

Probably Approximation Correct (PAC)

Prof. Ansar Sheikh

PAC learning is a framework for the mathematical analysis of machine learning

Goal of PAC: With high probability (“Probably”) , the selected hypothesis will have lower error (“Approximately Correct”)

In the PAC model, we specify two small parameters, ϵ and δ , and require that with probability at least $(1 - \delta)$ a system learn a concept with error at most ϵ .

ϵ and δ parameters

ϵ gives an upper bound on the error in accuracy with which h approximated (accuracy: $1 - \epsilon$)

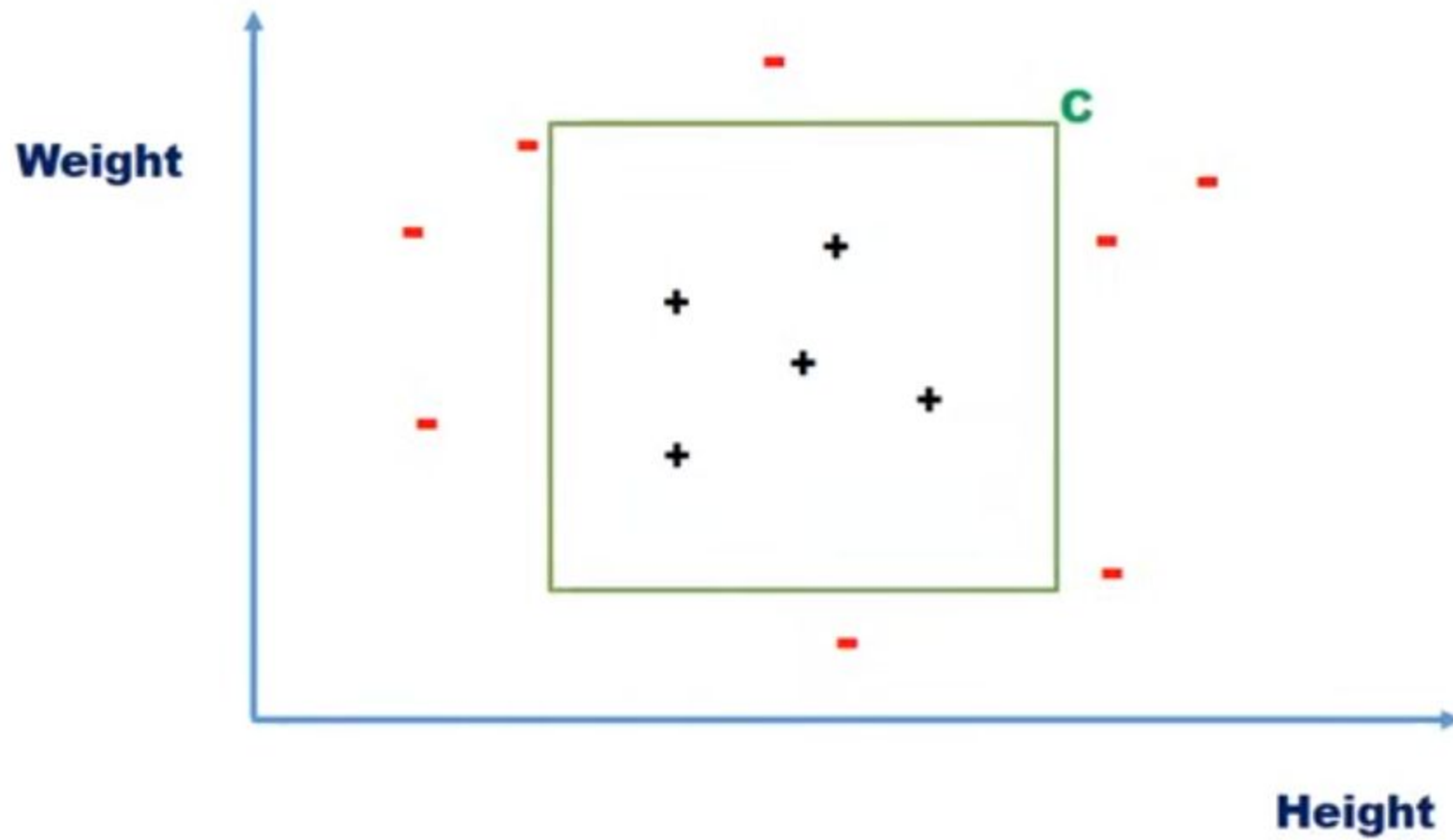
δ gives the probability of failure in achieving this accuracy (confidence : $1 - \delta$)

Example: Learn the concept "medium-built person"

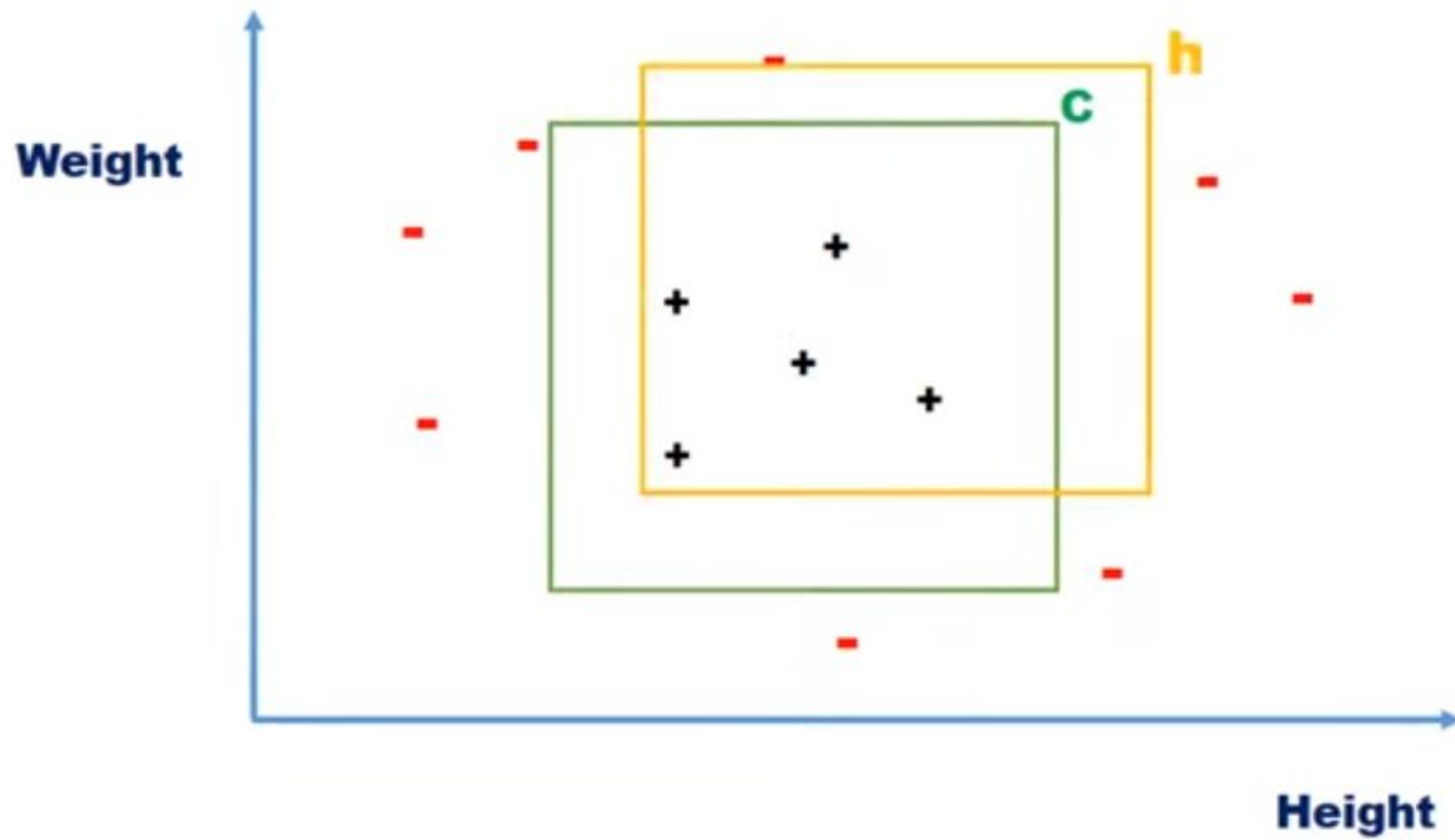
We are given the height and weight of m individuals, the training set.

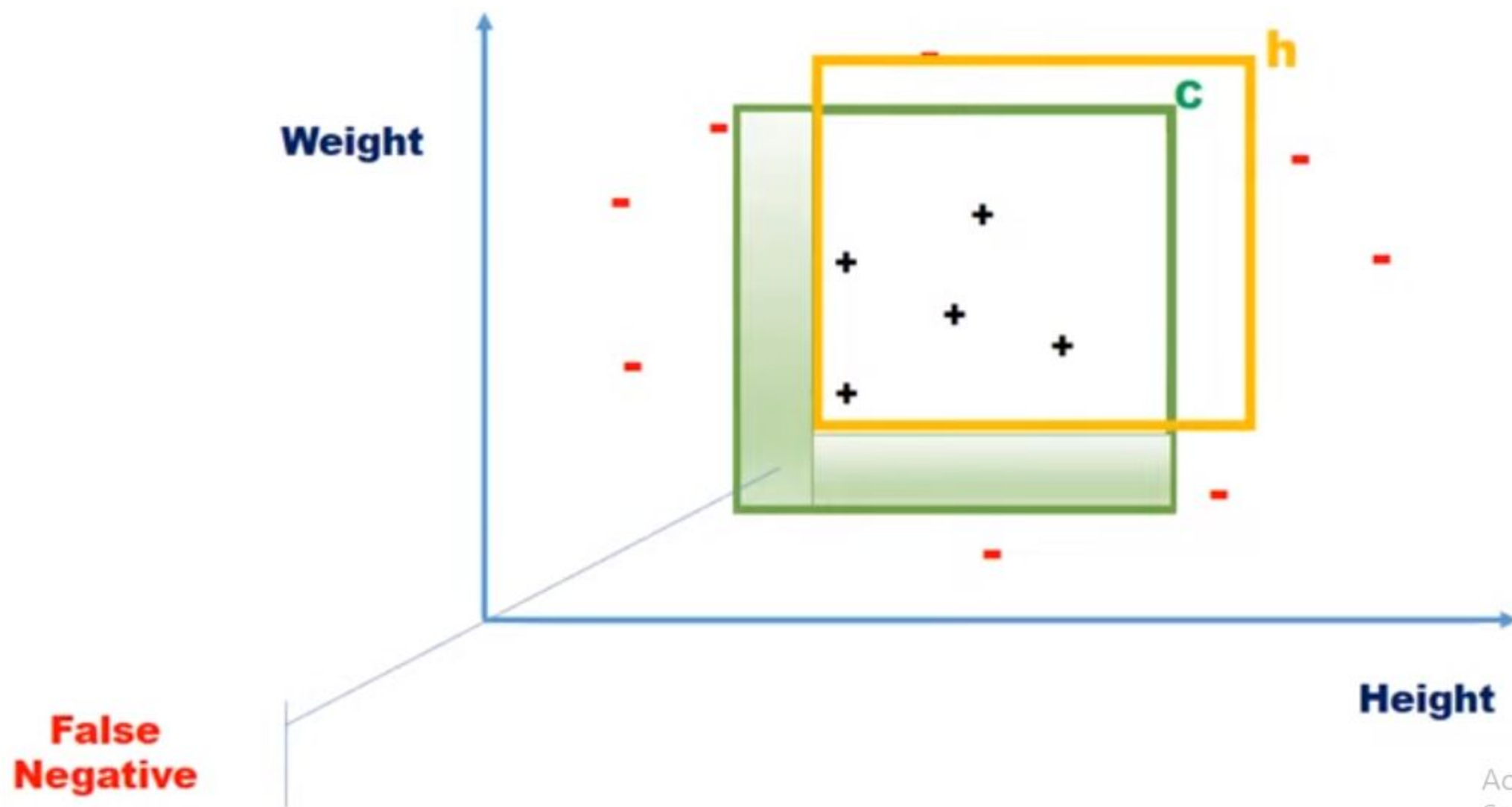
We are told for each [height, weight] pair if the person is medium built or not.

We would like to learn this concept, i.e. produce an algorithm that in future correctly answers if a pair [height, weight] represents a medium-built person or not.

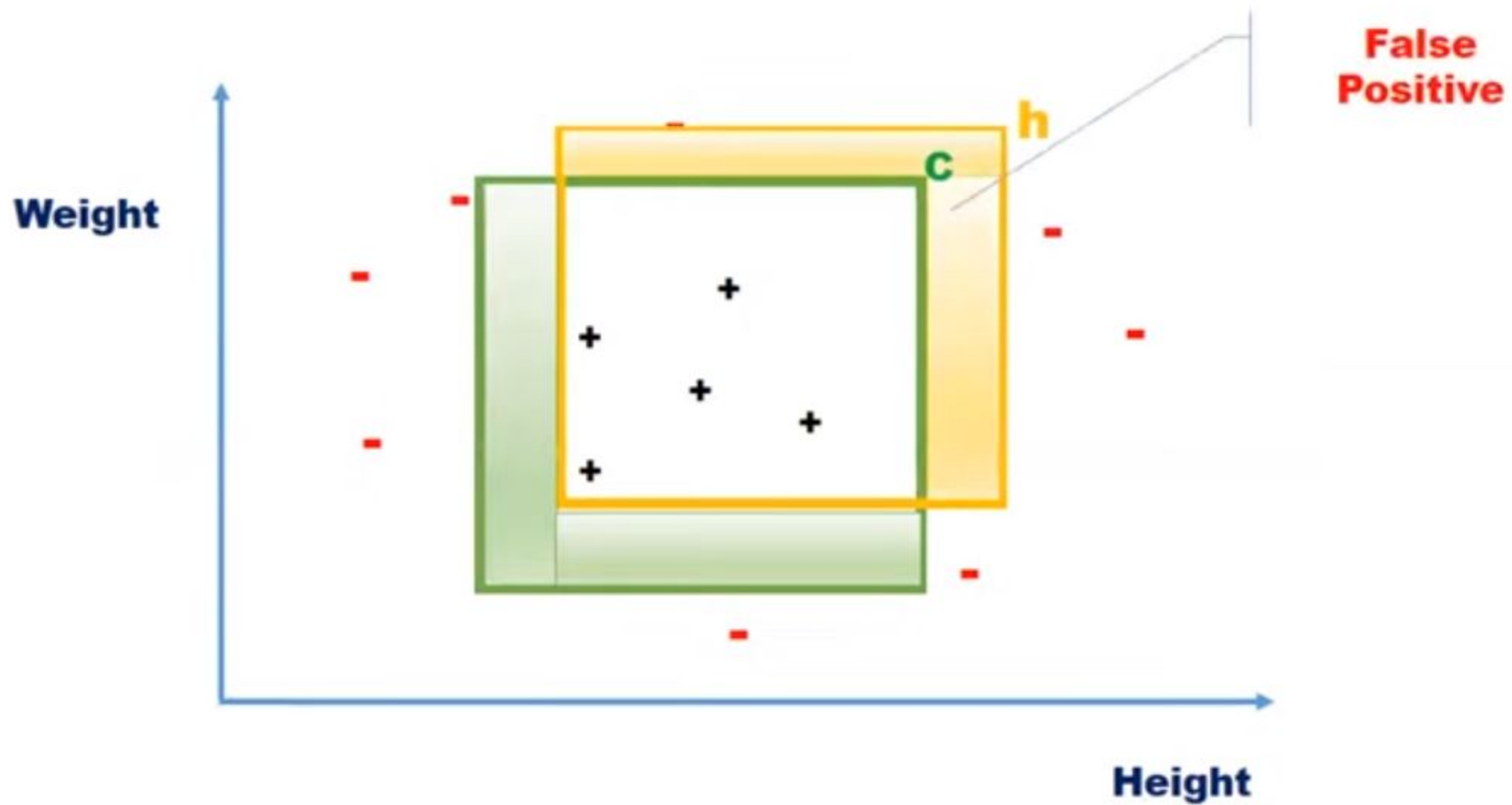


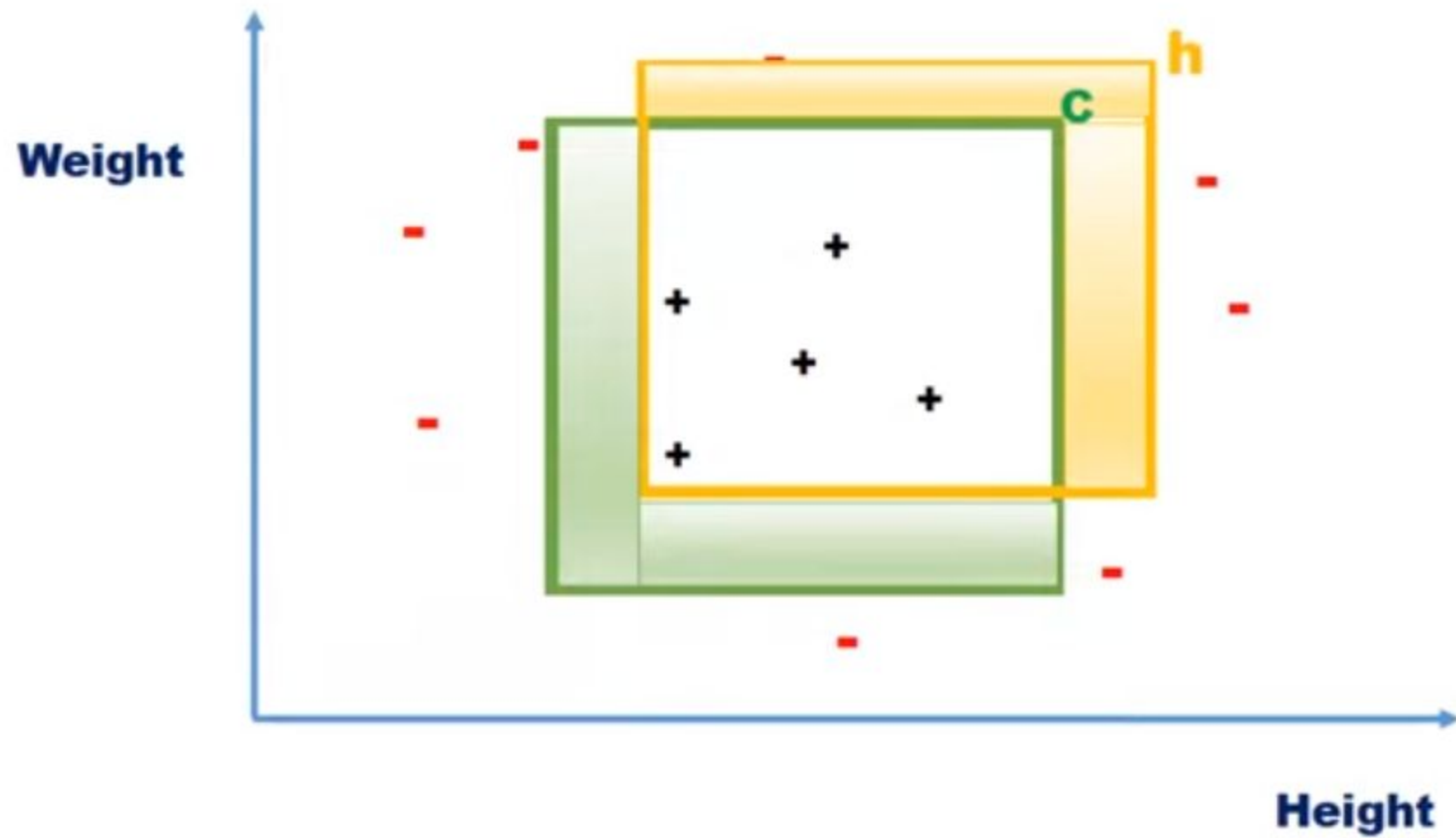
Activate Windows
Go to Settings to activate Windows.





Activate Windows
Go to Settings to activate Windows.





$$P(C \text{ XOR } h) \leq \epsilon$$

Approximately Correct

A hypothesis is said to be approximately correct , if the error is less than or equal to ϵ , where $0 \leq \epsilon \leq 1/2$

$$\text{i.e., } P(C \oplus h) \leq \epsilon$$

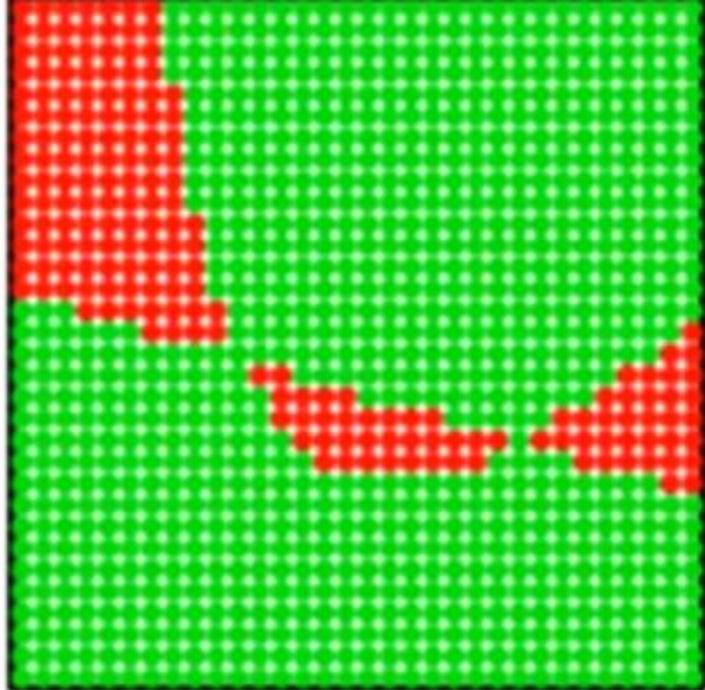
Probably Approximately Correct

The goal is to achieve low generalization error with high probability.

$$\Pr(\text{Error}(h) \leq \epsilon) > 1 - \delta$$

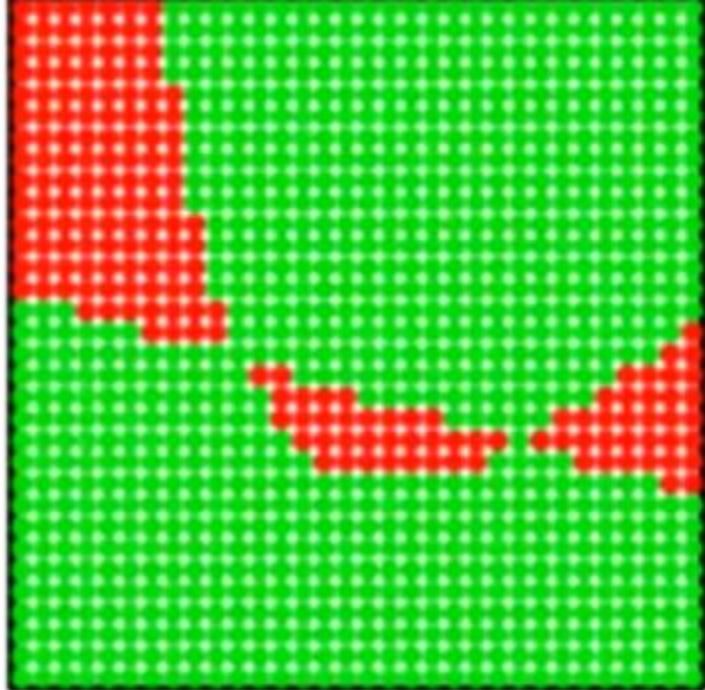
$$\text{i.e. } \Pr(P(C \oplus h) \leq \epsilon) > 1 - \delta$$

Weight



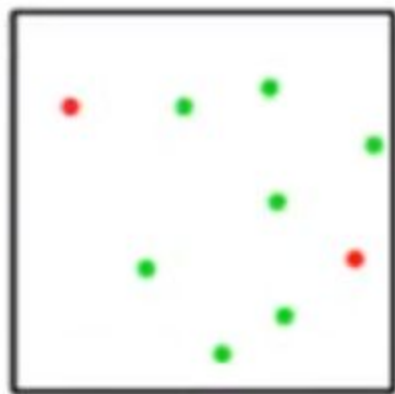
Height

Weight



Height

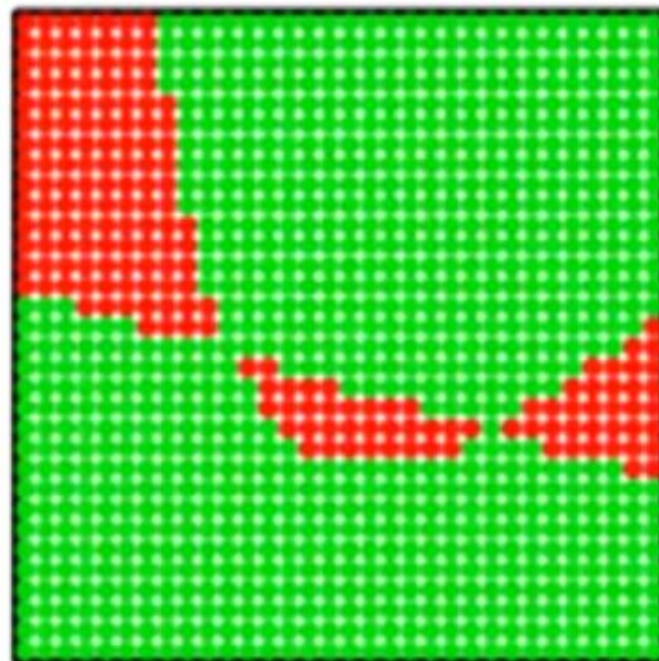
Weight



Height

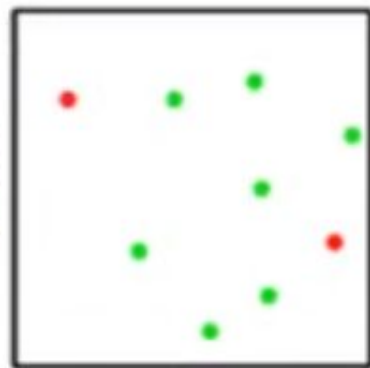
Activate Windows
Go to Settings to activate Windows.

Weight



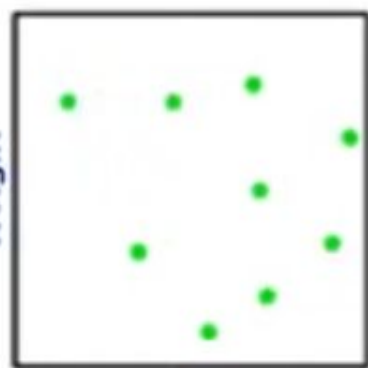
Height

Weight



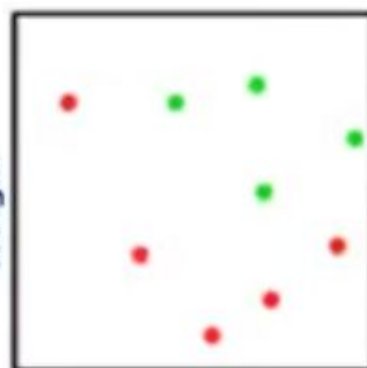
Height

Weight



Height

Weight



Height

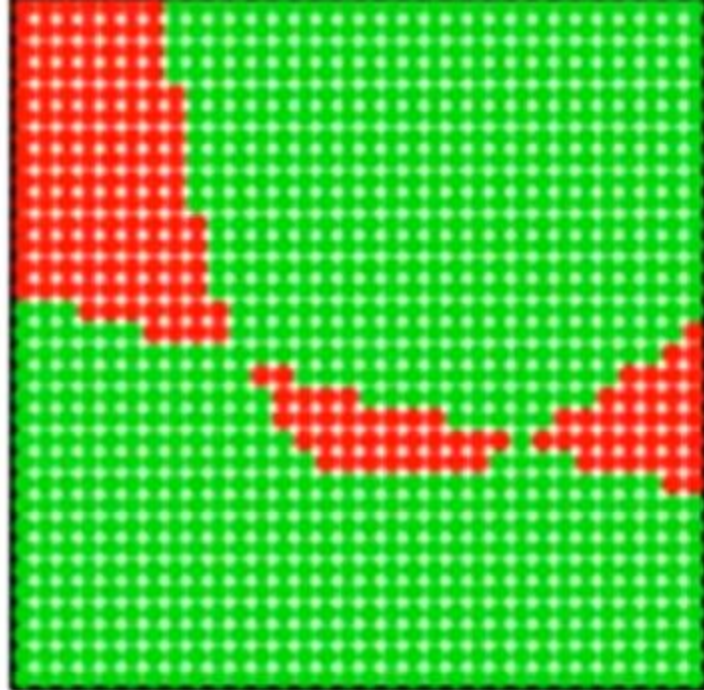
...

Weight

Height

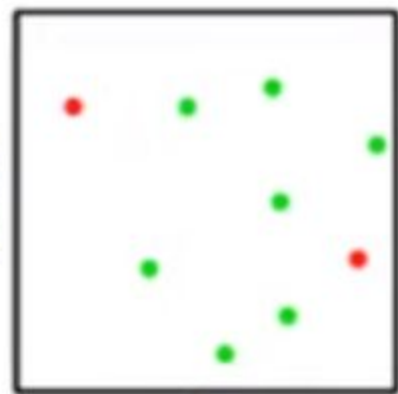
Activate Windows
Go to Settings to activate Windows.

Weight



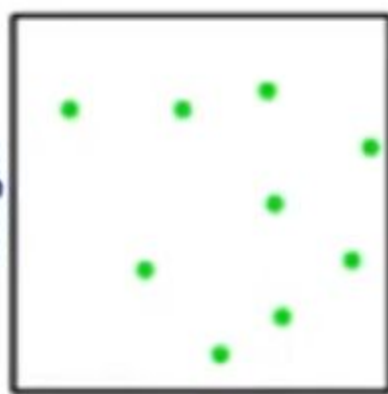
Height

Weight



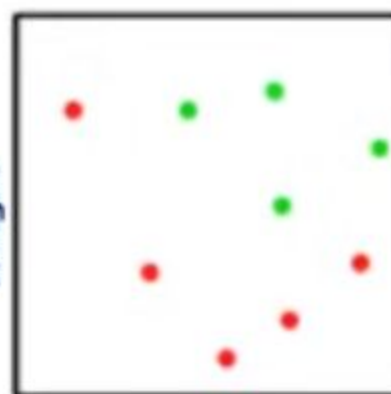
Height

Weight



Height

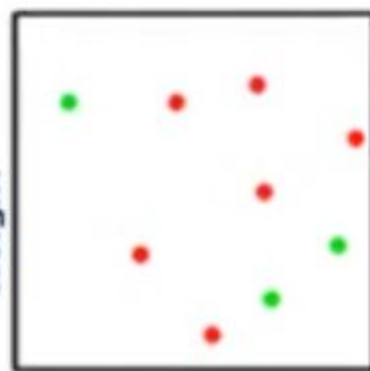
Weight



Height

...

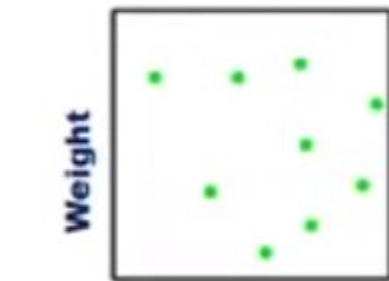
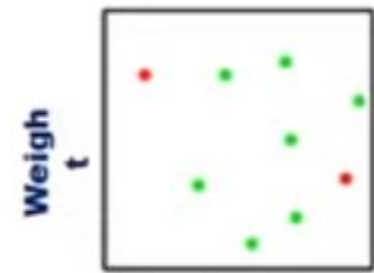
Weight



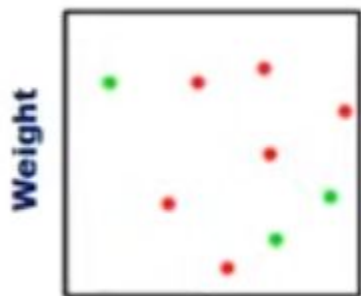
Height

Activate Windows
Go to Settings to activate Windows.

Error(H1)

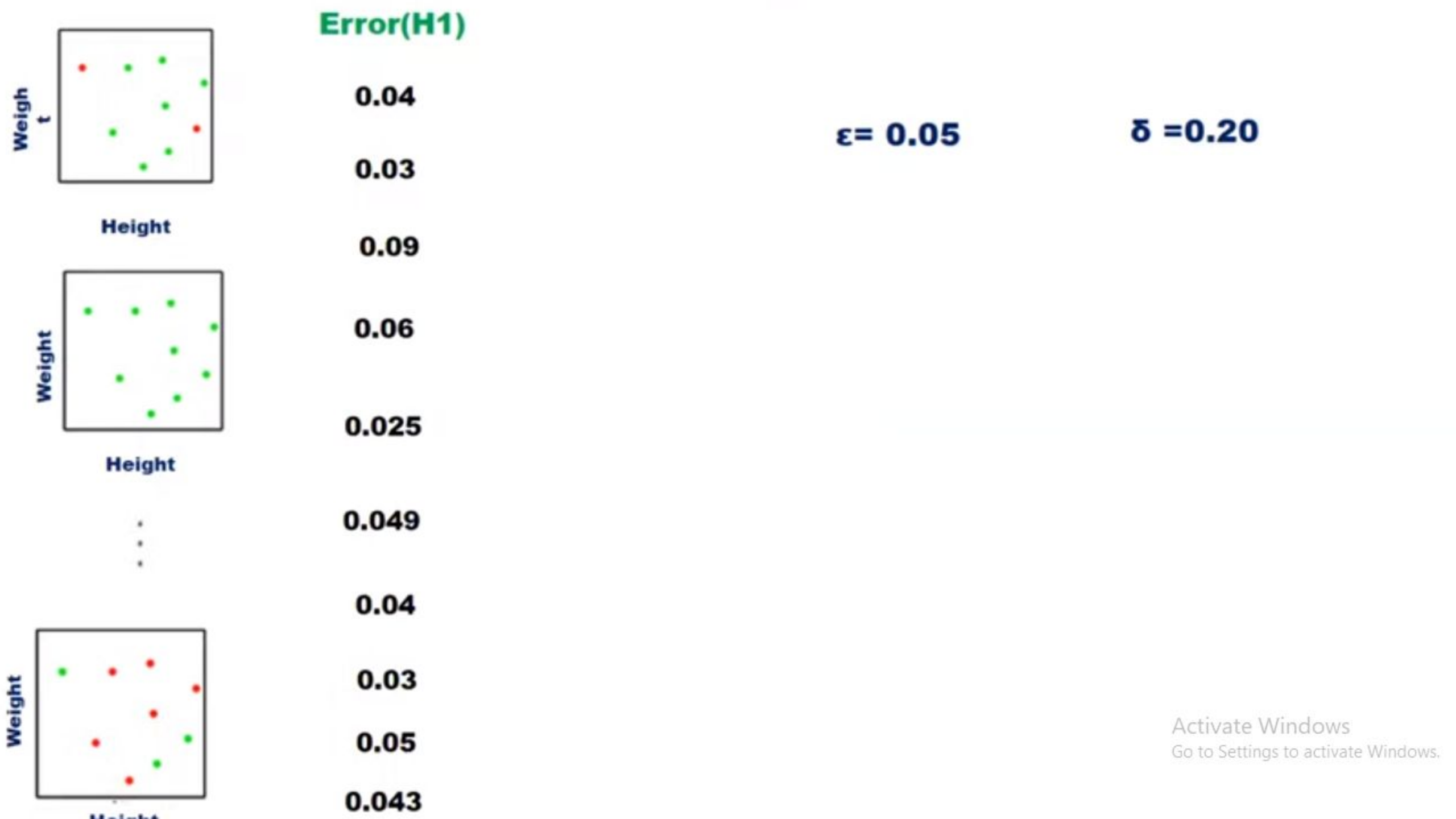


⋮

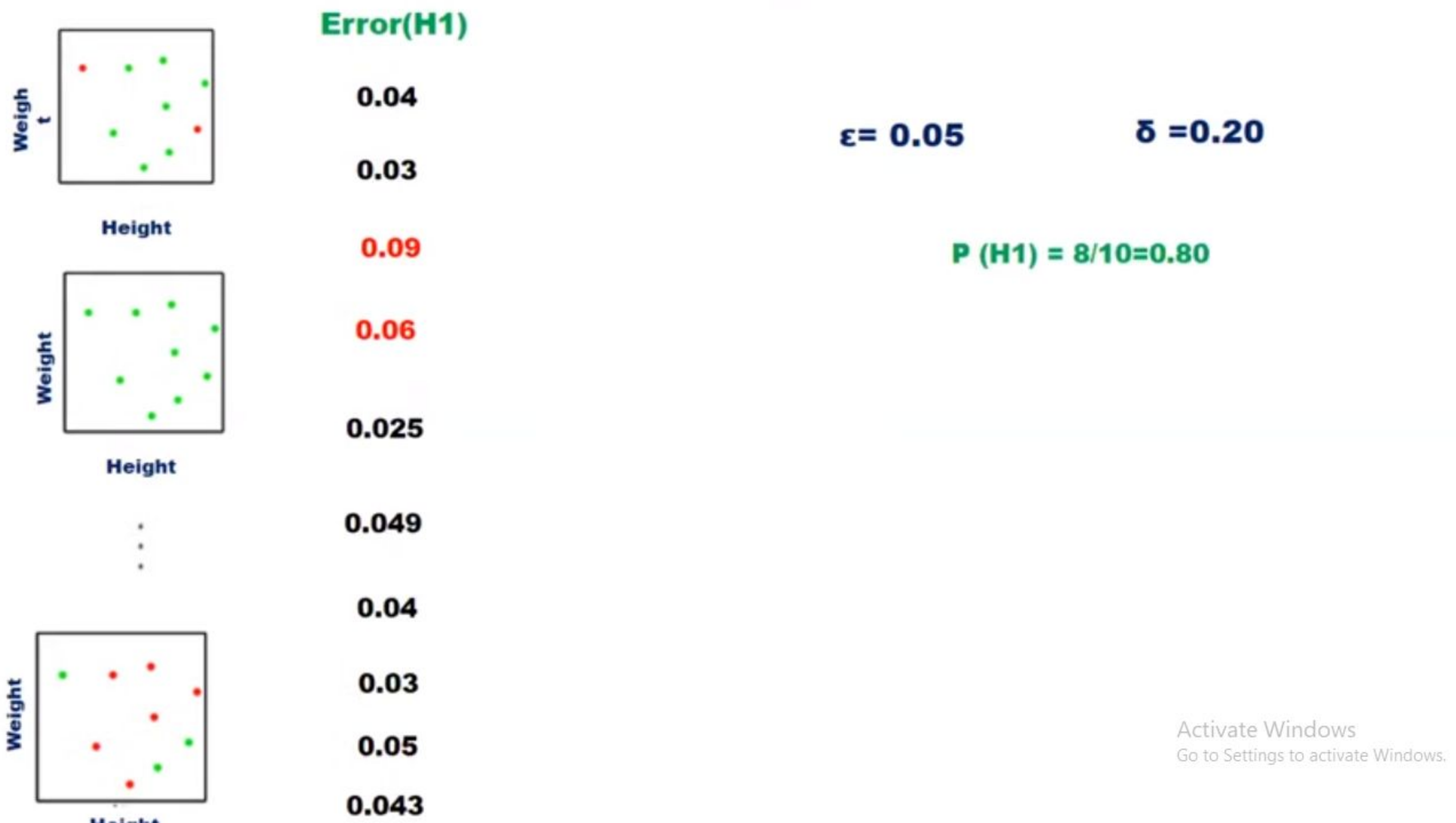


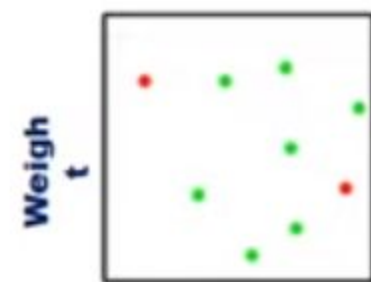
$$\epsilon = 0.05$$

$$\delta = 0.20$$









Error(H1)

0.04

0.03

0.09

0.06

0.025

0.049

0.04

0.03

0.05

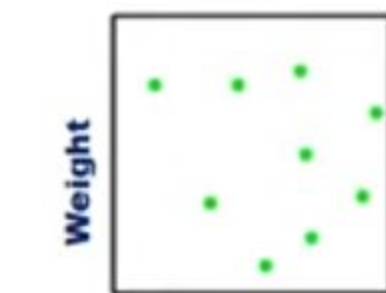
0.043

$\varepsilon = 0.05$

$\delta = 0.20$

P (H1) = 8/10=0.80

$P(H1) = 8/10 = 0.80 \geq 1 - 0.20$

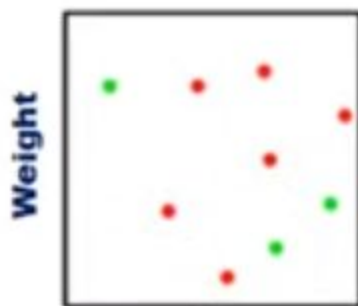


Height

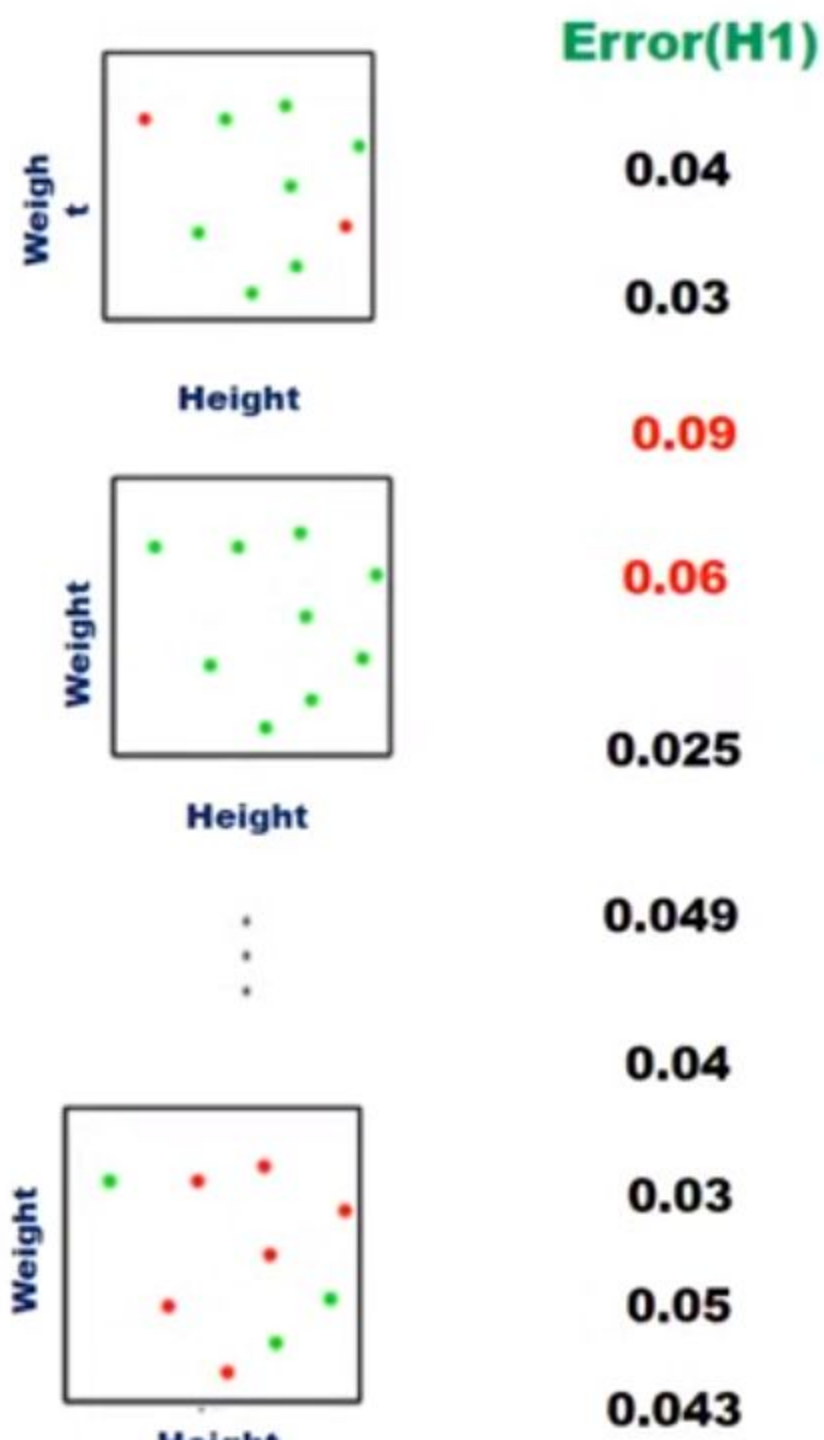
•

•

•



Abstract






$$\epsilon = 0.05$$

$$\delta = 0.20$$

$$P(H1) = 8/10 = 0.80$$

$$P(H1) = 8/10 = 0.80 \geq 1 - 0.20$$

Hence H1 is probably approximately correct

| | Error(H1) | Error(h2) | |
|--|-----------|-----------|--|
|  | 0.04 | 0.04 | $\epsilon = 0.05$ |
| | 0.03 | 0.035 | $\delta = 0.20$ |
|  | 0.09 | 0.039 | $P(H1) = 8/10 = 0.80$ |
| | 0.06 | 0.06 | $P(H1) = 8/10 = 0.80 \geq 1 - 0.20$ |
| | 0.025 | 0.025 | Hence H1 is probably approximately correct |
| | 0.049 | 0.059 | $P(H2) = 7/10 = 0.70$ |
| | 0.04 | 0.04 | $P(H2) = 7/10 = 0.70 < 1 - 0.20$ |
| | 0.03 | 0.03 | Hence H2 is not probably approximately correct |
| | 0.05 | 0.55 | |
|  | 0.043 | 0.043 | |

The PAC Learning Model: General Setting

Let \mathcal{X} be the set of all possible instances over which target functions are to be defined

The PAC Learning Model: General Setting

Let \mathcal{X} be the set of all possible instances over which target functions are to be defined

\mathcal{C} is the set of target concepts the learner may be asked to learn ,

where each $c \in \mathcal{C}$, c may be viewed as a boolean-valued function $c : \mathcal{X} \rightarrow \{0,1\}$

The PAC Learning Model: General Setting

Let X be the set of all possible instances over which target functions are to be defined

C is the set of target concepts the learner may be asked to learn ,

where each $c \in C$, c may be viewed as a boolean-valued function $c : X \rightarrow \{0,1\}$

If x is a positive example $c(x) = 1$; if x is a negative example $c(x) = 0$.

The PAC Learning Model: General Setting

Let \mathcal{X} be the set of all possible instances over which target functions are to be defined

\mathcal{C} is the set of target concepts the learner may be asked to learn ,

where each $c \in \mathcal{C}$, c may be viewed as a boolean-valued function $c : \mathcal{X} \rightarrow \{0,1\}$

If x is a positive example $c(x) = 1$; if x is a negative example $c(x) = 0$.

Examples are drawn at random from \mathcal{X} according to a probability distribution \mathcal{D}

The PAC Learning Model: General Setting

Let \mathcal{X} be the set of all possible instances over which target functions are to be defined

\mathcal{C} is the set of target concepts the learner may be asked to learn ,

where each $c \in \mathcal{C}$, c may be viewed as a boolean-valued function $c : \mathcal{X} \rightarrow \{0,1\}$

If x is a positive example $c(x) = 1$; if x is a negative example $c(x) = 0$.

Examples are drawn at random from \mathcal{X} according to a probability distribution \mathcal{D}

A learner L considers a set of hypotheses \mathcal{H} and, after observing some sequence of training examples, outputs a hypothesis $h \in \mathcal{H}$ which is its estimate of c .

The true error of hypothesis h (denoted $error_{\mathcal{D}}(h)$) with respect to target concept c and distribution \mathcal{D} is the probability that h will misclassify an instance drawn at random according to \mathcal{D} .

$$error_{\mathcal{D}}(h) \equiv \Pr_{x \in \mathcal{D}} [c(x) \neq h(x)]$$

Since for any set of training examples $T \subset X$ there may be multiple hypotheses consistent with T , however the learner cannot be guaranteed to choose the one corresponding to the target concept, unless it is trained on every instance in X (unrealistic)

Since for any set of training examples $T \subset X$ there may be multiple hypotheses consistent with T , however the learner cannot be guaranteed to choose the one corresponding to the target concept, unless it is trained on every instance in X (unrealistic)

Don't require learner to output a zero error hypothesis – only require that error be bounded by a constant ϵ that can be made arbitrarily small

Since training examples drawn at random, there must be a non-zero probability that they will be misleading

Don't require learner to succeed for every randomly drawn sequence of training examples – only require that its probability of failure be bounded by a constant δ that can also be made arbitrarily small

Formal Definition of PAC Learning

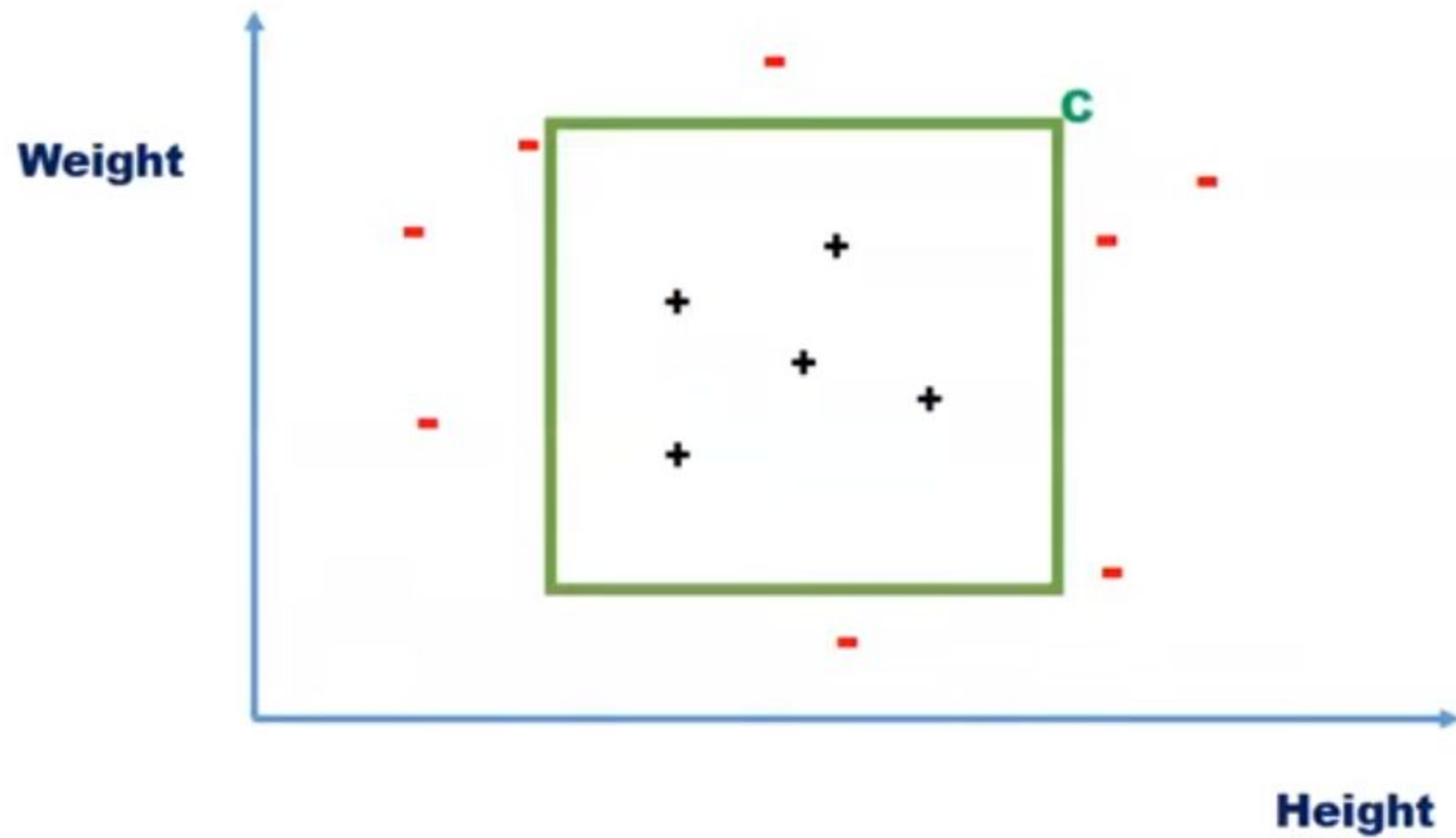
Consider a class C of possible target concepts defined over a set of instances X of length n , and a learner L using hypothesis space H .

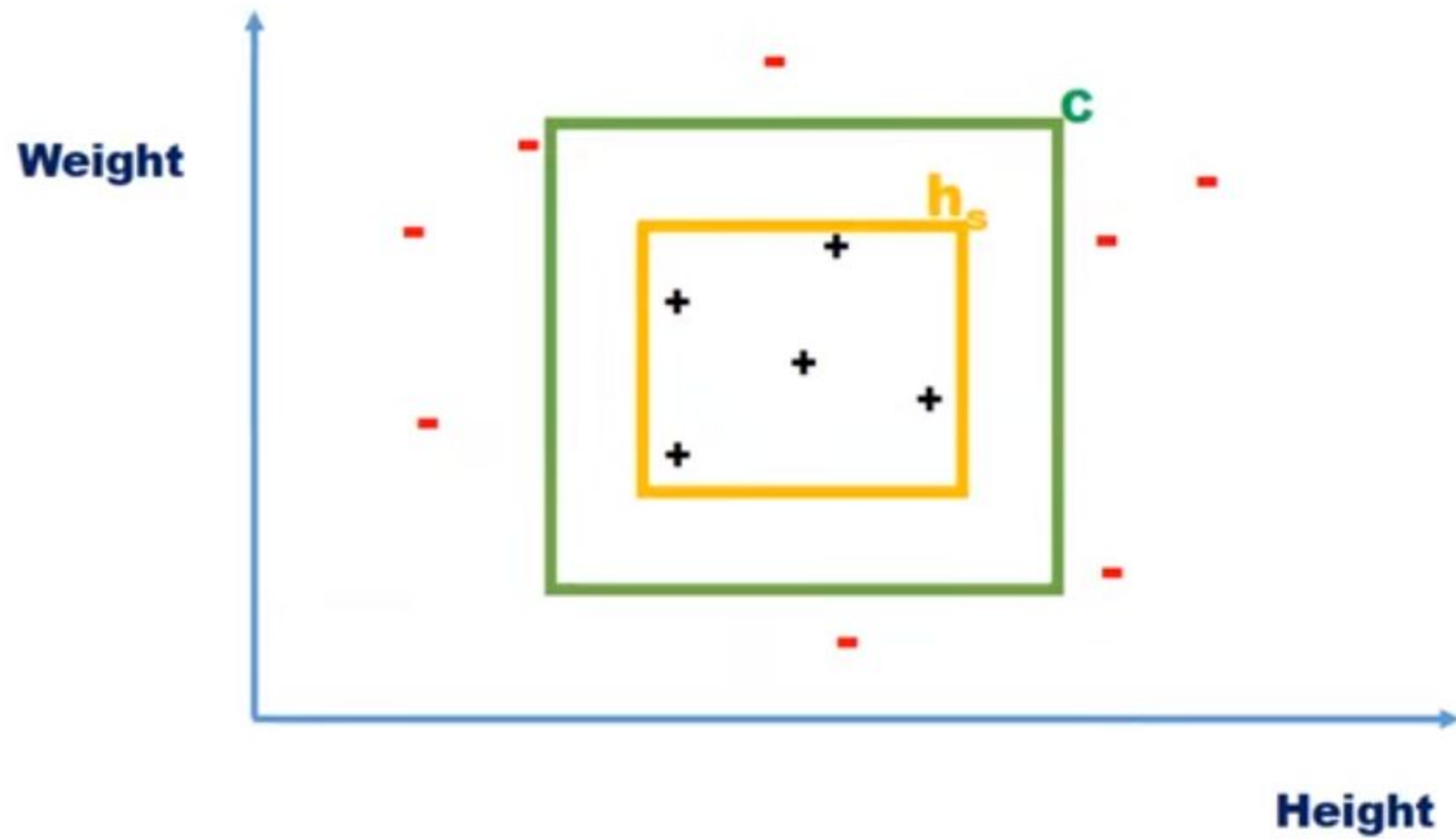
Formal Definition of PAC Learning

Consider a class C of possible target concepts defined over a set of instances X of length n , and a learner L using hypothesis space H .

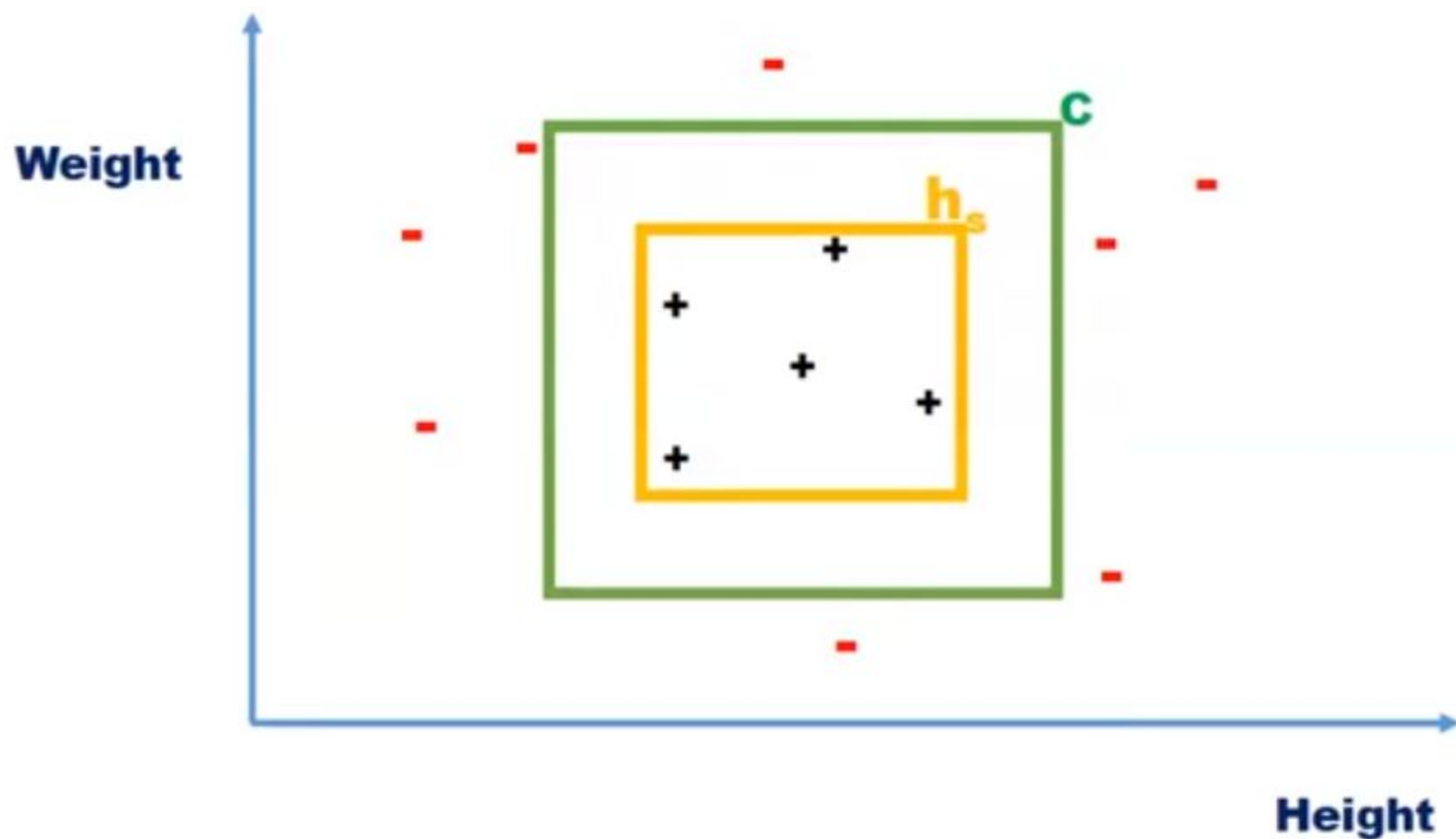
C is PAC-learnable by L using H if for all $c \in C$, distributions D over X , ϵ such that $0 < \epsilon < 1/2$, and δ such that $0 < \delta < 1/2$, learner L will output a hypothesis $h \in H$ such that $\text{error}_D(h) \leq \epsilon$ with probability at least $(1 - \delta)$, in time that is polynomial in $1 / \epsilon$, $1 / \delta$, n and $\text{size}(c)$

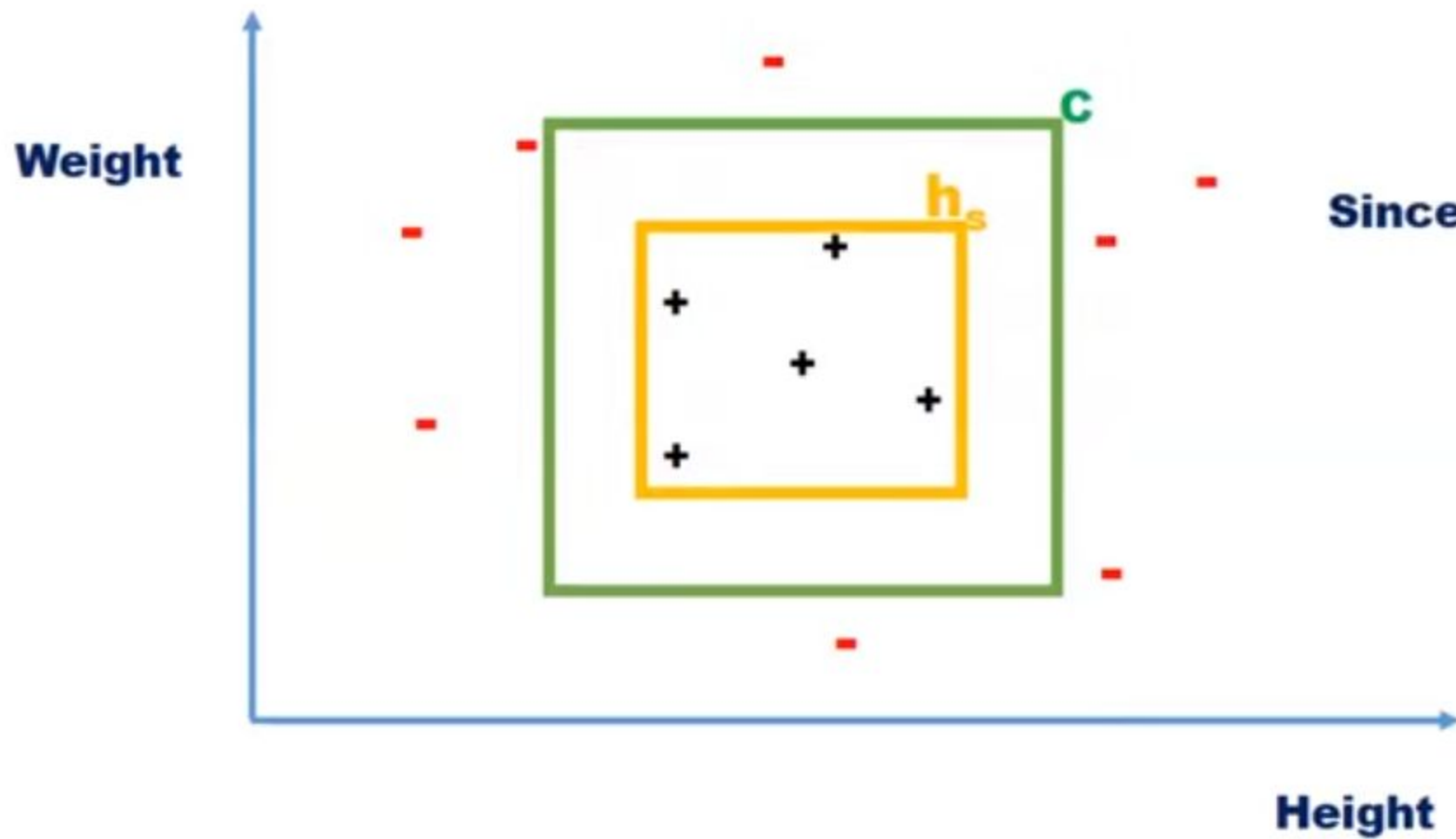
PAC-learnability for axis-aligned rectangles



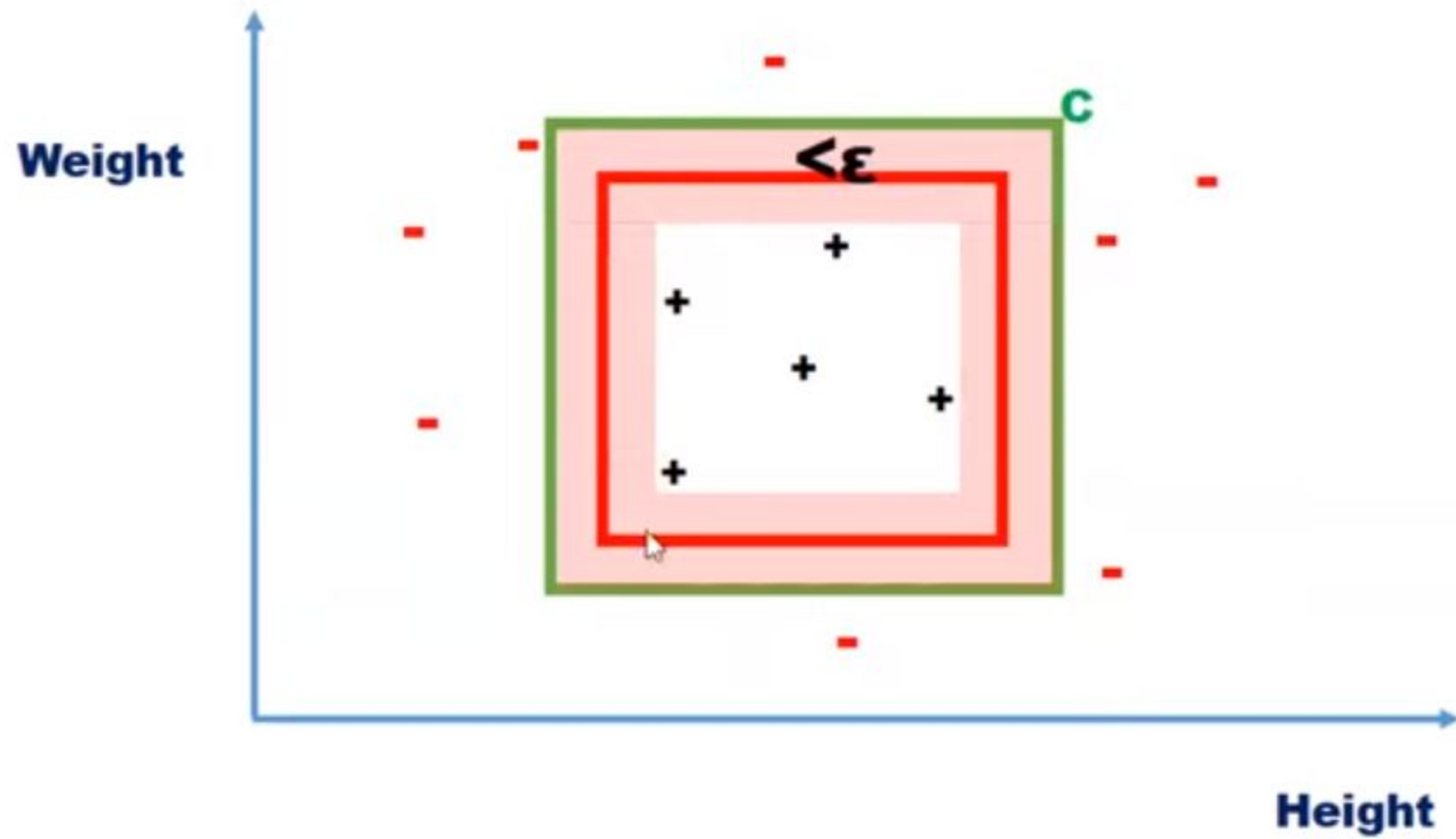


h_s is the tightest possible rectangle around a set of positive training examples

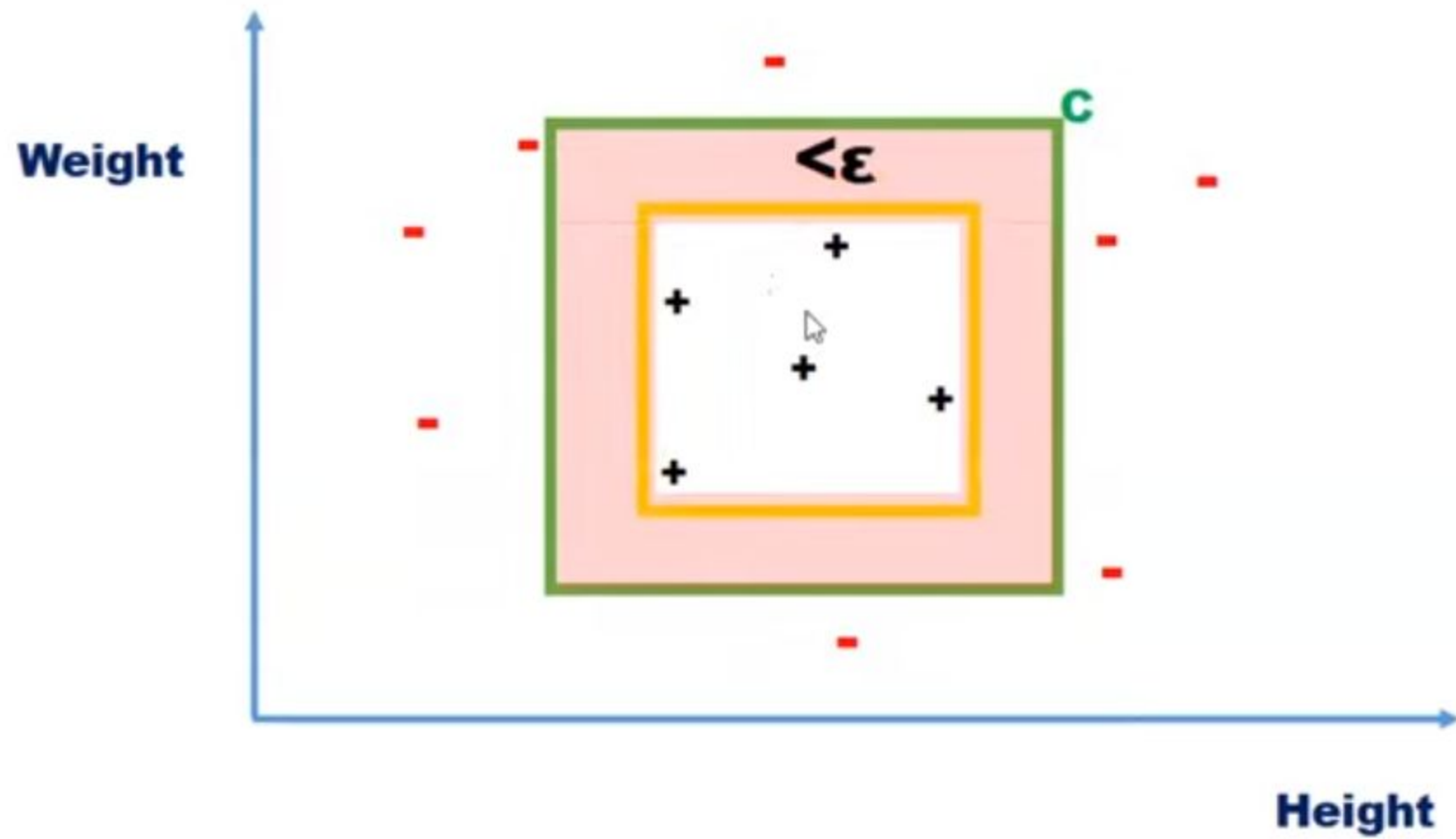


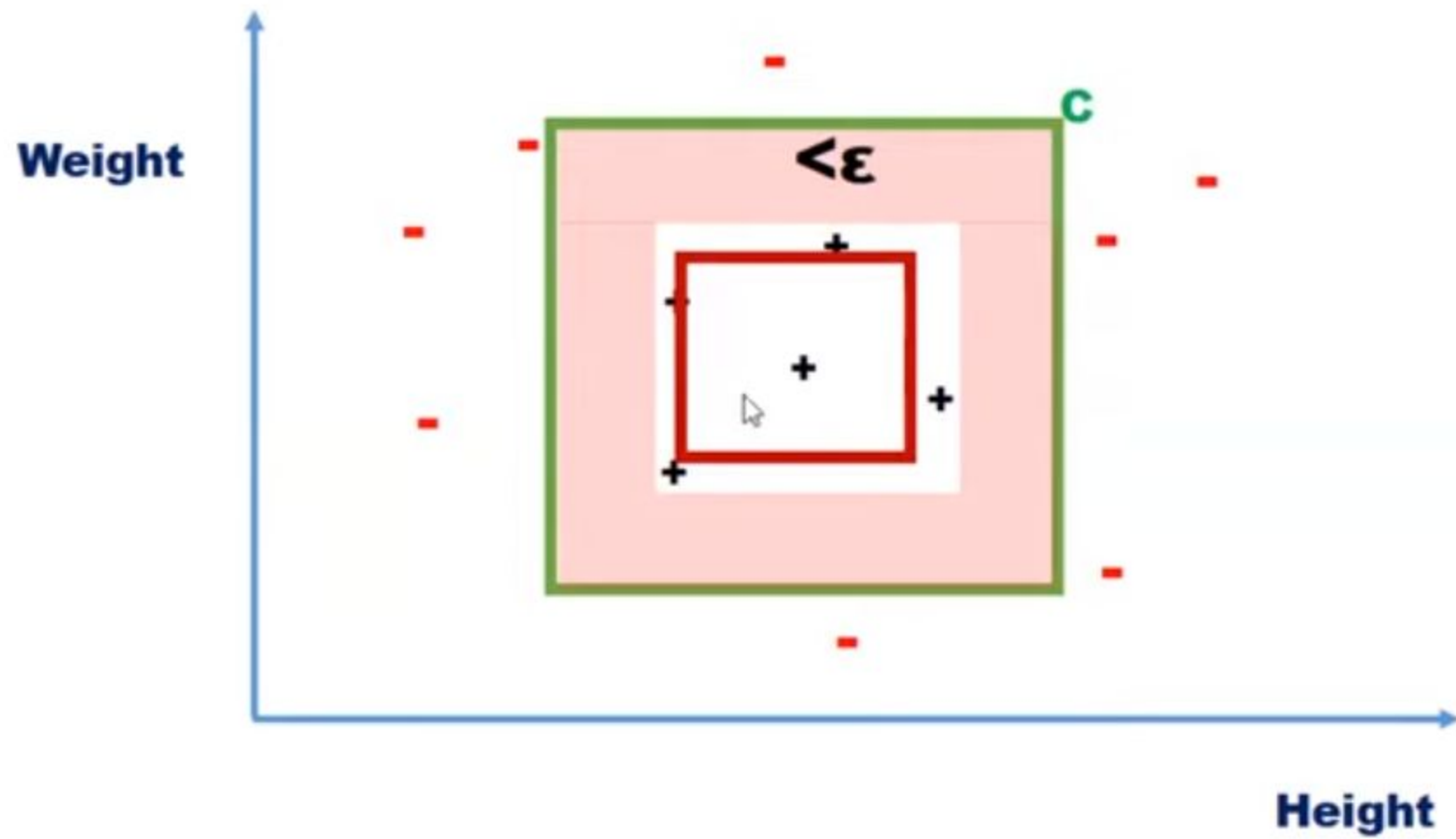


Since $h_s \subset C$, Error Region = $C-h$

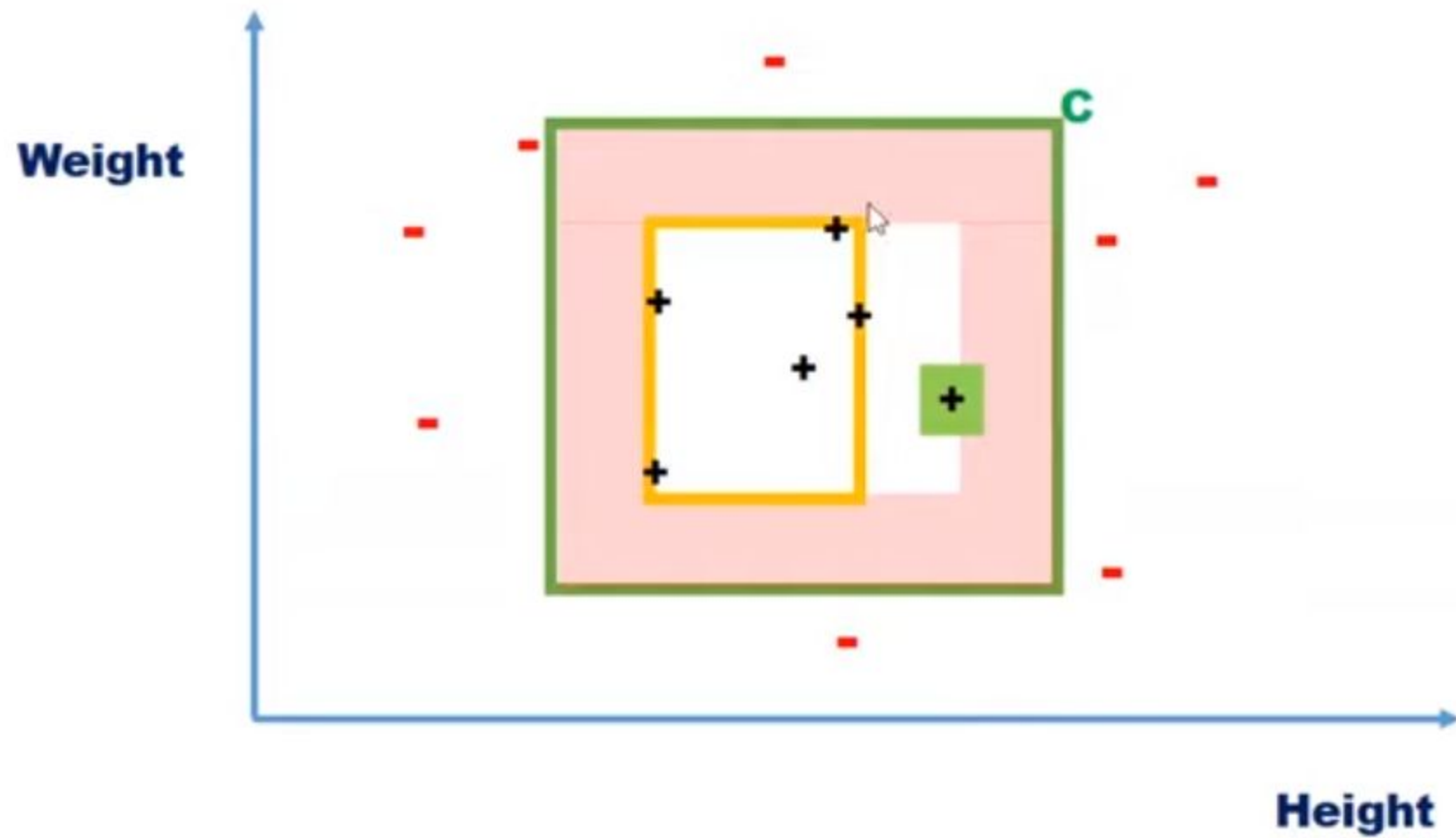


Activate Windows
Go to Settings to activate Windows.





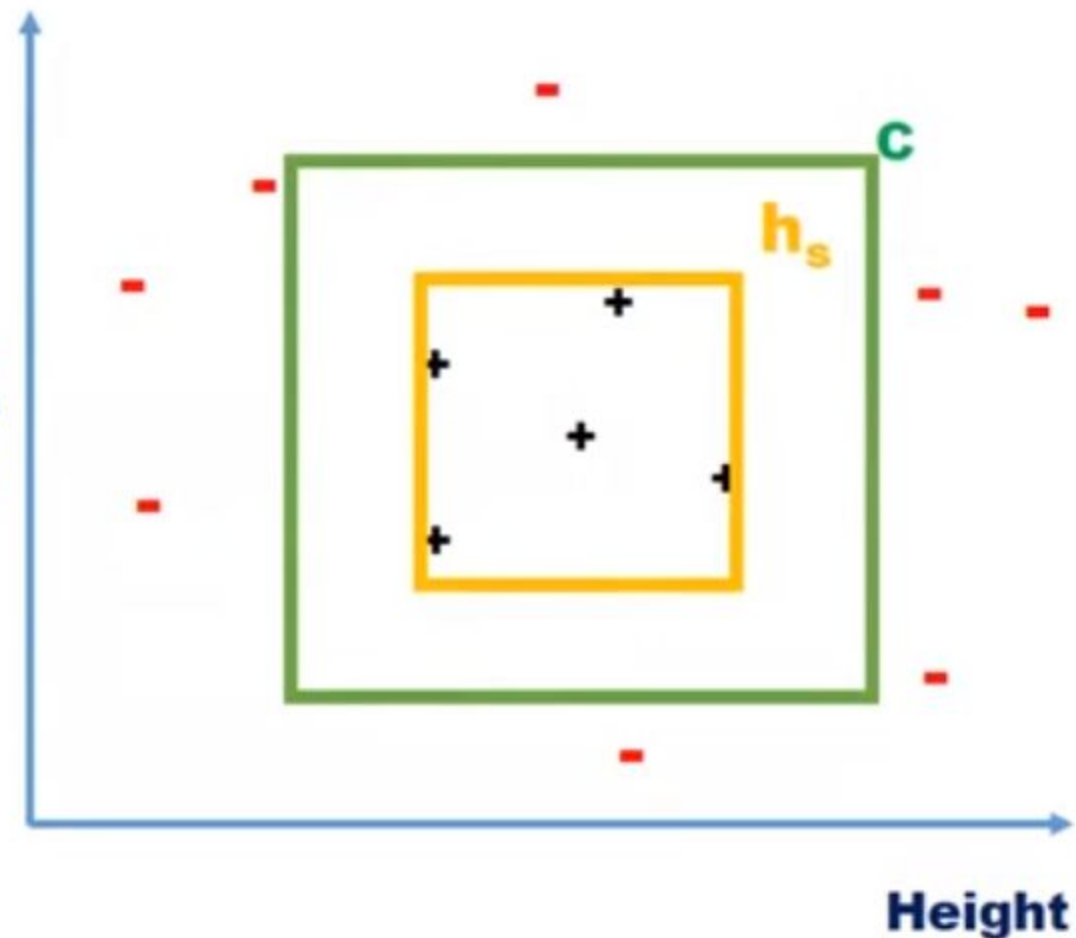
Activate Windows
Go to Settings to activate Windows.



Activate Windows
Go to Settings to activate Windows.

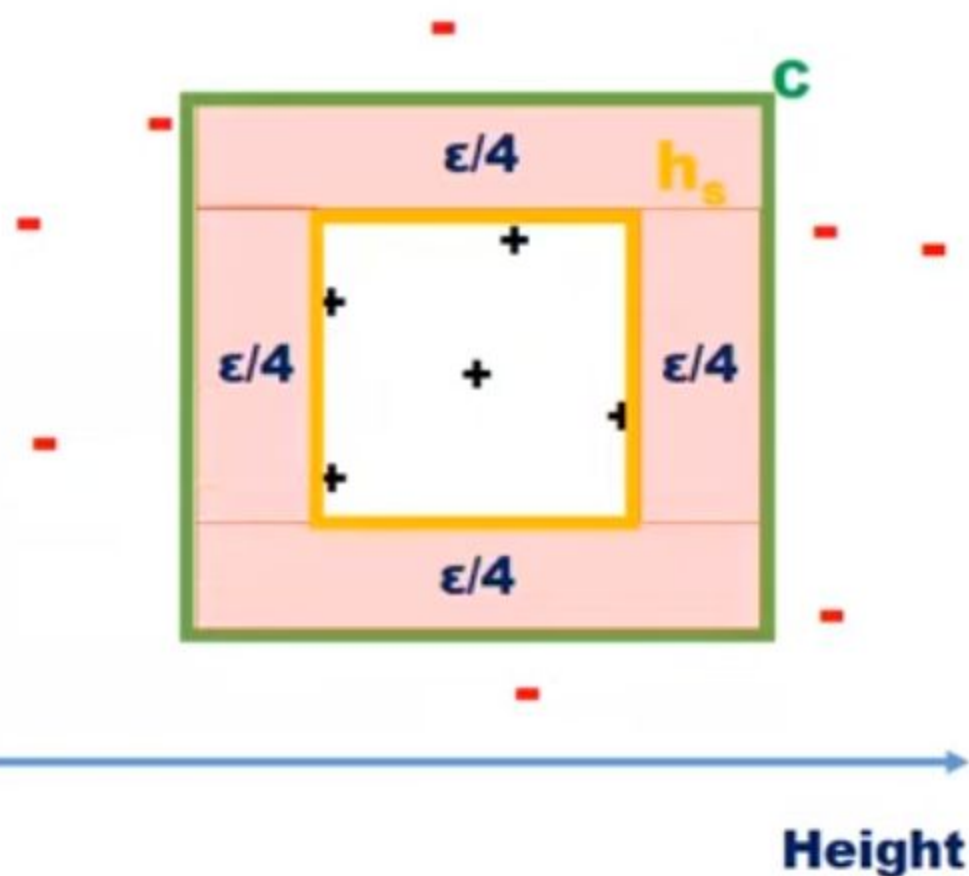
Error Region = Sum of four rectangular strips

Each strip is at most $\epsilon/4$



Error Region = Sum of four rectangular strips

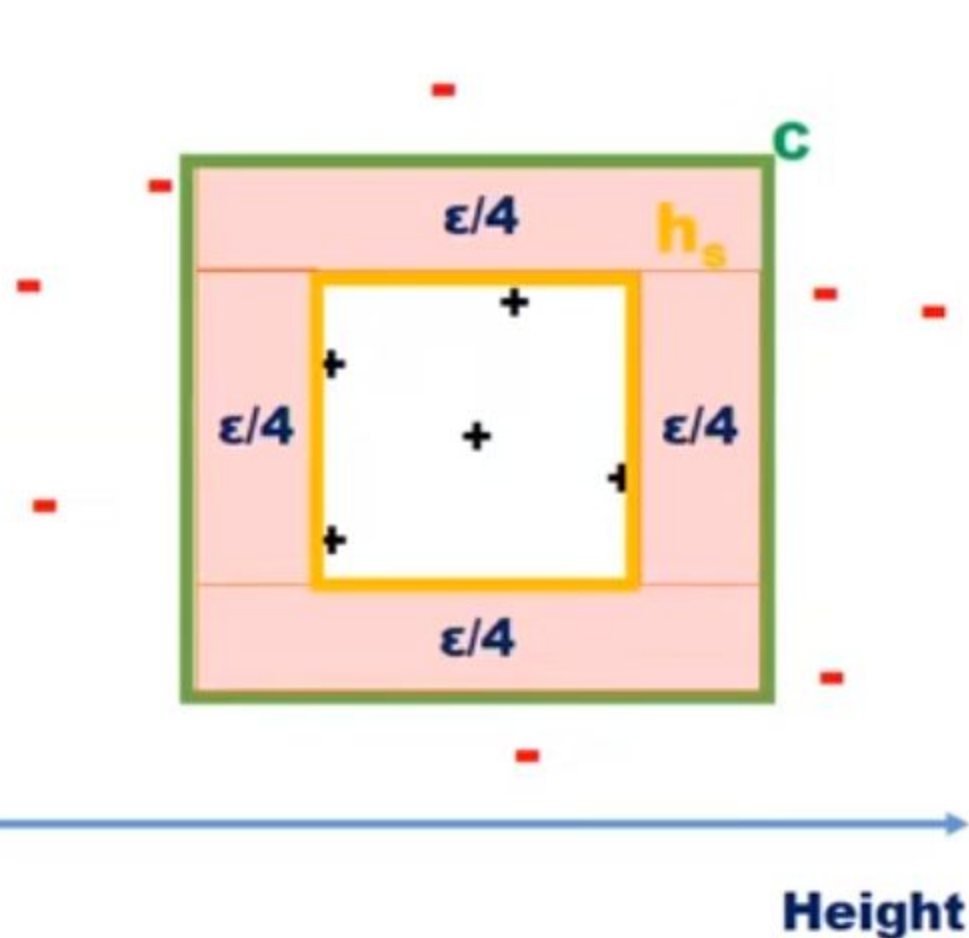
Each strip is at most $\epsilon/4$



Error Region = Sum of four rectangular strips

Each strip is at most $\epsilon/4$

Probability of a positive example falling in any one of the strip(error region)= $\epsilon /4$

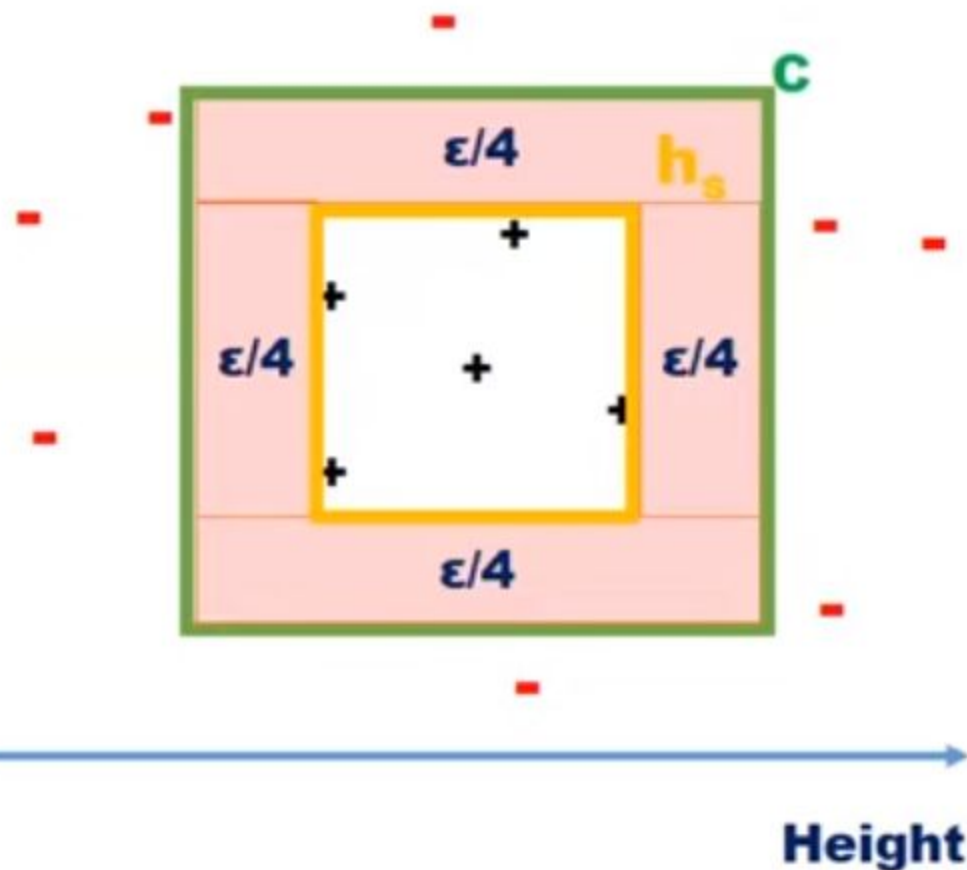


Error Region = Sum of four rectangular strips

Each strip is at most $\epsilon/4$

Probability of a positive example falling in any one of the strip(error region)= $\epsilon/4$

Probability that a randomly drawn positive example misses a strip = $1 - \epsilon/4$



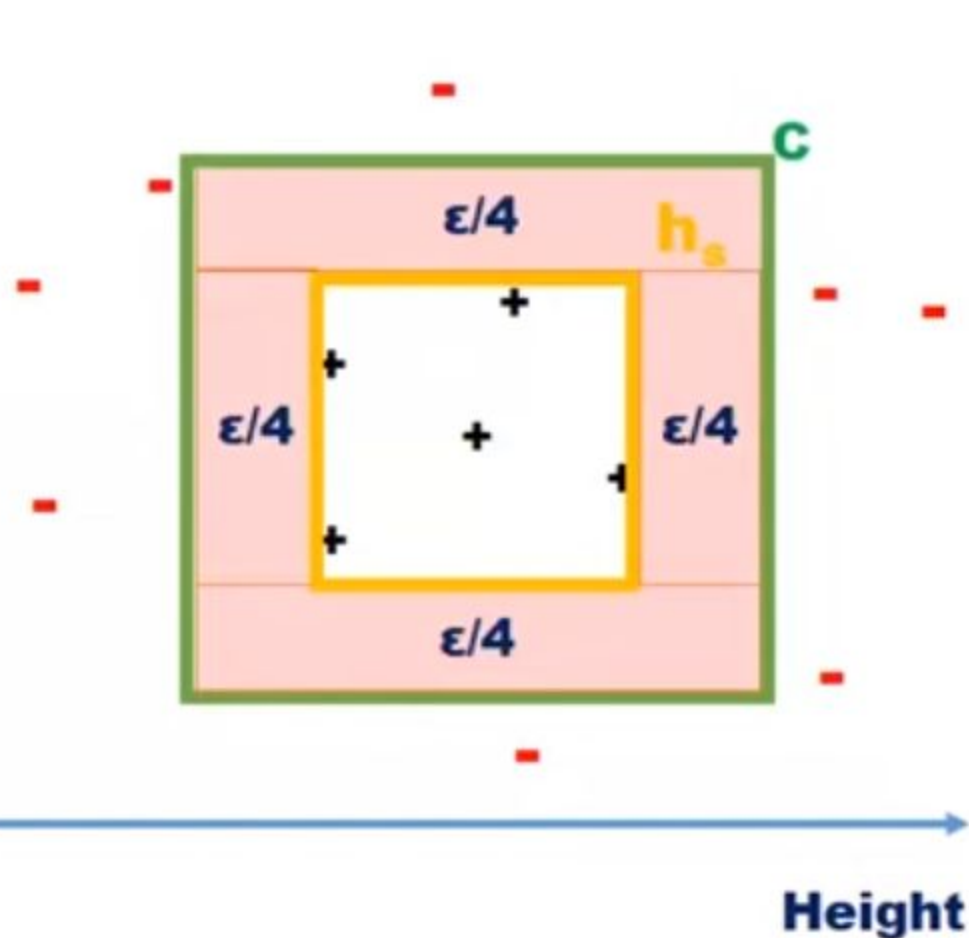
Error Region = Sum of four rectangular strips

Each strip is at most $\epsilon/4$

Probability of a positive example falling in any one of the strip(error region) = $\epsilon/4$

Probability that a randomly drawn positive example misses a strip = $1 - \epsilon/4$

$P(\text{m instances miss a strip}) = (1 - \epsilon/4)^m$



Error Region = Sum of four rectangular strips

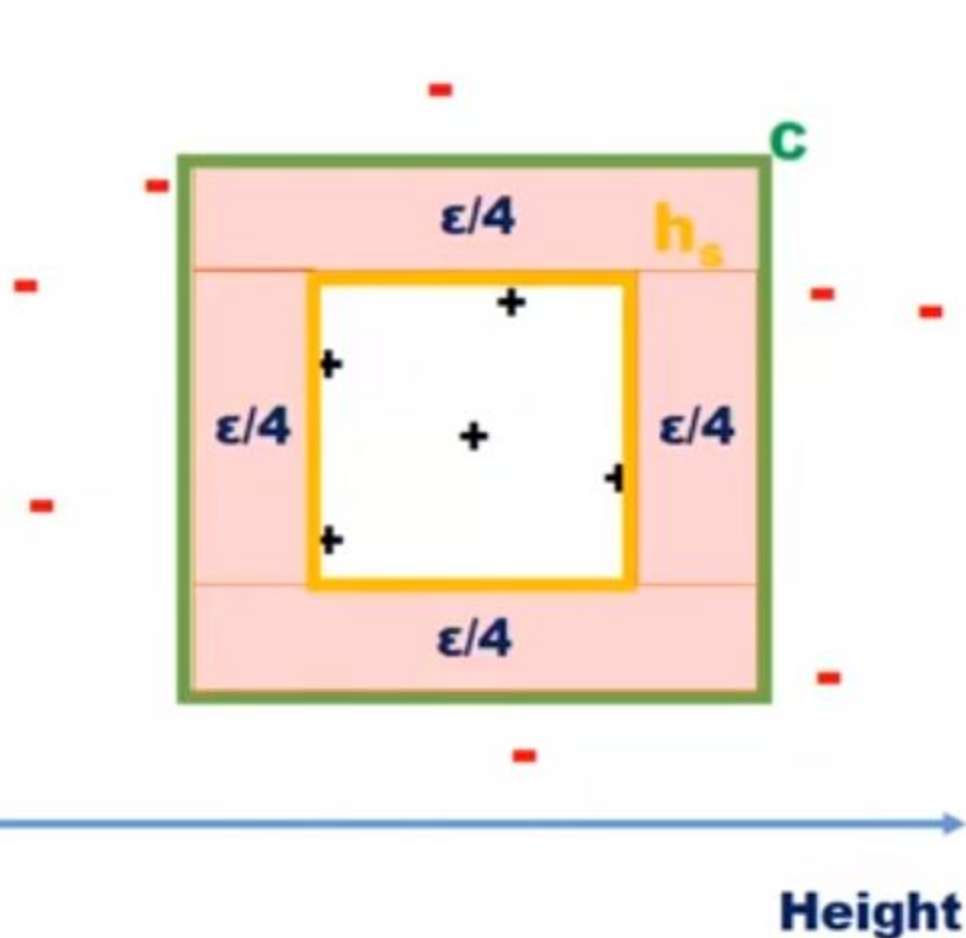
Each strip is at most $\epsilon/4$

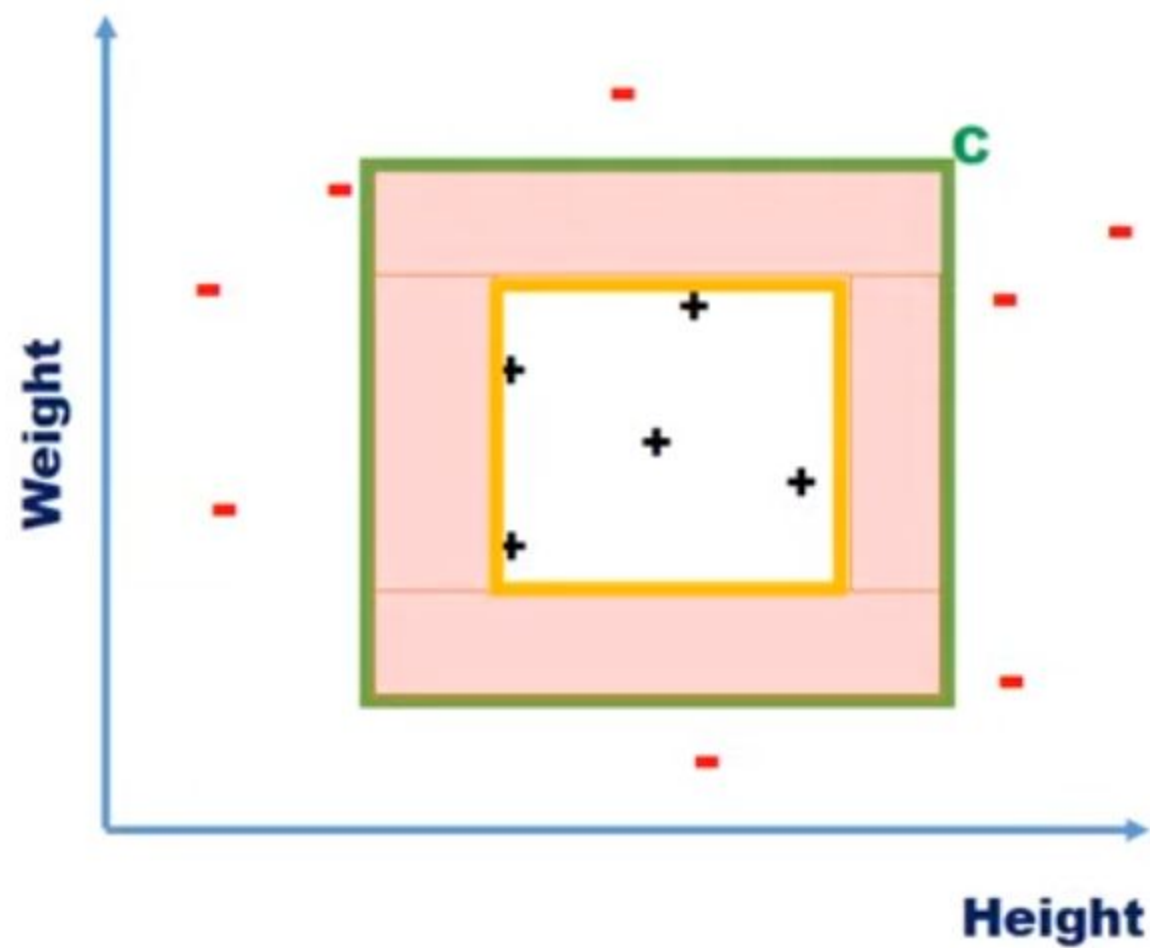
Probability of a positive example falling in any one of the strip(error region)= $\epsilon/4$

Probability that a randomly drawn positive example misses a strip = $1 - \epsilon/4$

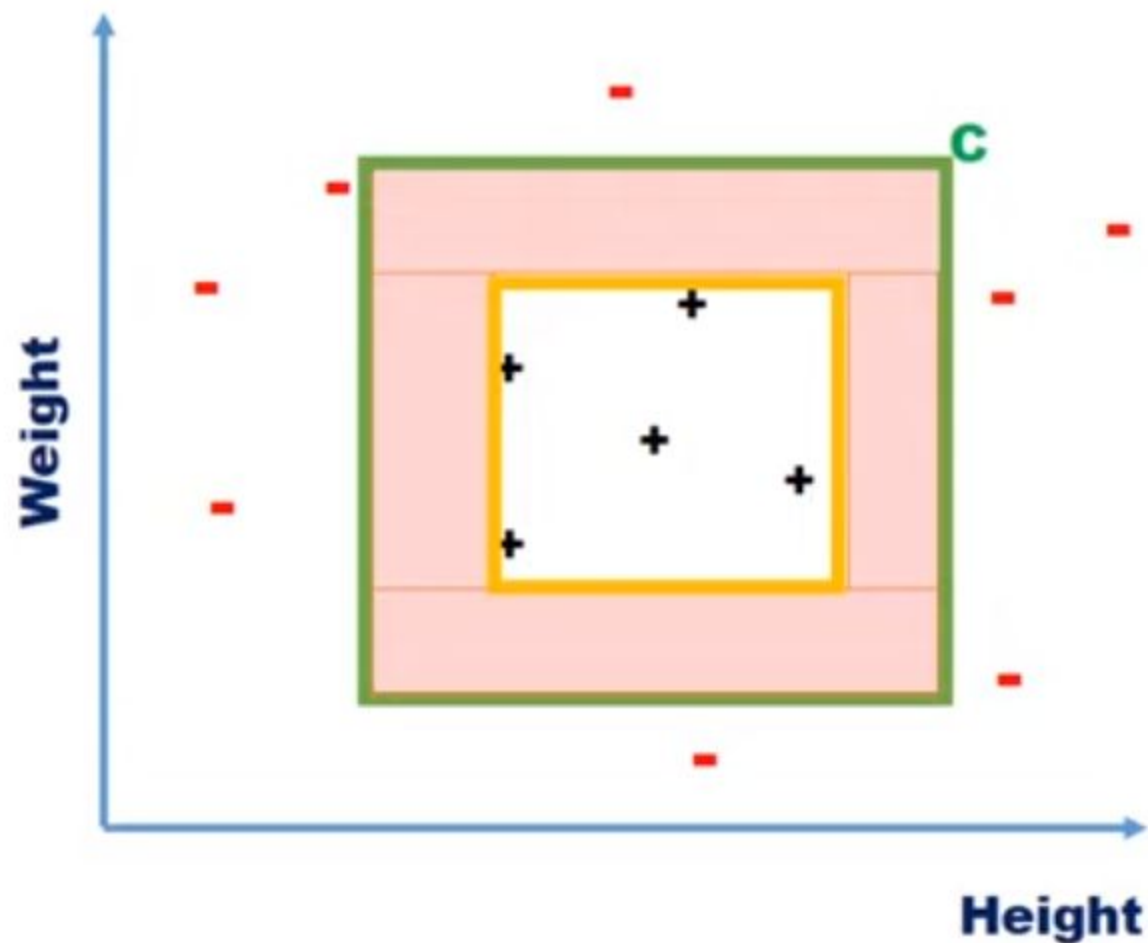
$P(\text{ m instances miss a strip }) = (1 - \epsilon/4)^m$

$P(\text{ m instances miss any strip }) < 4(1 - \epsilon/4)^m$





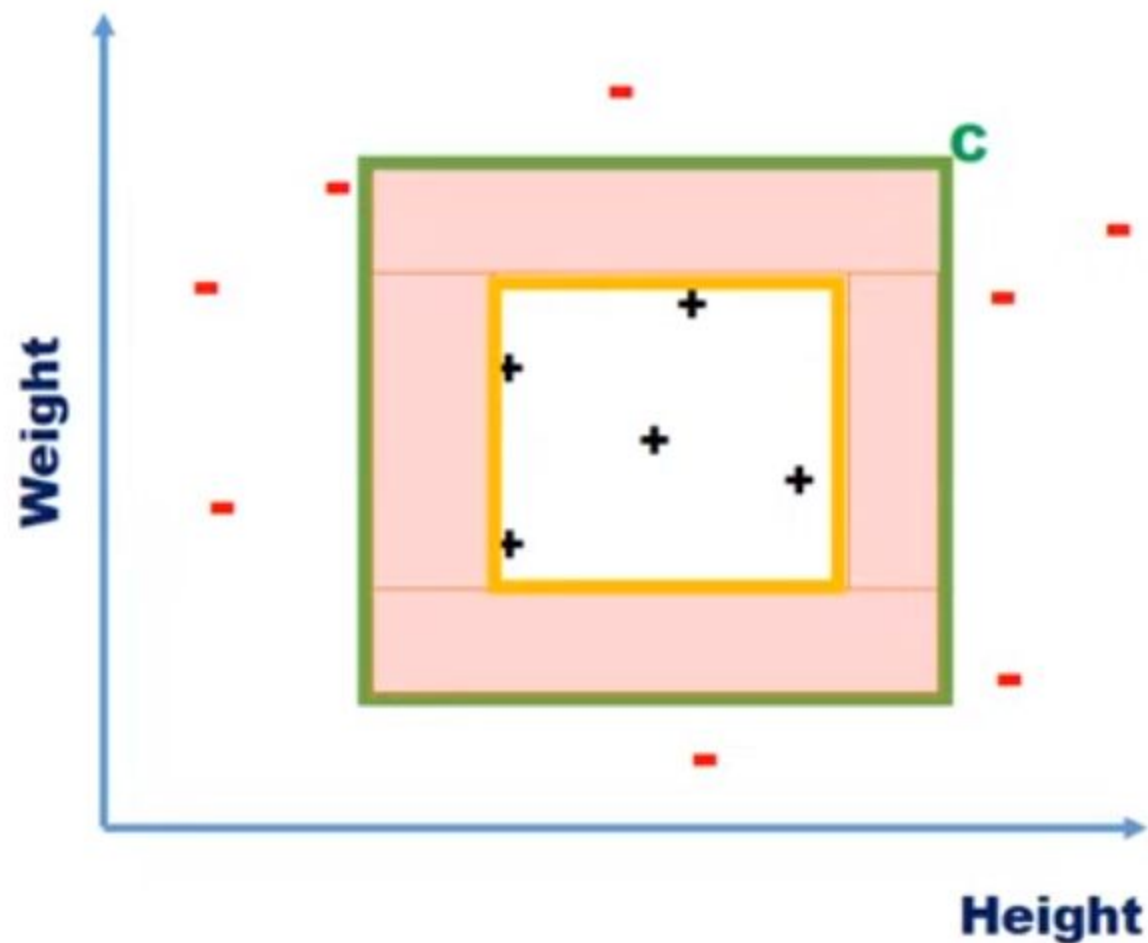
$$4(1 - \varepsilon/4)^m < \delta$$



$$4(1 - \varepsilon/4)^m < \delta$$

Using inequality

$$1-x \leq e^{-x}$$



$$4(1 - \varepsilon/4)^m < \delta$$

Using inequality

$$1-x \leq e^{-x}$$

$$4(1 - \varepsilon/4)^m \leq 4e^{-m\varepsilon/4} < \delta$$

$$m > \frac{4}{\varepsilon} \ln \frac{4}{\delta}$$

That is if we want to have an accuracy of ϵ and confidence of at least $1-\delta$ we have to choose a sample size m such that

$$m > \frac{4}{\epsilon} \ln \frac{4}{\delta}$$

For example, if we want to learn a concept with 99% correctness ($\epsilon = 0.01$) with a probability of 95% ($\delta = 0.05$), the number of examples that need to be shown to our learner to learn a rectangle hypothesis is:

$$m > \frac{4}{0.01} \ln \frac{4}{0.05} = 400 \times 4.38 = 1753$$