - for a new data point x_{new} do:
    - find the k nearest points from the training data to x_{new}
    - For classification: predict label $\hat{y}_{\text{new}}$ by taking a majority vote of the k neighbors's labels
    - For regression: predict label $\hat{y}_{\text{new}}$ by taking an average over the k neighbors' labels

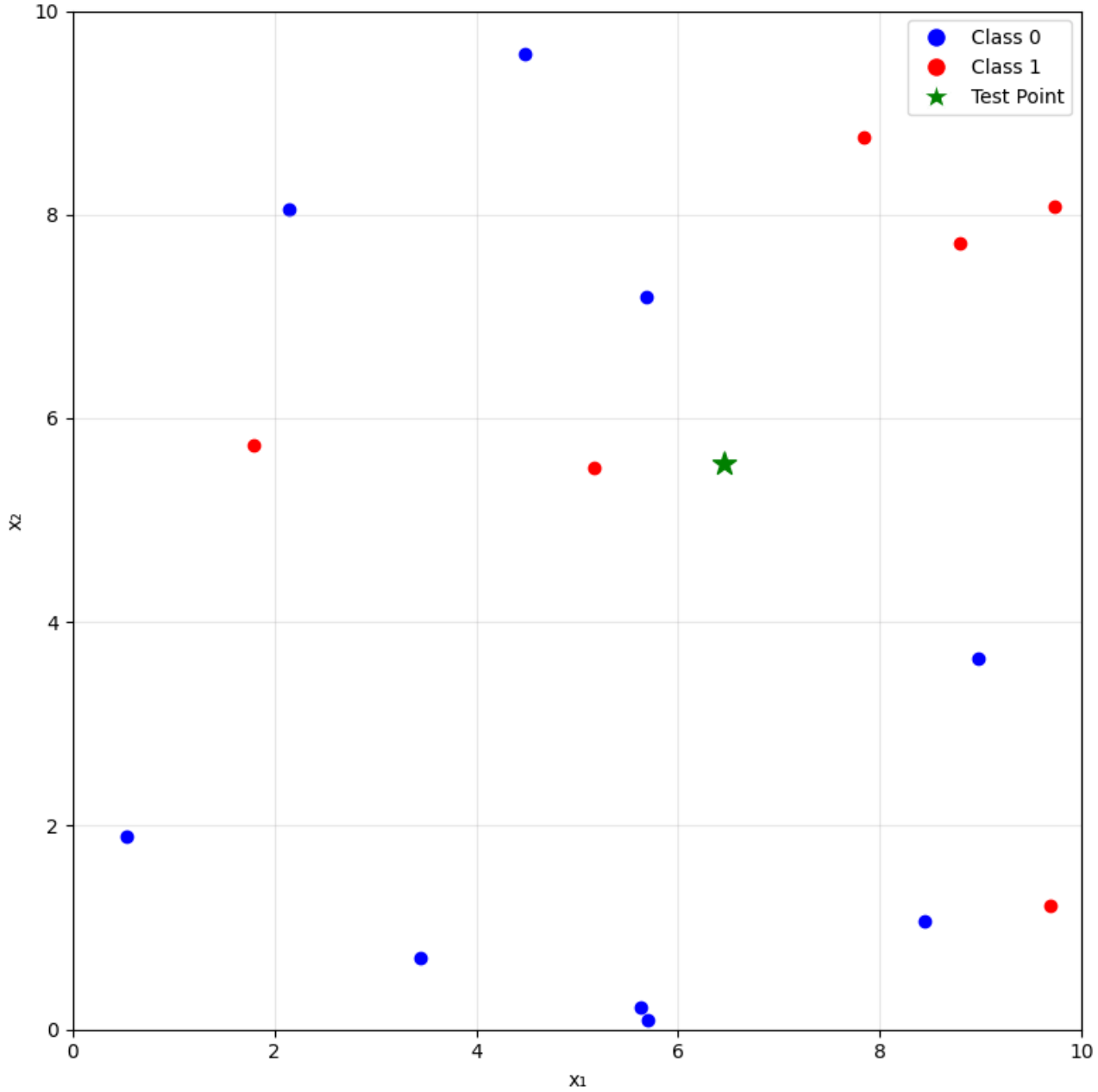
Hyper-parameter: k

Distance metric (typically Euclidean or Manhattan distance)

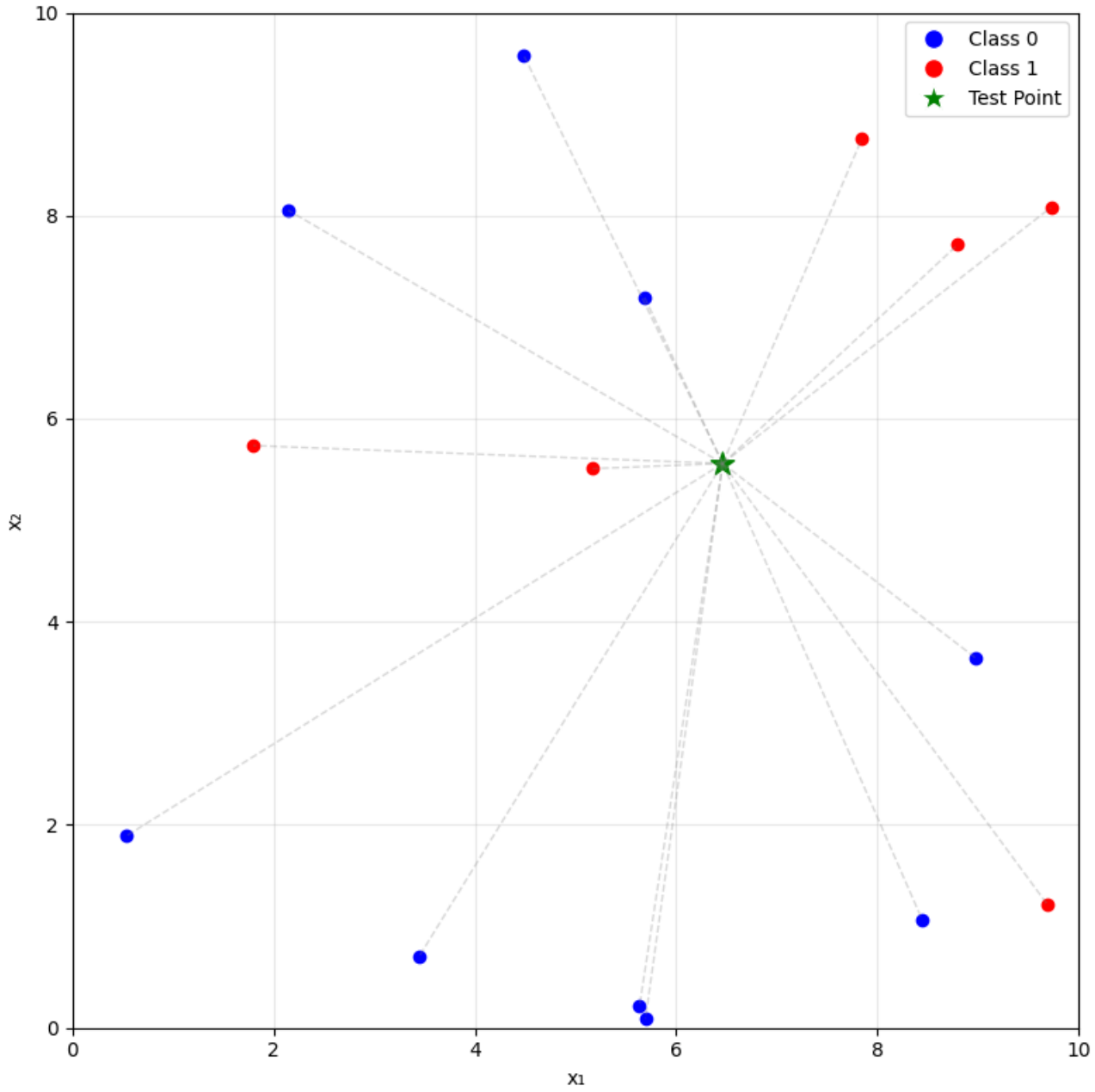
A tie-breaking scheme (typically at random)


Class 0
Class 1
10
8
6
4
2
0
0
2
4
6
8
10
x2
x1

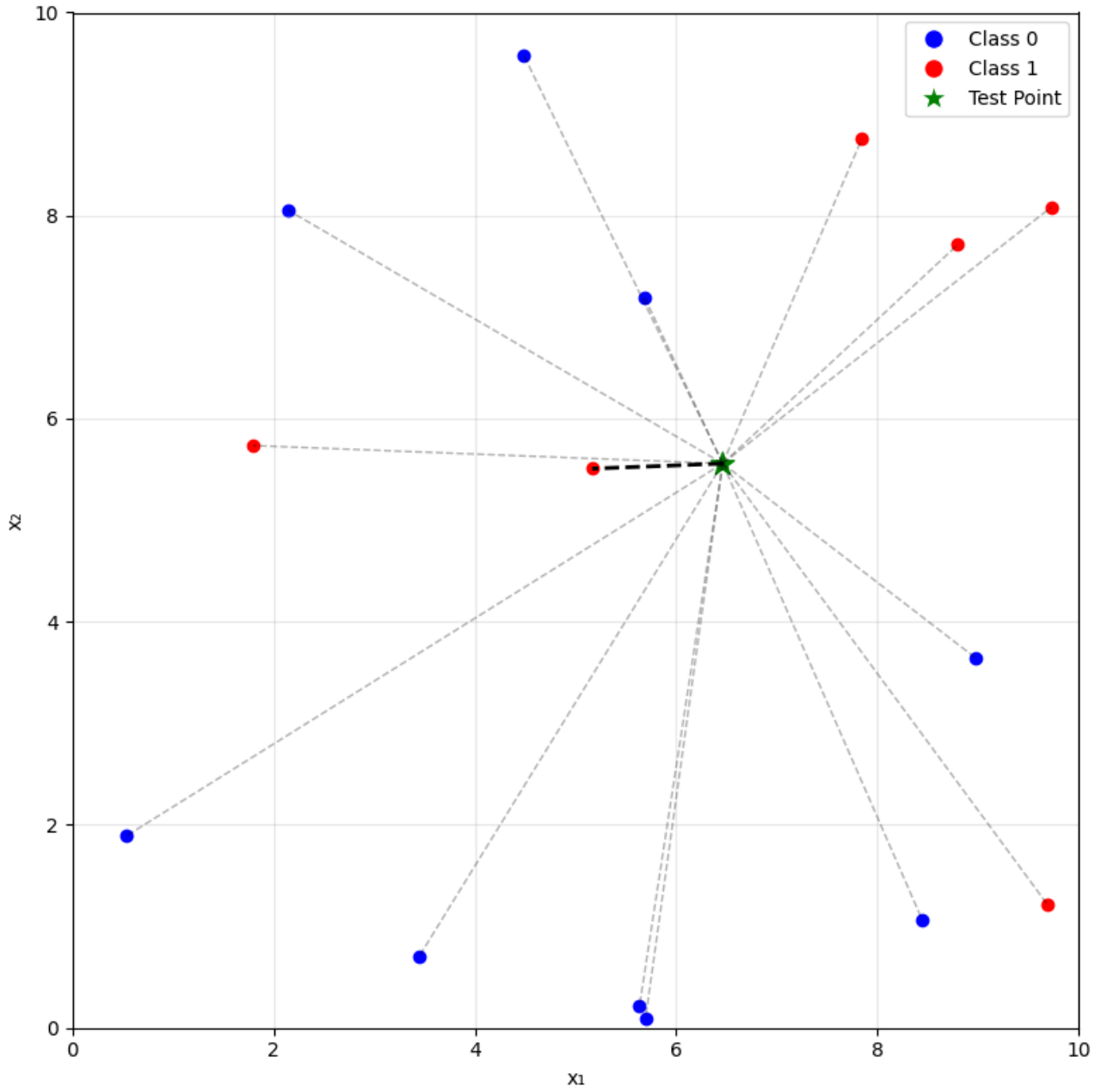
$k = 1$



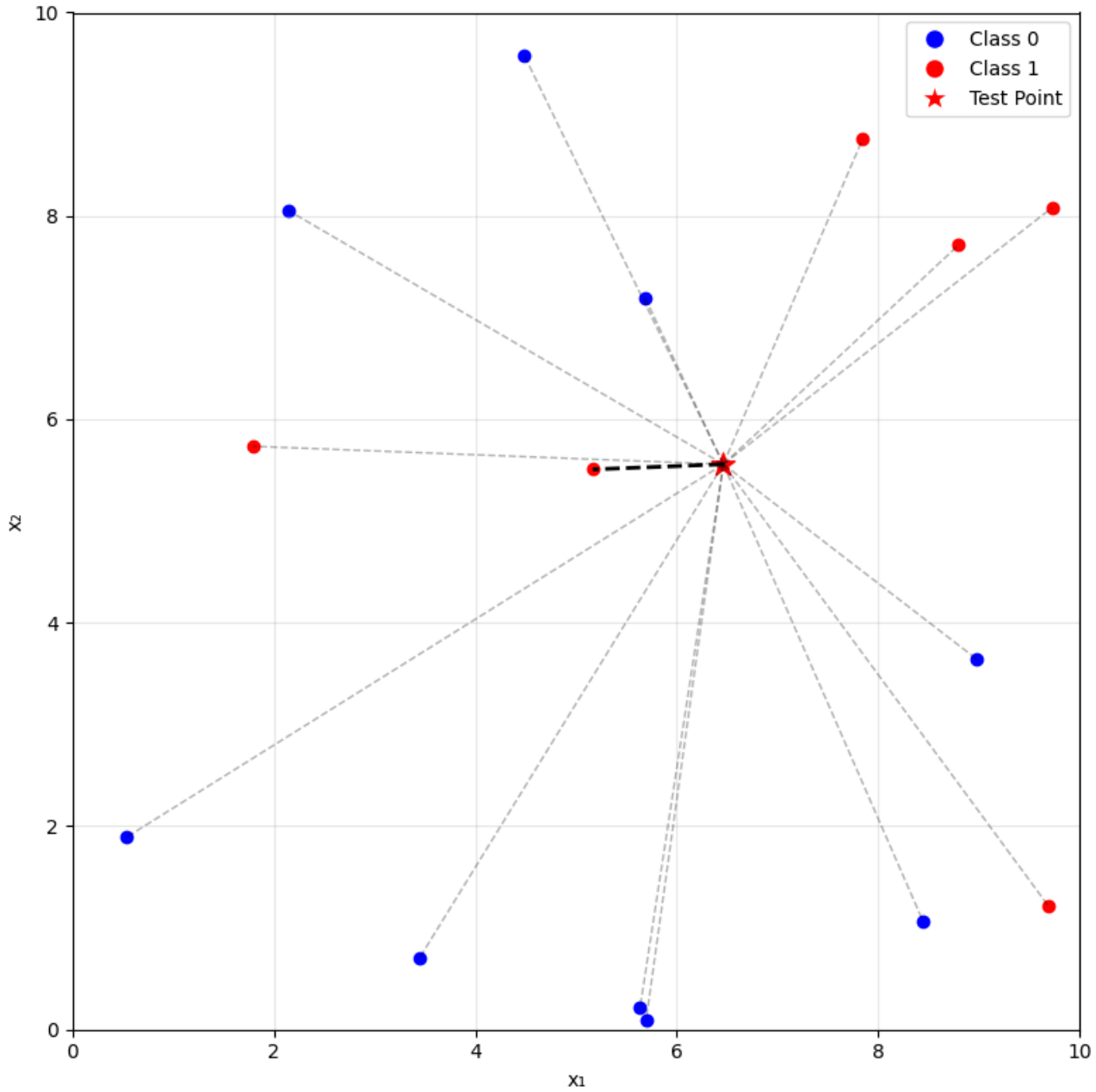
$k = 1$



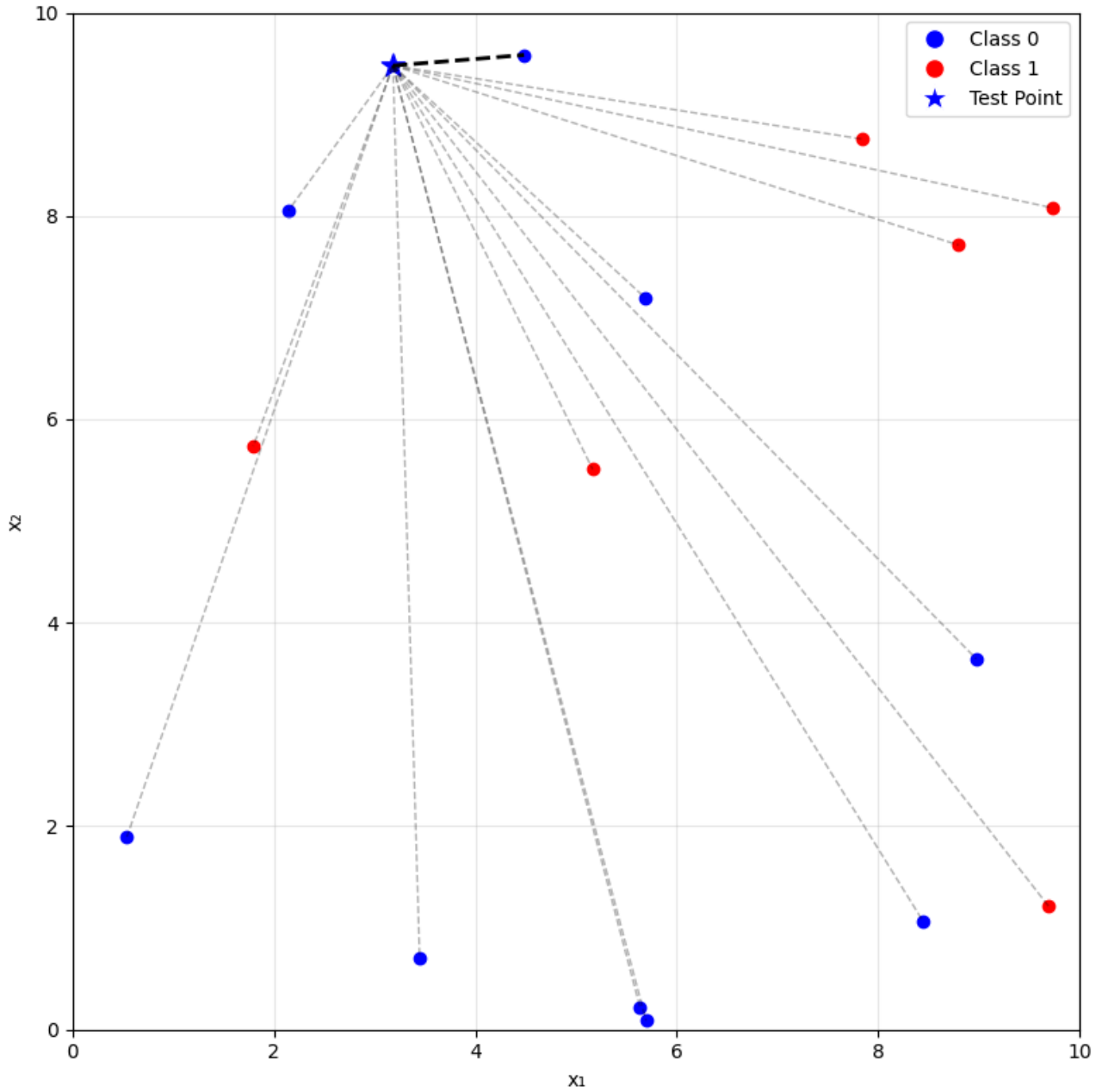
$k = 1$



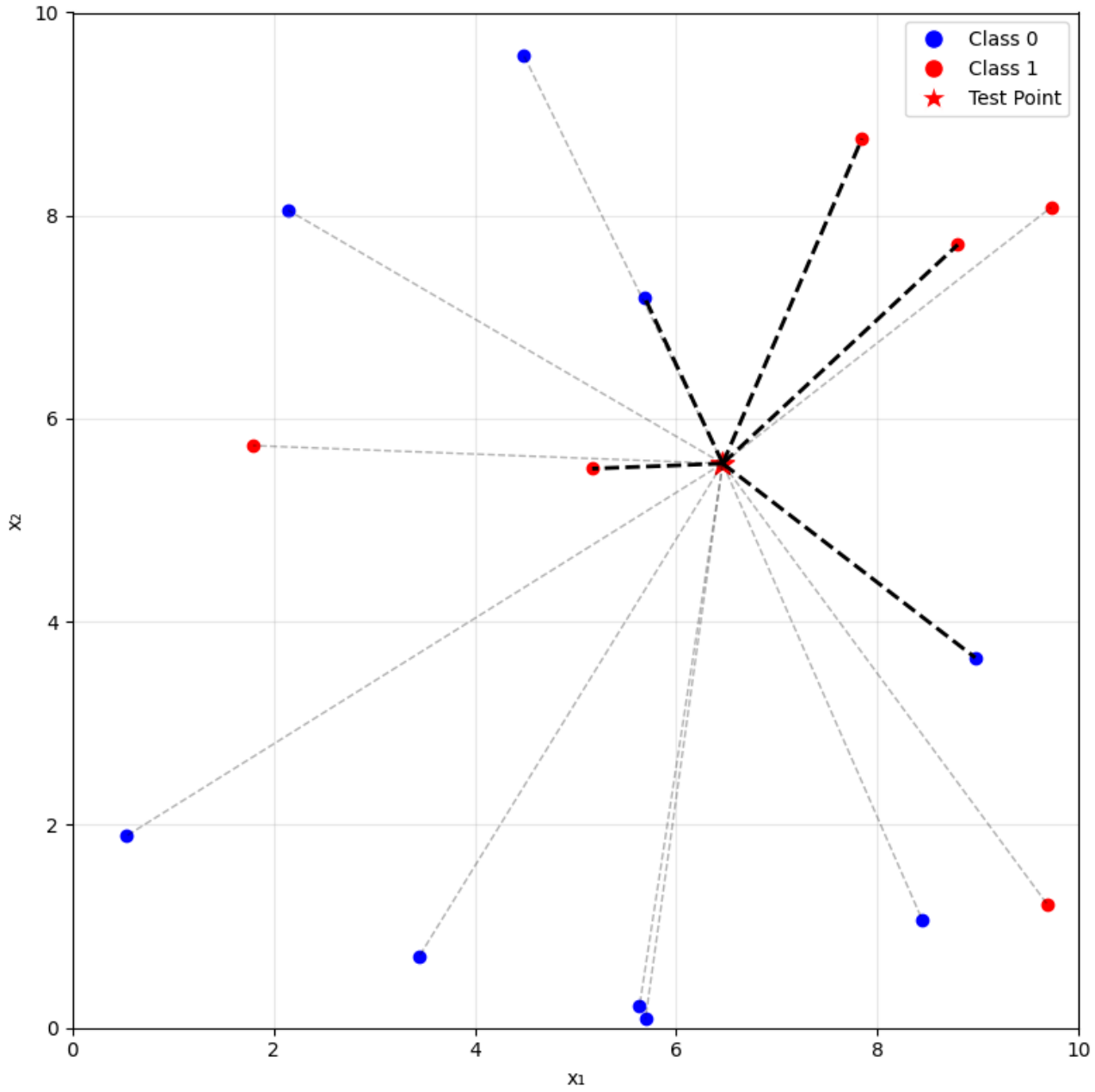
$k = 1$

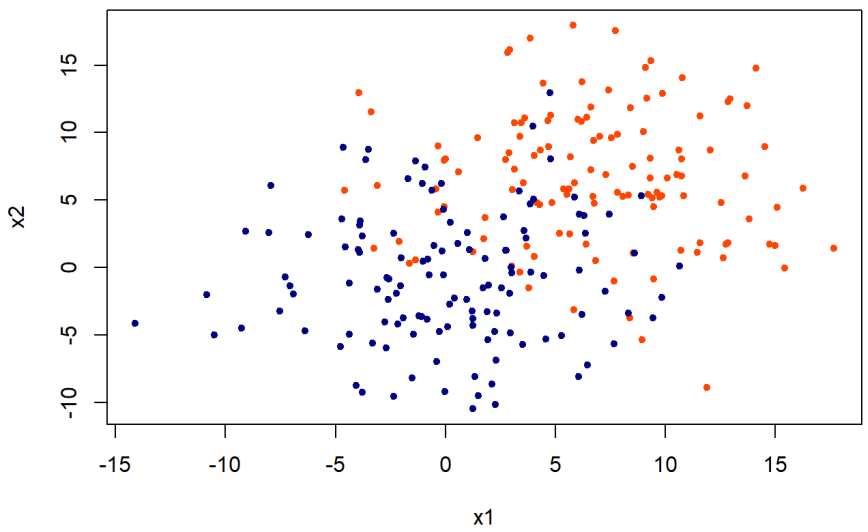


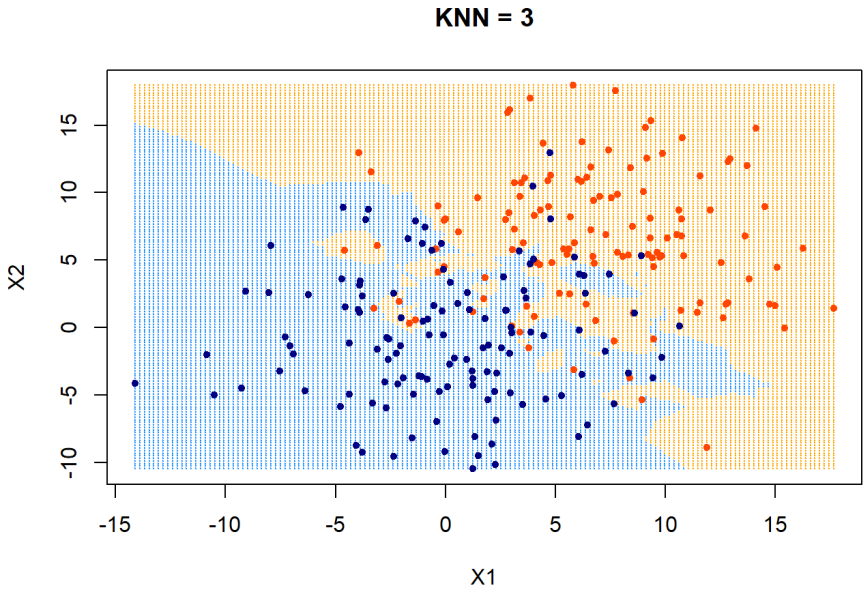
$k = 1$

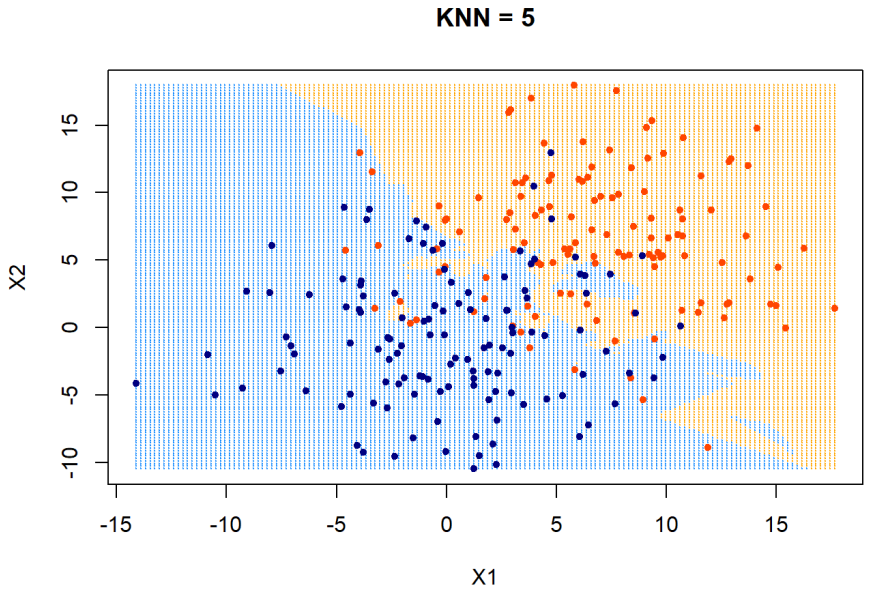


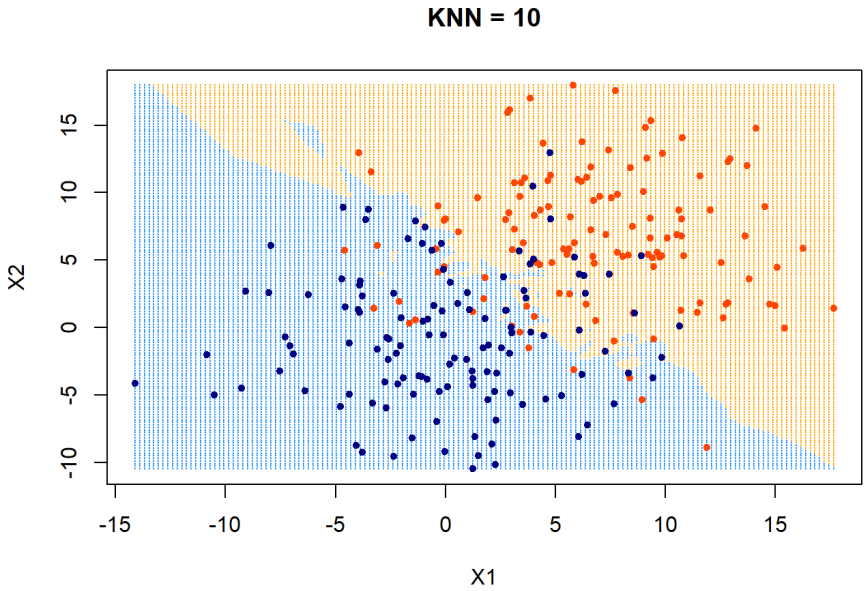
$k = 5$




x2
x1
15
10
5
0
-5
-10
-15


KNN = 3
X2
X1
15
10
5
0
-5
-10
-15

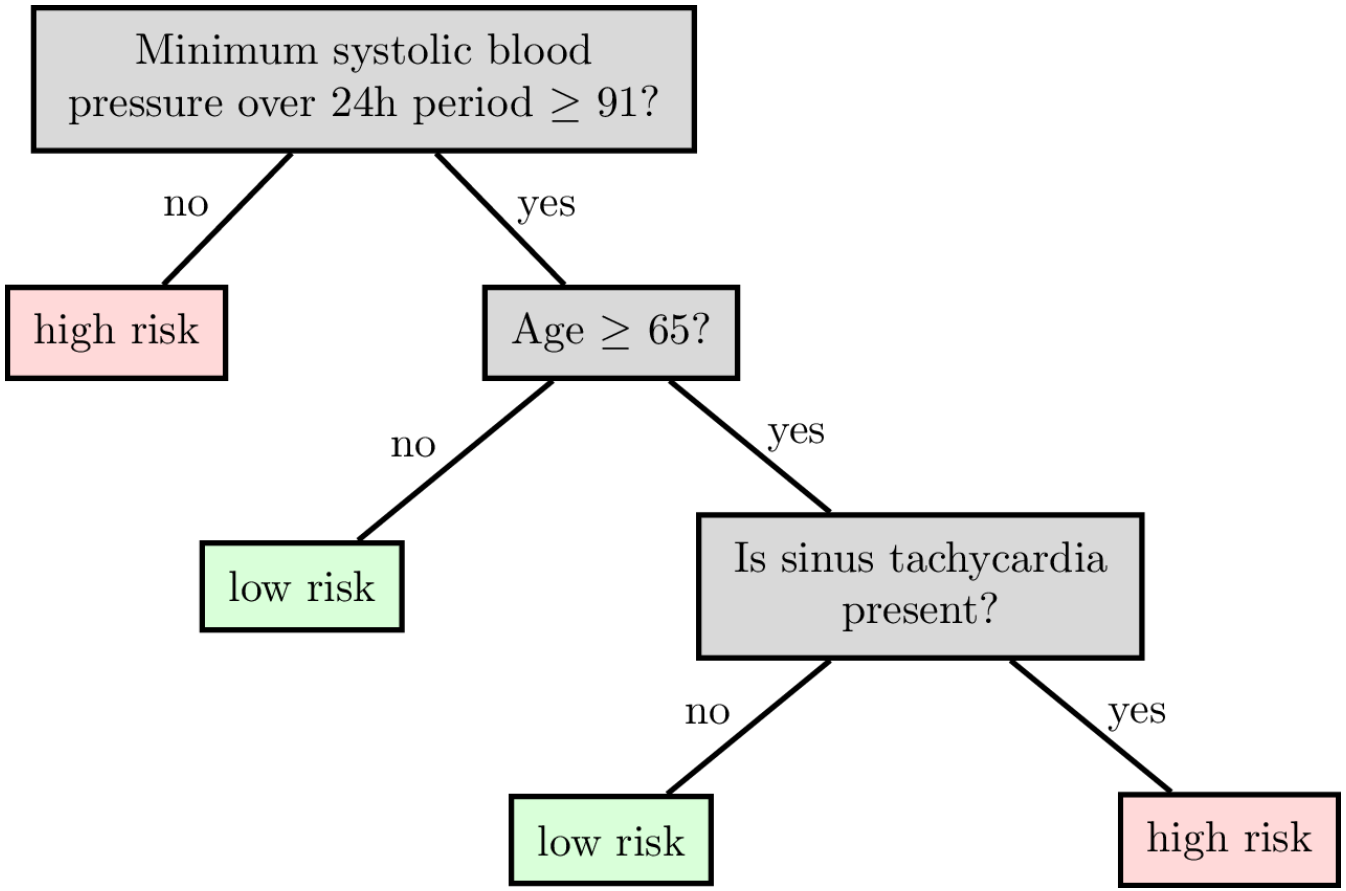

KNN = 5
X2
X1
15
10
5
0
-5
-10
-15


KNN = 10
X2
X1
15
10
5
0
-5
-10
-15

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

Reading a classification decision tree



features:

x_1 : date

x_2 : age

x_3 : height

x_4 : weight

x_5 : sinus tachycardia?

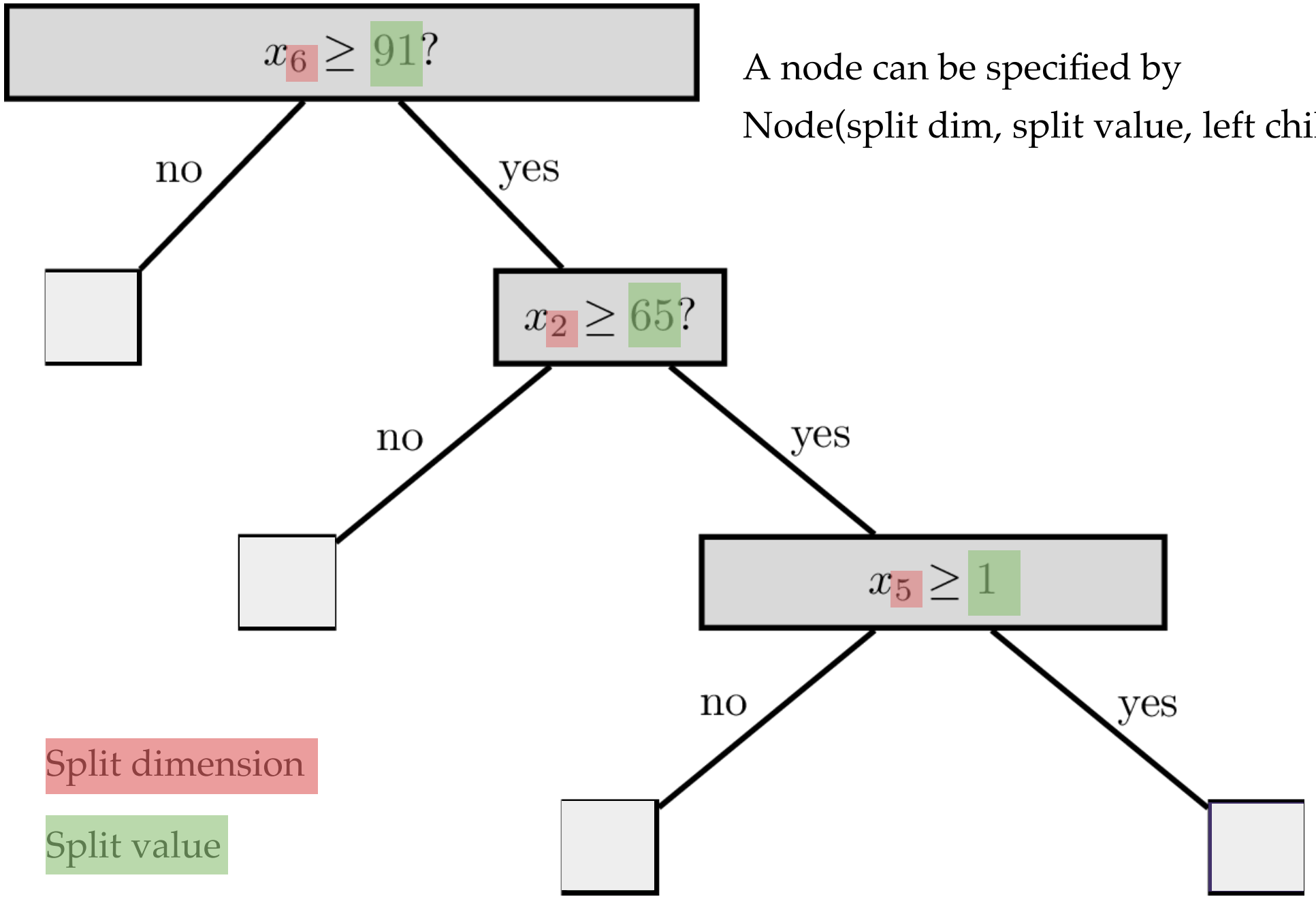
x_6 : min systolic bp, 24h

x_7 : latest diastolic bp

labels y :

1: high risk

-1: low risk

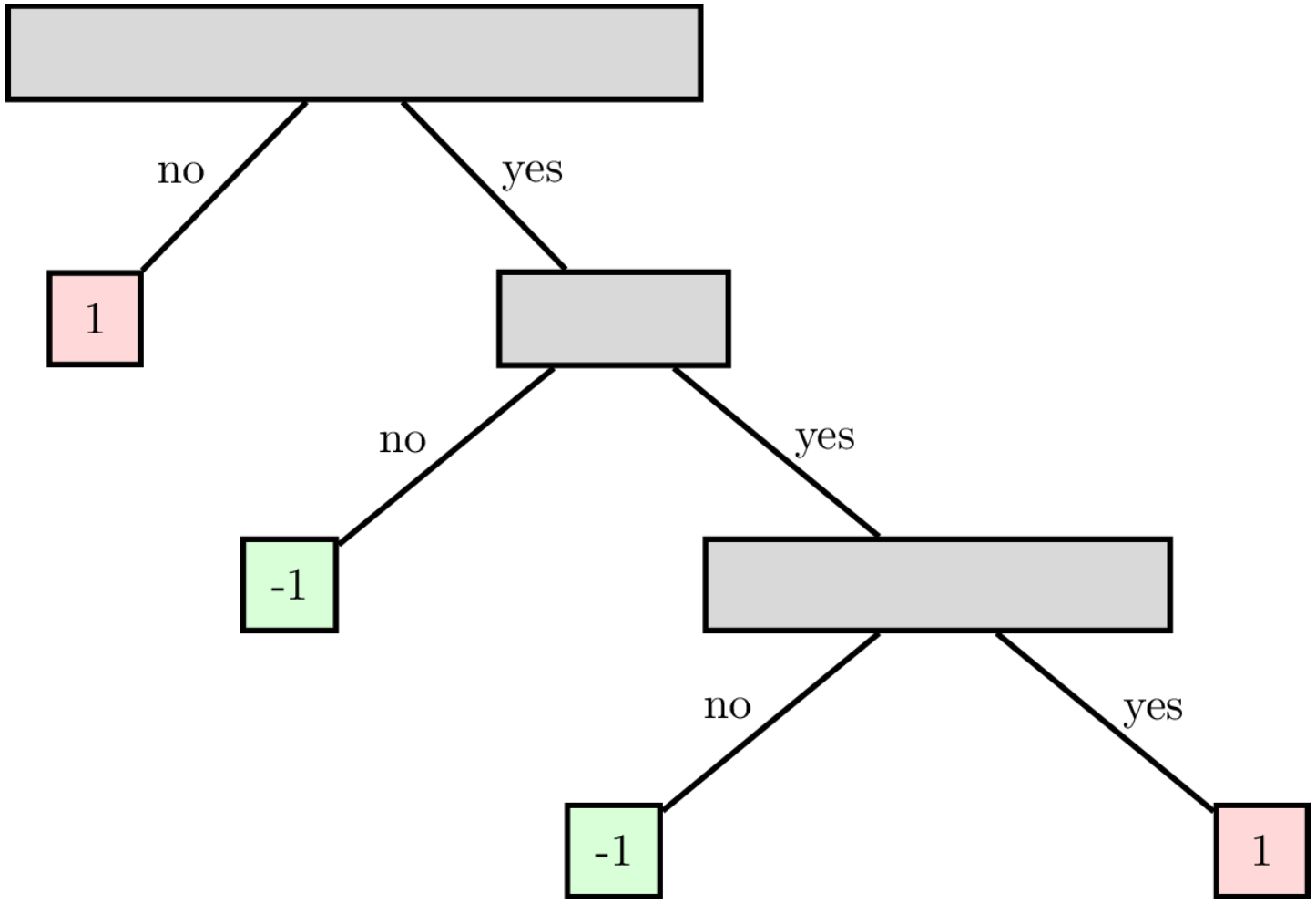


A node can be specified by

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

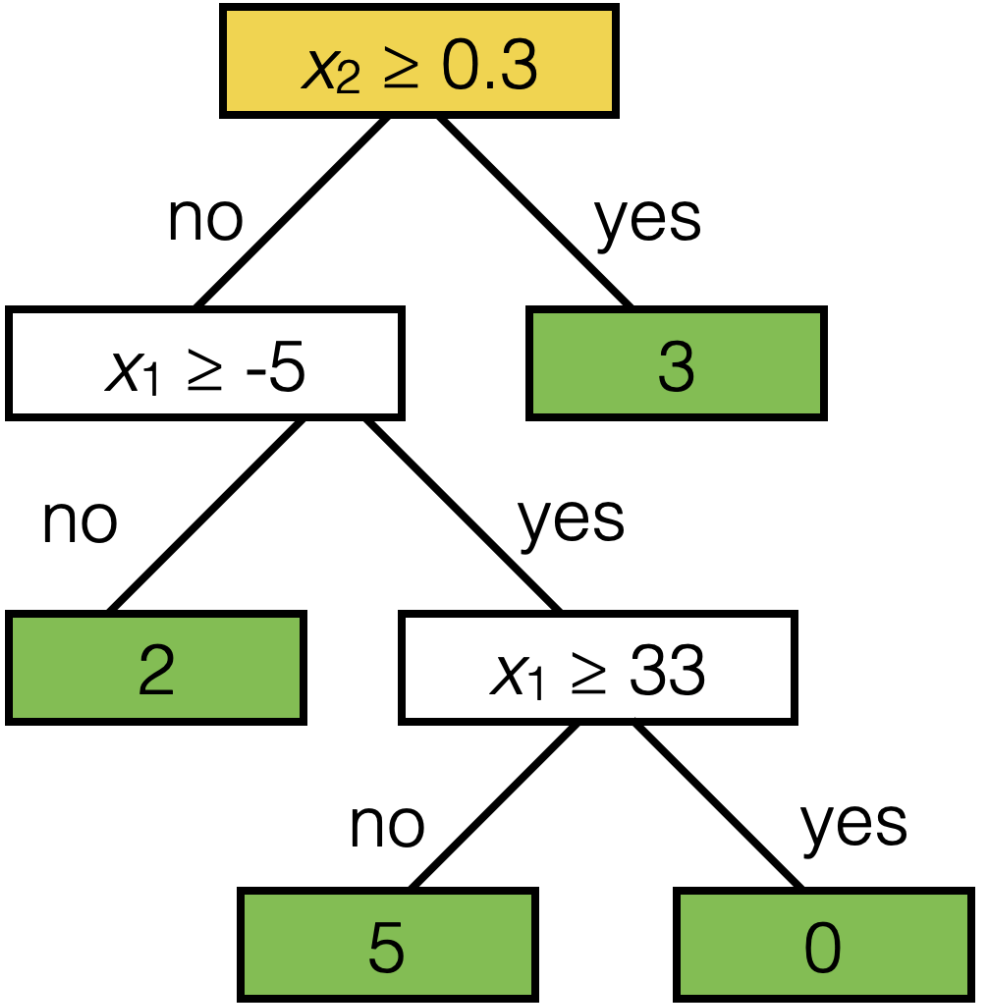
Split dimension

Split value



A leaf is specified by `Leaf(leaf_value)

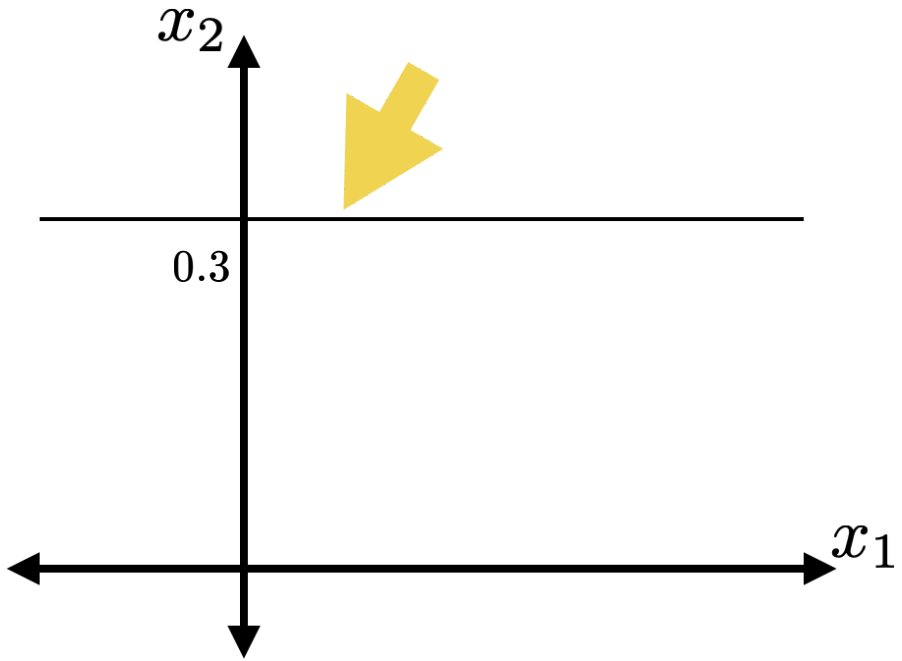
Reading a regression decision tree

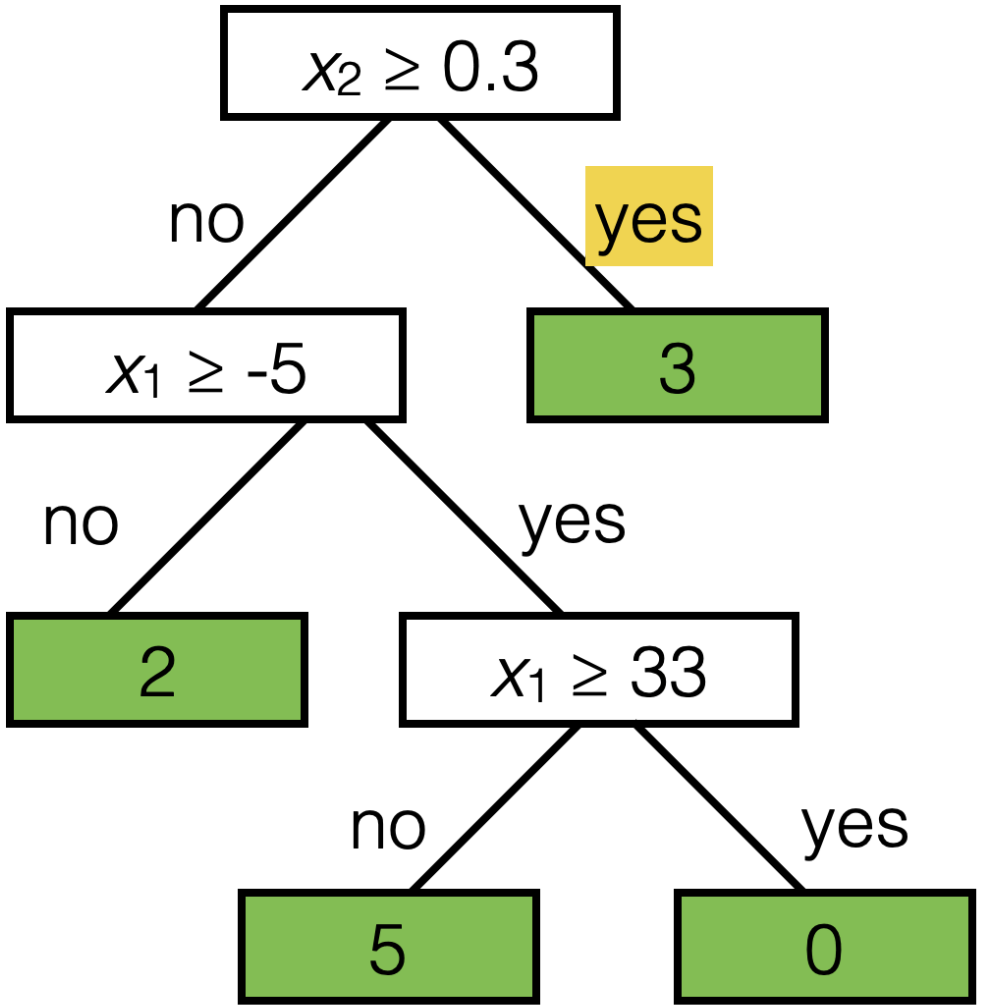


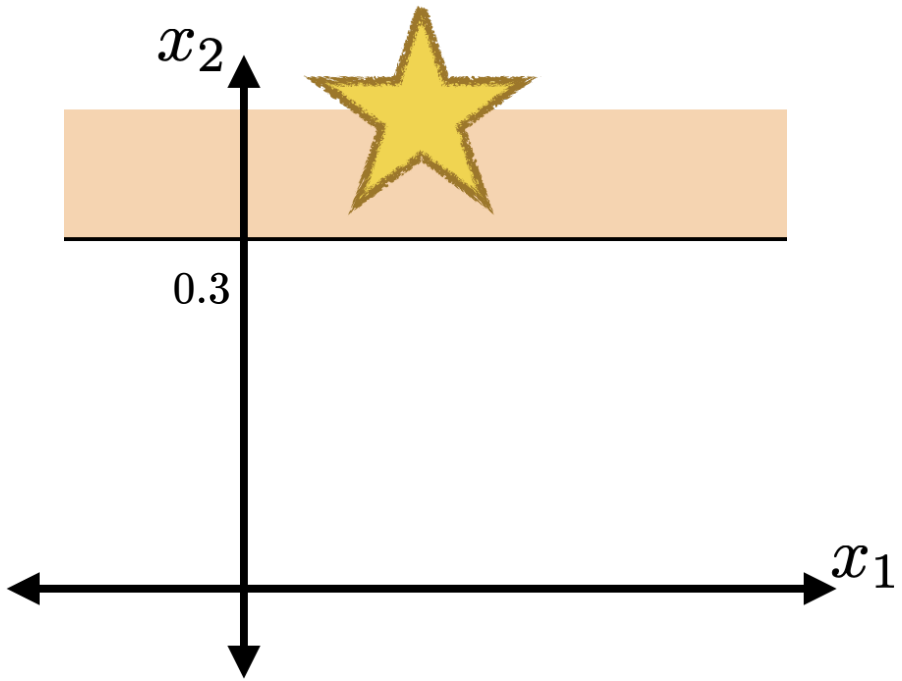
features:

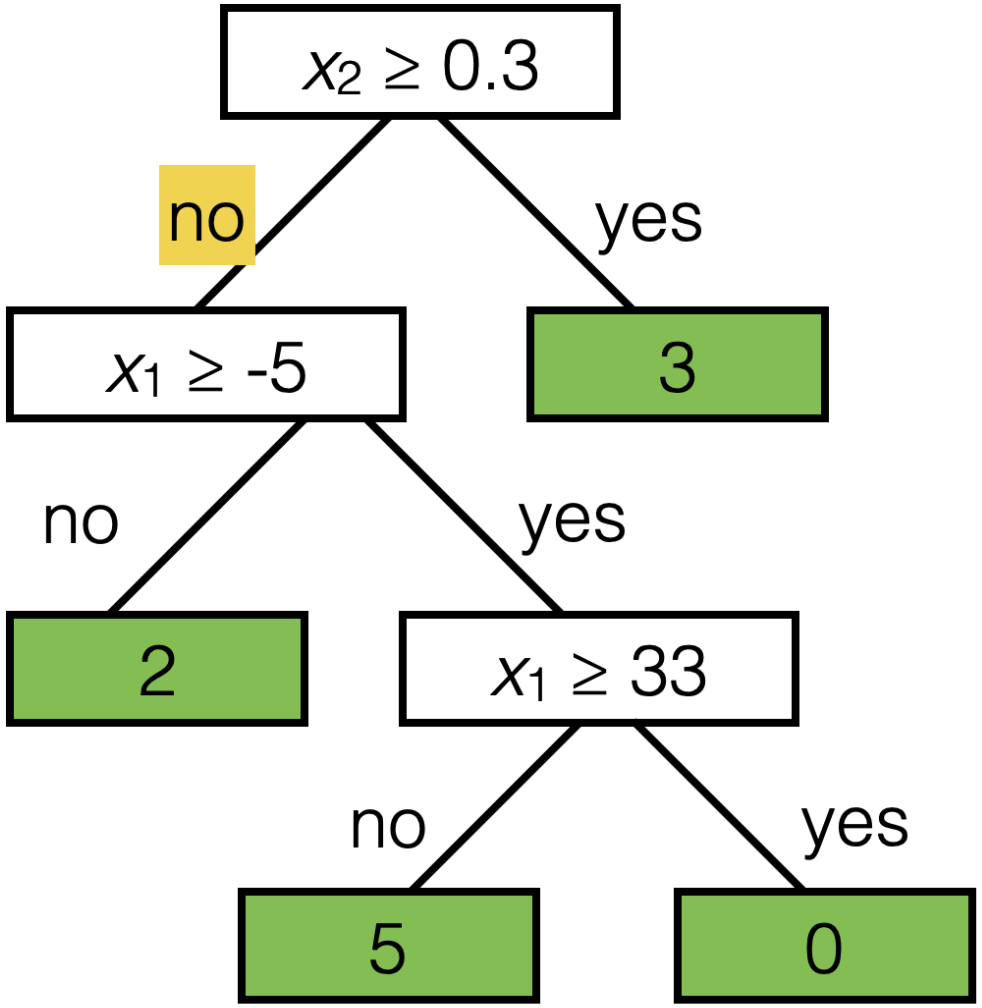
- x_1: temperature (deg C)
- x_2: precipitation (cm/hr)

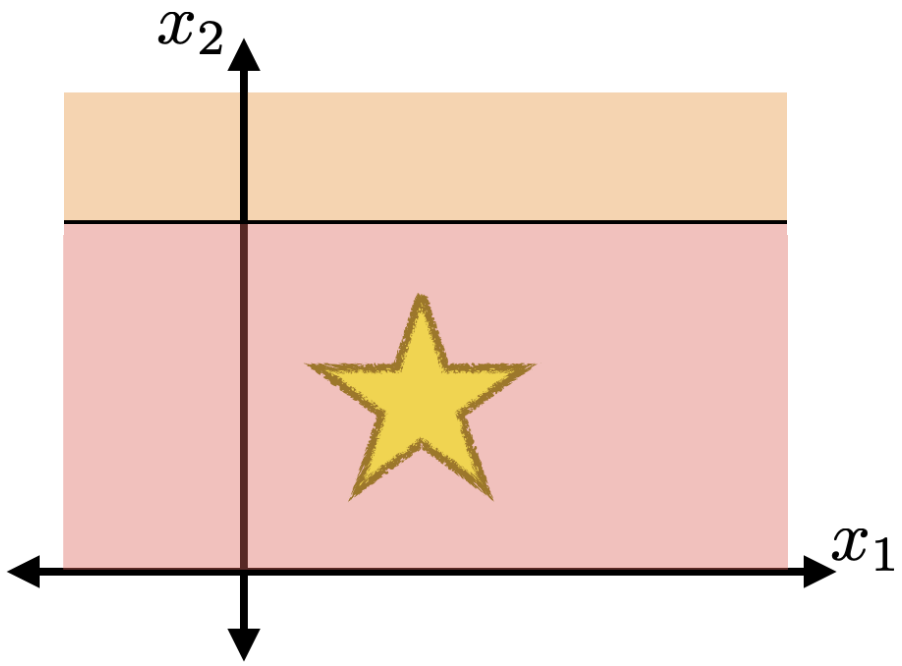
label: km run

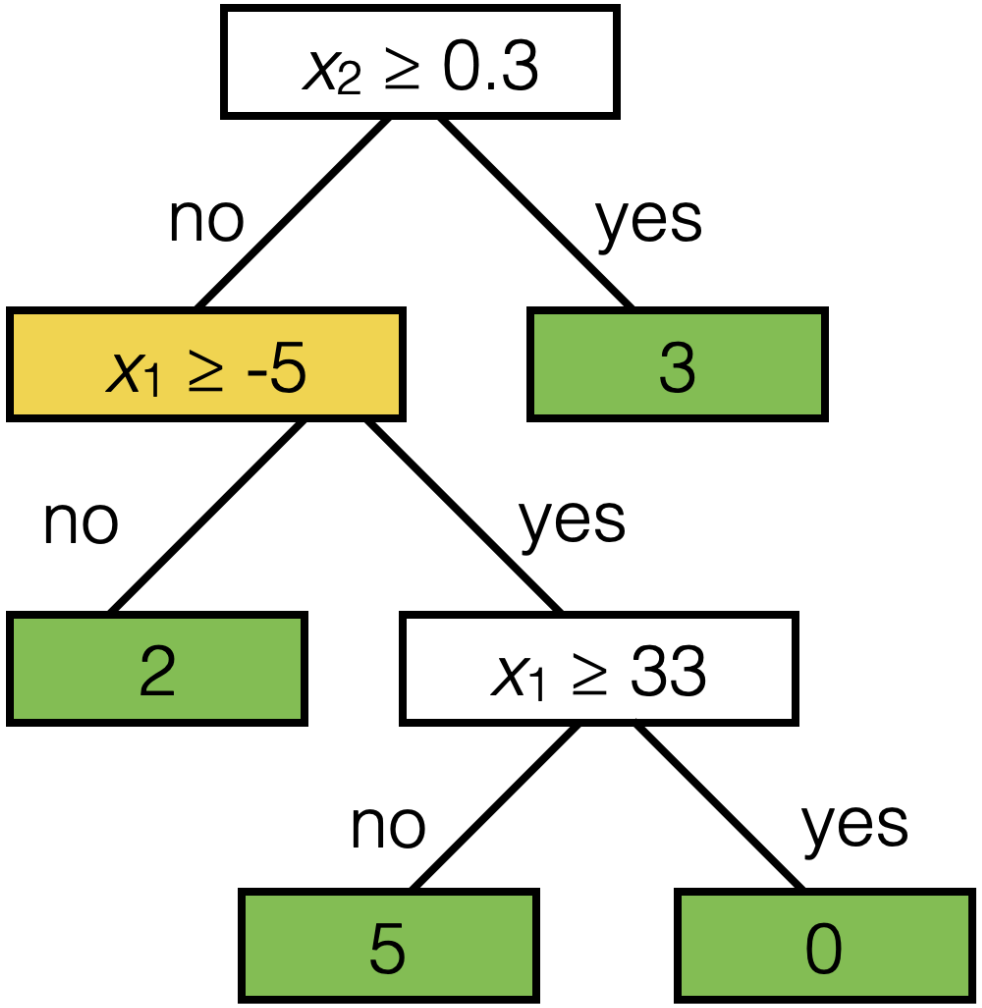


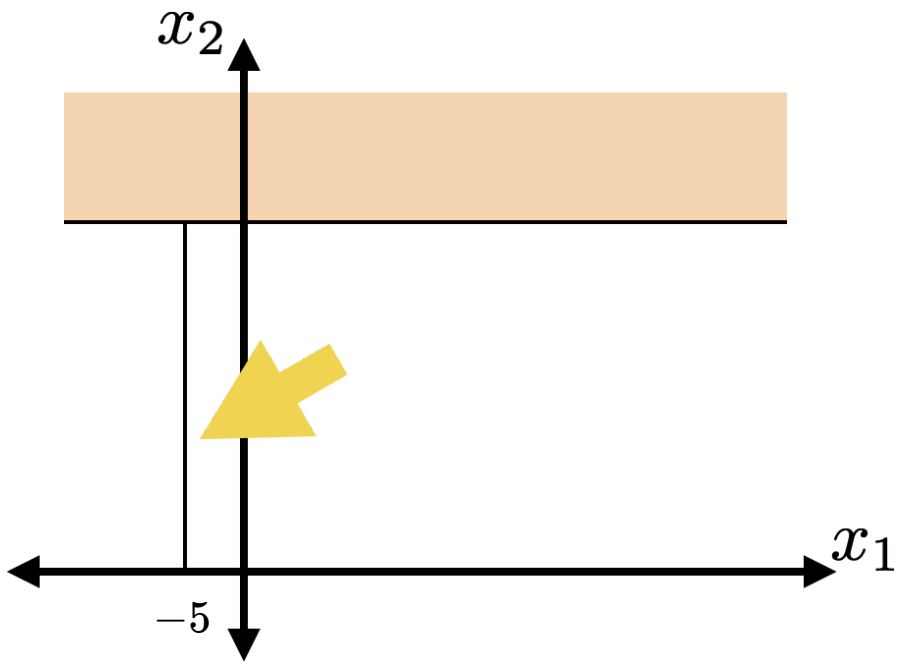

$x_2 \geq 0.3$
no
yes
$x_1 \geq -5$
3
no
yes
2
$x_1 \geq 33$
no
yes
5
0

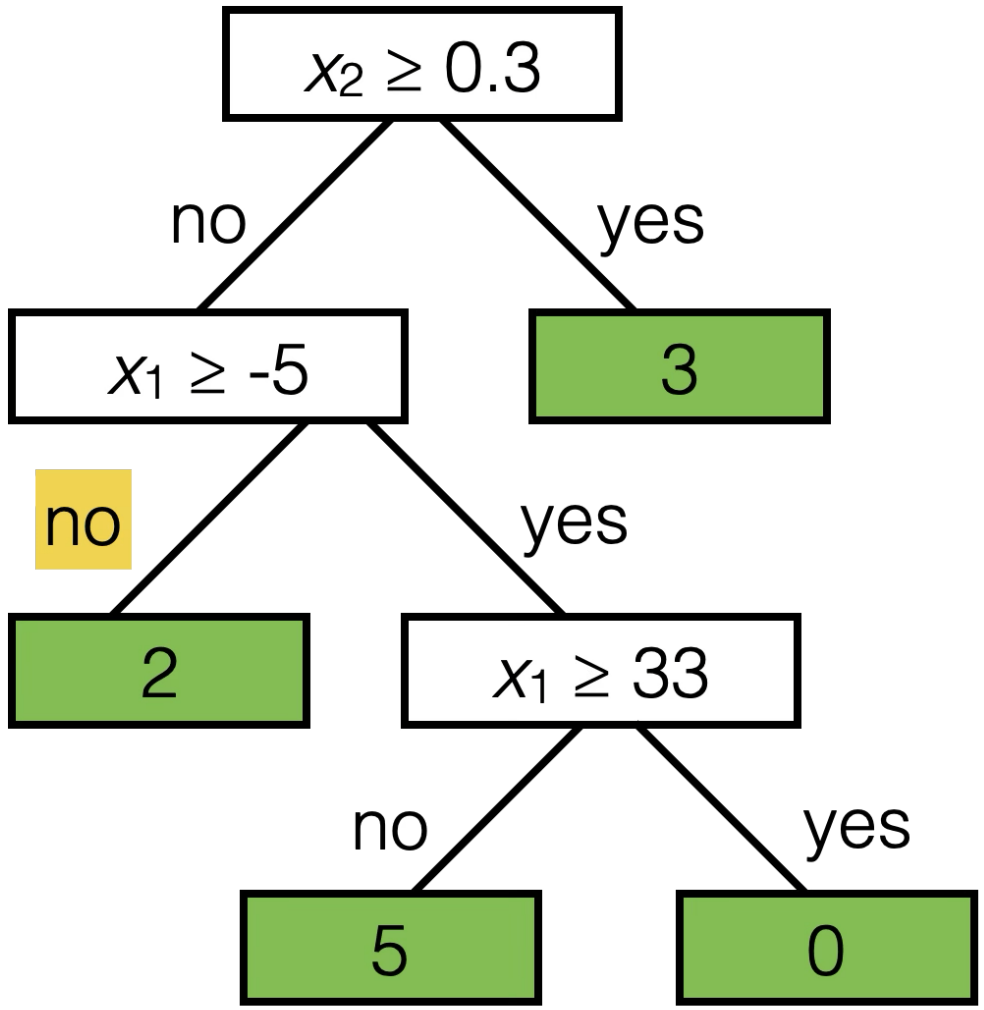

x_2
0.3
x_1

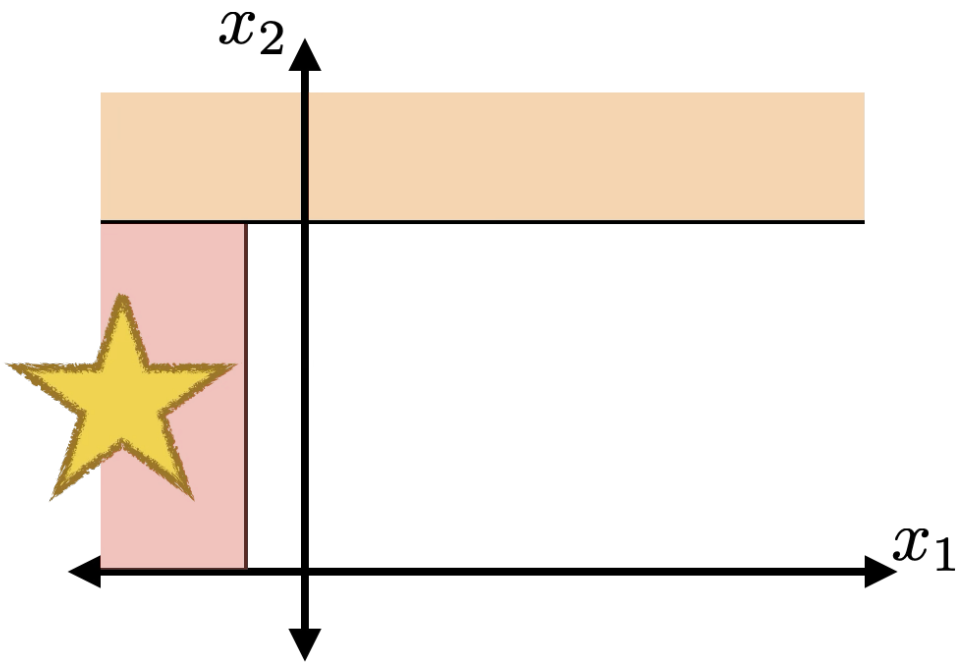

$x_2 \geq 0.3$
no
yes
$x_1 \geq -5$
3
no
yes
2
$x_1 \geq 33$
no
yes
5
0

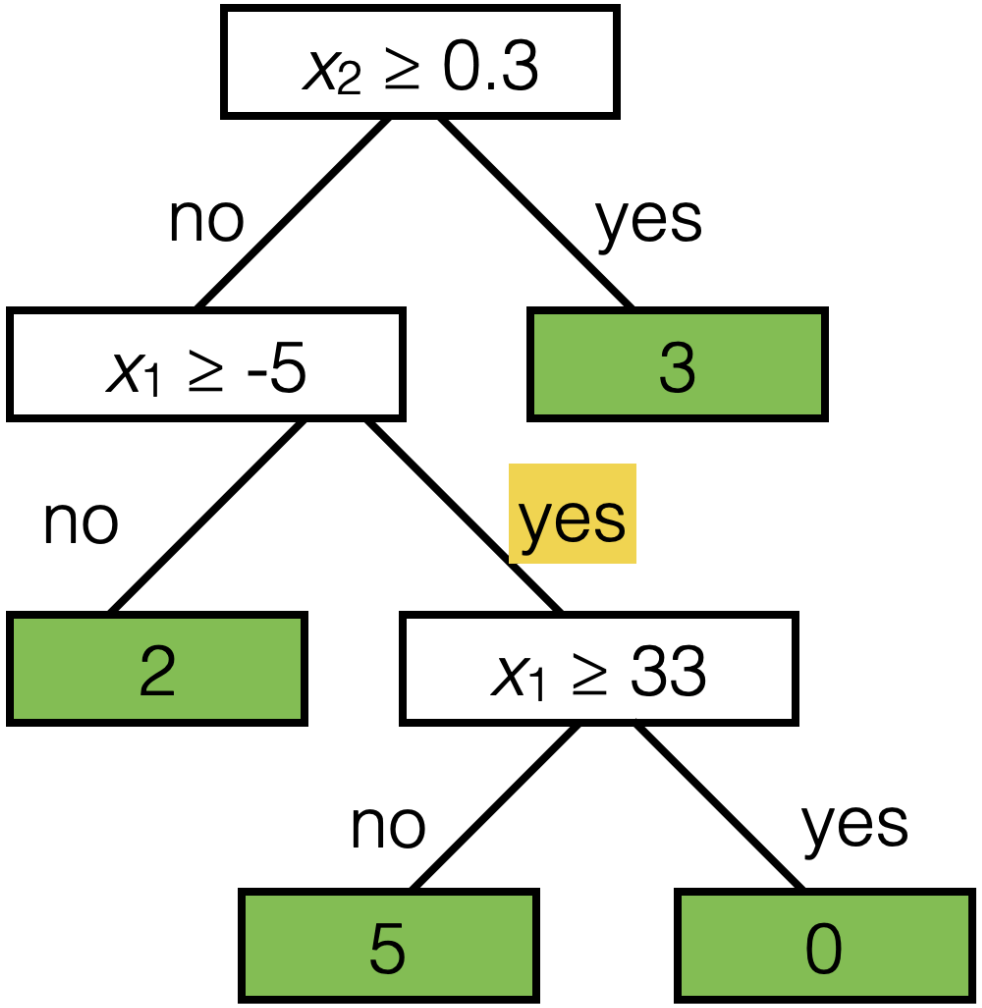

x_2
x_1

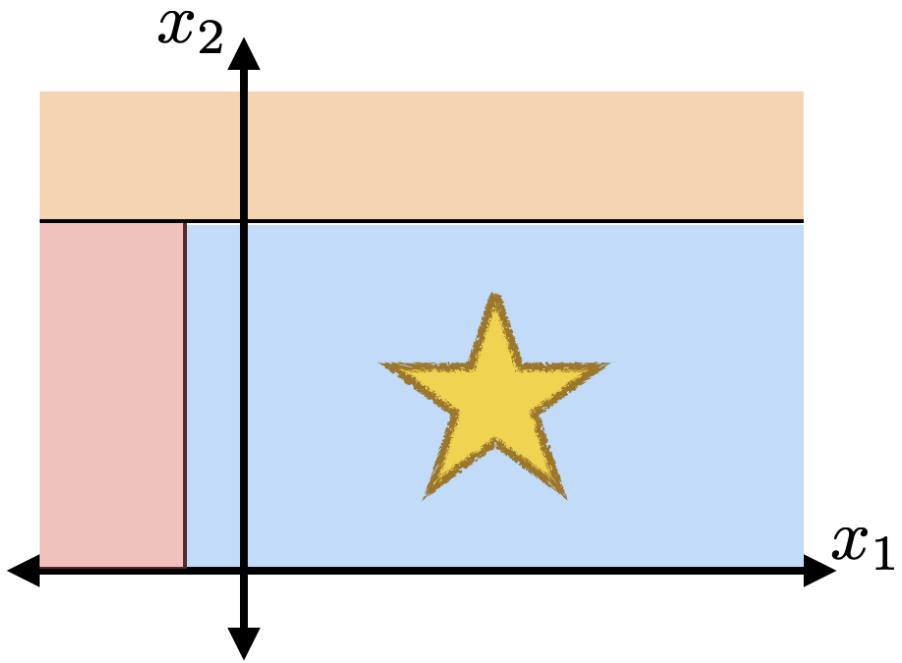

$x_2 \geq 0.3$
no
yes
$x_1 \geq -5$
3
no
yes
2
$x_1 \geq 33$
no
yes
5
0

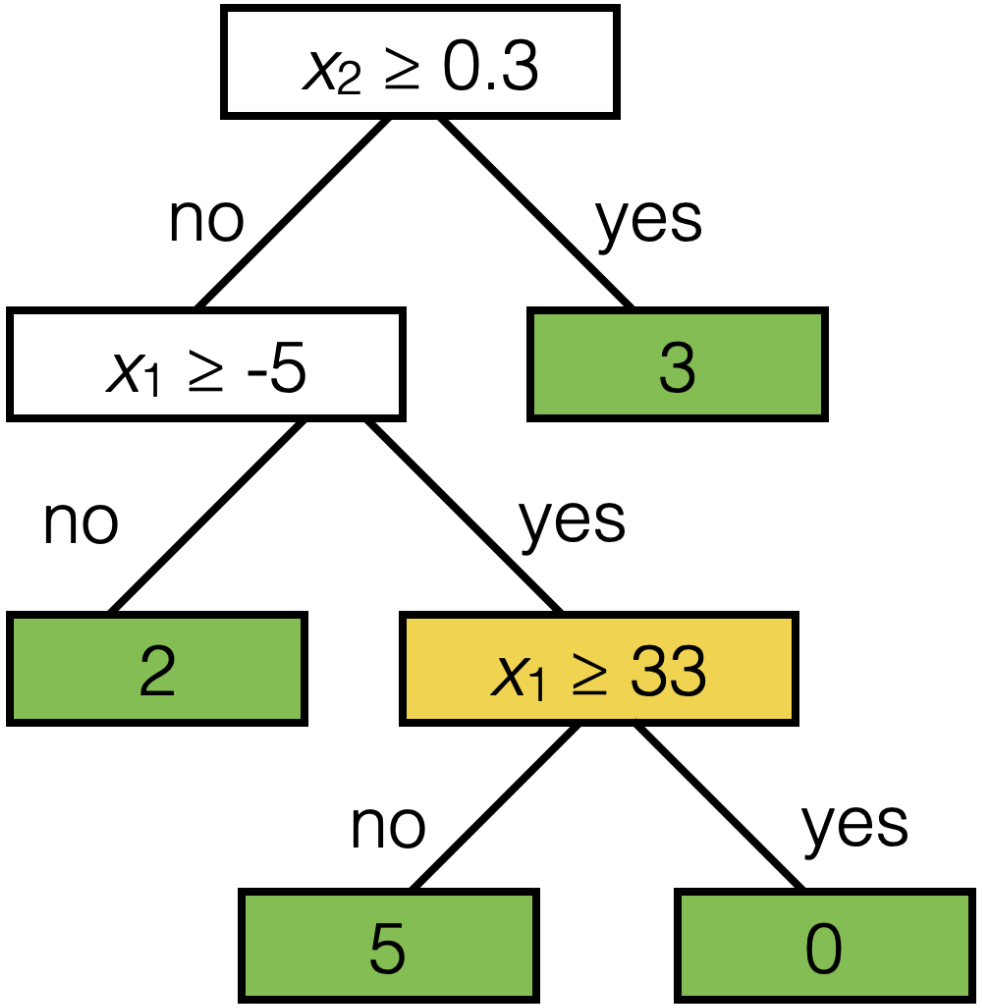

x_2
x_1
-5

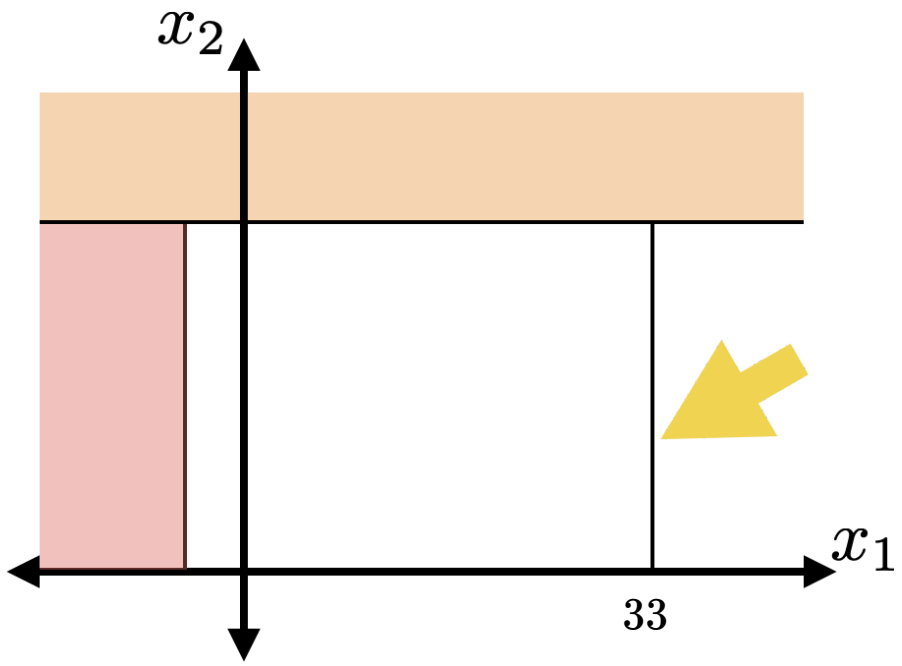

$x_2 \geq 0.3$
no
yes
$x_1 \geq -5$
3
no
yes
2
$x_1 \geq 33$
no
yes
5
0

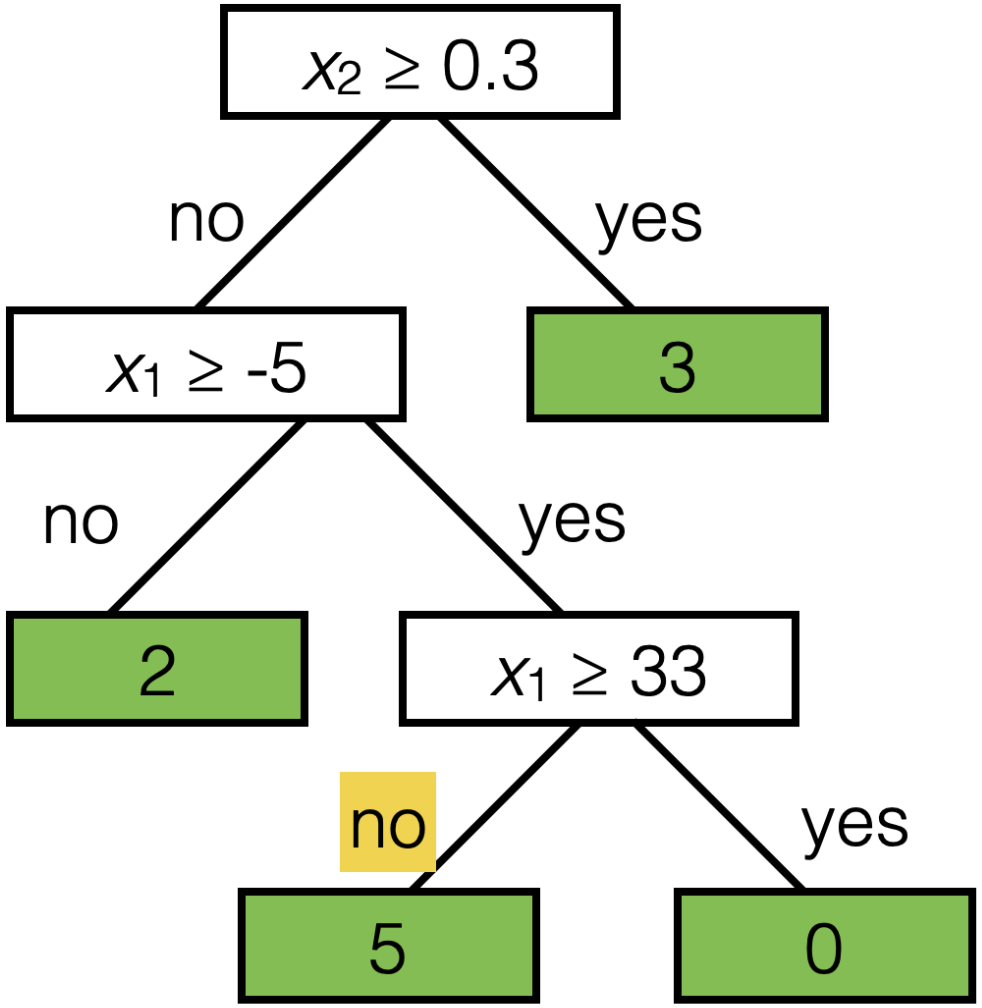

x_2
x_1

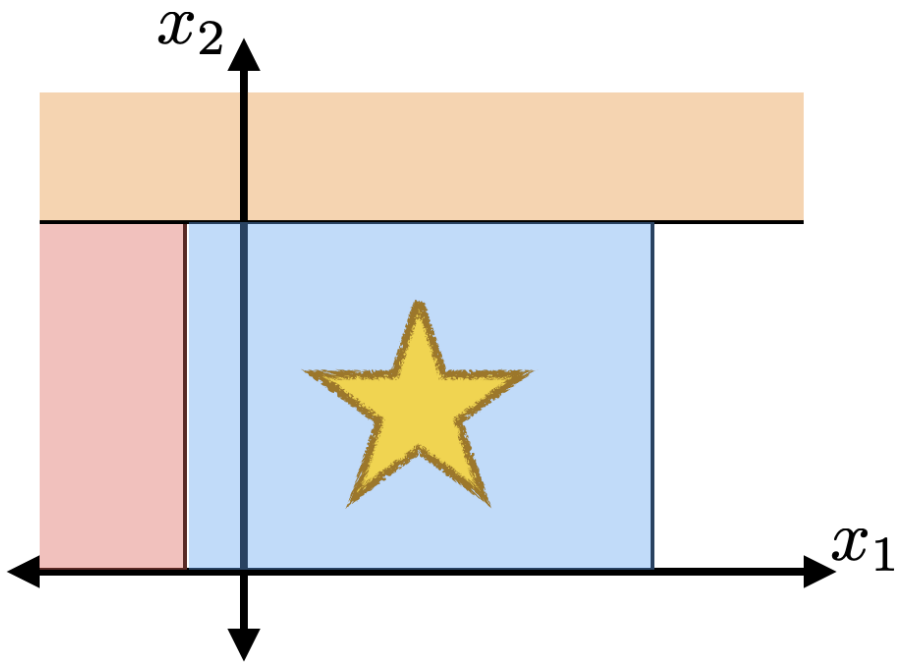

$x_2 \geq 0.3$
no
yes
$x_1 \geq -5$
3
no
yes
2
$x_1 \geq 33$
no
yes
5
0

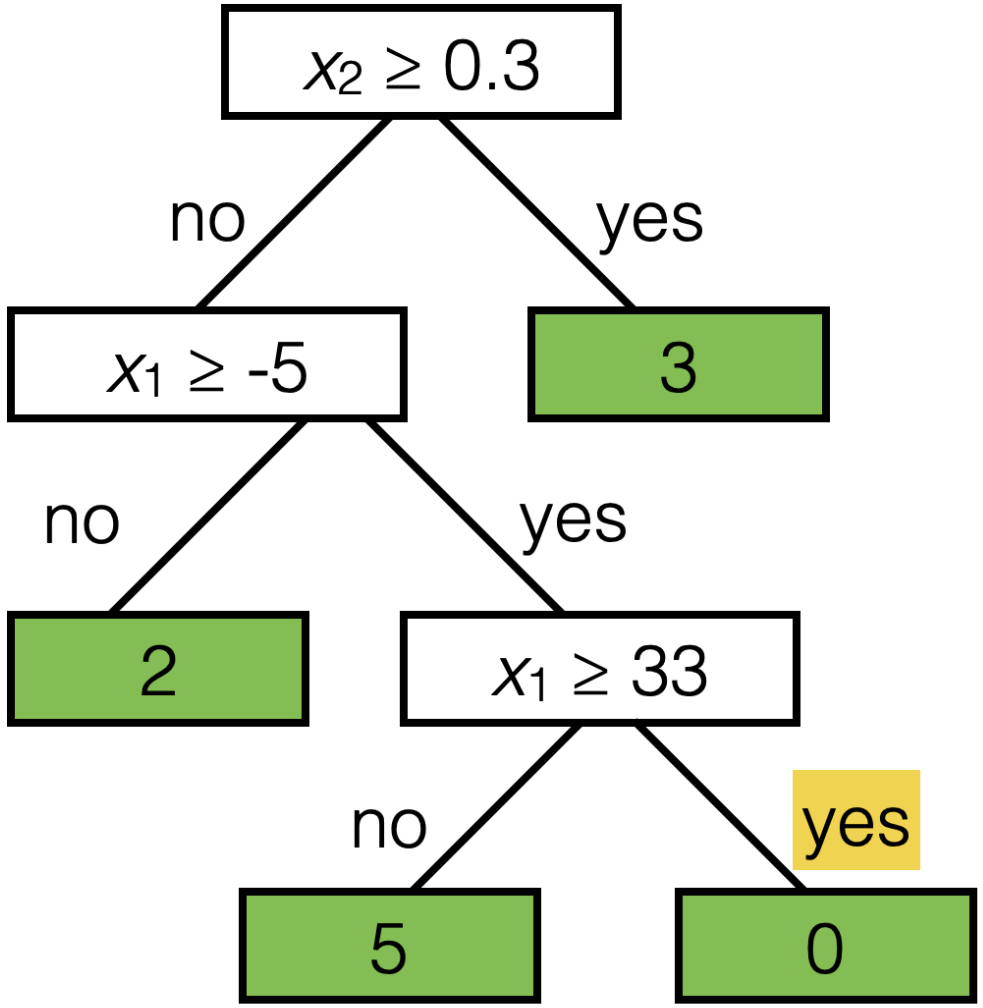

x_2
x_1

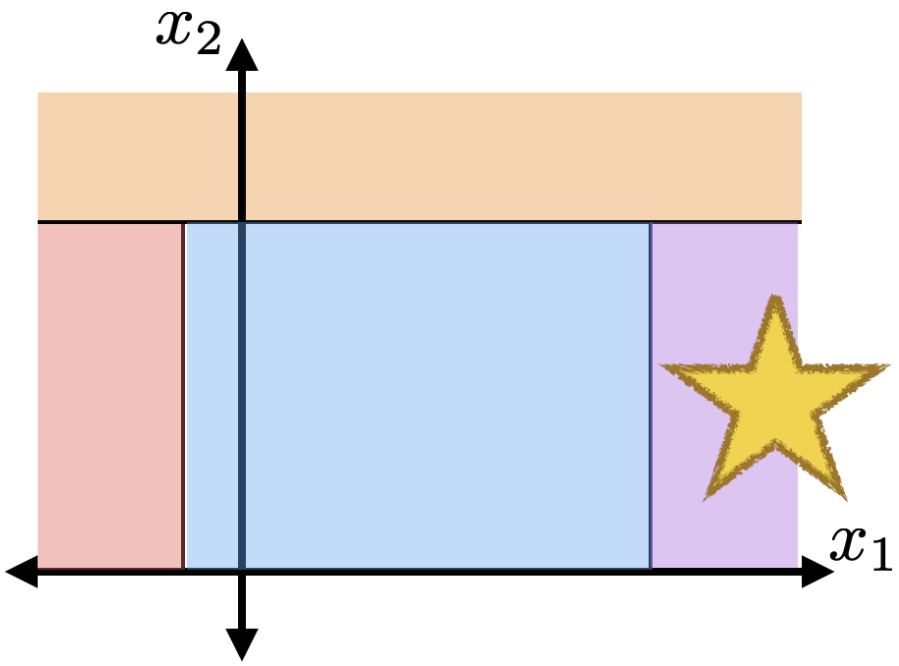

$x_2 \geq 0.3$
no
yes
$x_1 \geq -5$
3
no
yes
2
$x_1 \geq 33$
no
yes
5
0

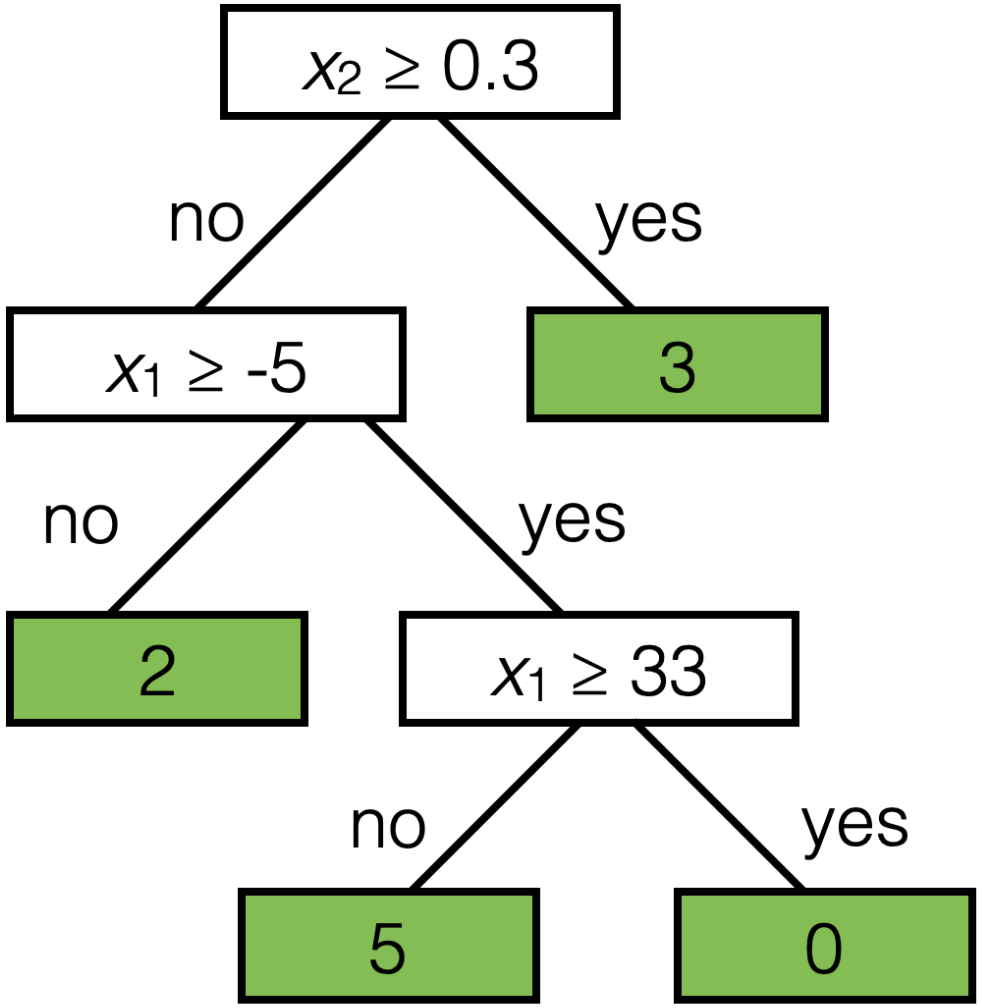

x_2
x_1
33


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

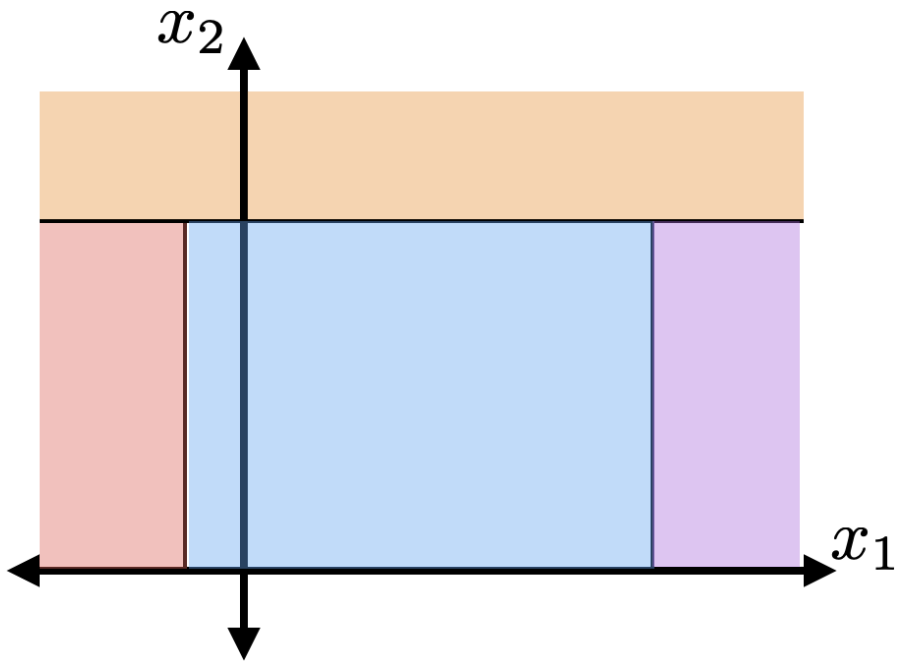

x_2
x_1

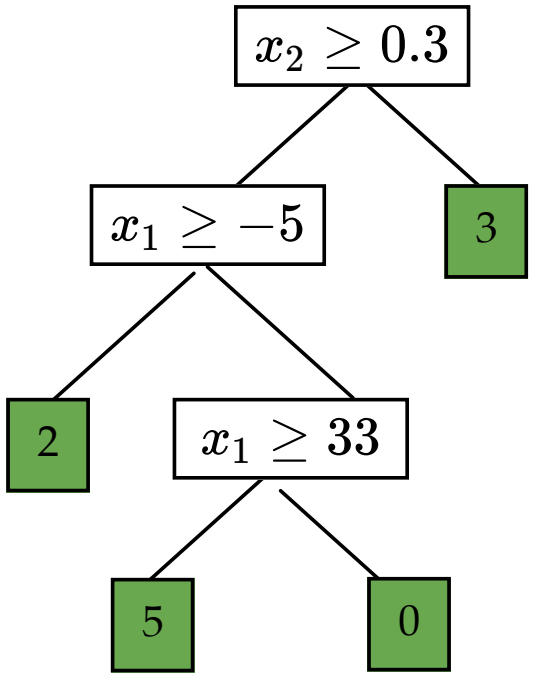

$x_2 \geq 0.3$
no
yes
$x_1 \geq -5$
3
no
yes
2
$x_1 \geq 33$
no
yes
5
0

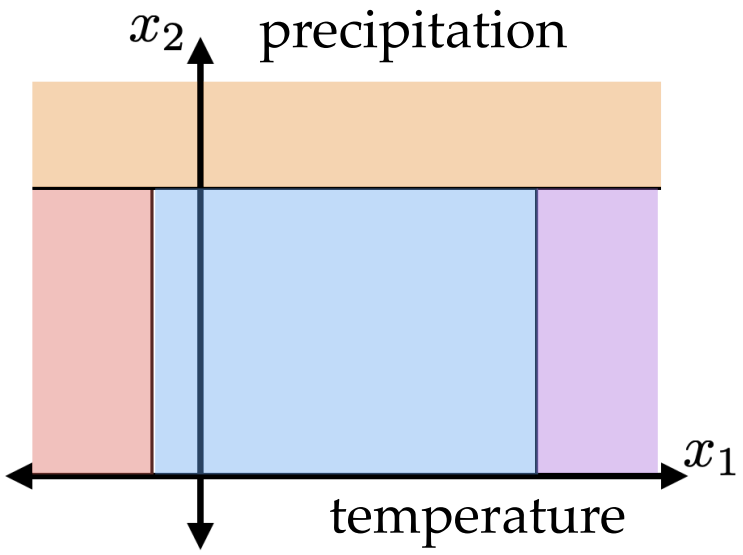

x_2
x_1



The same prediction applies to an axis-aligned ‘box’ or ‘volume’ in the feature space

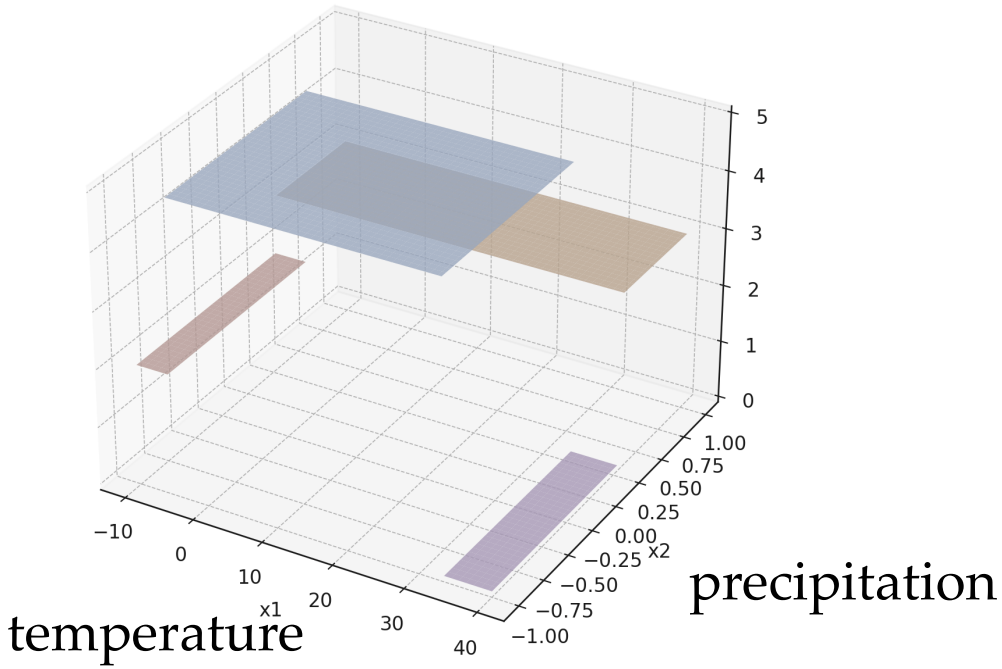






The same prediction applies to an axis-aligned ‘box’ or ‘volume’ in the feature space

label: km run



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

BuildTree for regression

Set of indices.

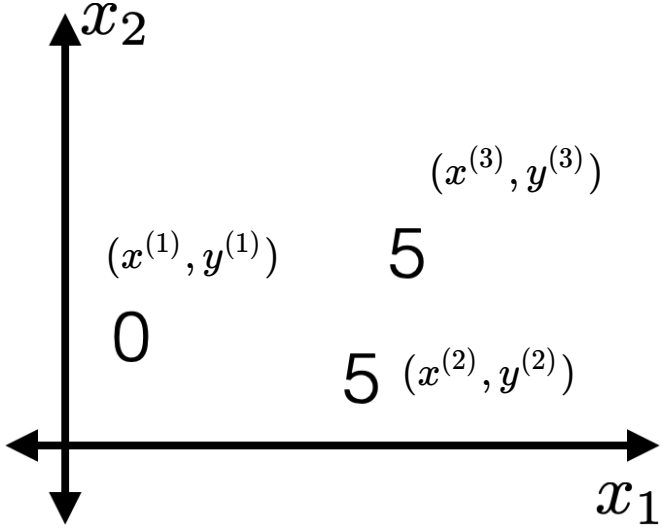
Hyper-parameter, largest leaf size (i.e. the maximum number of training data that can "flow" into that leaf).

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 $\left(j^*, s^*, \text{BuildTree}\left(I_{j^*,s^*}^{-}, k, \mathcal{D} \right), \text{BuildTree}\left(I_{j^*,s^*}^{+}, k, \mathcal{D} \right) \right)$

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

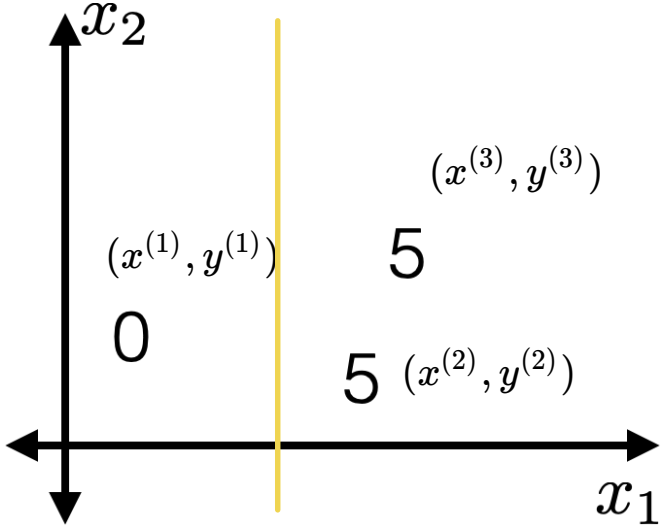
1. **if** $|I| > k$
2. **for** each split dim j and split value s
3. Set $I_{j,s}^{+} = \left\{ i \in I \mid x_j^{(i)} \geq s \right\}$
4. Set $I_{j,s}^{-} = \left\{ i \in I \mid x_j^{(i)} < s \right\}$
5. Set $\hat{y}_{j,s}^{+} = \text{average}_{\, i \in I_{j,s}^{+}} \, y^{(i)}$
6. Set $\hat{y}_{j,s}^{-} = \text{average}_{\, i \in I_{j,s}^{-}} \, y^{(i)}$
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$(I, k, \mathcal{D})$

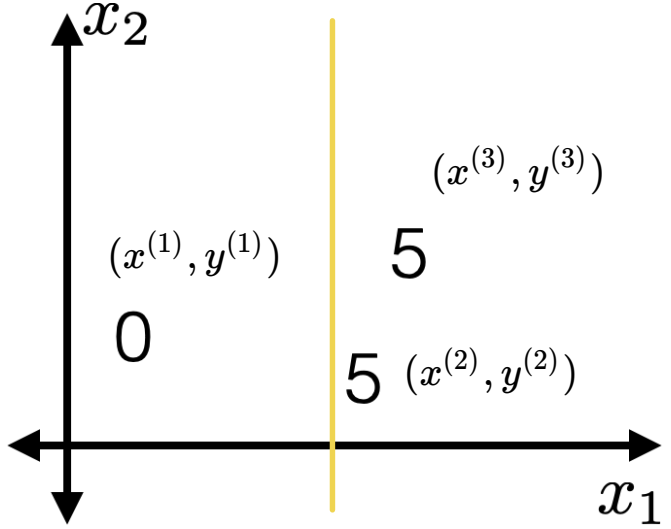
1. **if** $|I| > k$
2. **for** each split dim j and split value s
3. Set $I_{j,s}^{+} = \left\{ i \in I \mid x_j^{(i)} \geq s \right\}$
4. Set $I_{j,s}^{-} = \left\{ i \in I \mid x_j^{(i)} < s \right\}$
5. Set $\hat{y}_{j,s}^{+} = \text{average}_{\, i \in I_{j,s}^{+}} \, y^{(i)}$
6. Set $\hat{y}_{j,s}^{-} = \text{average}_{\, i \in I_{j,s}^{-}} \, y^{(i)}$
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$(I, k, \mathcal{D})$

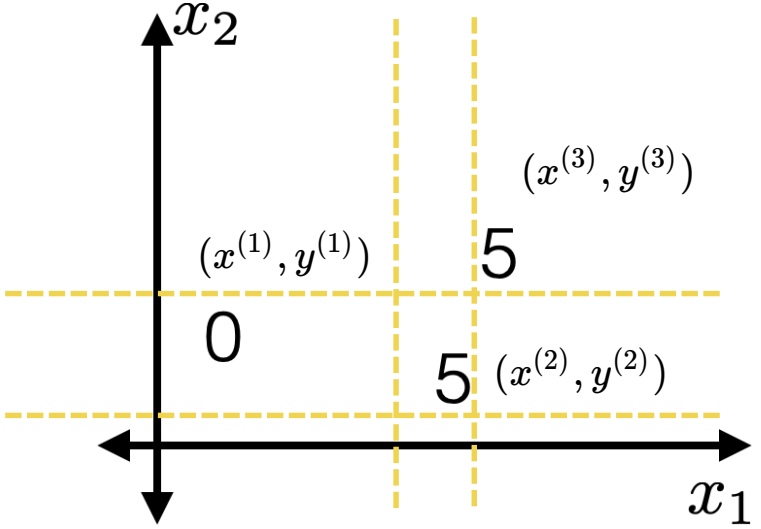
1. **if** $|I| > k$
2. **for** each split dim j and split value s
3. Set $I_{j,s}^{+} = \left\{ i \in I \mid x_j^{(i)} \geq s \right\}$
4. Set $I_{j,s}^{-} = \left\{ i \in I \mid x_j^{(i)} < s \right\}$
5. Set $\hat{y}_{j,s}^{+} = \text{average}_{\, i \in I_{j,s}^{+}} \, y^{(i)}$
6. Set $\hat{y}_{j,s}^{-} = \text{average}_{\, i \in I_{j,s}^{-}} \, y^{(i)}$
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$(I, k, \mathcal{D})$

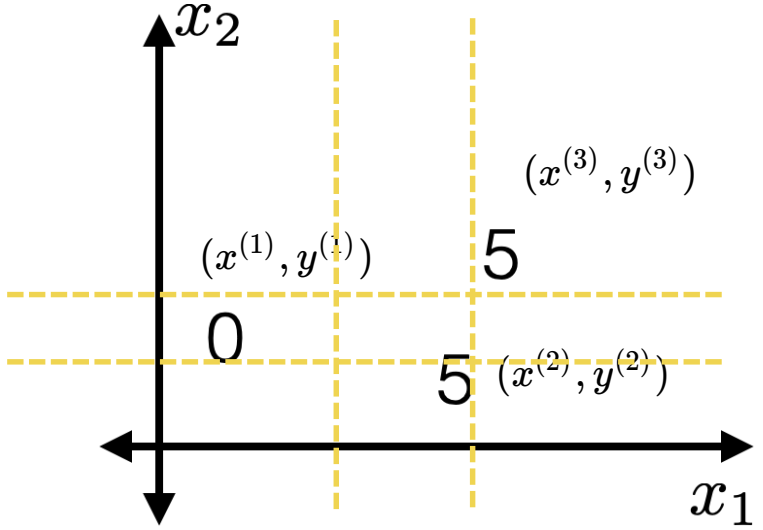
1. **if** $|I| > k$
2. **for** each split dim j and split value s
3. Set $I^+_{j,s} = \left\{ i \in I \mid x^{(i)}_j \geq s \right\}$
4. Set $I^-_{j,s} = \left\{ i \in I \mid x^{(i)}_j < s \right\}$
5. Set $\hat{y}^+_{j,s} = \text{average}_{\ i \in I^+_{j,s}}\ y^{(i)}$
6. Set $\hat{y}^-_{j,s} = \text{average}_{\ i \in I^-_{j,s}}\ y^{(i)}$
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$(I, k, \mathcal{D})$

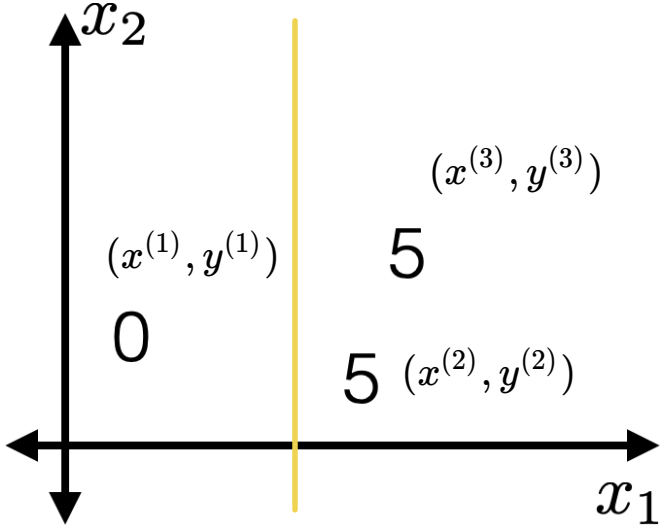
1. **if** $|I| > k$
2. **for** each split dim j and split value s
3. Set $I_{j,s}^{+} = \left\{ i \in I \mid x_j^{(i)} \geq s \right\}$
4. Set $I_{j,s}^{-} = \left\{ i \in I \mid x_j^{(i)} < s \right\}$
5. Set $\hat{y}_{j,s}^{+} = \text{average}_{\, i \in I_{j,s}^{+}} \, y^{(i)}$
6. Set $\hat{y}_{j,s}^{-} = \text{average}_{\, i \in I_{j,s}^{-}} \, y^{(i)}$
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 $\left(j^*, s^*, \text{BuildTree}\left(I_{j^*,s^*}^{-}, k \right), \text{BuildTree}\left(I_{j^*,s^*}^{+}, k \right) \right)$



- 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 $\left(j^*, s^*, \text{BuildTree}\left(I^-_{j^*,s^*}, k\right), \text{BuildTree}\left(I^+_{j^*,s^*}, k\right)\right)$

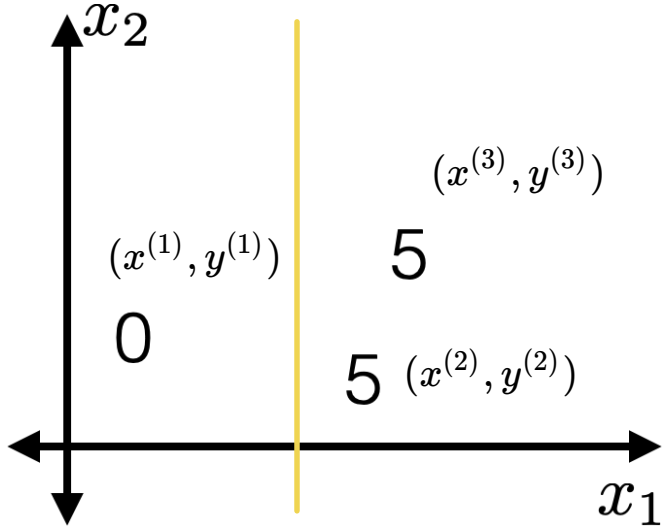


Suppose line 8 sets this (j^*, s^*),

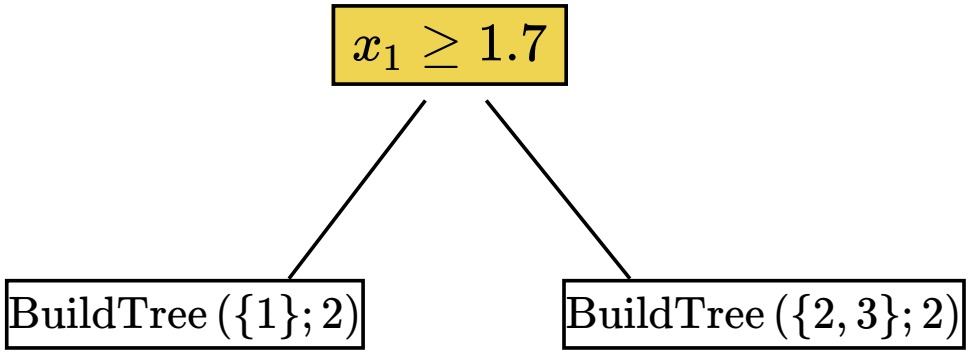
say $= (j^*, s^*) = (1, 1.7)$

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 $\left(j^*, s^*, \text{BuildTree}\left(I^-_{j^*,s^*}, k\right), \text{BuildTree}\left(I^+_{j^*,s^*}, k\right)\right)$

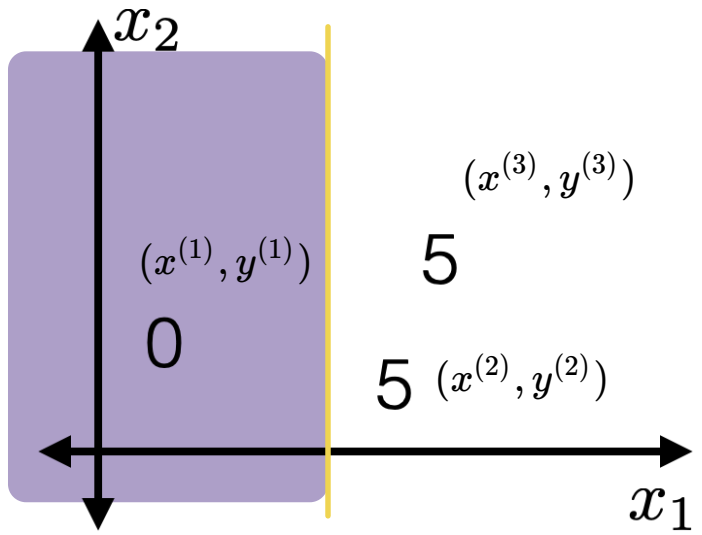


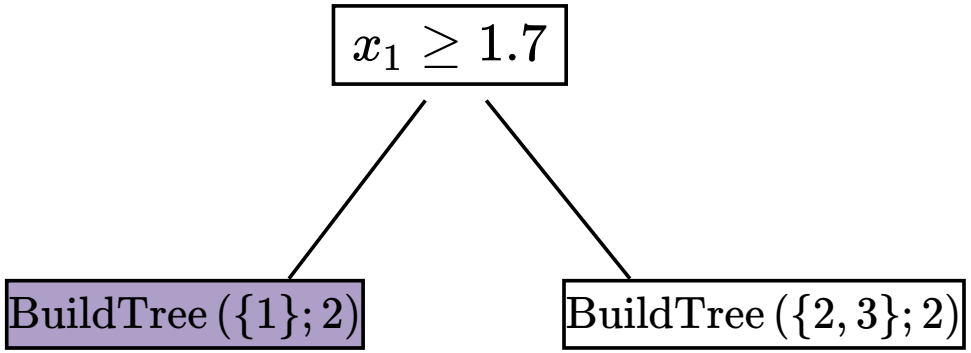
Line 12 recursion



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

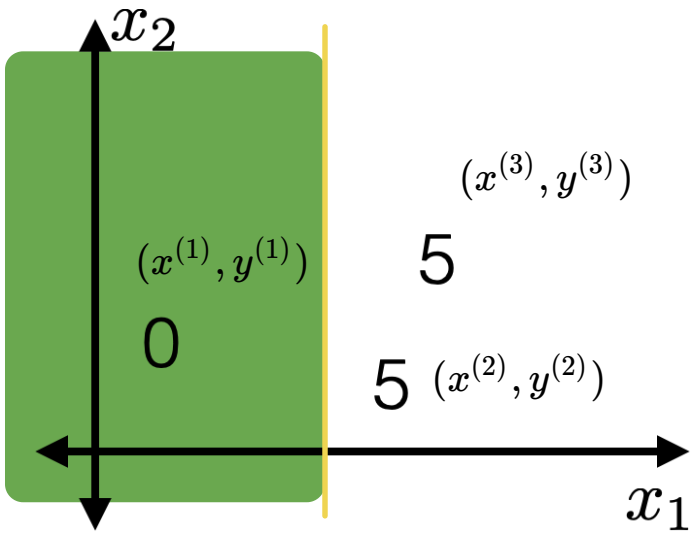
1. **if** $|I| > k$
2. **for** each split dim j and split value s
3. Set $I^+_{j,s} = \left\{ i \in I \mid x^{(i)}_j \geq s \right\}$
4. Set $I^-_{j,s} = \left\{ i \in I \mid x^{(i)}_j < s \right\}$
5. Set $\hat{y}^+_{j,s} = \text{average}_{\ i \in I^+_{j,s}}\ y^{(i)}$
6. Set $\hat{y}^-_{j,s} = \text{average}_{\ i \in I^-_{j,s}}\ y^{(i)}$
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 $\left(j^*, s^*, \text{BuildTree}\left(I^-_{j^*,s^*}, k\right), \text{BuildTree}\left(I^+_{j^*,s^*}, k\right)\right)$

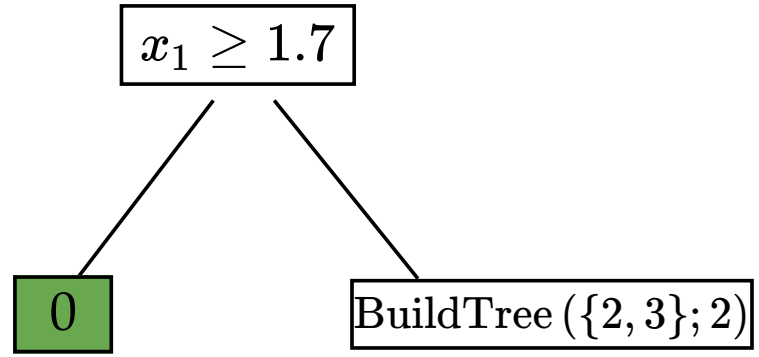




BuildTree$(I, k, \mathcal{D})$

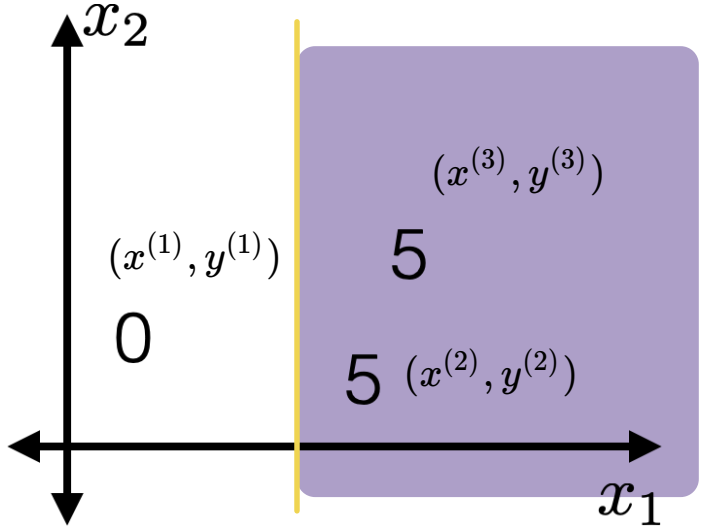
1. **if** $|I| > k$
2. **for** each split dim j and split value s
3. Set $I^+_{j,s} = \left\{ i \in I \mid x^{(i)}_j \geq s \right\}$
4. Set $I^-_{j,s} = \left\{ i \in I \mid x^{(i)}_j < s \right\}$
5. Set $\hat{y}^+_{j,s} = \text{average}_{\ i \in I^+_{j,s}}\ y^{(i)}$
6. Set $\hat{y}^-_{j,s} = \text{average}_{\ i \in I^-_{j,s}}\ y^{(i)}$
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 $\left(j^*, s^*, \text{BuildTree}\left(I^-_{j^*,s^*}, k\right), \text{BuildTree}\left(I^+_{j^*,s^*}, k\right)\right)$

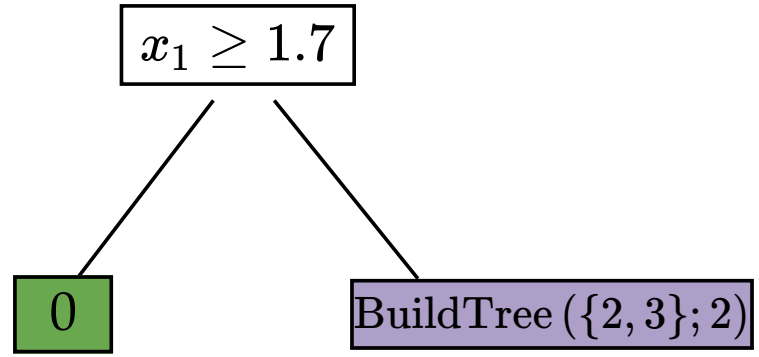




BuildTree$(I, k, \mathcal{D})$

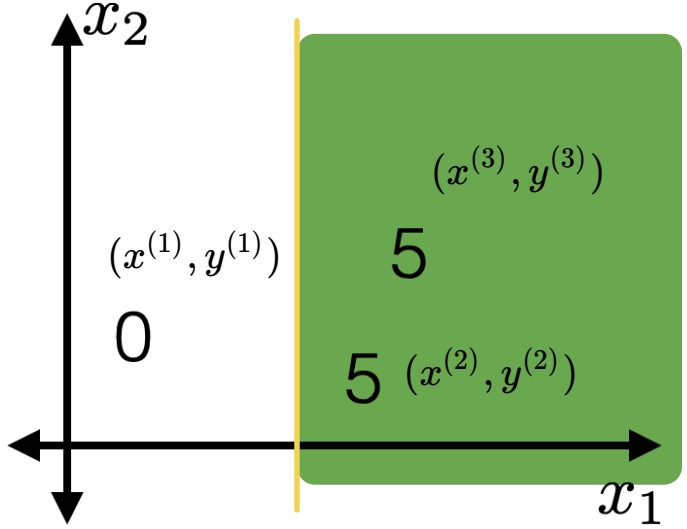
1. **if** $|I| > k$
2. **for** each split dim j and split value s
3. Set $I^+_{j,s} = \left\{ i \in I \mid x^{(i)}_j \geq s \right\}$
4. Set $I^-_{j,s} = \left\{ i \in I \mid x^{(i)}_j < s \right\}$
5. Set $\hat{y}^+_{j,s} = \text{average}_{\ i \in I^+_{j,s}}\ y^{(i)}$
6. Set $\hat{y}^-_{j,s} = \text{average}_{\ i \in I^-_{j,s}}\ y^{(i)}$
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 $\left(j^*, s^*, \text{BuildTree}\left(I^-_{j^*,s^*}, k\right), \text{BuildTree}\left(I^+_{j^*,s^*}, k\right)\right)$

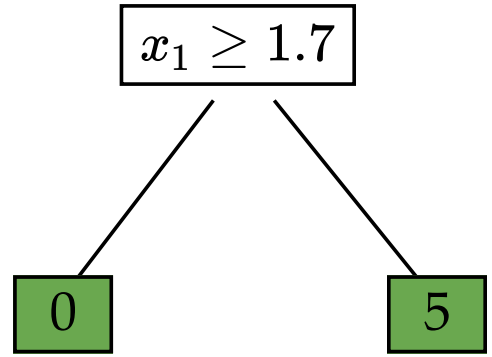




BuildTree$(I, k, \mathcal{D})$

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 $\left(j^*, s^*, \text{BuildTree}\left(I^-_{j^*,s^*}, k\right), \text{BuildTree}\left(I^+_{j^*,s^*}, k\right)\right)$





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

BuildTree for classification

For classification

majority vote as the (intermediate) prediction

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

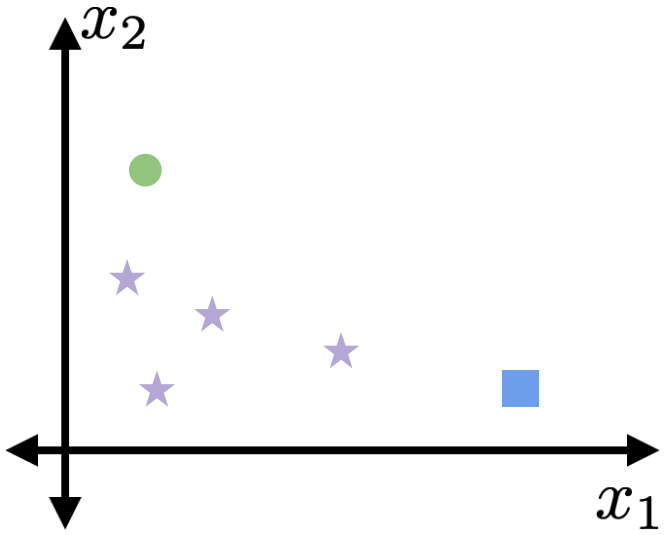
1. **if** $|I| > k$
2. **for** each split dim j and split value s
3. Set $I_{j,s}^{+} = \left\{ i \in I \mid x_j^{(i)} \geq s \right\}$
4. Set $I_{j,s}^{-} = \left\{ i \in I \mid x_j^{(i)} < s \right\}$
5. Set $\hat{y}_{j,s}^{+} = \text{majority}_{\ i \in I_{j,s}^{+}}\ y^{(i)}$
6. Set $\hat{y}_{j,s}^{-} = \text{majority}_{\ i \in I_{j,s}^{-}}\ y^{(i)}$
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

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

empirical probability

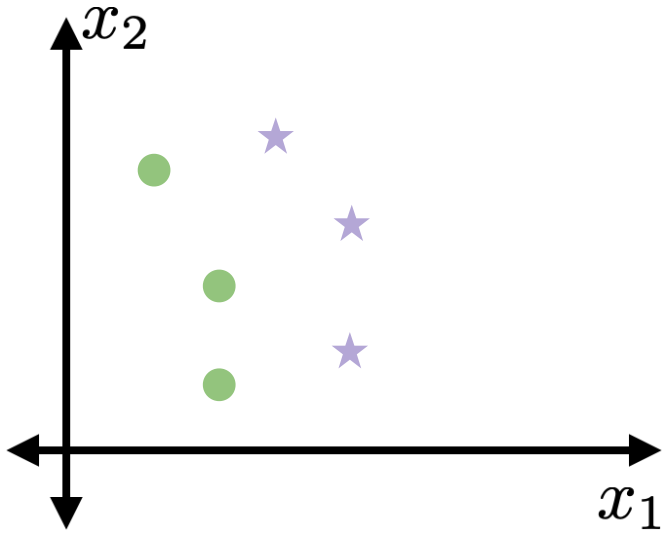
e.g.: c iterates over 3 classes 



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

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

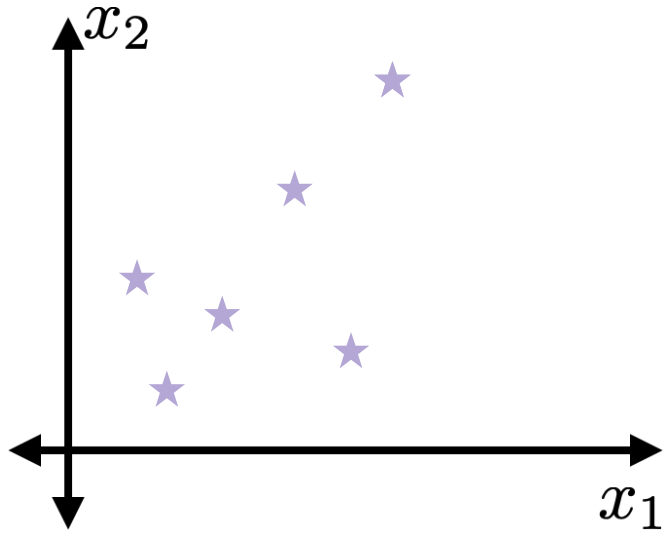
(about 1.252)



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

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

(about 1.1)



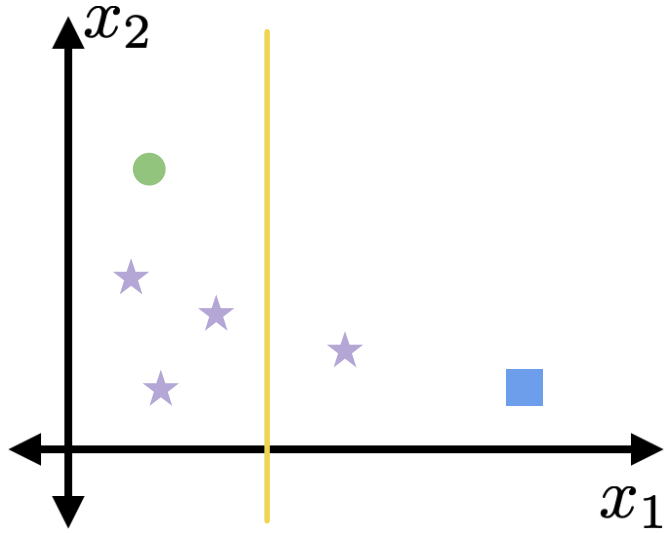
$\hat{P}_c$:  $\frac{6}{6}$  0  0

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

(= 0)

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 $\left(j^*, s^*, \text{BuildTree}\left(I_{j^*,s^*}^{-}, k\right), \text{BuildTree}\left(I_{j^*,s^*}^{+}, k\right)\right)$



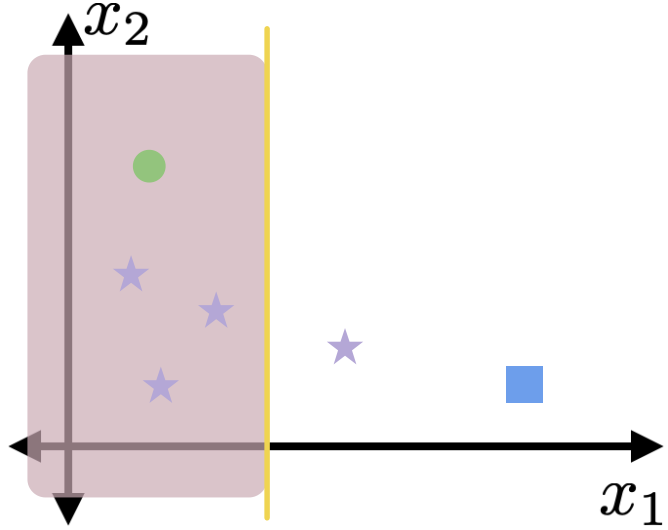
$$E_{j,s} = \frac{4}{6} \cdot H\left(I_{j,s}^{-}\right) + \frac{2}{6} \cdot H\left(I_{j,s}^{+}\right)$$

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 $\left(j^*, s^*, \text{BuildTree}\left(I_{j^*,s^*}^{-}, k\right), \text{BuildTree}\left(I_{j^*,s^*}^{+}, k\right)\right)$



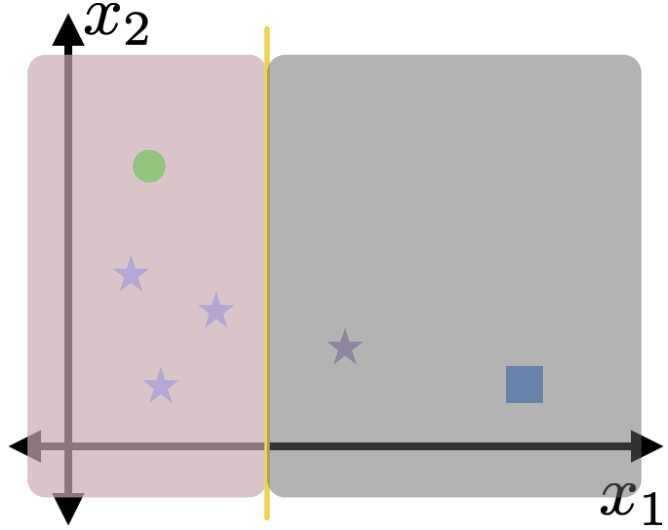
$$\frac{4}{6} \cdot H\left(I_{j,s}^{-}\right) + \frac{2}{6} \cdot H\left(I_{j,s}^{+}\right)$$

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

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

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 $\left(j^*, s^*, \text{BuildTree}\left(I_{j^*,s^*}^{-}, k\right), \text{BuildTree}\left(I_{j^*,s^*}^{+}, k\right)\right)$



≈ 0.811

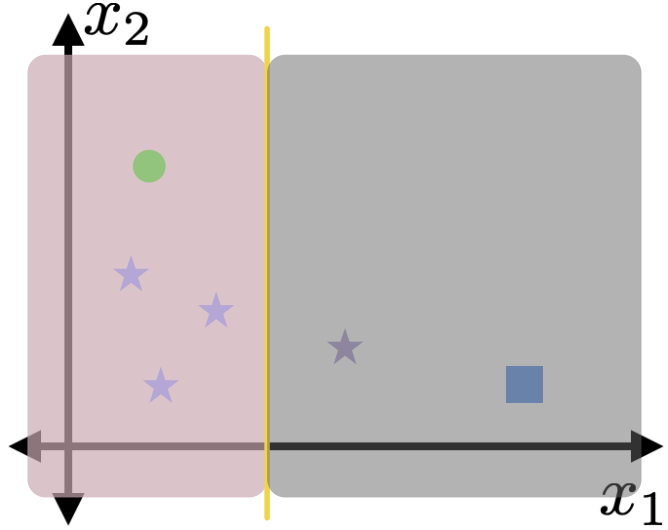
$$\frac{4}{6} \cdot H\left(I_{j,s}^{-}\right) + \frac{2}{6} \cdot H\left(I_{j,s}^{+}\right)$$

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

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

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 $\left(j^*, s^*, \text{BuildTree}\left(I_{j^*,s^*}^{-}, k\right), \text{BuildTree}\left(I_{j^*,s^*}^{+}, k\right)\right)$



≈ 0.811 $= 1$

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

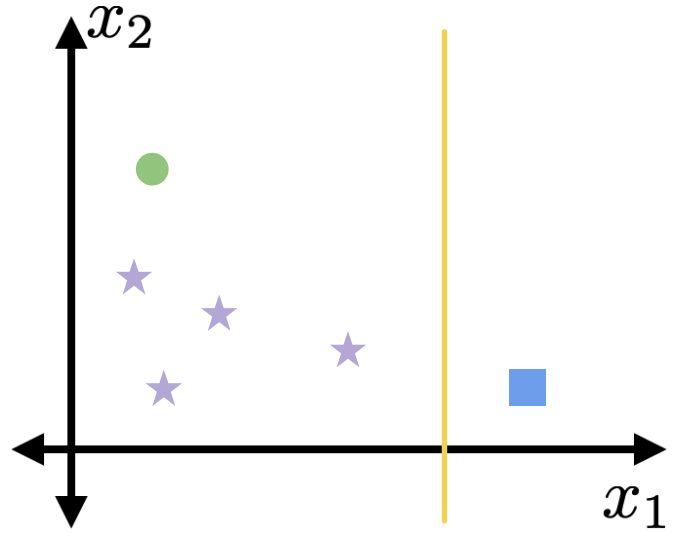
$$\approx \frac{4}{6} \cdot 0.811 + \frac{2}{6} \cdot 1$$

$$= 0.874$$

(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 $\left(j^*, s^*, \text{BuildTree}\left(I_{j^*,s^*}^{-}, k\right), \text{BuildTree}\left(I_{j^*,s^*}^{+}, k\right)\right)$



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

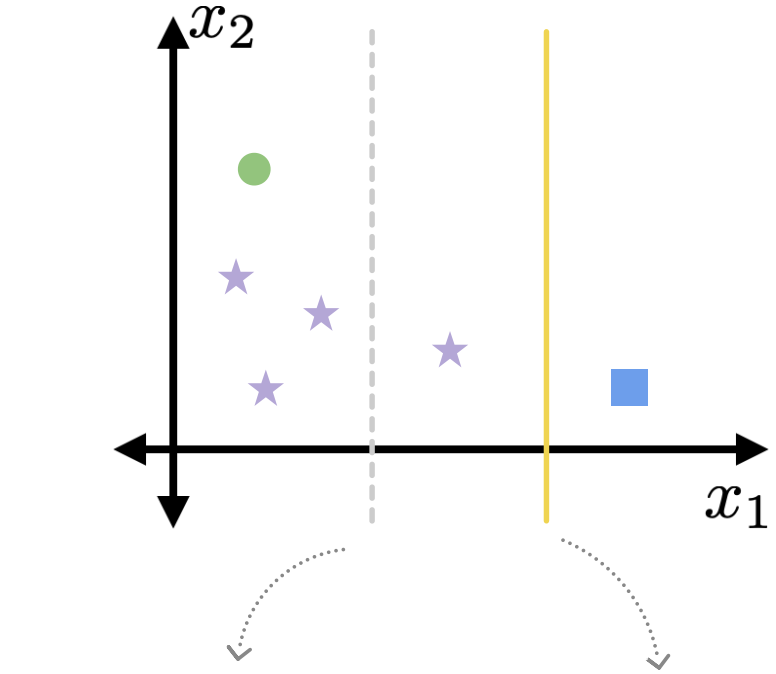
$$\tfrac{5}{6} \cdot H\left(I_{j,s}^{-}\right) + \tfrac{1}{6} \cdot H\left(I_{j,s}^{+}\right)$$

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

(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)

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

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 Netflix movie recommenda [...]

ELSEVIER International Journal of Forecasting 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

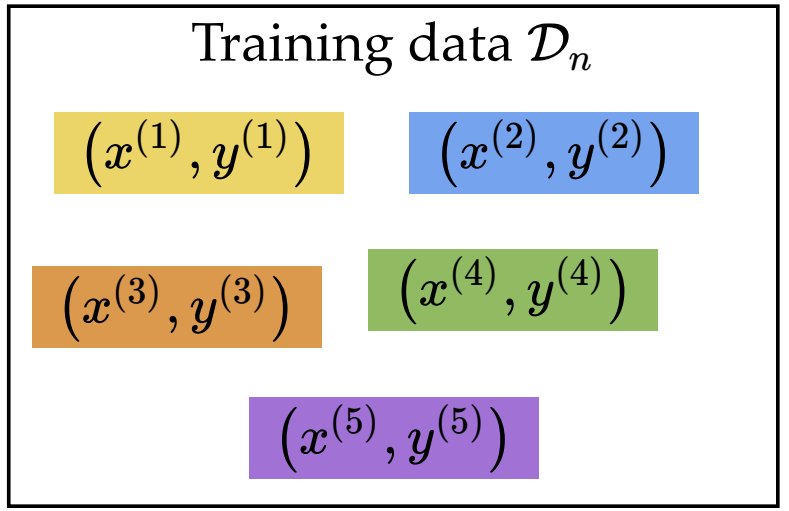
Observed and forecasted new and total reported COVID-19 deaths as of November 9, 2020.

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.

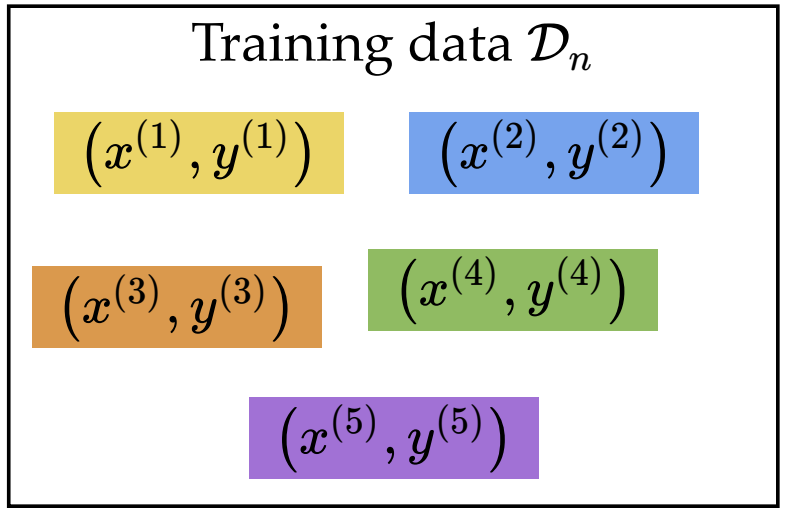
Bagging

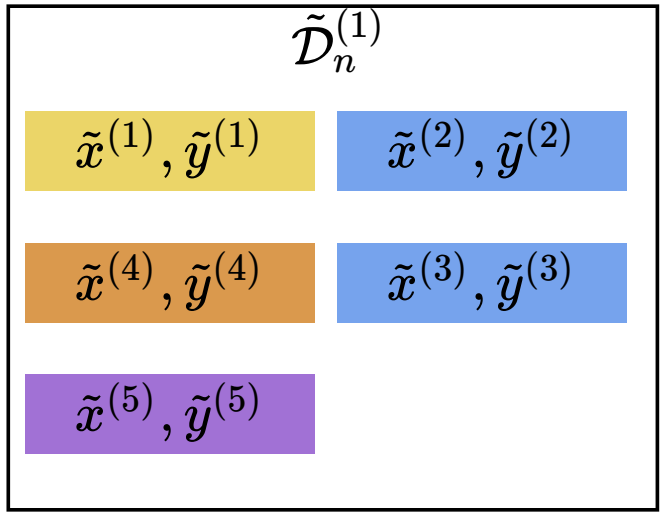
- One of multiple ways to make and use an ensemble
- Bagging = **B**ootstrap **agg**regat**ing**



Bagging

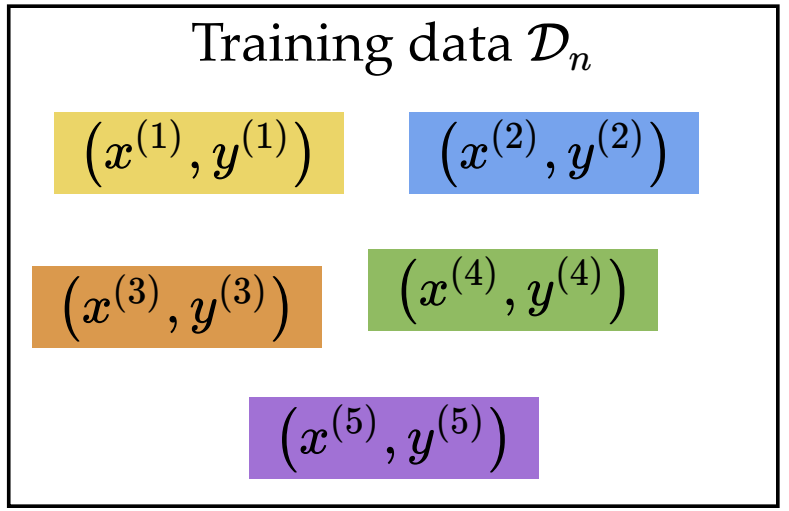
- Training data $\mathcal{D}_n$
- For $b = 1, \ldots, B$
  - Draw a new "data set" $\tilde{\mathcal{D}}_n^{(b)}$ of size n by sampling with replacement from $\mathcal{D}_n$

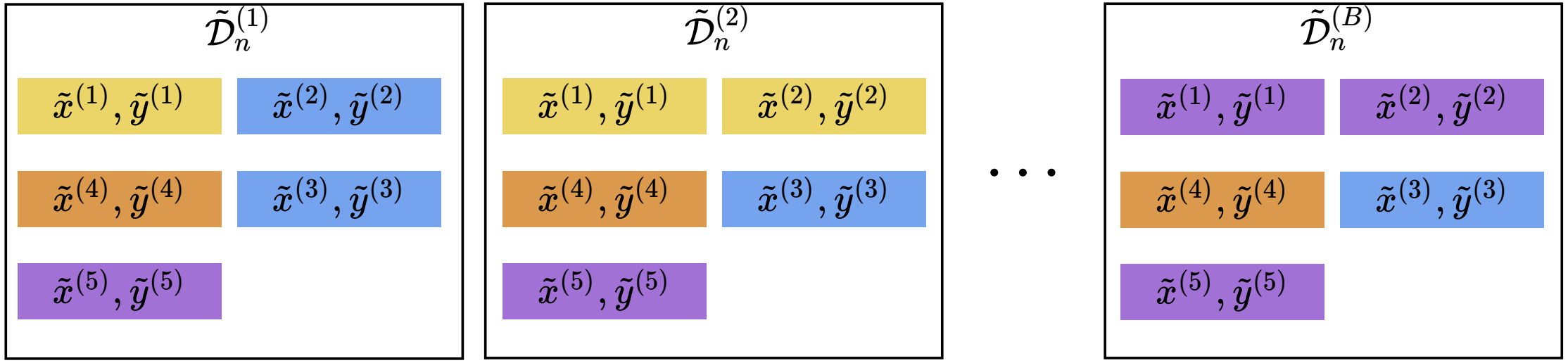




Bagging

- Training data $\mathcal{D}_n$
- For $b = 1, \ldots, B$
  - Draw a new "data set" $\tilde{\mathcal{D}}_n^{(b)}$ of size n by sampling with replacement from $\mathcal{D}_n$
  - Train a predictor $\hat{f}^{(b)}$ on $\tilde{\mathcal{D}}_n^{(b)}$



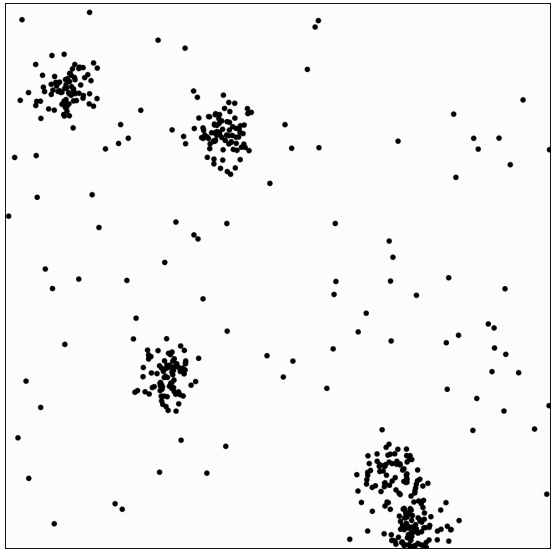


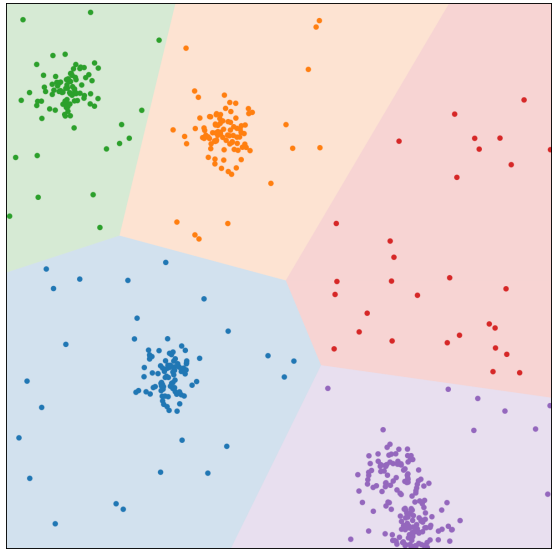
Bagging

- Training data $\mathcal{D}_n$
- For $b = 1, \ldots, B$
  - Draw a new "data set" $\tilde{\mathcal{D}}_n^{(b)}$ of size n by sampling with replacement from $\mathcal{D}_n$
  - Train a predictor $\hat{f}^{(b)}$ on $\tilde{\mathcal{D}}_n^{(b)}$
- Return
  - Regression: $\hat{f}_{\text{bag}}(x) = \frac{1}{B} \sum_{b=1}^{B} \hat{f}^{(b)}(x)$
  - For classification: Predict the class that receives the most votes among the B classifiers.

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





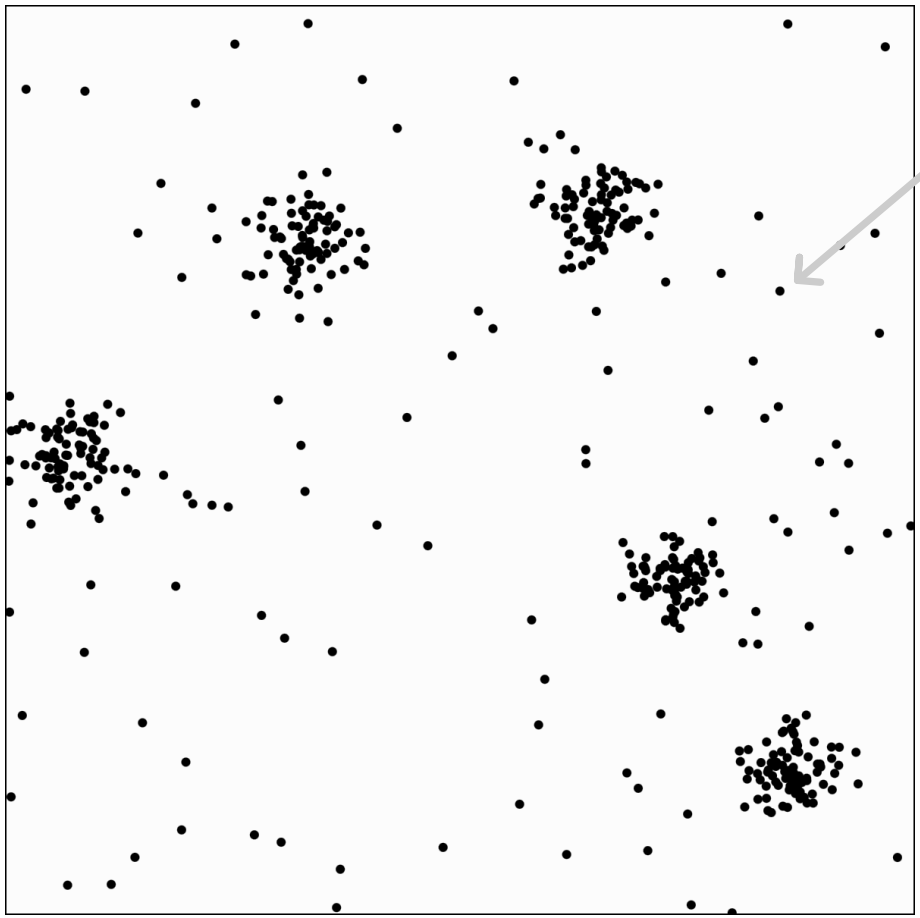
$$\{x^{(1)}\}$$

$$\{x^{(2)}\}$$

$$\{x^{(3)}\}$$

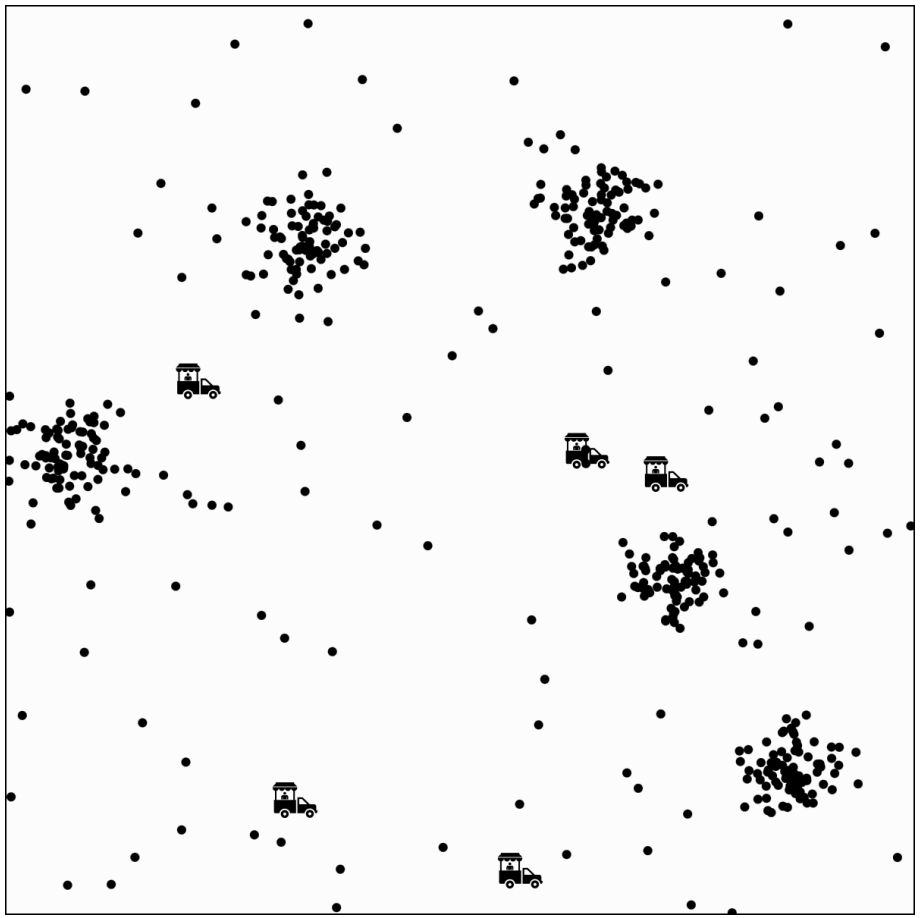
$$\ldots$$

Food-truck placement



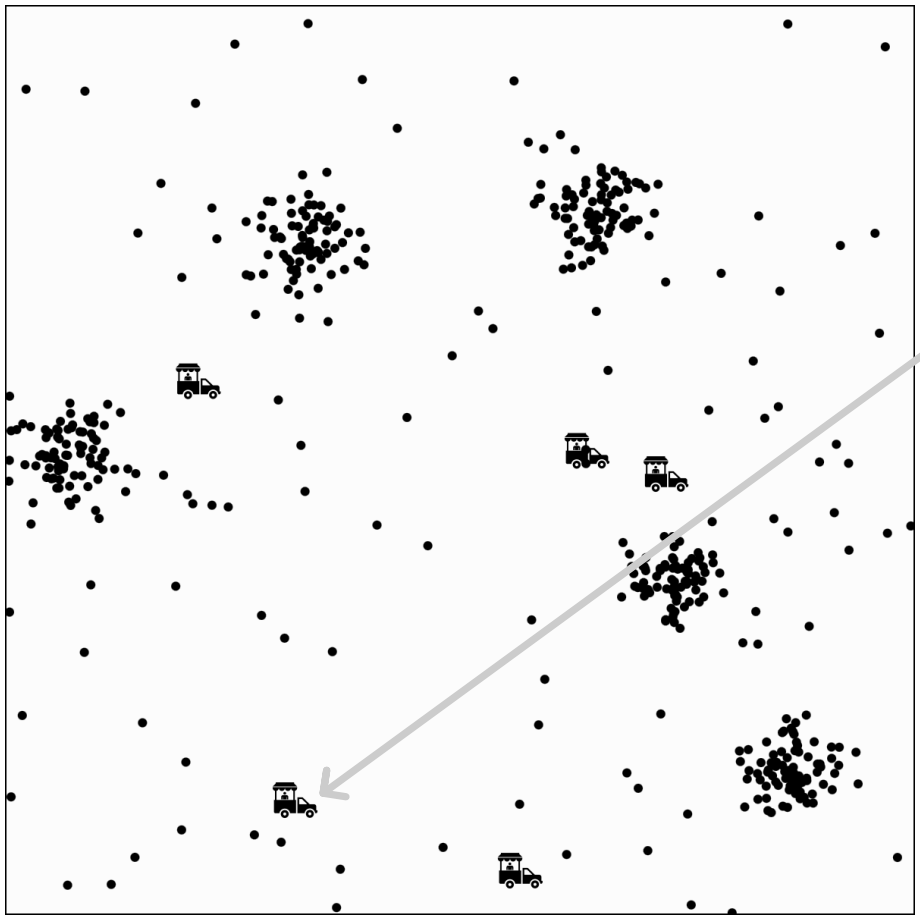
- x_1: longitude, x_2: latitude
- Person i location $x^{(i)}$

Food-truck placement



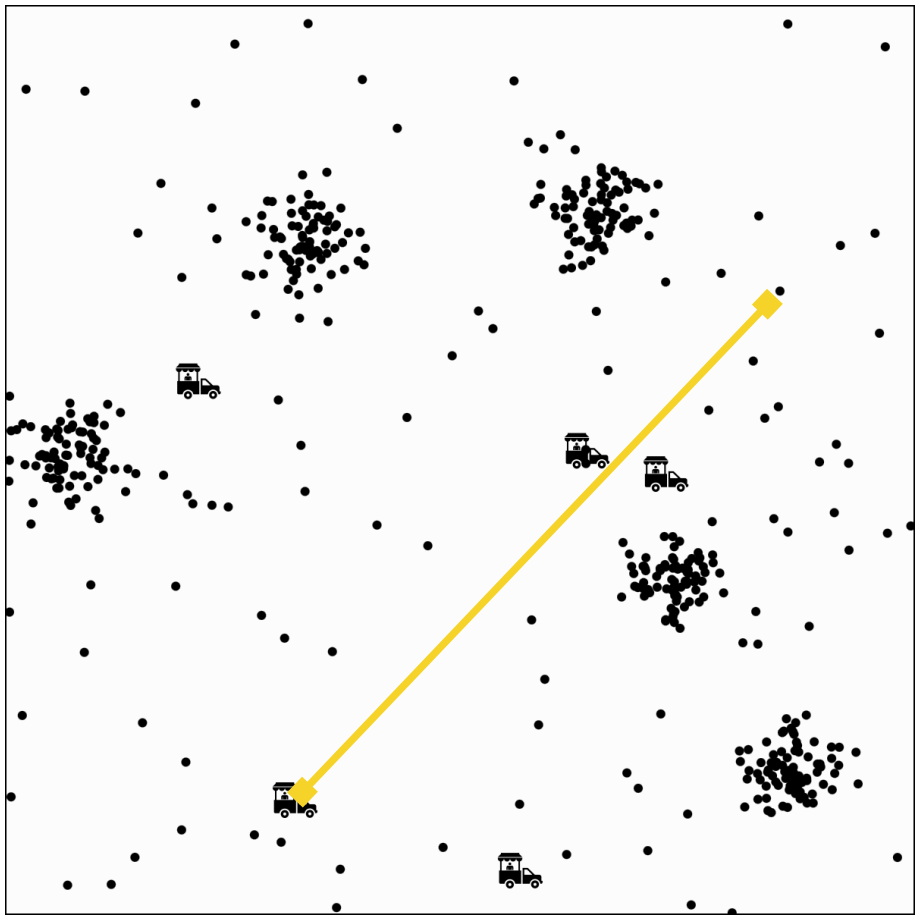
- x_1: longitude, x_2: latitude
- Person i location $x^{(i)}$
- Q: where should I have k food trucks park?

Food-truck placement



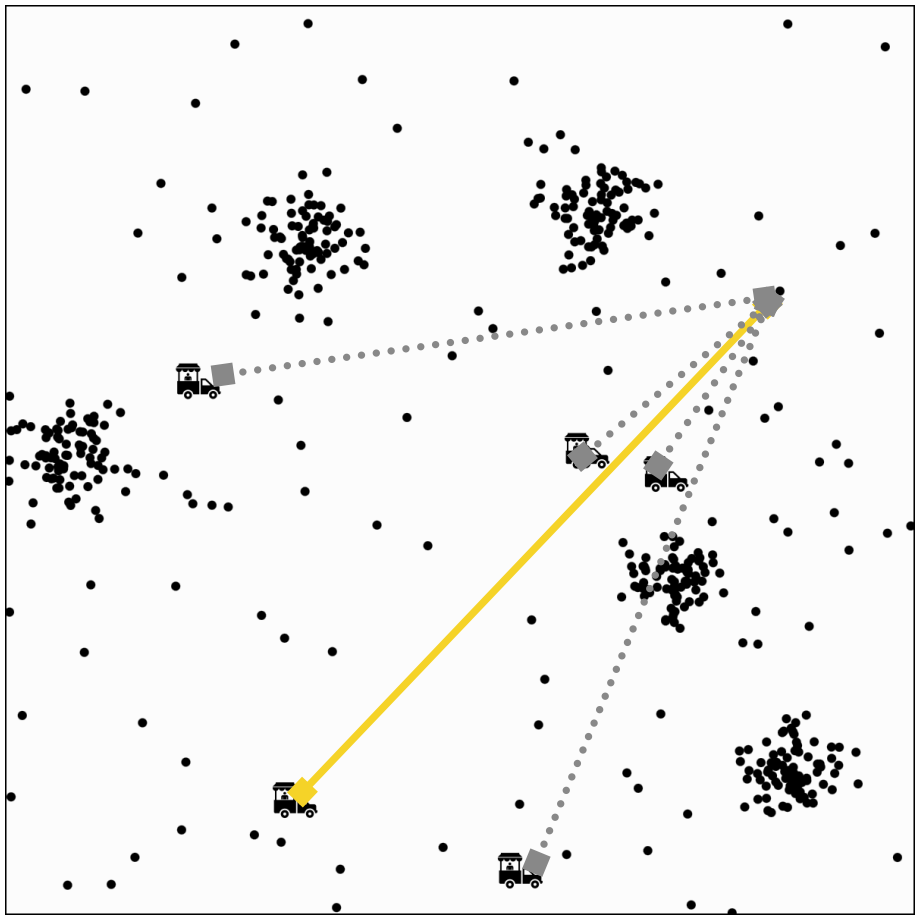
- x_1: longitude, x_2: latitude
- Person i location $x^{(i)}$
- Q: where should I have k food trucks park?
- Food truck j location $\mu^{(j)}$

Food-truck placement



- x_1: longitude, x_2: latitude
- Person i location $x^{(i)}$
- Q: where should I have k food trucks park?
- Food truck j location $\mu^{(j)}$
- Loss if i walks to truck j : $\left\|x^{(i)} - \mu^{(j)}\right\|_2^2$

Food-truck placement



- x_1: longitude, x_2: latitude
- Person i location $x^{(i)}$
- Q: where should I have k food trucks park?
- Food truck j location $\mu^{(j)}$
- Loss if i walks to truck j : $\left\|x^{(i)} - \mu^{(j)}\right\|_2^2$
- Index of the truck where person 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, otherwise 0.

k-means objective

what we learn

clustering membership

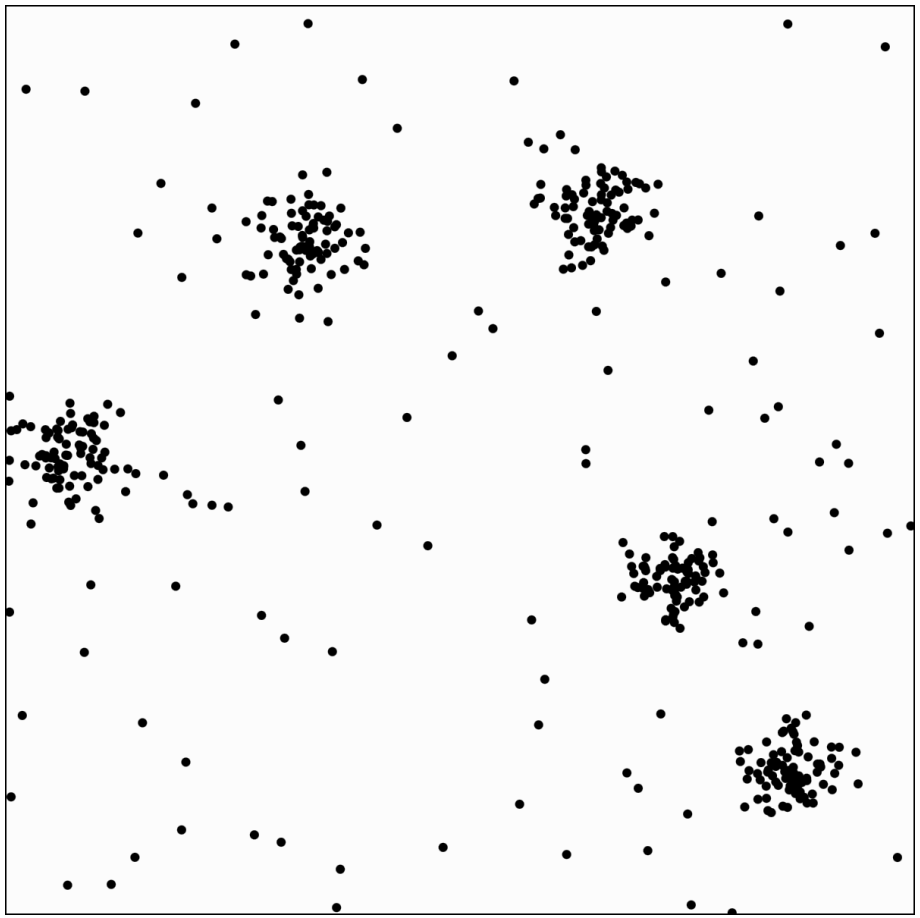
clustering centroid location

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

enumerates over data

enumerates 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 τ

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

    for $i = 1$ to n

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

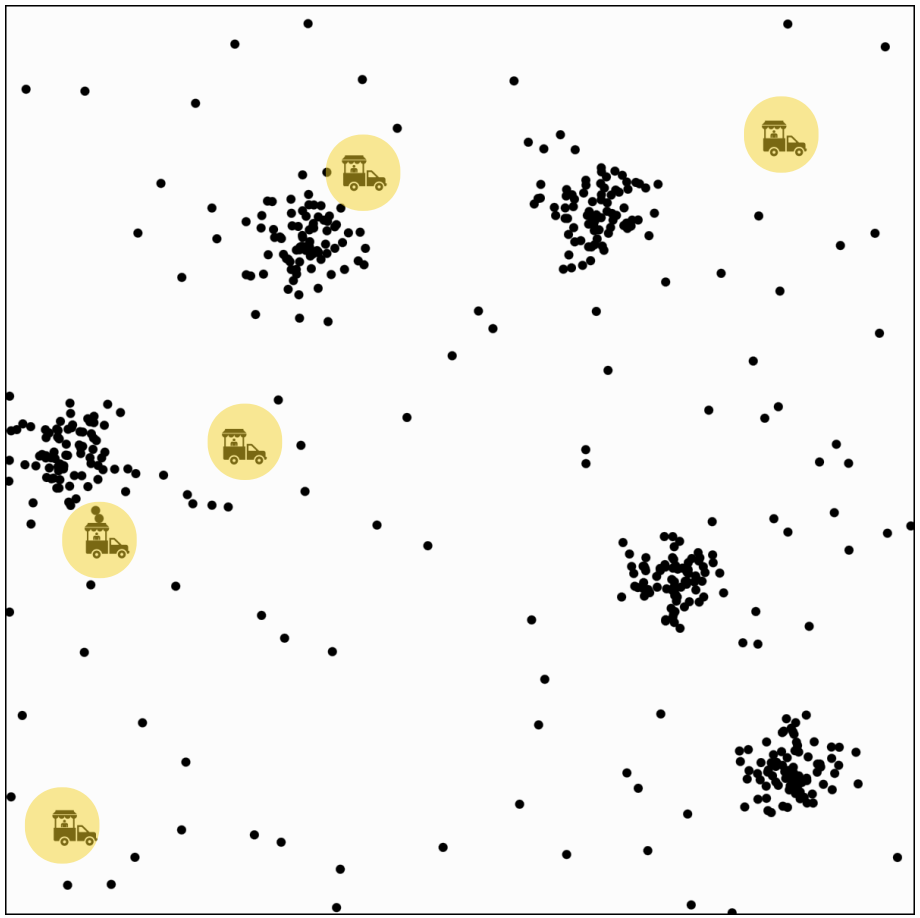
    for $j = 1$ to k

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

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

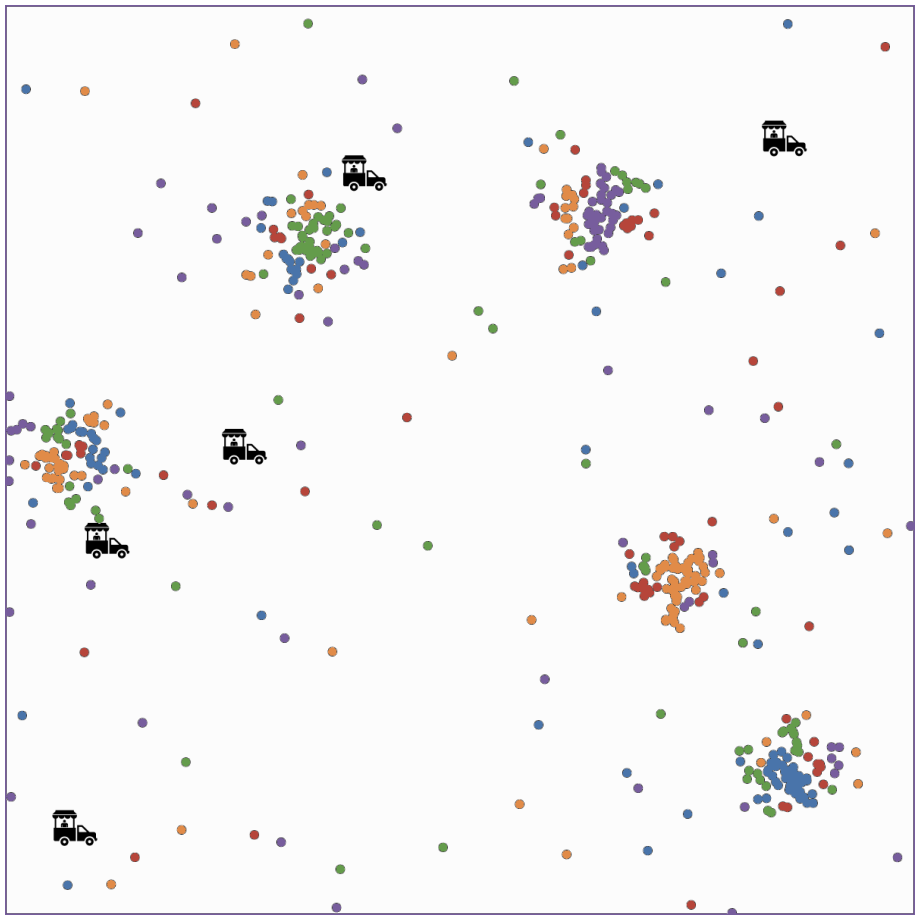
        break

return μ, y



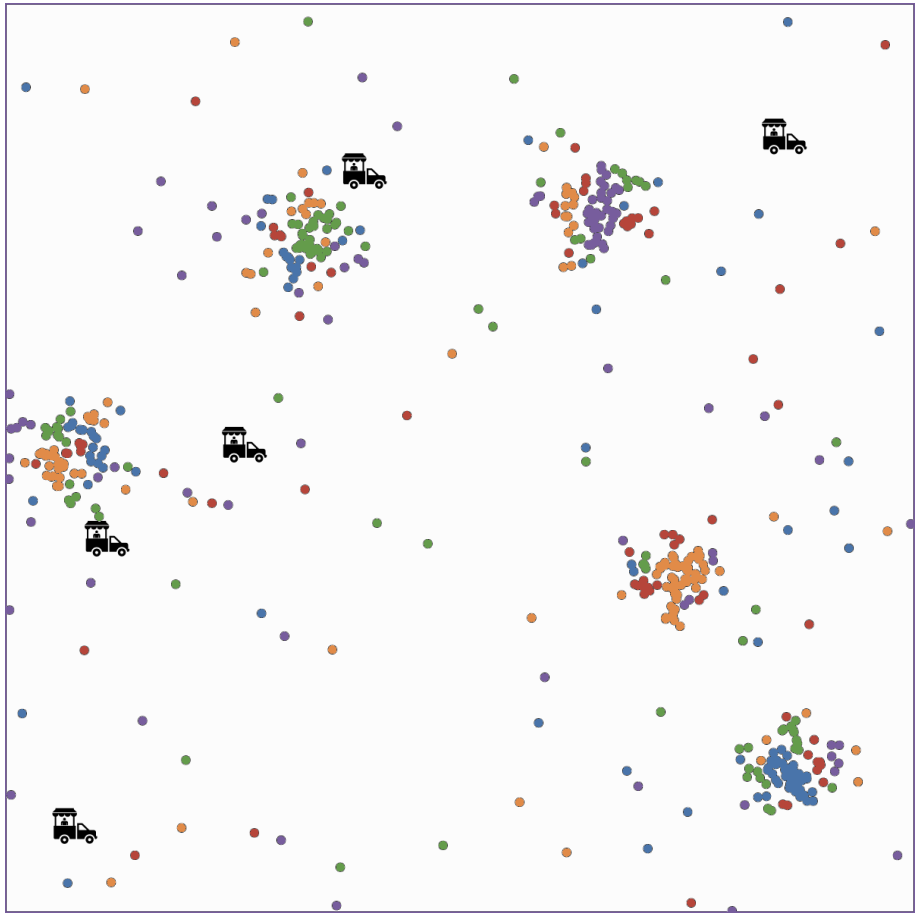
K-MEANS$(k, \tau, \{x^{(i)}\}_{i=1}^{n})$

1 $\mu, y =$ random initialization
2 **for** $t = 1$ to τ
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 μ, y



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

1 μ, y = random initialization
2 **for** $t = 1$ to τ
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 μ, y



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

$\mu, y =$ random initialization

for $t = 1$ to τ

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

  for $i = 1$ to n

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

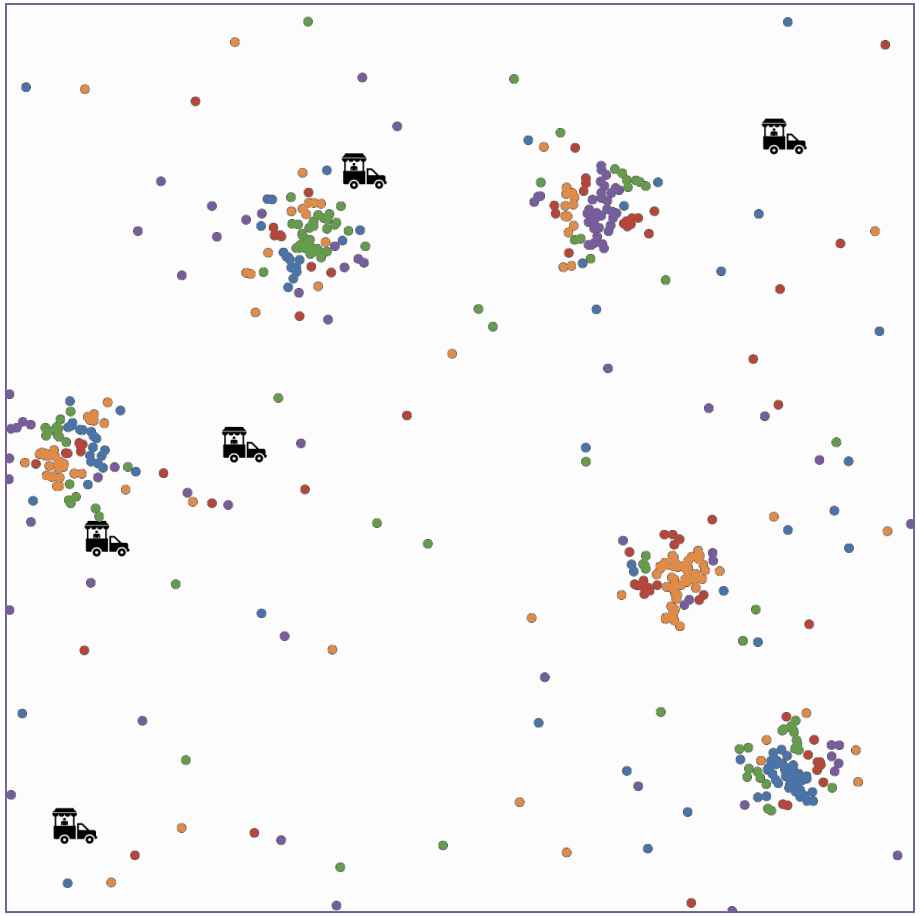
  for $j = 1$ to k

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

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

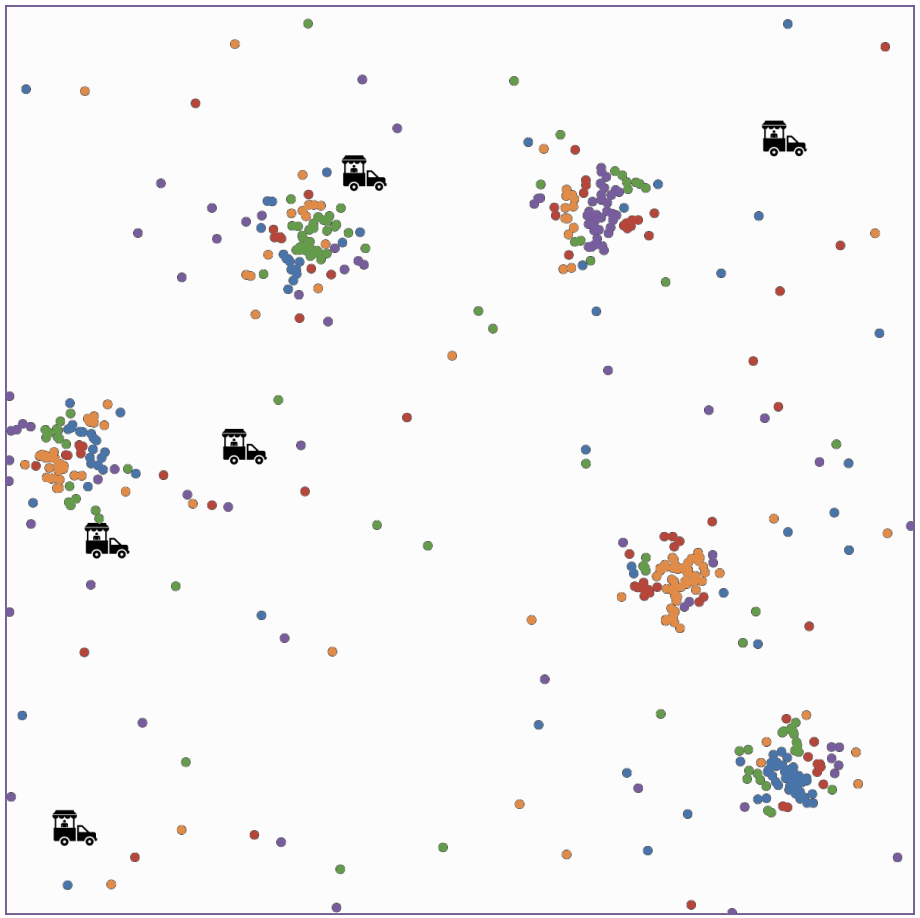
    break

return μ, y



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

1 $\mu, y =$ random initialization
2 **for** $t = 1$ to τ
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 μ, y



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

1 $\mu, y =$ random initialization

2 **for** $t = 1$ to τ

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

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

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

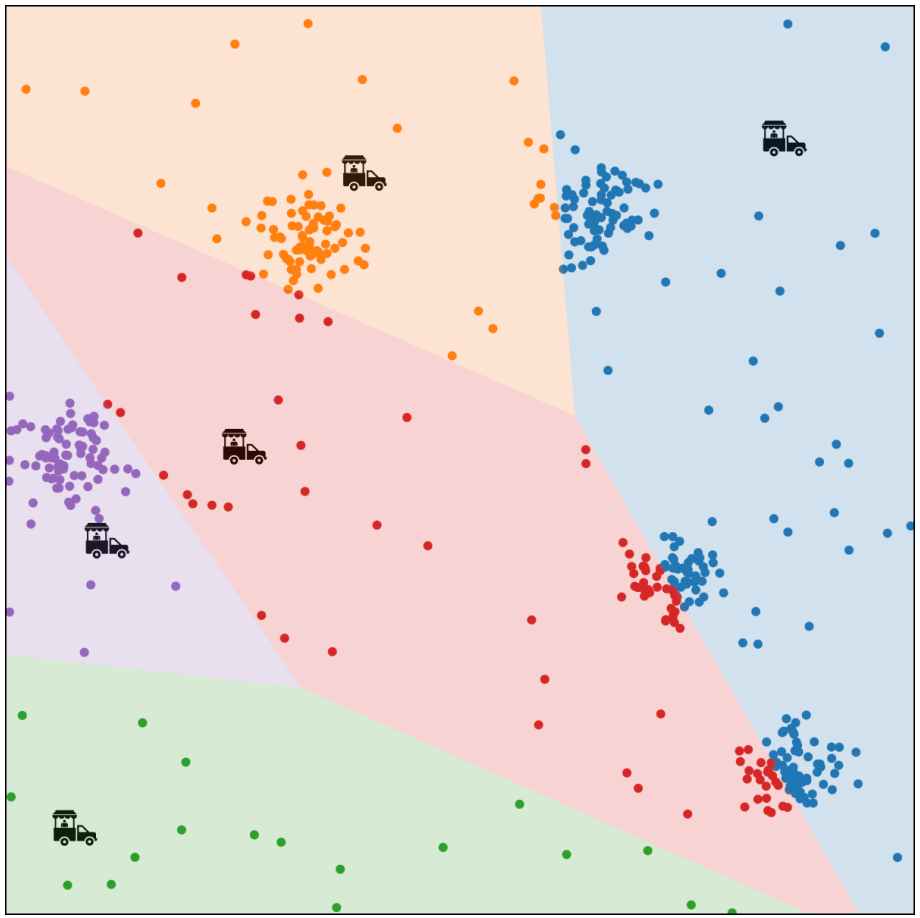
6 $\quad$ **for** $j = 1$ to k

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 μ, y



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

1 $\mu, y =$ random initialization

2 **for** $t = 1$ to τ

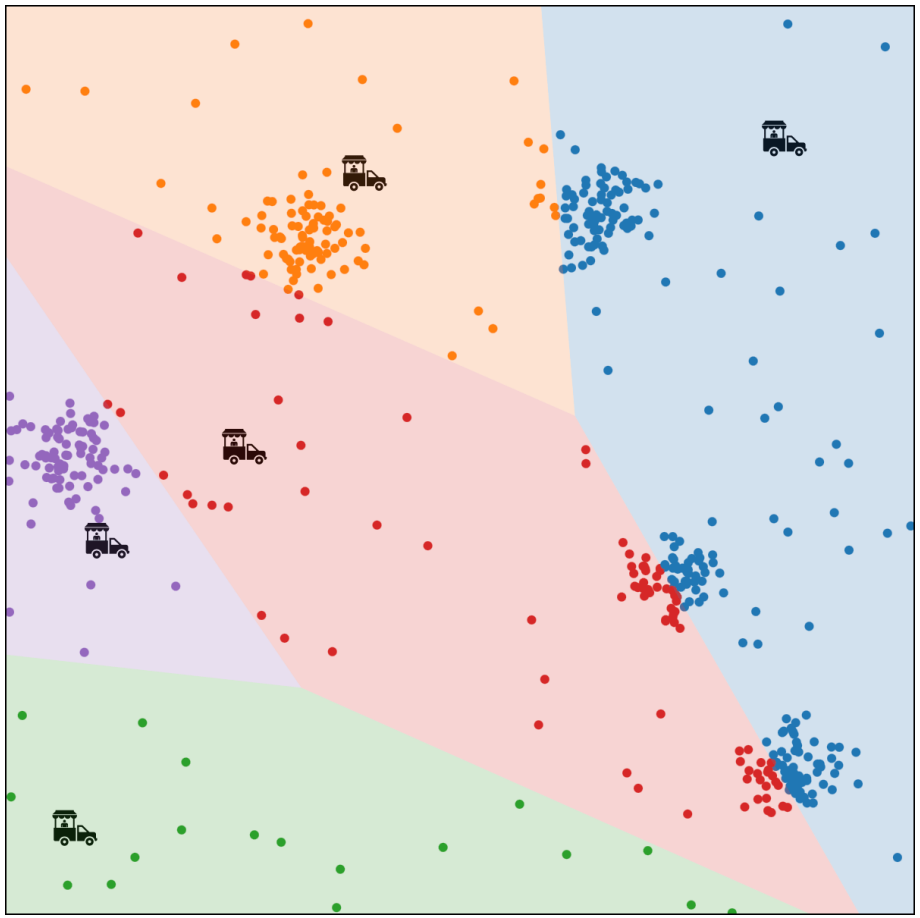
3 $\quad y_{\text{old}} = y$

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

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

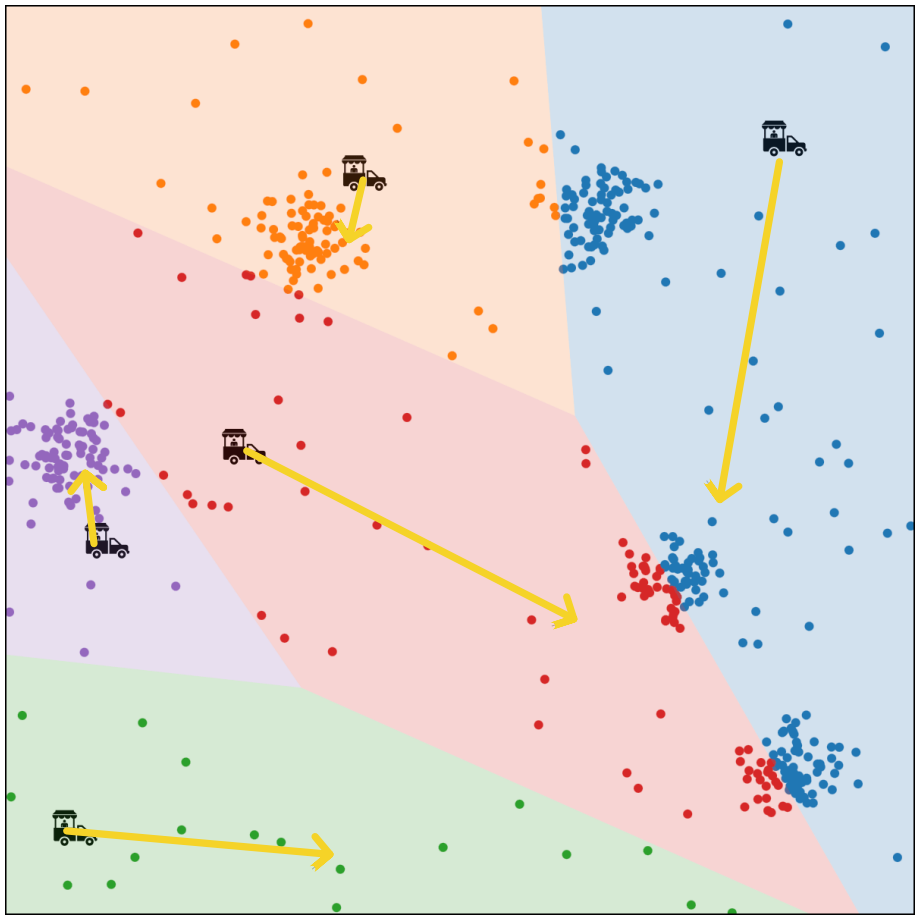
$\quad\quad \ldots$

each person i gets assigned to food truck j, color-coded.



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

1 $\mu, y =$ random initialization
2 **for** $t = 1$ to τ
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 μ, y



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

1 $\mu, y =$ random initialization

2 **for** $t = 1$ to τ

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

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

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

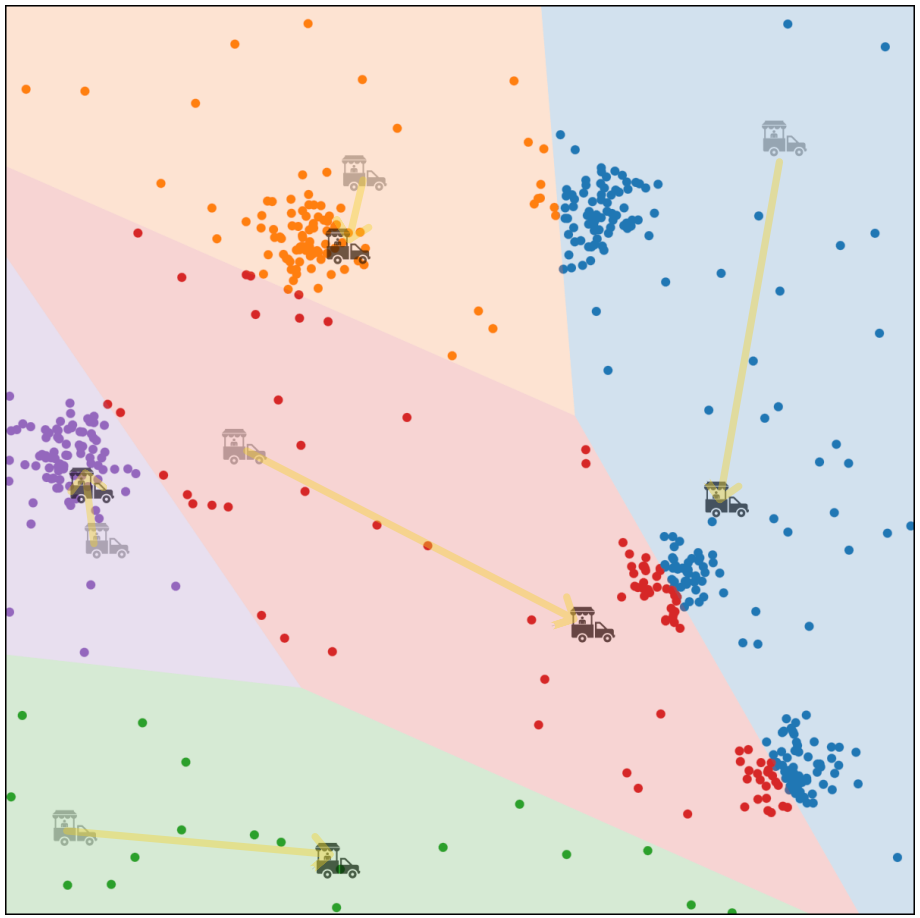
6 $\quad$ **for** $j = 1$ to k

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

...

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

food truck j gets moved to the "central" location of all ppl assigned to it



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

$\mu, y =$ random initialization

for $t = 1$ to τ

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

  for $i = 1$ to n

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

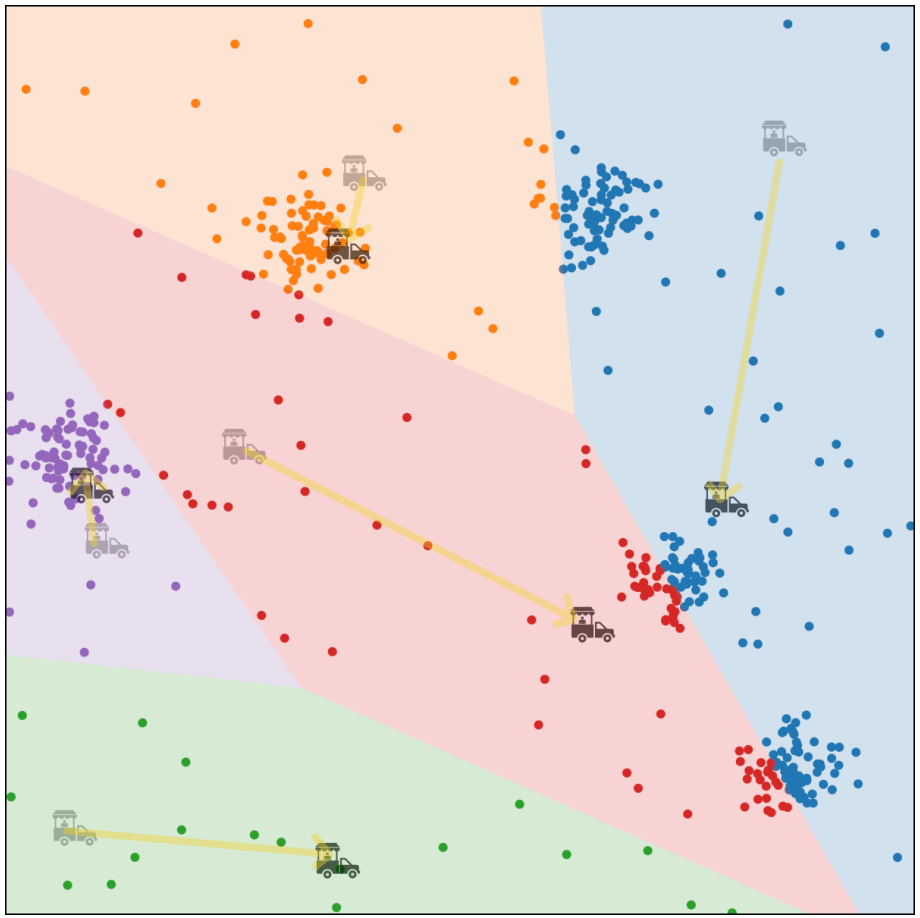
  for $j = 1$ to k

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

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

    break

return μ, y



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

$\mu, y =$ random initialization

for $t = 1$ to τ

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

  for $i = 1$ to n

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

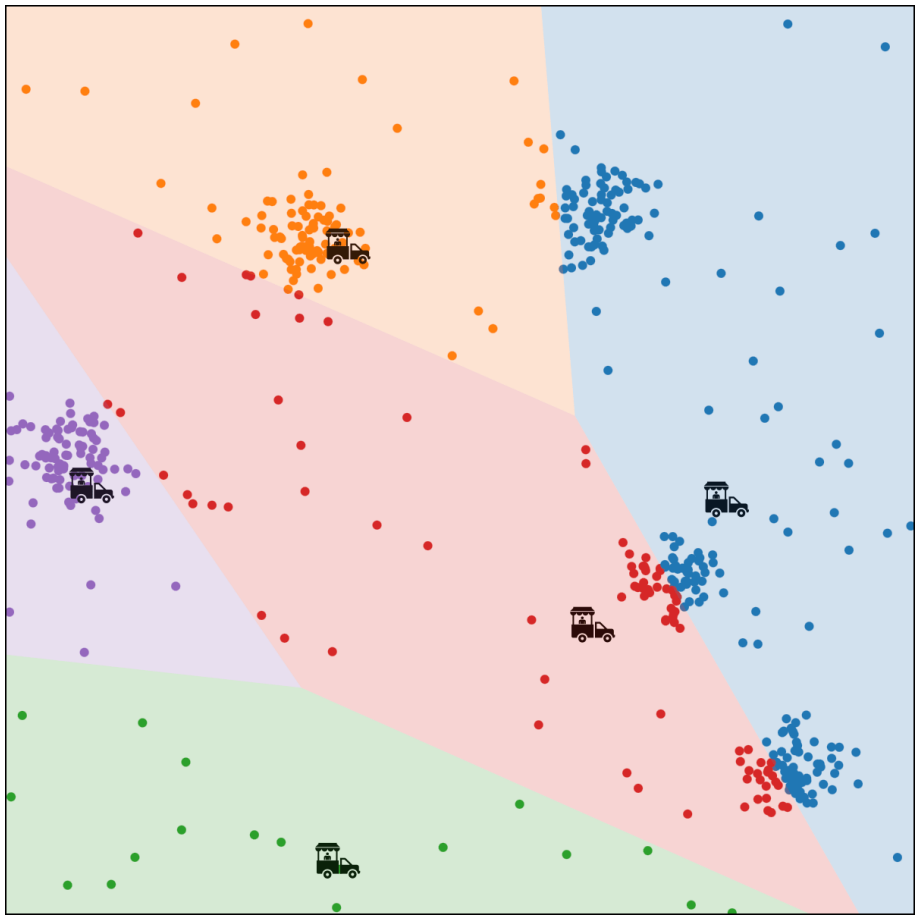
  for $j = 1$ to k

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

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

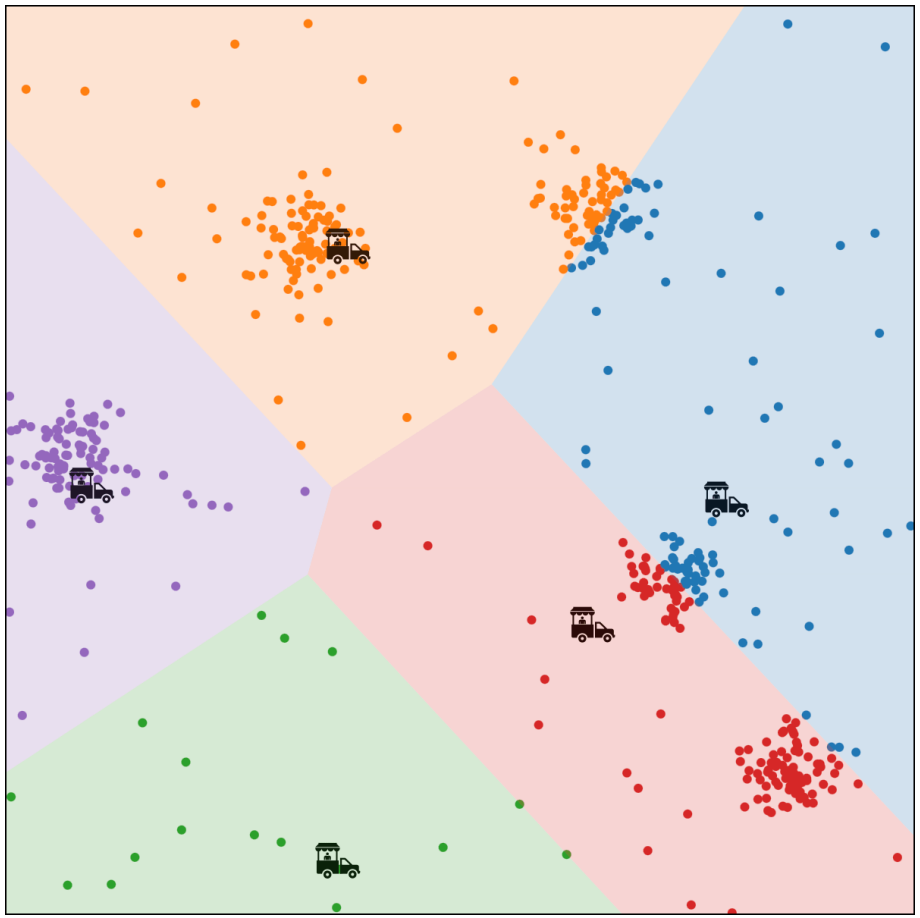
    break

return μ, y



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

$\mu, y =$ random initialization
for $t = 1$ to τ
    $y_{\text{old}} = y$
    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 μ, y



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

1 $\mu, y =$ random initialization

2 **for** $t = 1$ to τ

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

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

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

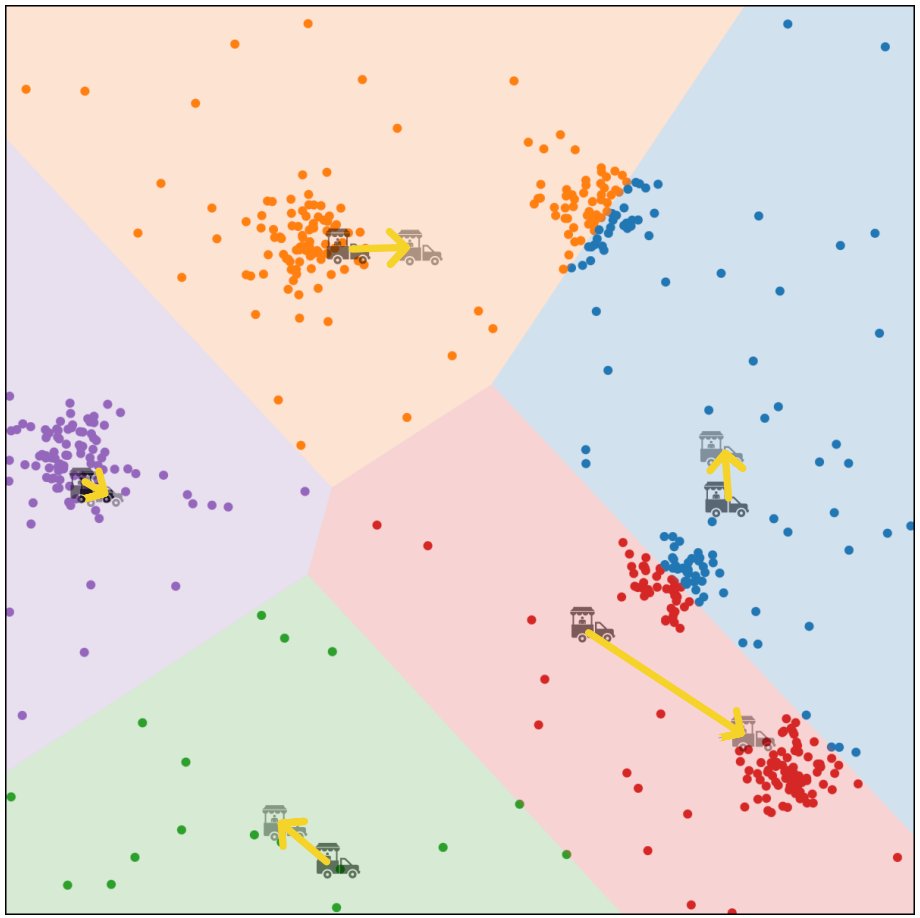
6 $\quad$ **for** $j = 1$ to k

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 μ, y



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

1 $\mu, y =$ random initialization

2 **for** $t = 1$ to τ

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

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

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

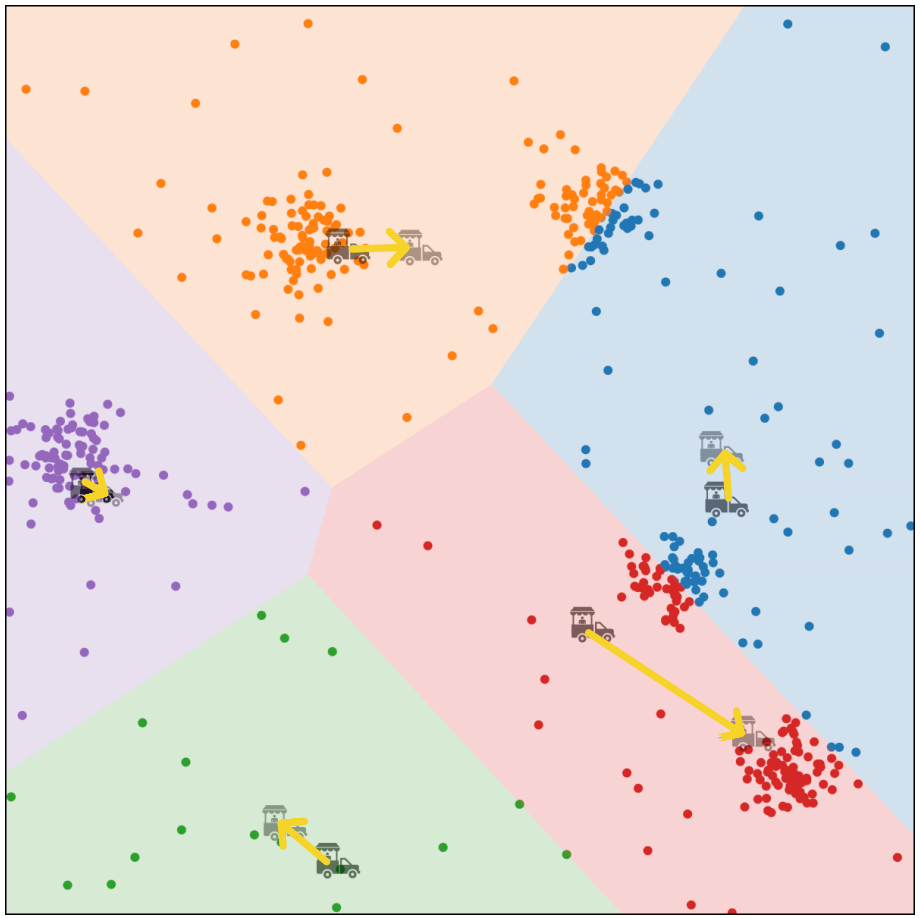
6 $\quad$ **for** $j = 1$ to k

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

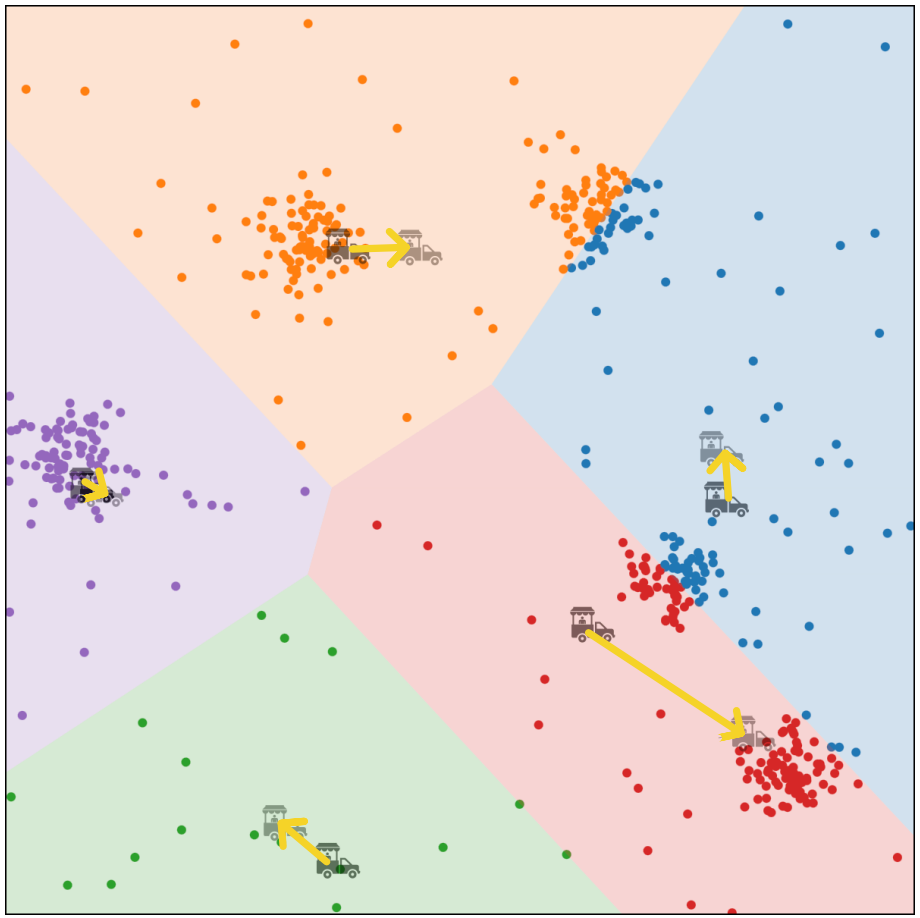
9 $\quad\quad$ break

10 return μ, y



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

$\mu, y =$ random initialization
for $t = 1$ to τ
    $y_{\text{old}} = y$
    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 μ, y



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

$\mu, y =$ random initialization

for $t = 1$ to τ

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

  for $i = 1$ to n

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

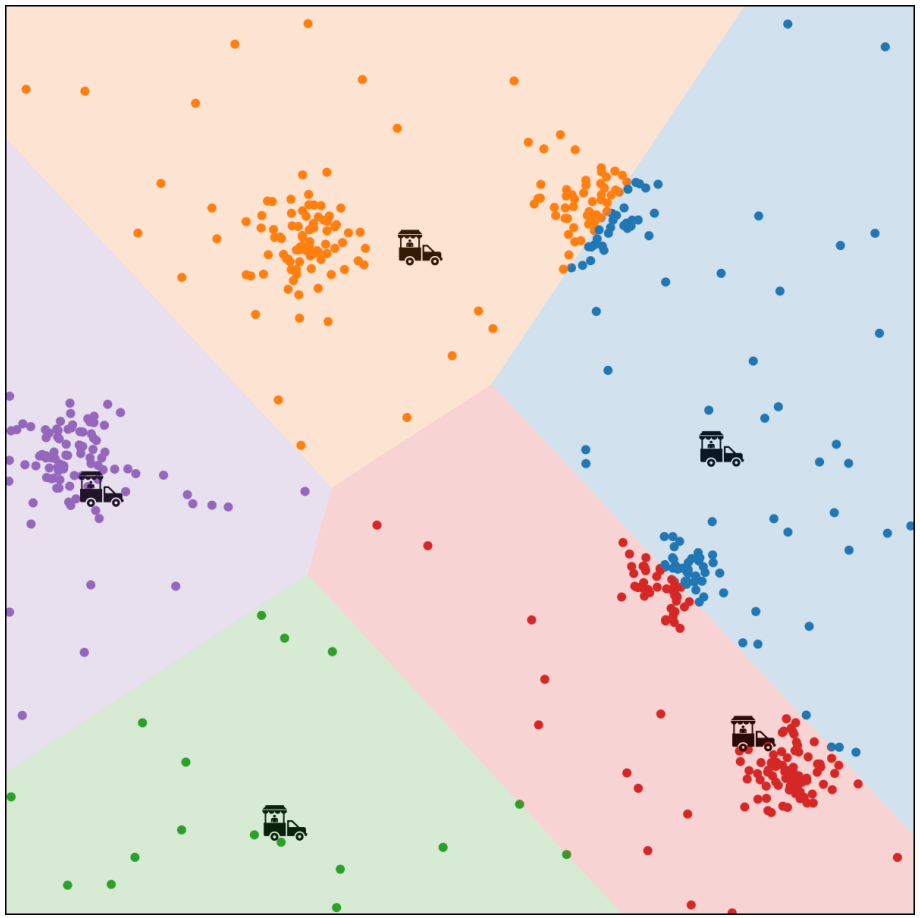
  for $j = 1$ to k

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

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

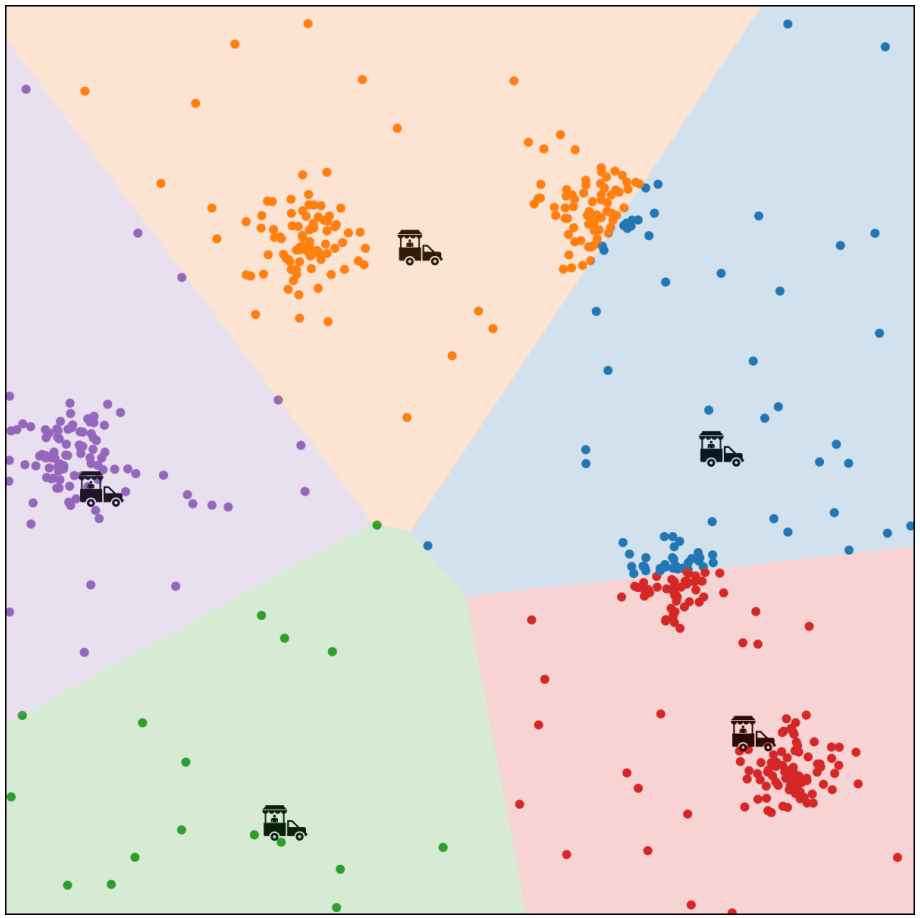
    break

return μ, y



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

$\mu, y =$ random initialization
for $t = 1$ to τ
    $y_{\text{old}} = y$
    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 μ, y



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

1 $\mu, y =$ random initialization

2 **for** $t = 1$ to τ

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

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

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

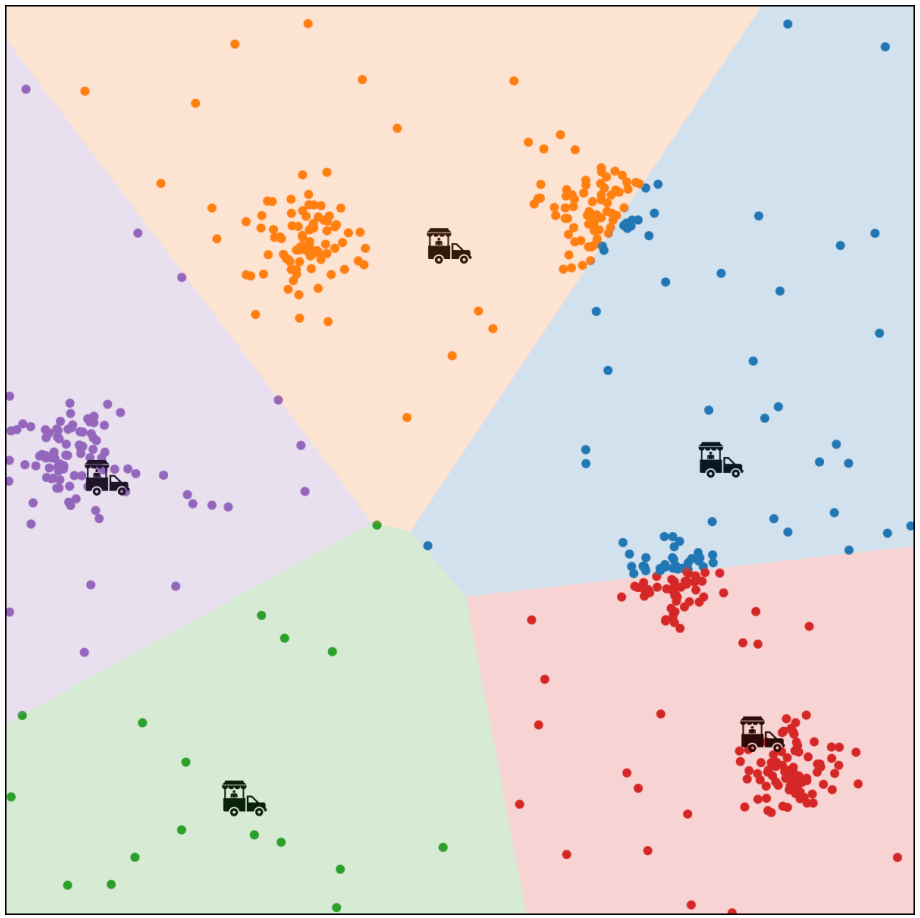
6 $\quad$ **for** $j = 1$ to k

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 μ, y



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

1 $\mu, y =$ random initialization

2 **for** $t = 1$ to τ

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

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

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

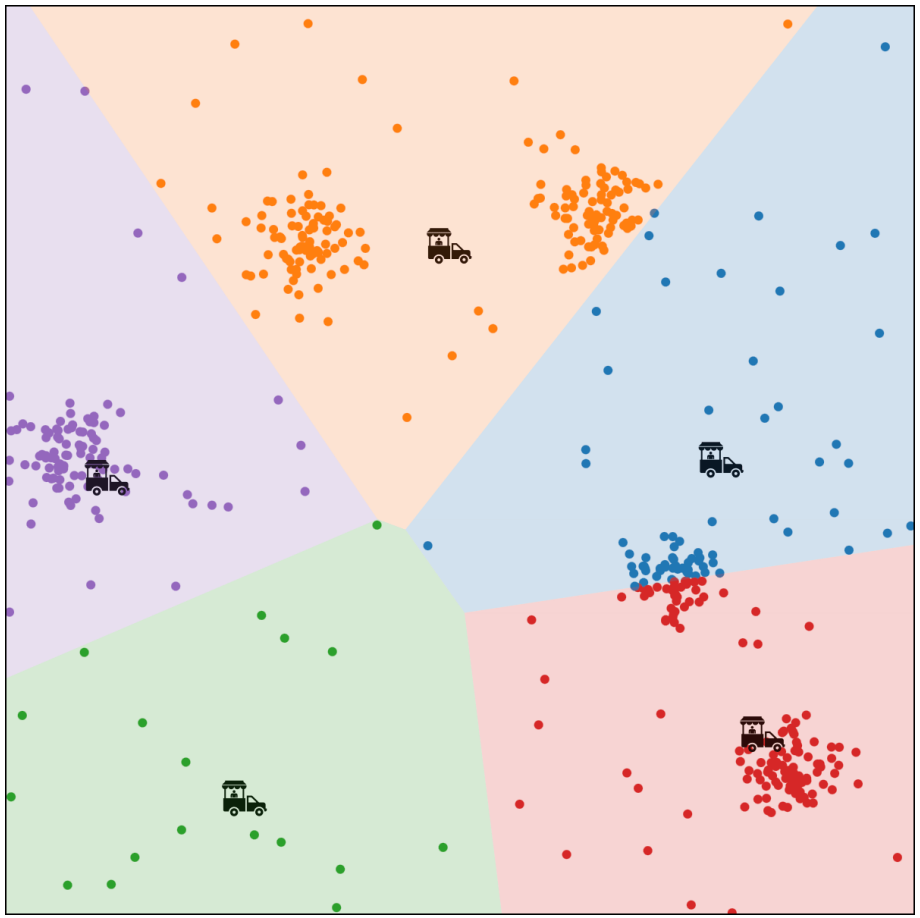
6 $\quad$ **for** $j = 1$ to k

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 μ, y



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

1 $\mu, y =$ random initialization

2 **for** $t = 1$ to τ

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

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

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

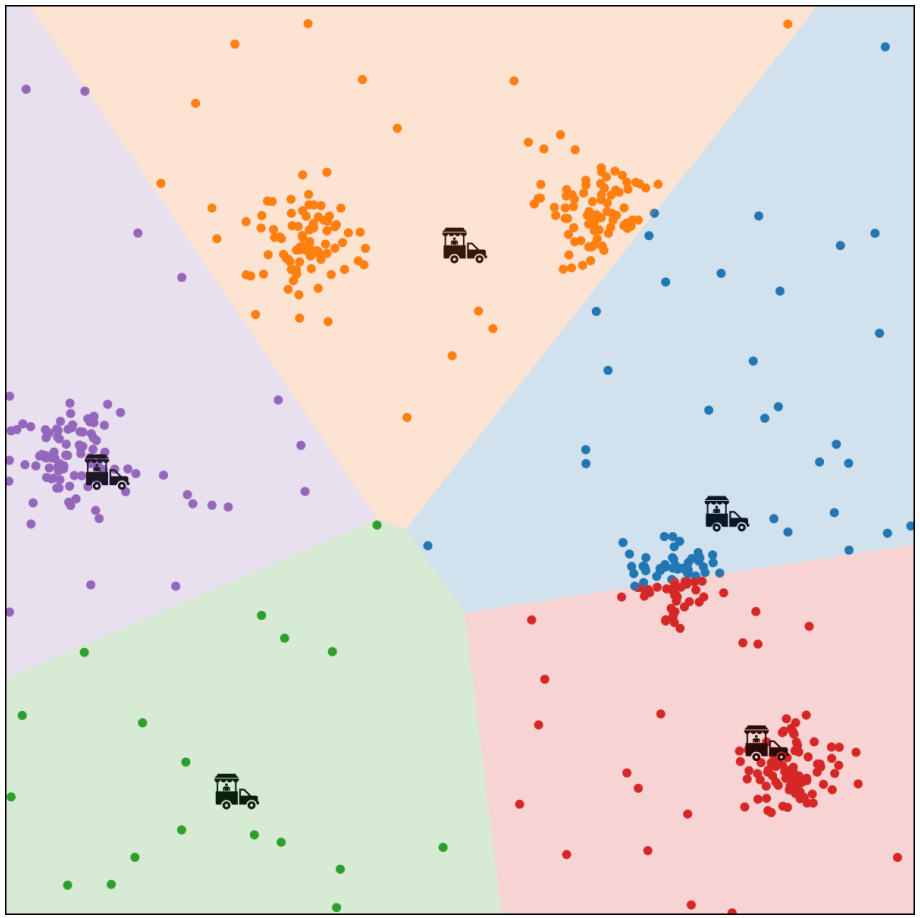
6 $\quad$ **for** $j = 1$ to k

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 μ, y



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

1 $\mu, y =$ random initialization

2 **for** $t = 1$ to τ

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

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

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

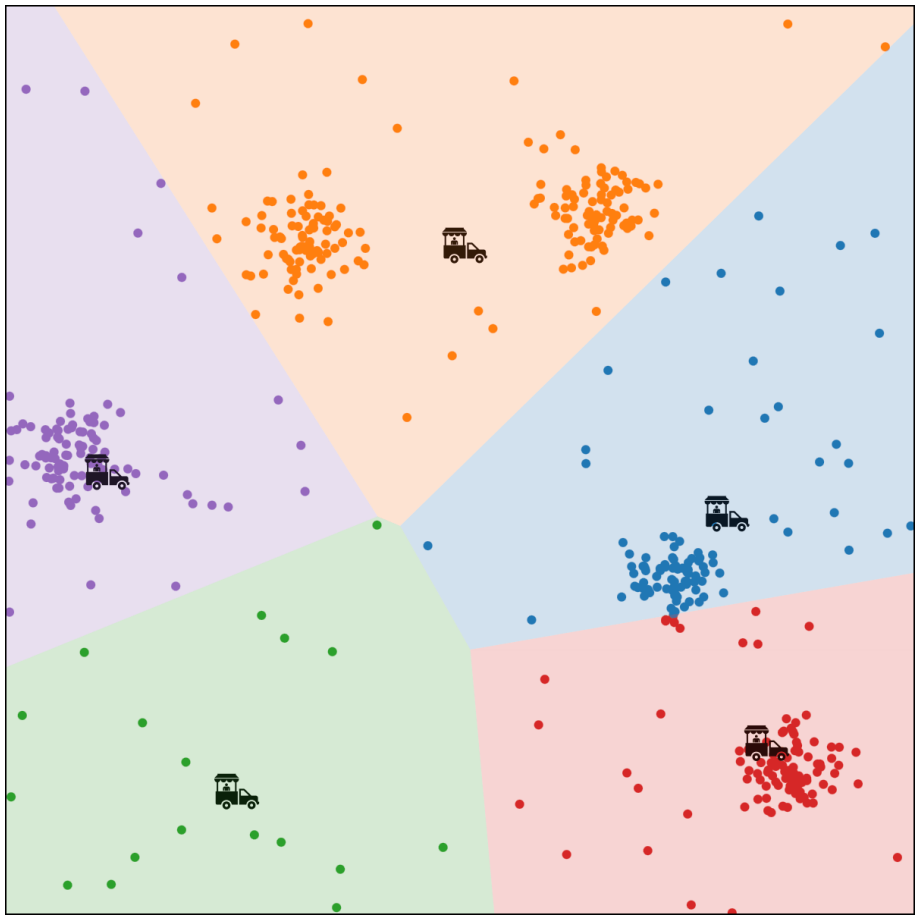
6 $\quad$ **for** $j = 1$ to k

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 μ, y



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

1 $\mu, y =$ random initialization

2 **for** $t = 1$ to τ

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

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

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

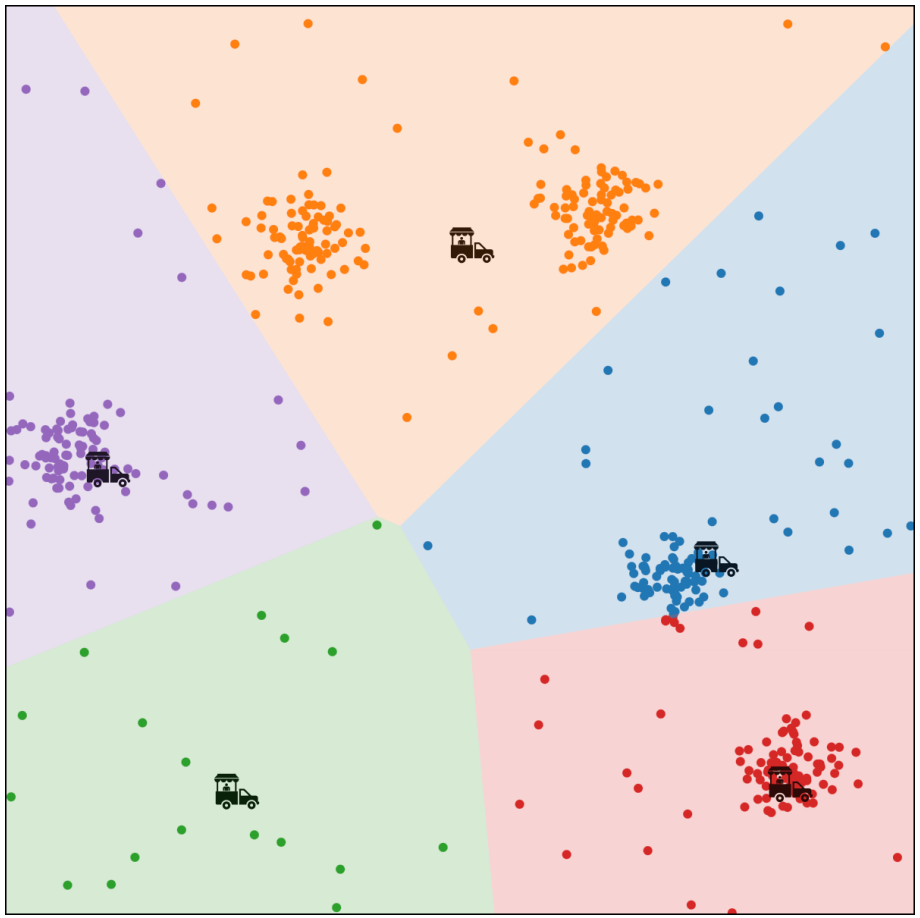
6 $\quad$ **for** $j = 1$ to k

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 μ, y



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

1 $\mu, y =$ random initialization

2 **for** $t = 1$ to τ

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

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

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

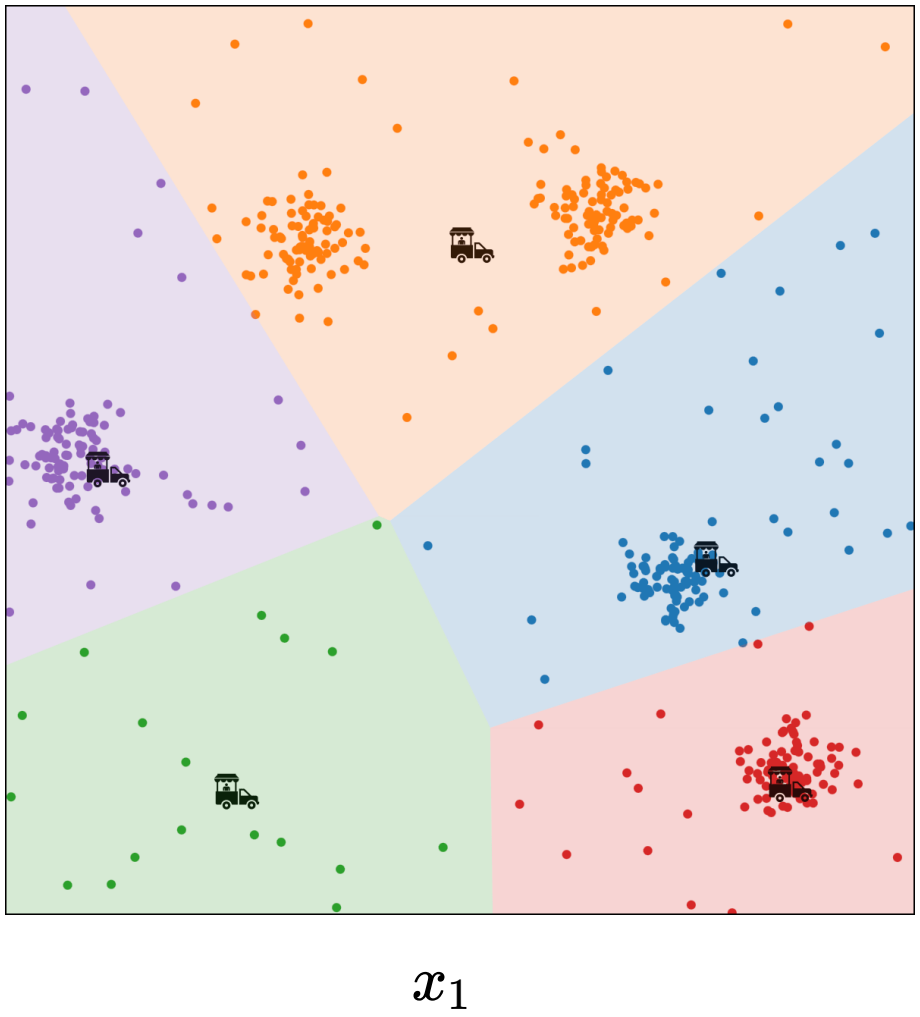
6 $\quad$ **for** $j = 1$ to k

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 μ, y



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

1 $\mu, y =$ random initialization

2 **for** $t = 1$ to τ

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

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

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

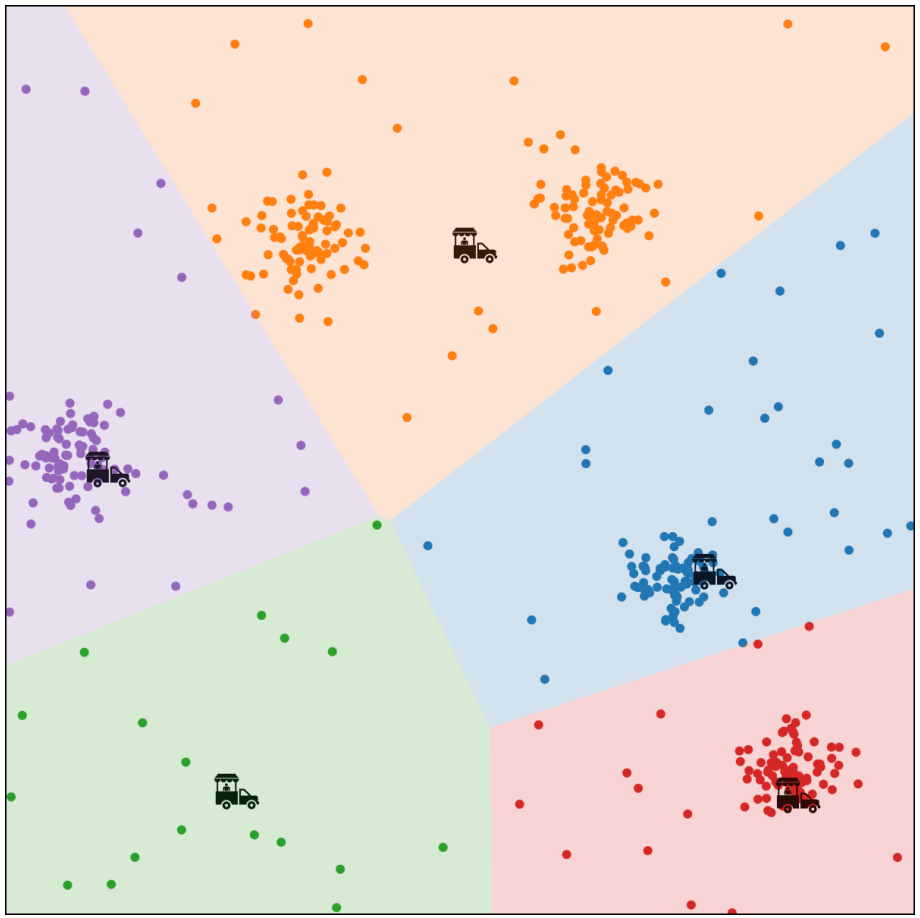
6 $\quad$ **for** $j = 1$ to k

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

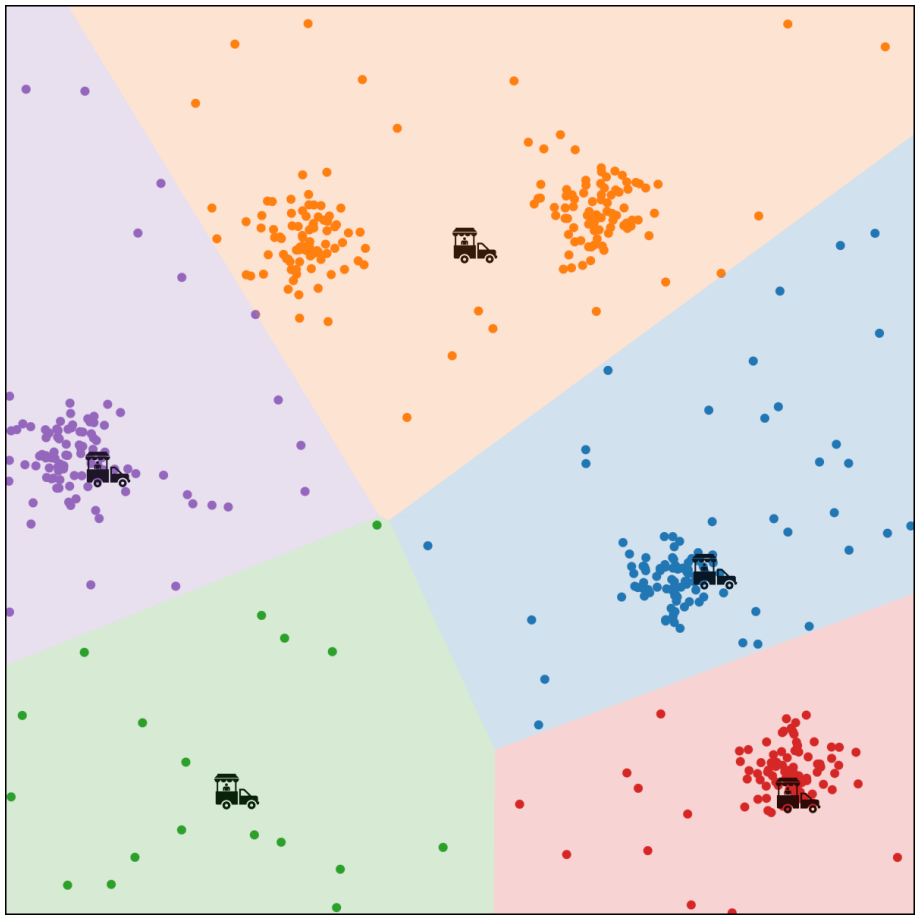
9 $\quad\quad$ break

10 return μ, y



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

1 $\mu, y =$ random initialization
2 **for** $t = 1$ to τ
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 μ, y



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

1 $\mu, y =$ random initialization

2 **for** $t = 1$ to τ

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

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

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

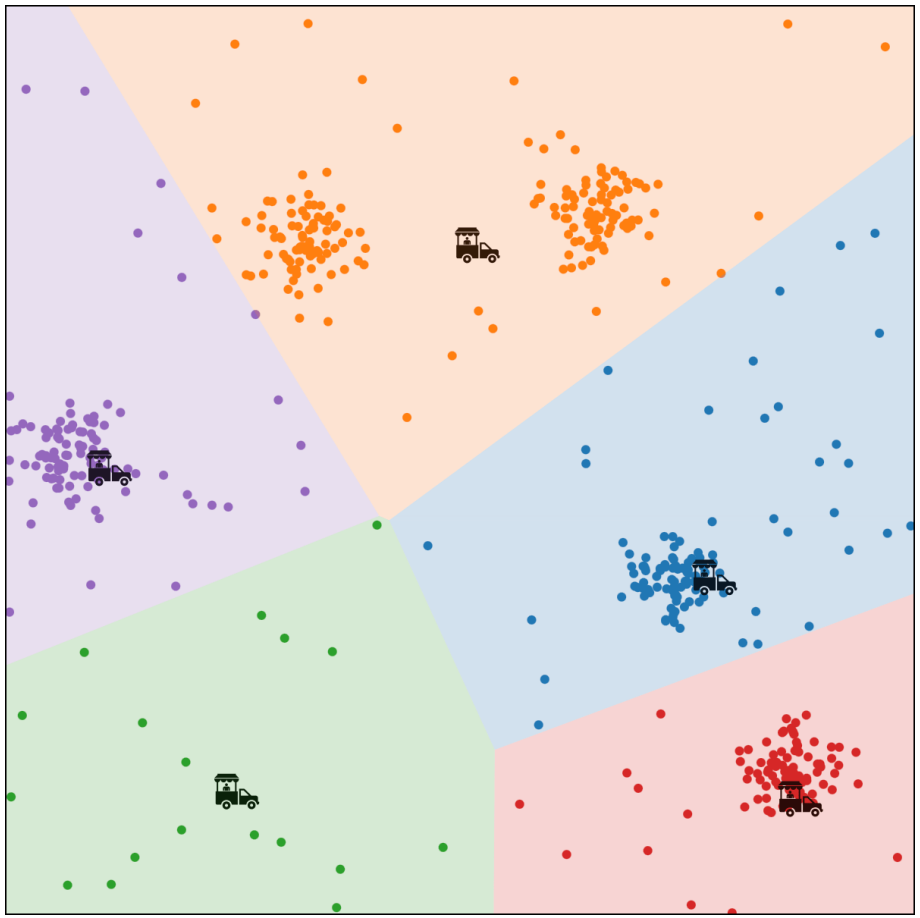
6 $\quad$ **for** $j = 1$ to k

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

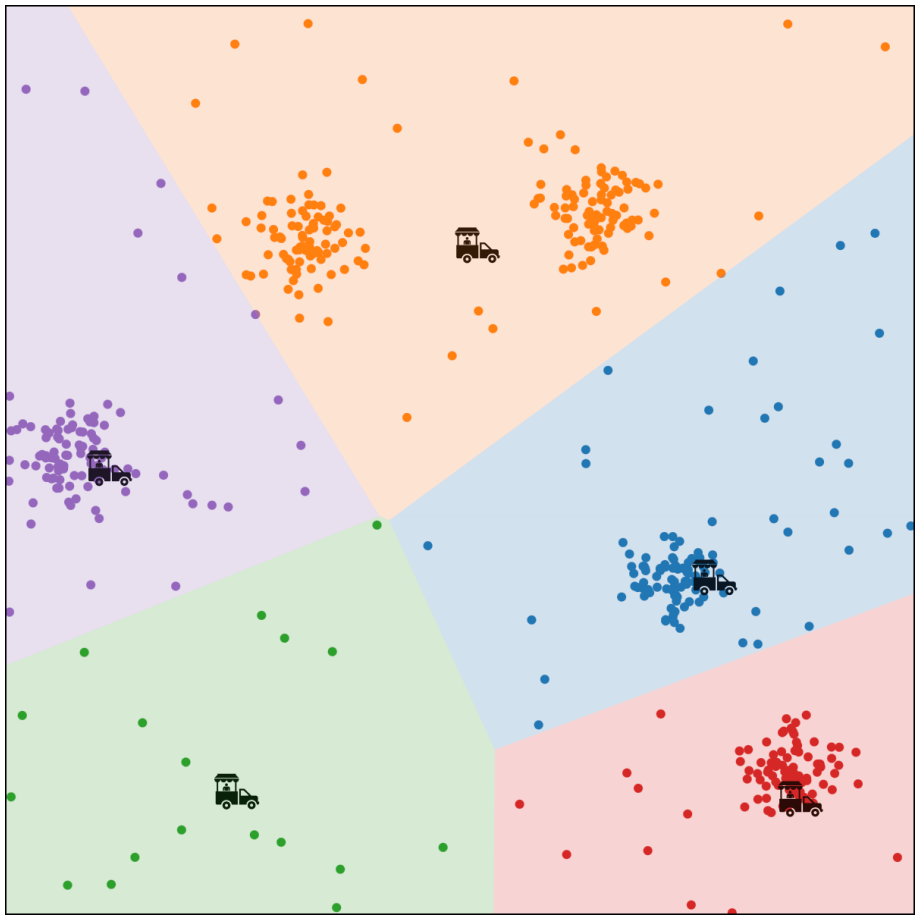
9 $\quad\quad$ break

10 return μ, y



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

1 $\mu, y =$ random initialization
2 **for** $t = 1$ to τ
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 μ, y



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

1 $\mu, y =$ random initialization

2 **for** $t = 1$ to τ

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

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

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

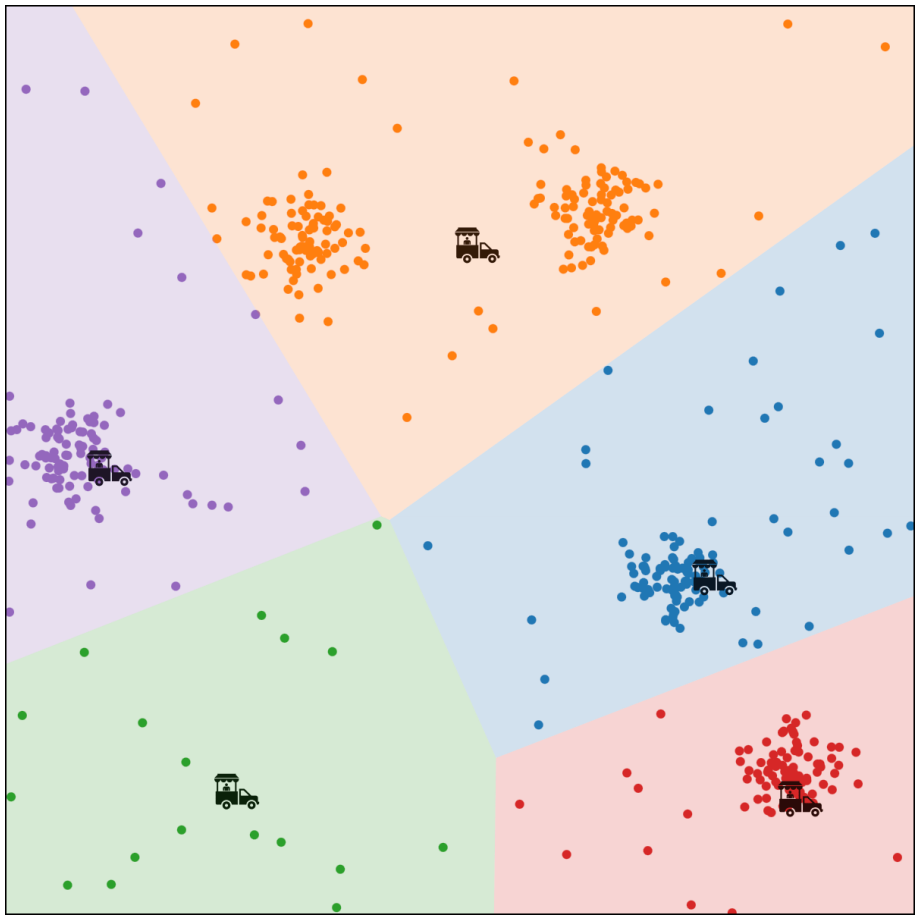
6 $\quad$ **for** $j = 1$ to k

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 μ, y



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

1 $\mu, y =$ random initialization

2 **for** $t = 1$ to τ

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

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

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

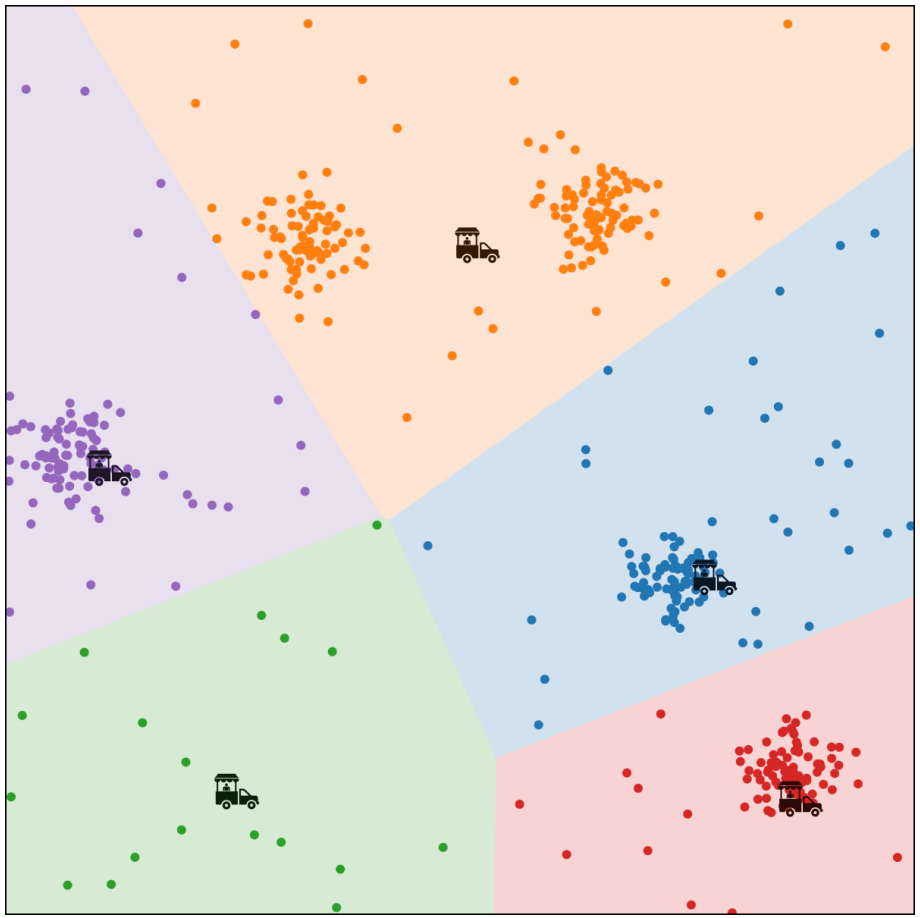
6 $\quad$ **for** $j = 1$ to k

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 μ, y



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

1 $\mu, y =$ random initialization

2 **for** $t = 1$ to τ

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

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

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

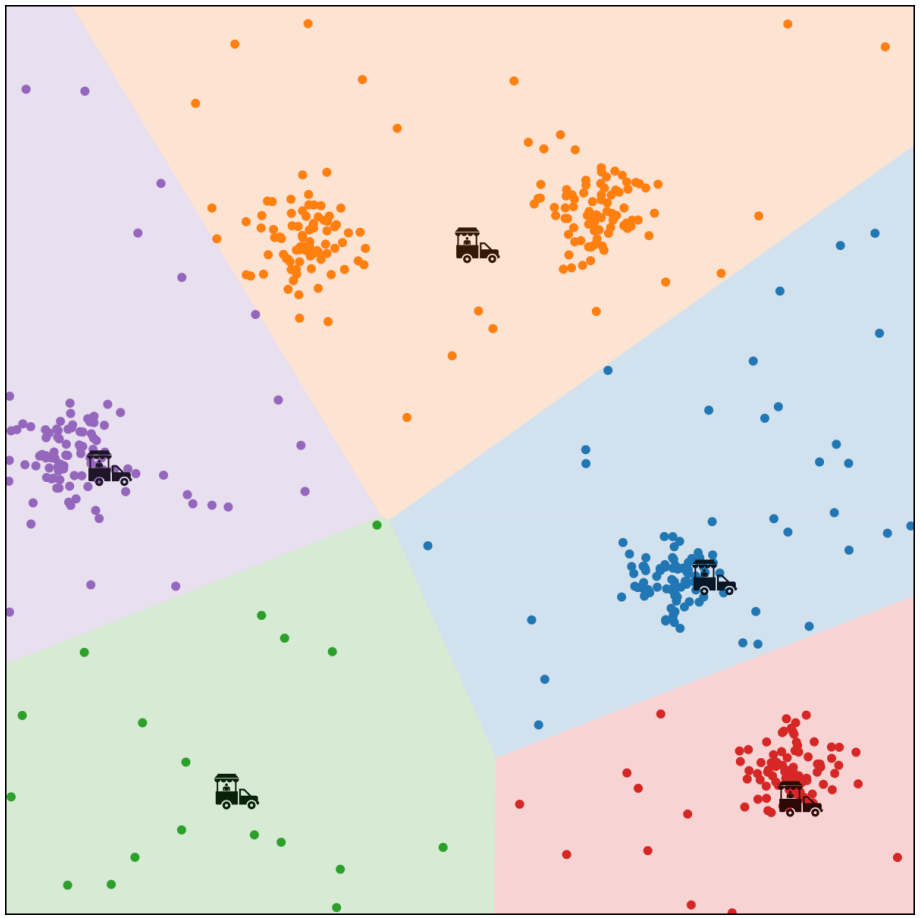
6 $\quad$ **for** $j = 1$ to k

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 μ, y



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

1 $\mu, y =$ random initialization

2 **for** $t = 1$ to τ

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

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

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

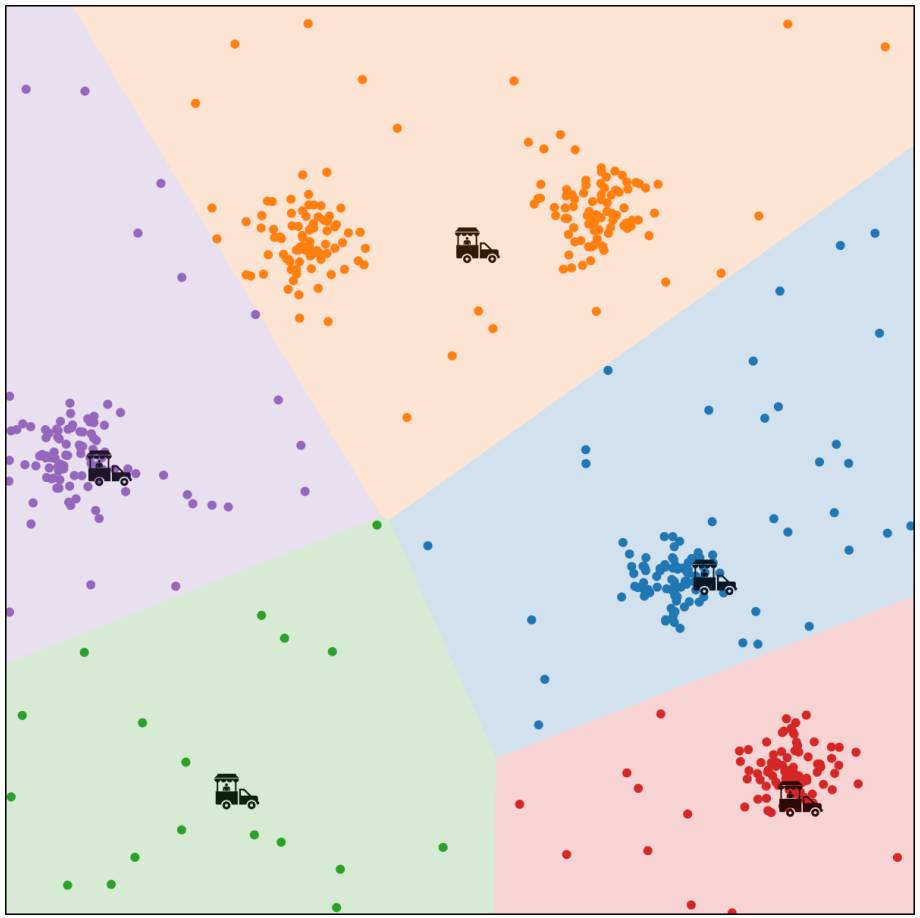
6 $\quad$ **for** $j = 1$ to k

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 μ, y



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

$\mu, y =$ random initialization

for $t = 1$ to τ

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

  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 μ, y

Summary

- One really important class of ML models is called “non-parametric”.
- Decision trees are kind of like creating a flow chart. These hypotheses are the most human-understandable of any we have worked with.We regularize by first growing trees that are very big and then “pruning” them.
- Ensembles: sometimes it’s useful to come up with a lot of simple hypotheses and then let them “vote” to make a prediction for a new example.
- Nearest neighbor remembers all the training data for prediction. Depends crucially on our notion of “closest” (standardize data is important). Can do fancier things (weighted kNN). Less good in high dimensions (computationally expensive).

Summary

- Clustering is an important kind of unsupervised learning in which we try to divide the x's into a finite set of groups that are in some sense similar.
- A widely used clustering objective is the k-means. It also requires a distance metric on x's.
- There's a convenient special-purpose method for finding a local optimum: the k-means algorithm.
- The solution obtained by k-means algorithm is sensitive to initialization.
- The solution obtained by k-means algorithm is sensitive to the number of clusters chosen.

https://forms.gle/YLZKTx8T4h42Pf4E7

We'd love to hear
your thoughts.

Thanks!