Machine Learning for Large-Scale Data Analysis and Decision Making (MATH80629A)
Fall 2021

Week #4 - **Summary**

Announcement

- **Hybrid classroom**: Mondays 8:30 am - 11:30 am
  Class room: Manuvie. This classroom is located on the 1st floor of Côte-Sainte-Catherine building.
  Zoom: Zoom link.

- **Hybrid office hour**: Mondays 11:30 am - 1 pm
  Office:4.834
  Zoom: Zoom link.

- **Lab session** on week #5 (September 27)
  Lab room: Laboratoire Lachute

Today

- **Second Quiz** on Gradescope!
- Summary of Machine learning fundamental
- Q&A
- Hands-on session

Quiz 1

Login to your Gradescope account

Models for supervised learning

- (Mostly) linear models
- Focus on **classification**

1. **Non-Probabilistic Models**
   - Nearest Neighbor, Support Vector Machines (SVMs)
2. **Probabilistic Models**
   - Naive Bayes

Supervised learning

Train Data

$$\underbrace{\begin{array}{c|ccccc|}
 & \text{Nb.bed.} & \text{Area} & \text{Neigh.} & \cdot & \cdot \\
x_0 & 1 & 0 & 0 & 0 & 0 \\
x_1 & 1 & 100 & 1 & .2 & .5 \\
x_2 & 3 & 200 & 0 & .1 & .2 \\
x_3 & 1 & 150 & 1 & .4 & .1 \\
x_4 & 2 & 210 & 2 & .5 & 1.1
\end{array}}_{X}
\quad
\underbrace{\begin{array}{c|c|}
 & \text{Sell-ability} \\
y_0 & 1 \\
y_1 & 2 \\
y_2 & 0 \\
y_3 & 2 \\
y_4 & 1
\end{array}}_{Y}$$

Task

$$f : \mathbb{R}^n \to \{0, 1, 2\}$$

Test Data

$$\underbrace{\begin{array}{c|ccccc|}
 & \text{Nb.bed.} & \text{Area} & \text{Neigh.} & \cdot & \cdot \\
x_0 & 1 & 0 & 0 & 0 & 0 \\
x_1 & 2 & 50 & 1 & .3 & .8 \\
x_2 & 1 & 100 & 1 & .5 & 1.4 \\
x_3 & 4 & 170 & 0 & .7 & .4 \\
x_4 & 1 & 120 & 3 & .9 & .5
\end{array}}_{X^{new}}
\quad
\underbrace{\begin{array}{c|c|}
 & \text{Sell-ability} \\
y_0 & ? \\
y_1 & ? \\
y_2 & ? \\
y_3 & ? \\
y_4 & ?
\end{array}}_{Y^{new}}$$

Supervised learning

Train Data

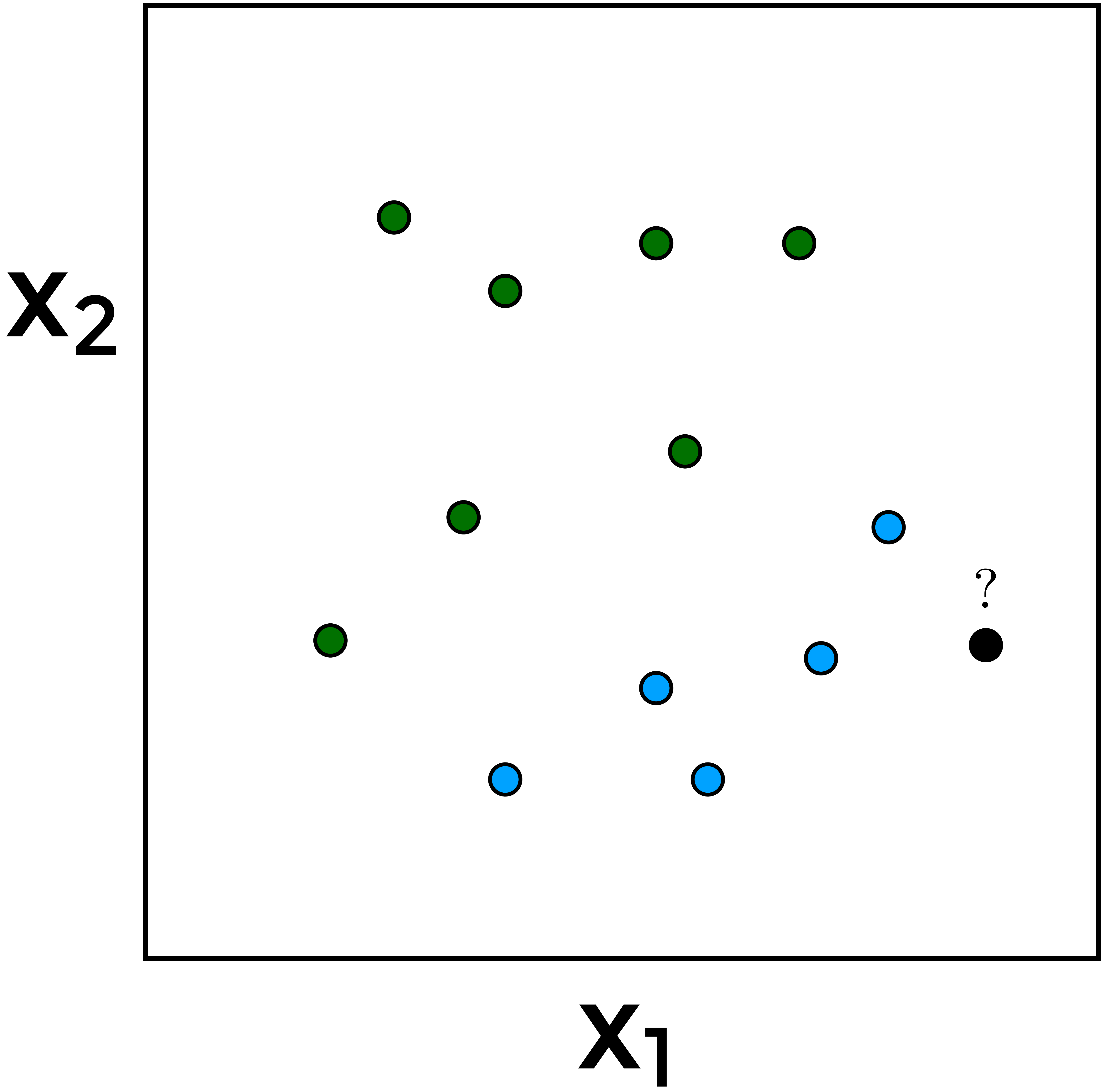
$$\underbrace{\begin{array}{c} \\ x_0 \\ x_1 \\ x_2 \\ x_3 \\ x_4 \end{array}\begin{array}{c} \begin{array}{ccccc} \text{Nb.bed.} & \text{Area} & \text{Neigh.} & \cdot & \cdot \end{array} \\ \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 1 & 100 & 1 & .2 & .5 \\ 3 & 200 & 0 & .1 & .2 \\ 1 & 150 & 1 & .4 & .1 \\ 2 & 210 & 2 & .5 & 1.1 \end{bmatrix}\end{array}}_{X} \quad \underbrace{\begin{array}{c} \\ y_0 \\ y_1 \\ y_2 \\ y_3 \\ y_4 \end{array}\begin{array}{c} \text{Sell-ability} \\ \begin{bmatrix} 1 \\ 2 \\ 0 \\ 2 \\ 1 \end{bmatrix}\end{array}}_{Y}$$

Task

Models $\boxed{f}\ \mathbb{R}^n \rightarrow \{0, 1, 2\}$

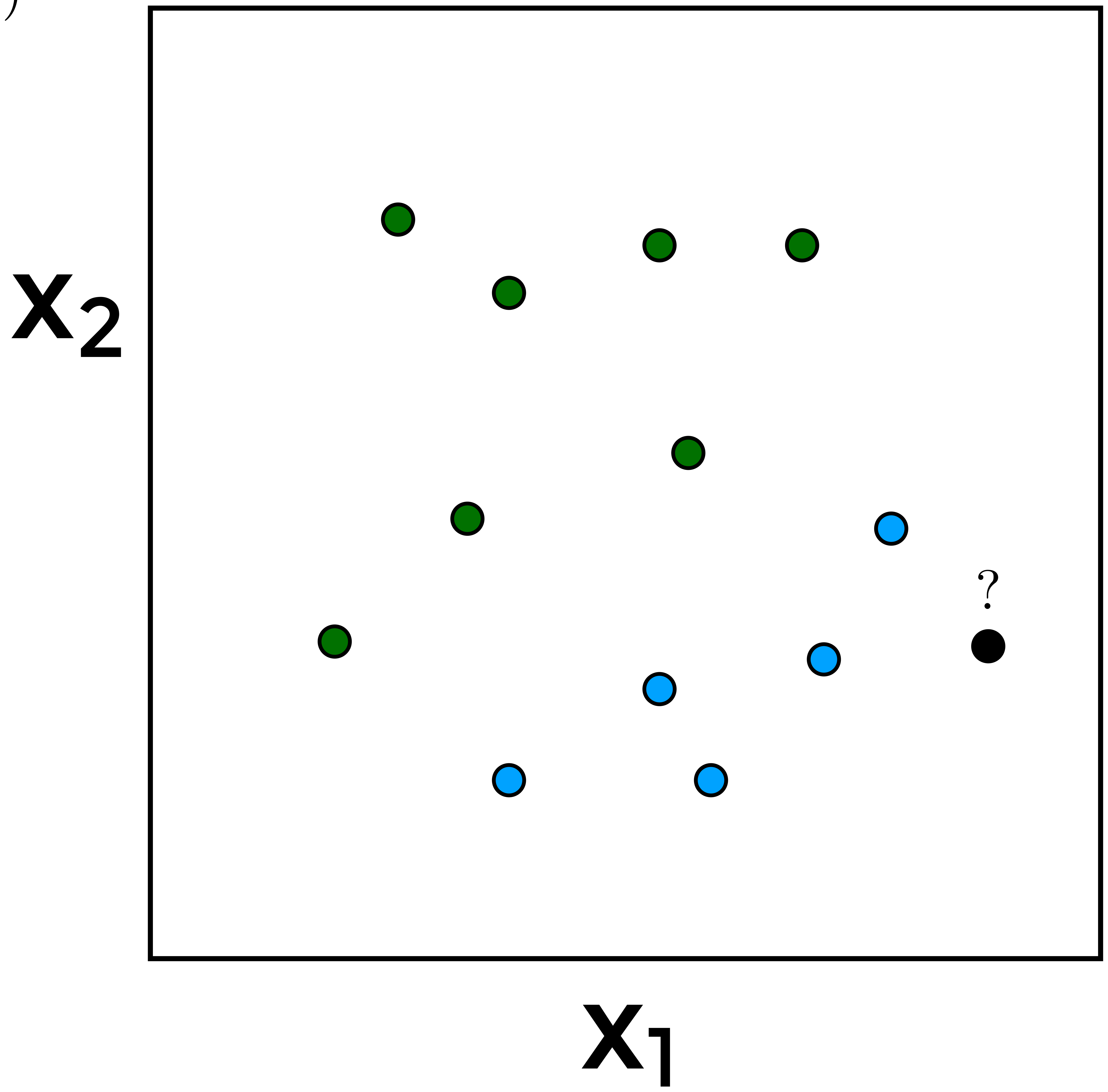
Test Data

$$\underbrace{\begin{array}{c} \\ x_0 \\ x_1 \\ x_2 \\ x_3 \\ x_4 \end{array}\begin{array}{c} \begin{array}{ccccc} \text{Nb.bed.} & \text{Area} & \text{Neigh.} & \cdot & \cdot \end{array} \\ \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 2 & 50 & 1 & .3 & .8 \\ 1 & 100 & 1 & .5 & 1.4 \\ 4 & 170 & 0 & .7 & .4 \\ 1 & 120 & 3 & .9 & .5 \end{bmatrix}\end{array}}_{X^{new}} \quad \underbrace{\begin{array}{c} \\ y_0 \\ y_1 \\ y_2 \\ y_3 \\ y_4 \end{array}\begin{array}{c} \text{Sell-ability} \\ \begin{bmatrix} ? \\ ? \\ ? \\ ? \\ ? \end{bmatrix}\end{array}}_{Y^{new}}$$



$$i' = \arg\min_i \mathsf{dist}(x_i, x_j)$$

$$y_j = y_{i'}$$

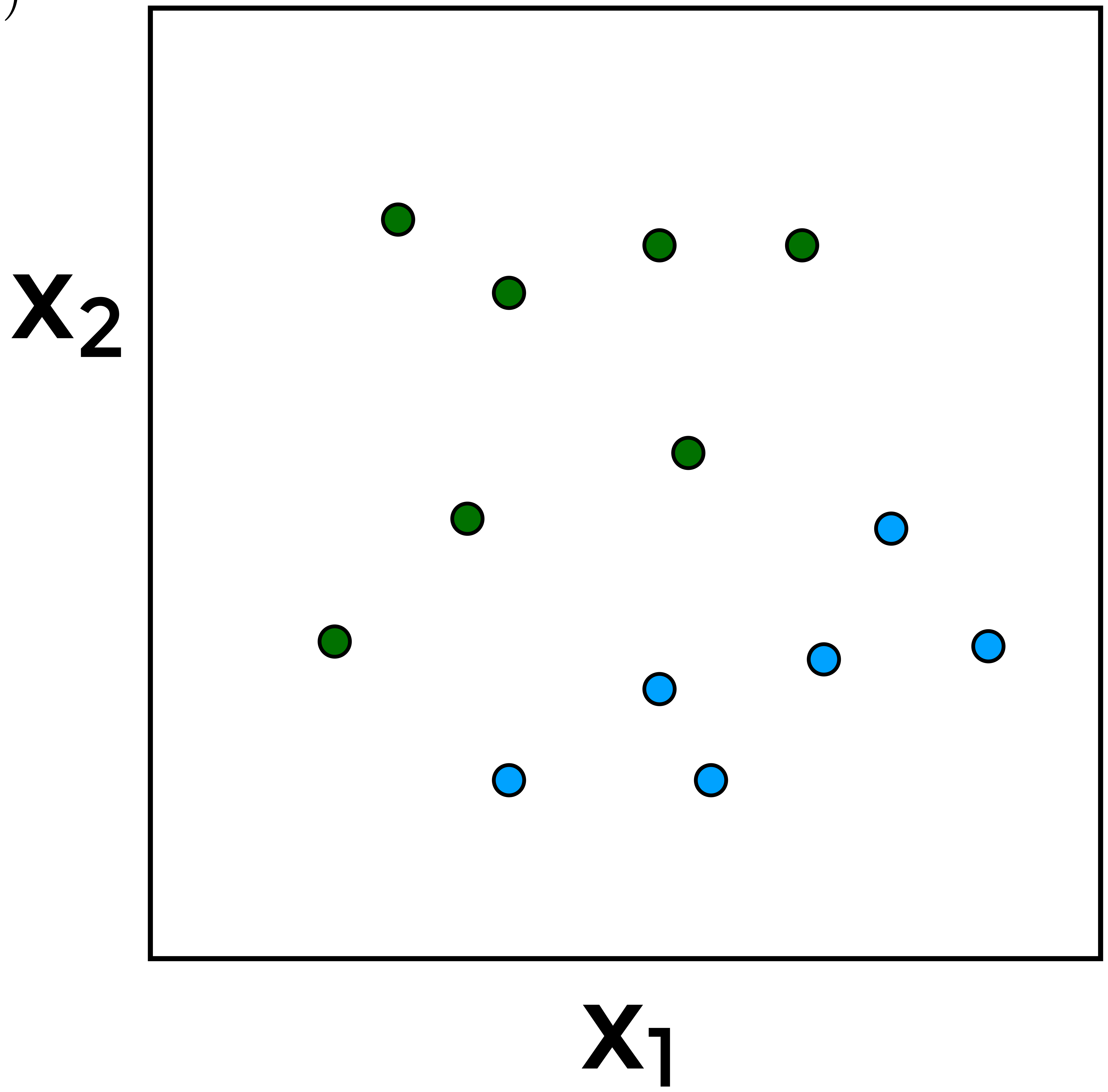


- **1-NN**

Instance classified according to its nearest neighbor

$$i' = \arg\min_i \mathsf{dist}(x_i, x_j)$$

$$y_j = y_{i'}$$

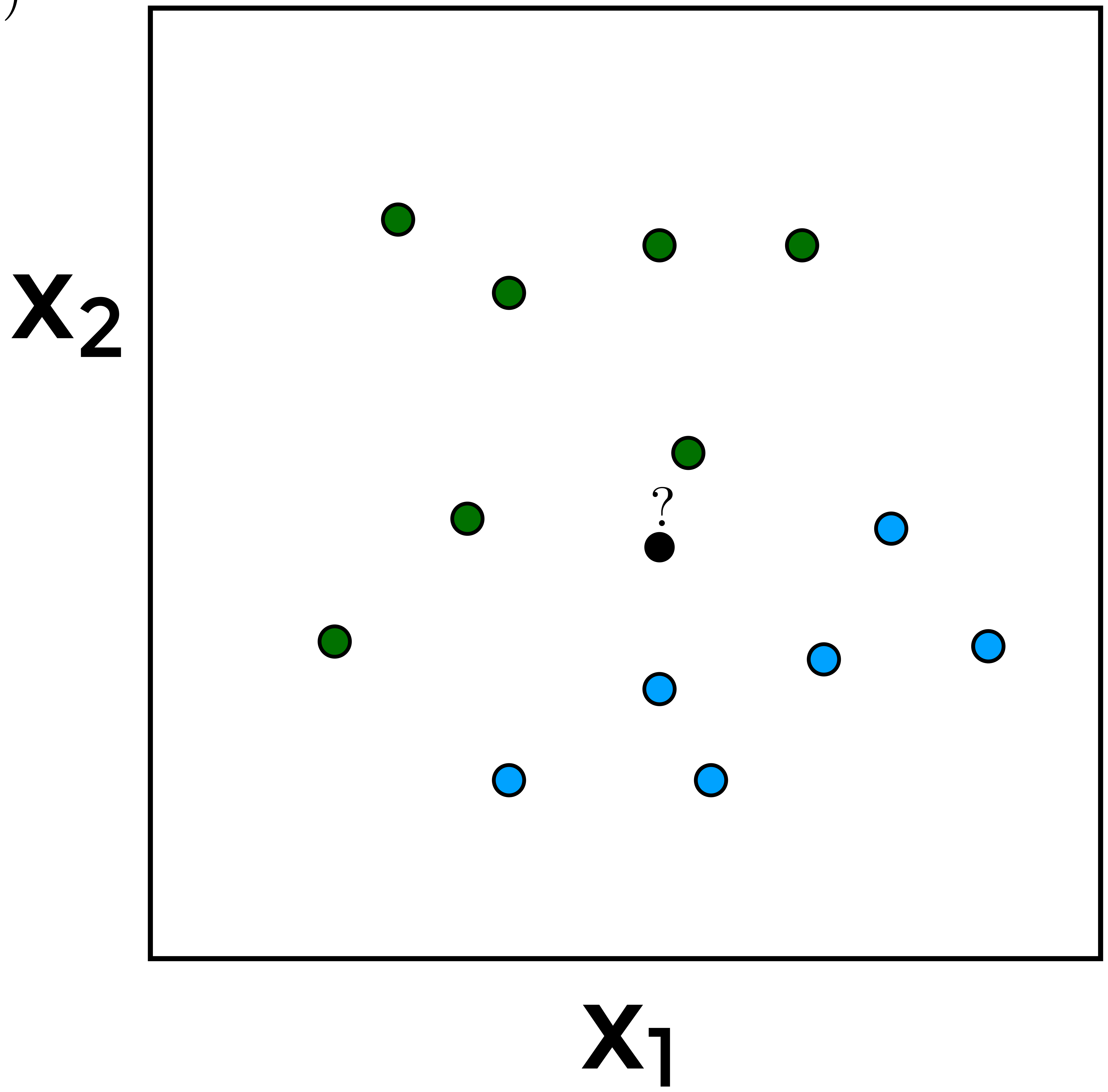


- **1-NN**

Instance classified according to its nearest neighbor

$$i' = \arg\min_i \mathrm{dist}(x_i, x_j)$$

$$y_j = y_{i'}$$



- **1-NN**

Instance classified according to its nearest neighbor

- **K-NN**

Instance classified according to the majority of its K nearest neighbors

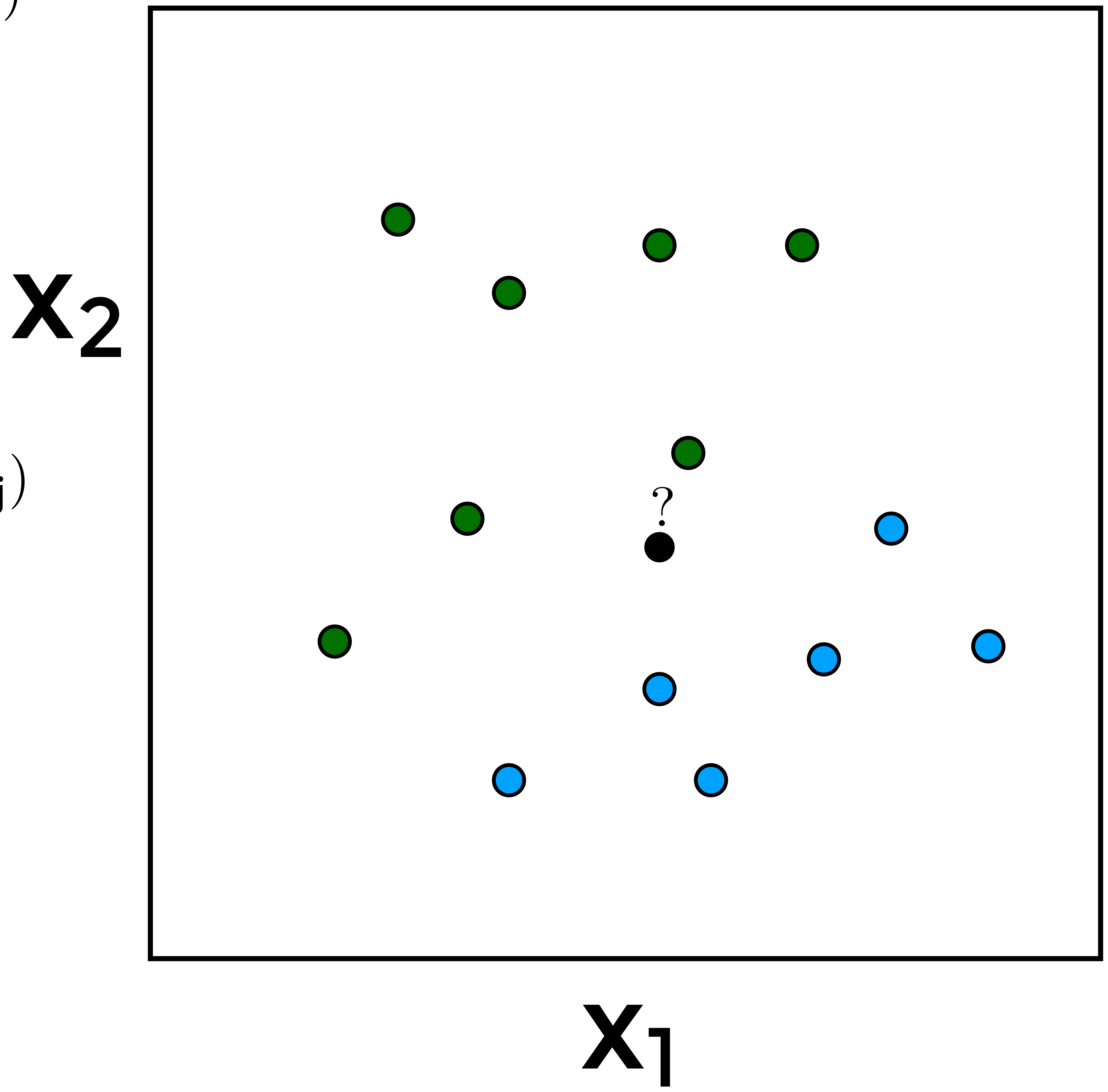
$$i' = \arg\min_i \text{dist}(x_i, x_j)$$

$$y_j = y_{i'}$$

$$k = 5 \text{ (assumption)}$$

$$i = \arg\text{sort}_i \, \text{dist}(x_i, x_j)$$

$$y_j = \text{majority}(i_{:5})$$



- **1-NN**

Instance classified according to its nearest neighbor

- **K-NN**

Instance classified according to the majority of its K nearest neighbors

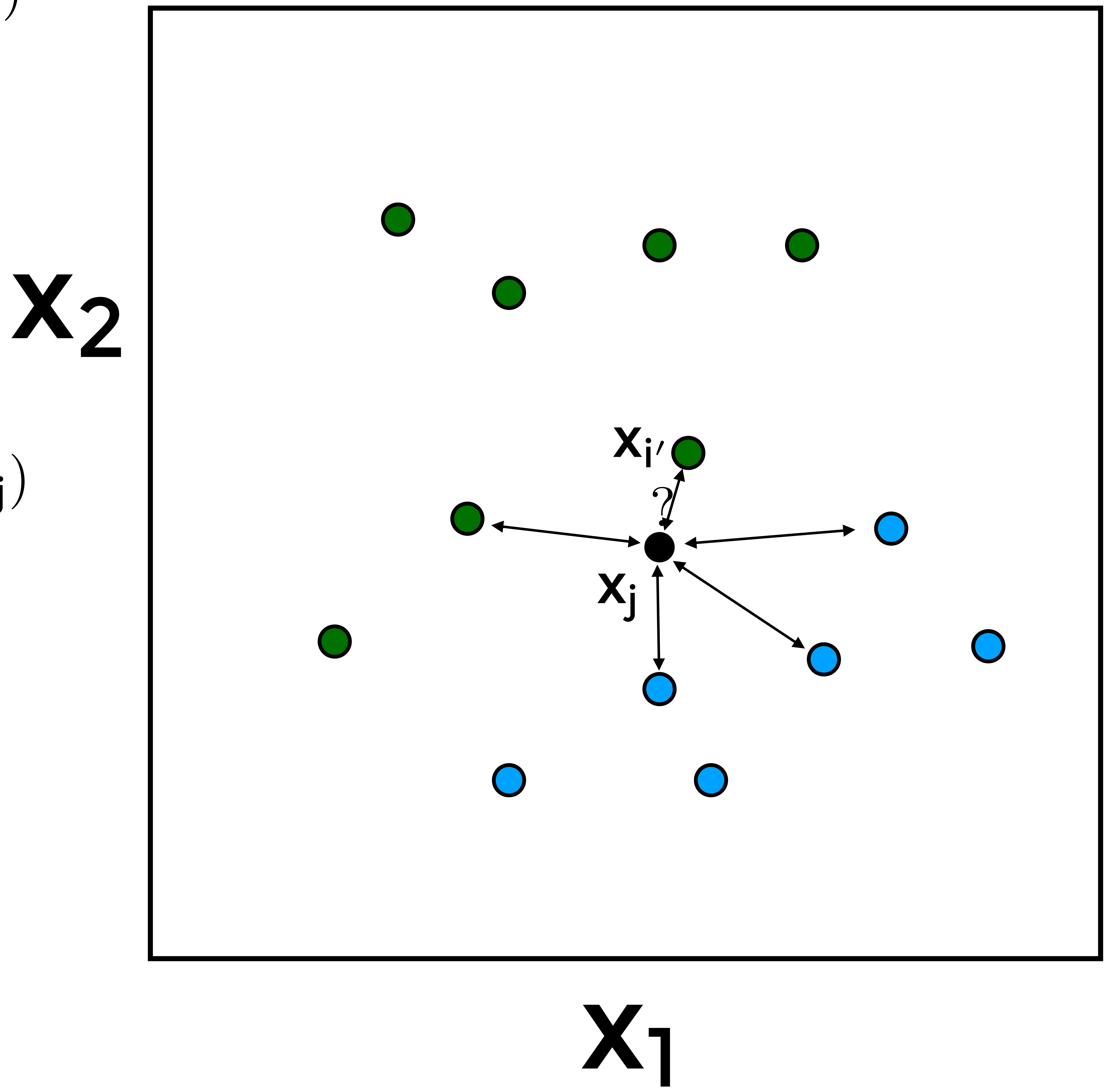
$$i' = \arg\min_i \text{dist}(x_i, x_j)$$

$$y_j = y_{i'}$$

$$k = 5 \text{ (assumption)}$$

$$i = \arg\text{sort}_i \, \text{dist}(x_i, x_j)$$

$$y_j = \text{majority}(i_{:5})$$



- **1-NN**

Instance classified according to its nearest neighbor

- **K-NN**

Instance classified according to the majority of its K nearest neighbors

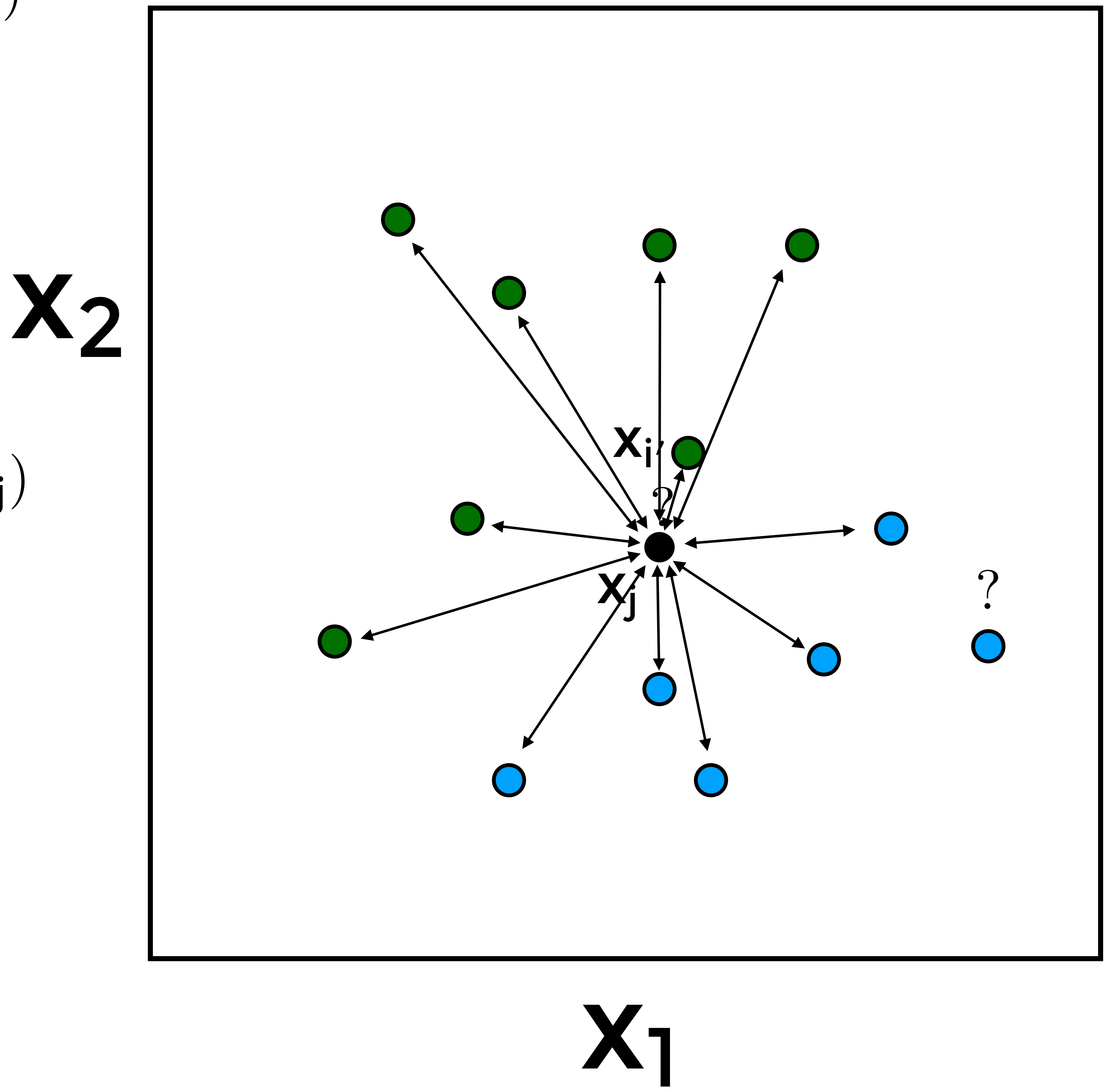
$$i' = \arg\min_i \mathsf{dist}(x_i, x_j)$$

$$y_j = y_{i'}$$

$$k = 5 \text{ (assumption)}$$

$$i = \arg\mathrm{sort}_i \, \mathsf{dist}(x_i, x_j)$$

$$y_j = \mathsf{majority}(i_{:5})$$



- **1-NN**

Instance classified according to its nearest neighbor

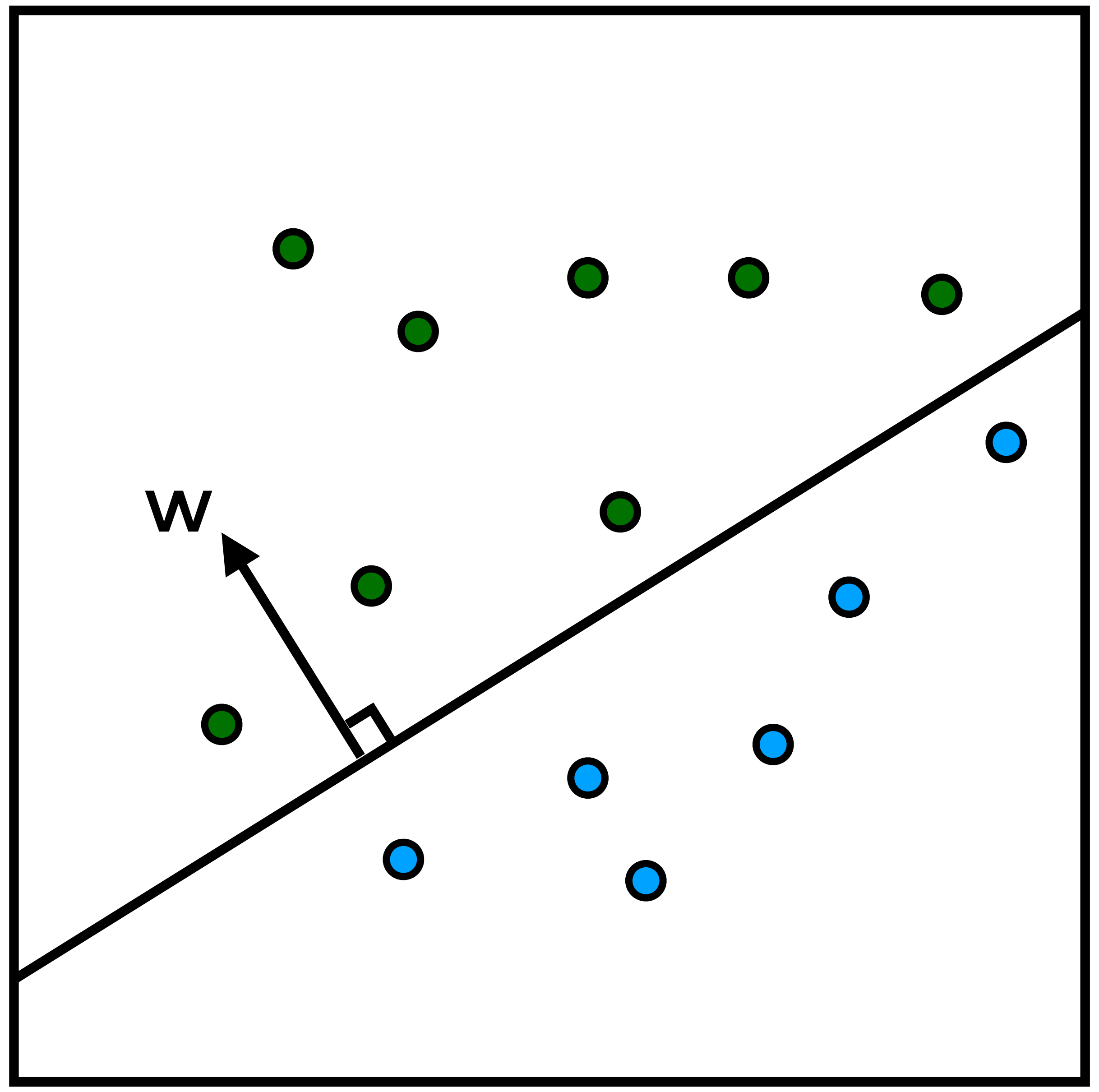
- **K-NN**

Instance classified according to the majority of its K nearest neighbors

- **weighted-NN**

Instance classified according to all neighbors. The contribution of each neighbor is weighted by its distance.

Linear Classification



decision boundary: $y(\mathbf{x}) = 0$

take two points on the boundary: $\mathbf{x}_a, \mathbf{x}_b$

then: $\mathbf{w}^\top \mathbf{x}_a + w_0 = \mathbf{w}^\top \mathbf{x}_b + w_0$

$\implies \mathbf{w}^\top (\mathbf{x}_a - \mathbf{x}_b) = 0$

$\implies$ w is perpendicular to the decision boundary

w represents the orientation of the decision boundary

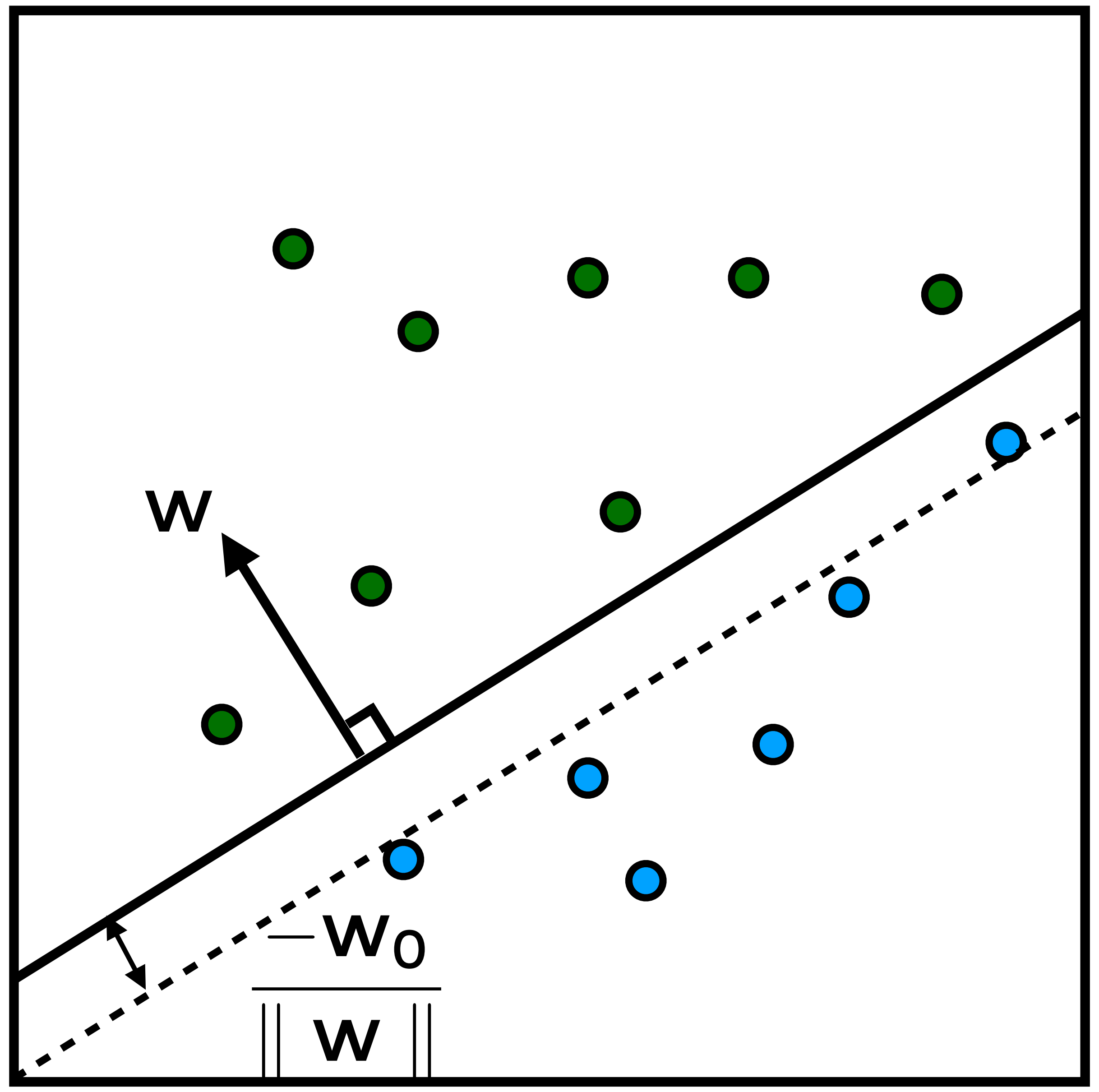
$$y(\mathbf{x}) = \mathbf{w}^\top \mathbf{x} + w_0$$

Decision

$$(\mathbf{w}^\top \mathbf{x} + w_0) > 0 \implies \text{green}$$

$$(\mathbf{w}^\top \mathbf{x} + w_0) < 0 \implies \text{blue}$$

Linear Classification



$$y(\mathbf{x}) = \mathbf{w}^\top \mathbf{x} + w_0$$

Decision

$$(\mathbf{w}^\top \mathbf{x} + w_0) > 0 \implies \text{green}$$

$$(\mathbf{w}^\top \mathbf{x} + w_0) < 0 \implies \text{blue}$$

w_0 is a scalar

you can think of it like an intercept

take $\mathbf{x}'$ as the closest point on the decision boundary to the origin

$\mathbf{x}' = \beta \mathbf{w}$

$$\implies y(\mathbf{x}') = \mathbf{w}^\top \mathbf{x}' + w_0$$

$$\implies y(\mathbf{x}') = \mathbf{w}^\top (\beta \mathbf{w}) + w_0$$

$$\implies 0 = \beta \| \mathbf{w} \|^2 + w_0$$

$$\implies \beta = \frac{-w_0}{\| \mathbf{w} \|^2}$$

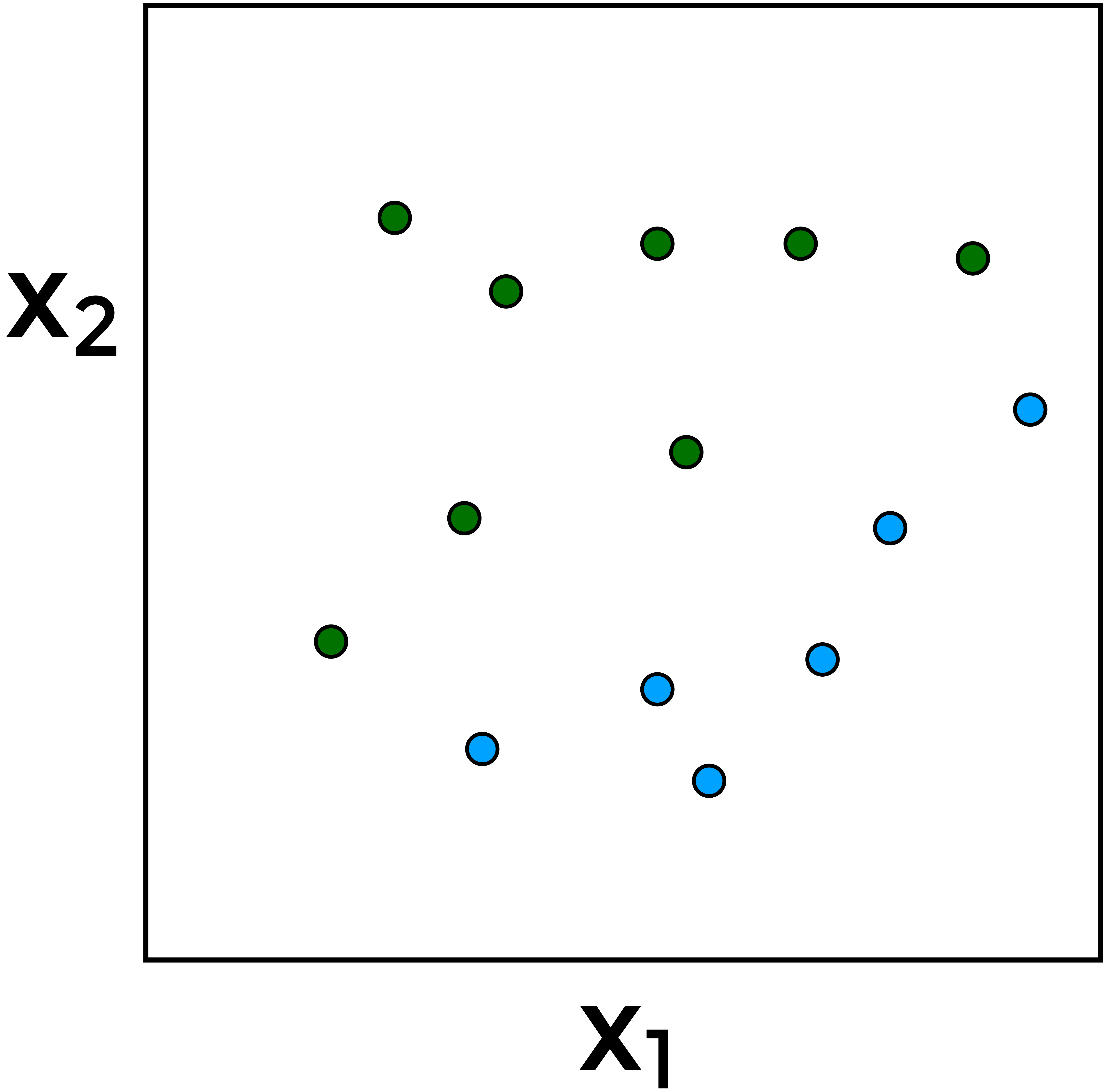
Then you know that the distance from the origin to $\mathbf{x}'$ is:

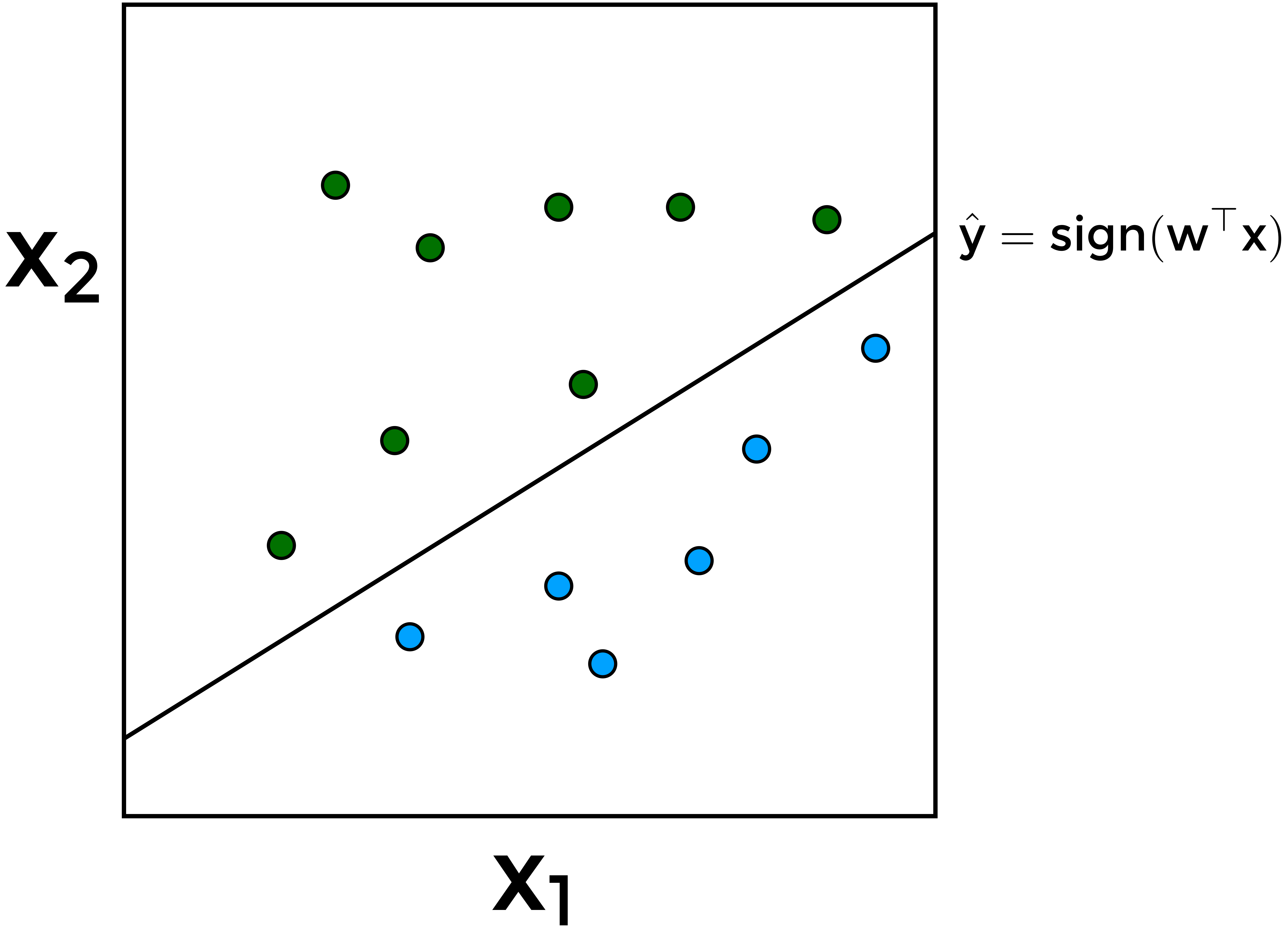
$\| \mathbf{x}' \| = \| \beta \mathbf{w} \|$

$$\implies \| \mathbf{x}' \| = \beta \| \mathbf{w} \|$$

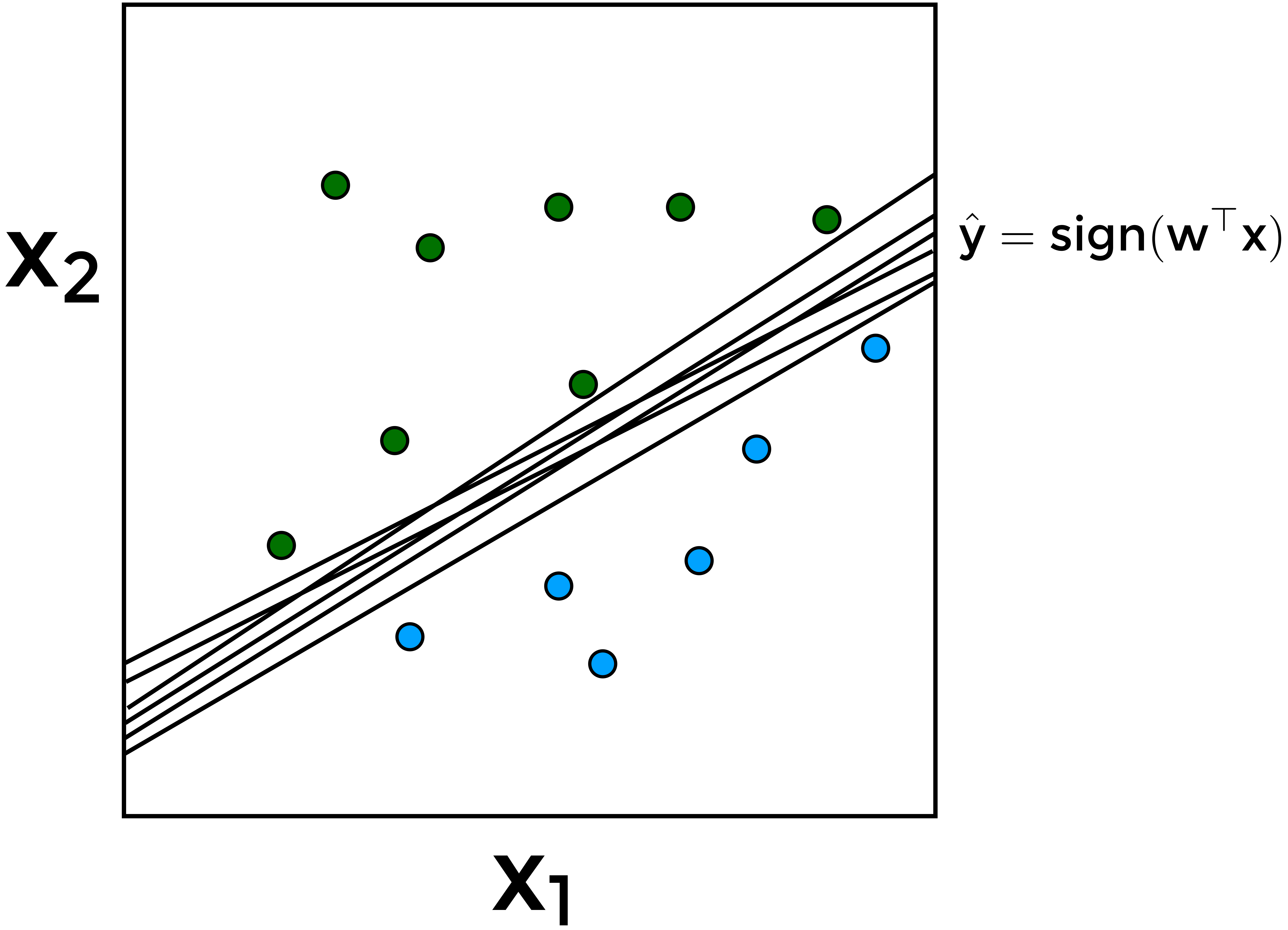
$$\implies \| \mathbf{x}' \| = \frac{-w_0}{\| \mathbf{w} \|^2} \| \mathbf{w} \|$$

$$\implies \| \mathbf{x}' \| = \frac{-w_0}{\| \mathbf{w} \|}$$

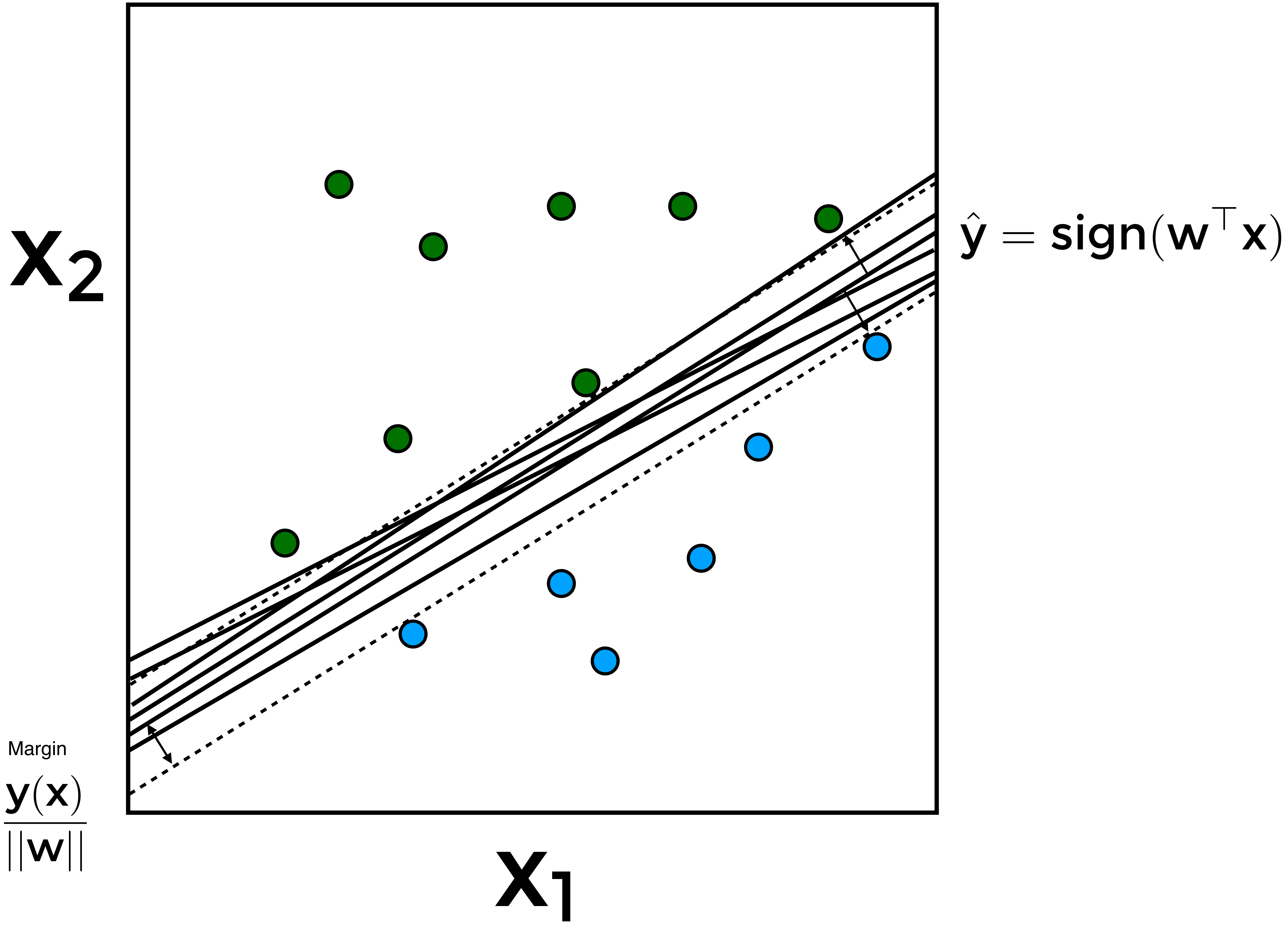

x_2
x_1



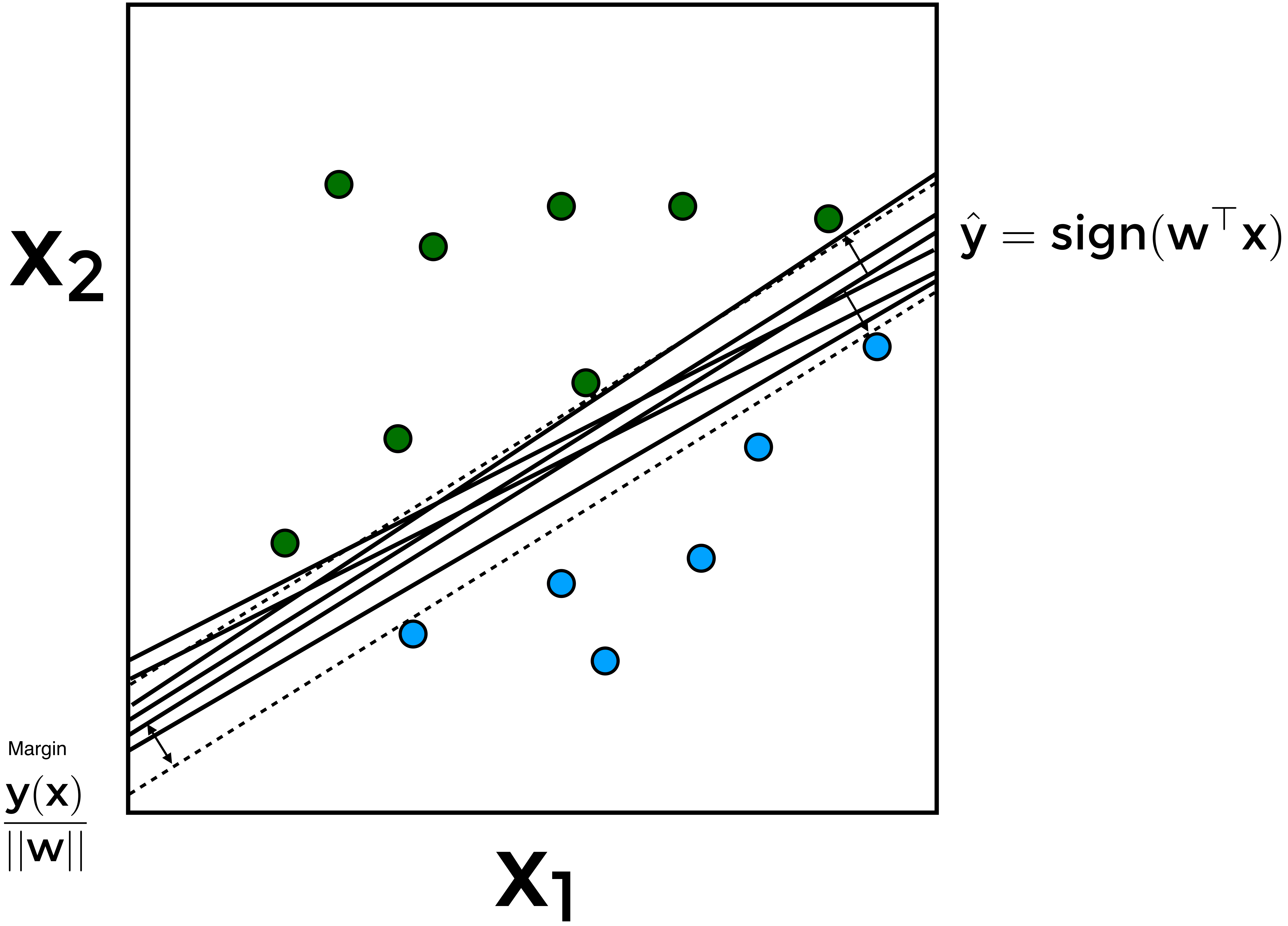
$\mathbf{x}_2$
$\hat{y} = \text{sign}(\mathbf{w}^\top \mathbf{x})$
$\mathbf{x}_1$


x_2
$\hat{y} = \text{sign}(w^\top x)$
x_1

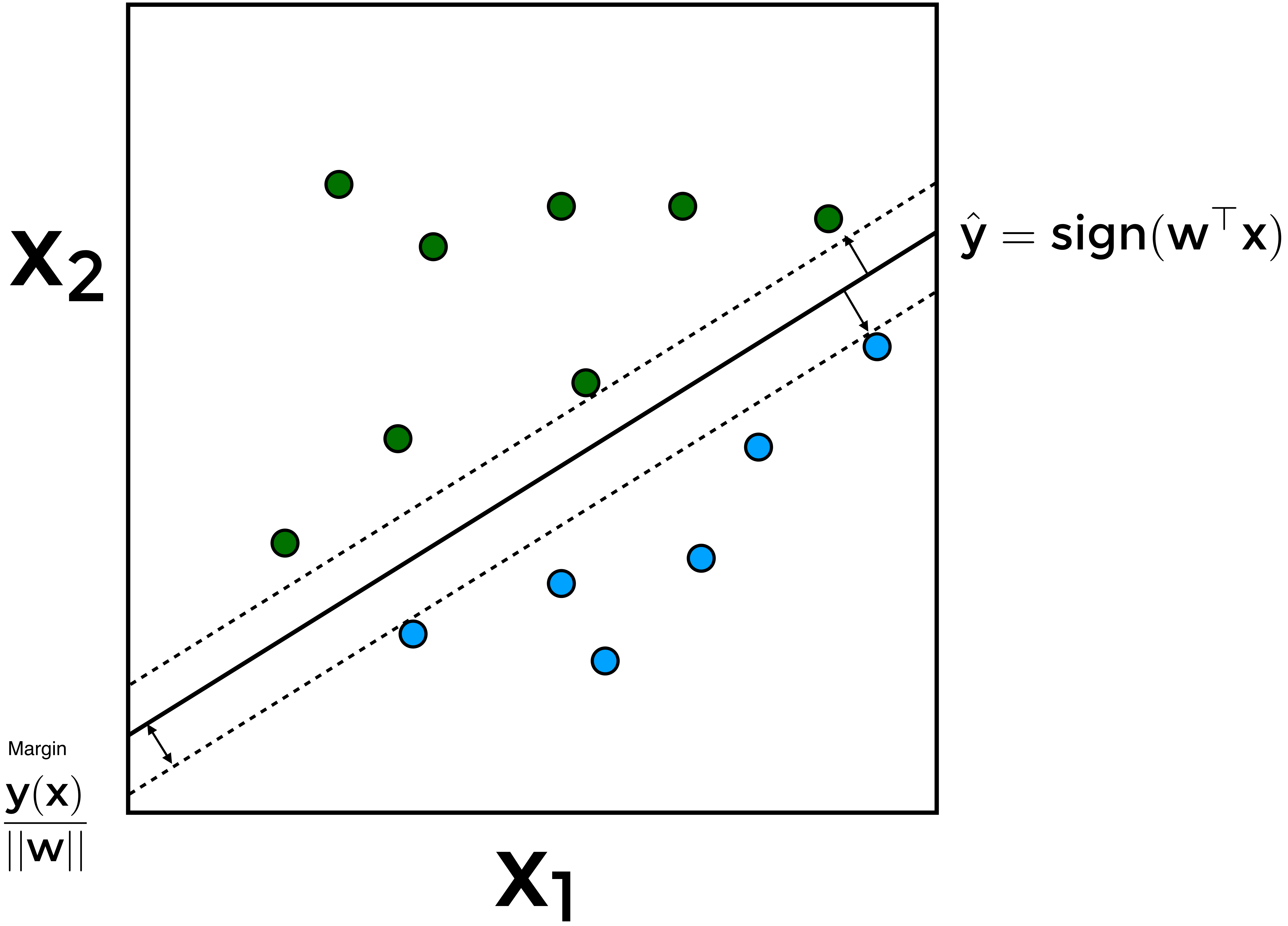
The objective is to find the separating boundary that maximizes the margin



The objective is to find the separating boundary that maximizes the margin

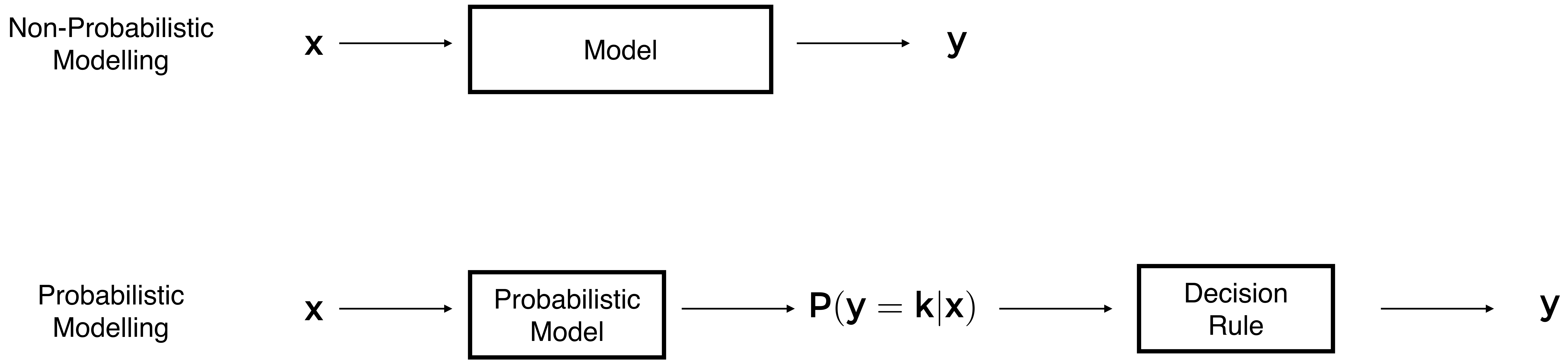


The objective is to find the separating boundary that maximizes the margin



Probabilistic Models for Classification

Probabilistic Models separate Decision and Inference



Probabilistic models

1. **Model the conditional directly:**

$$P(y = k|x)$$

2. **Model the joint (or the prior and the class conditionals):**

$$\underbrace{P(y = k|x)}_{\text{posterior}} \propto \underbrace{P(y = k, x)}_{\text{joint}}$$

$$= \underbrace{P(x \mid y = k)}_{\text{class conditional densities}} \underbrace{P(y = k)}_{\text{class prior}}$$

Bayes' Theorem

Probabilistic Modelling

Often
inteertwinned

1. **Posit a model: P(X, Y)**
    - How the data is generated
2. **Parametrize the distributions: P(X, Y | Parameters)**
3. **Set the objective (e.g., MLE)**
4. **Learn the parameters of the model:**
    - E.g., Naive Bayes: learn the parameters of the class conditional P(X | Y) and of the prior P(Y)
5. **Use the model (e.g., for predictions)**