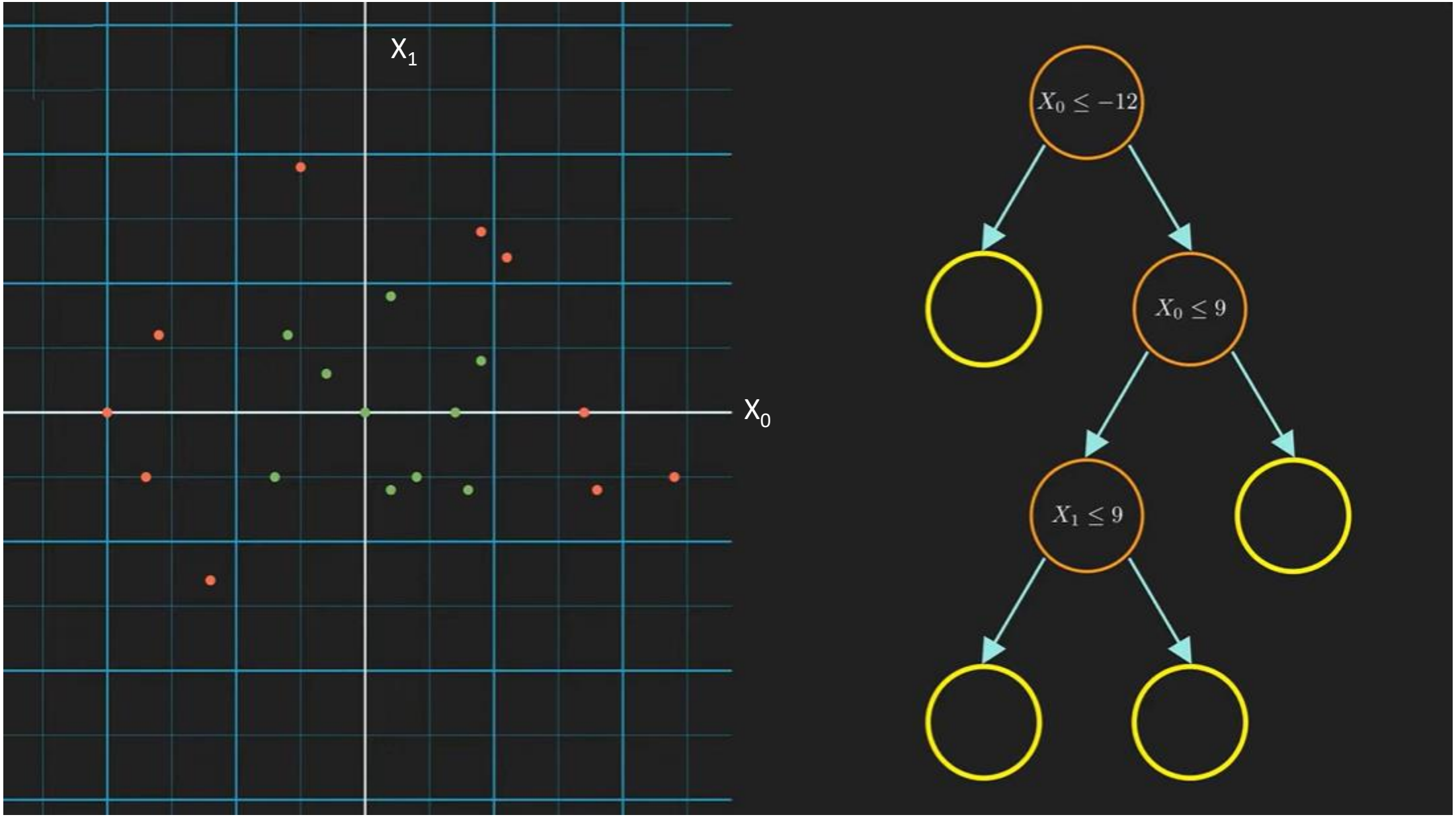
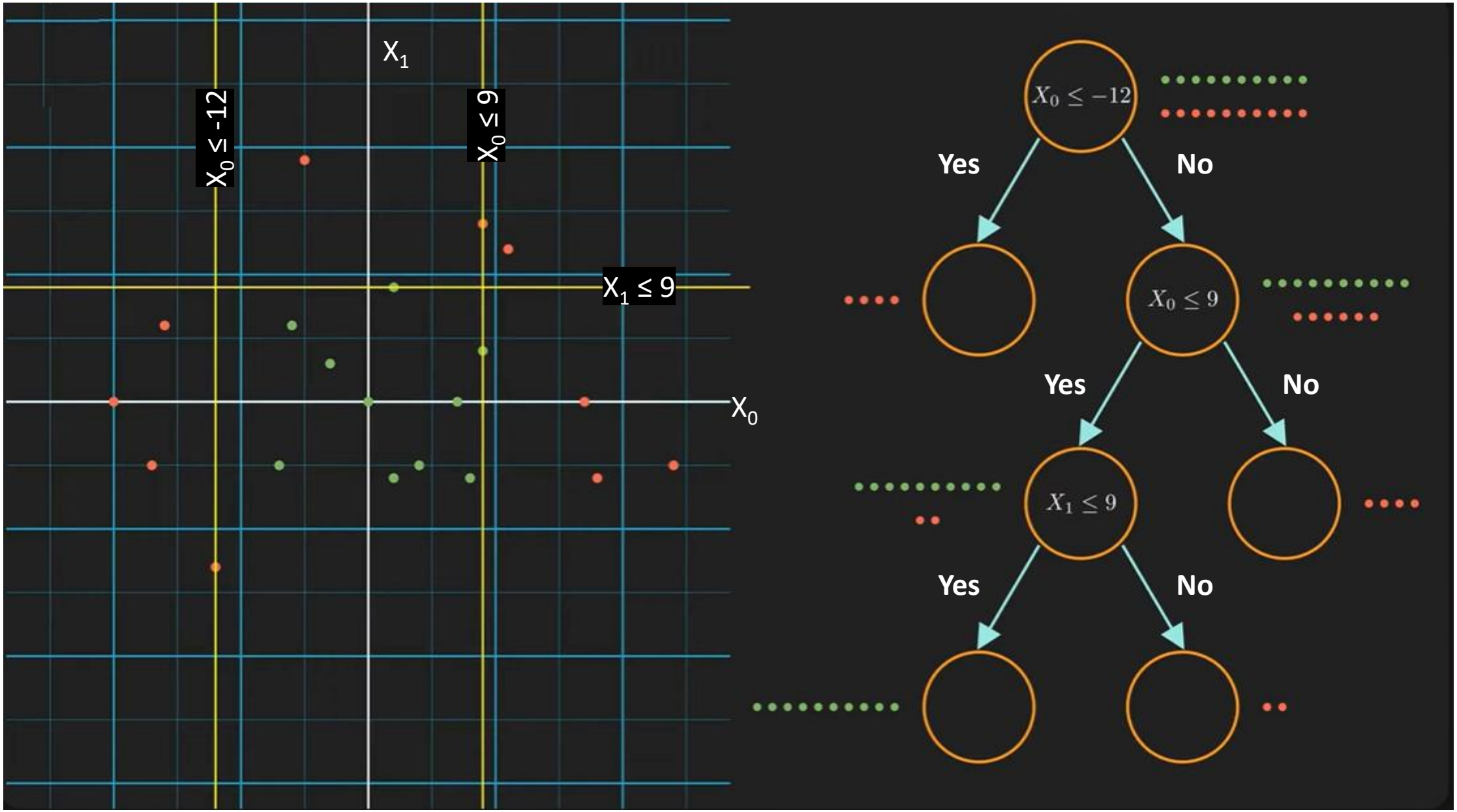
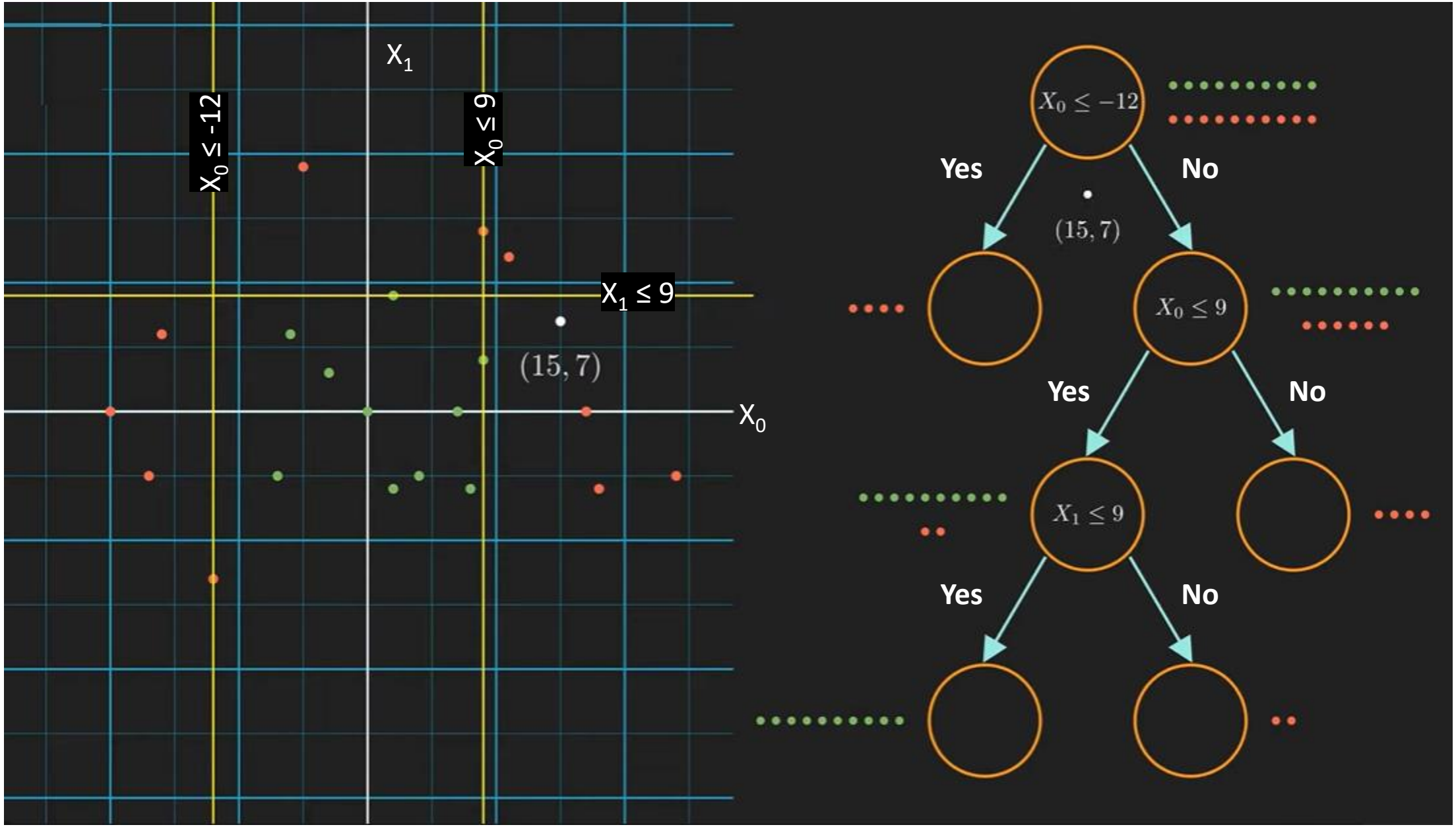
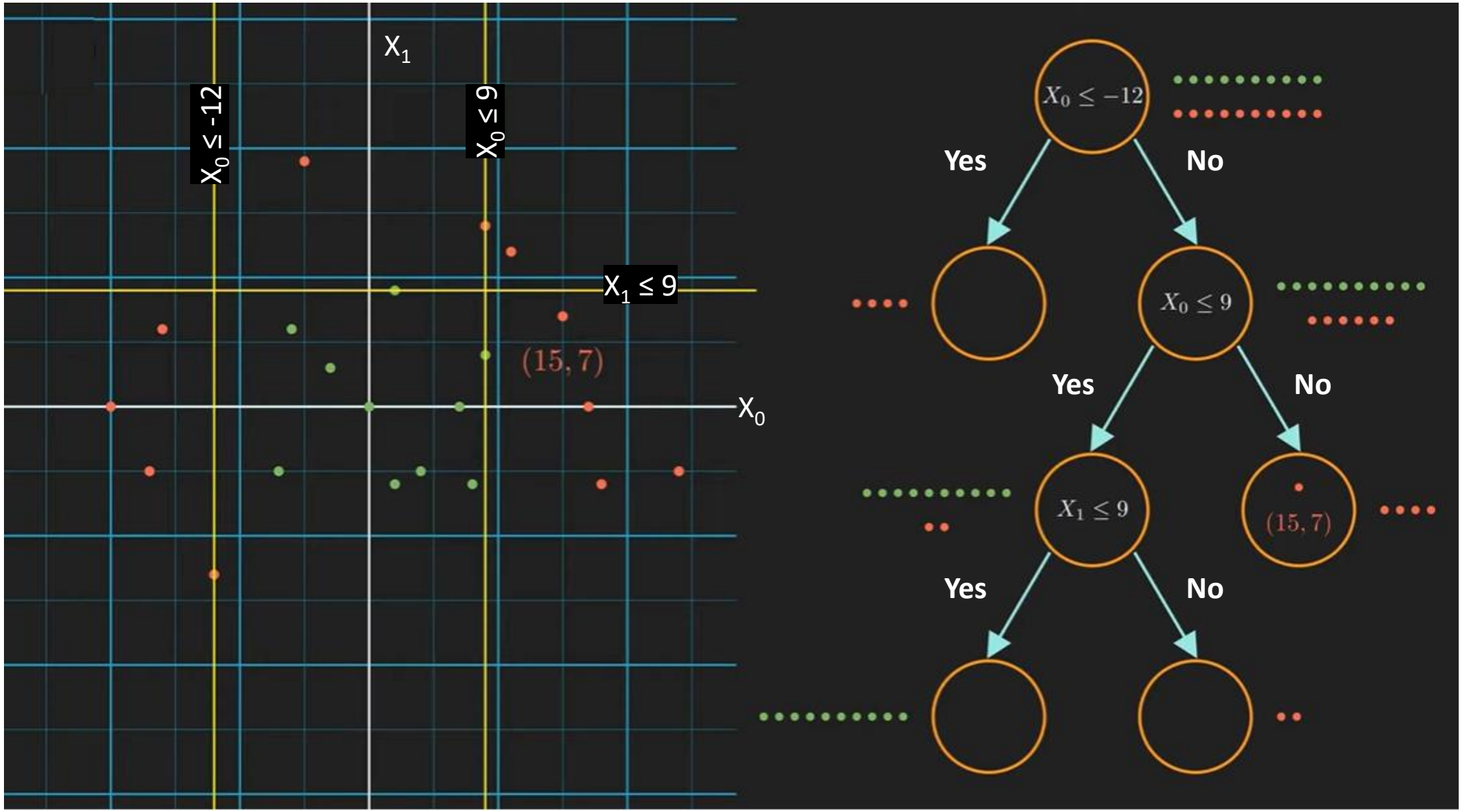
Decision Tree

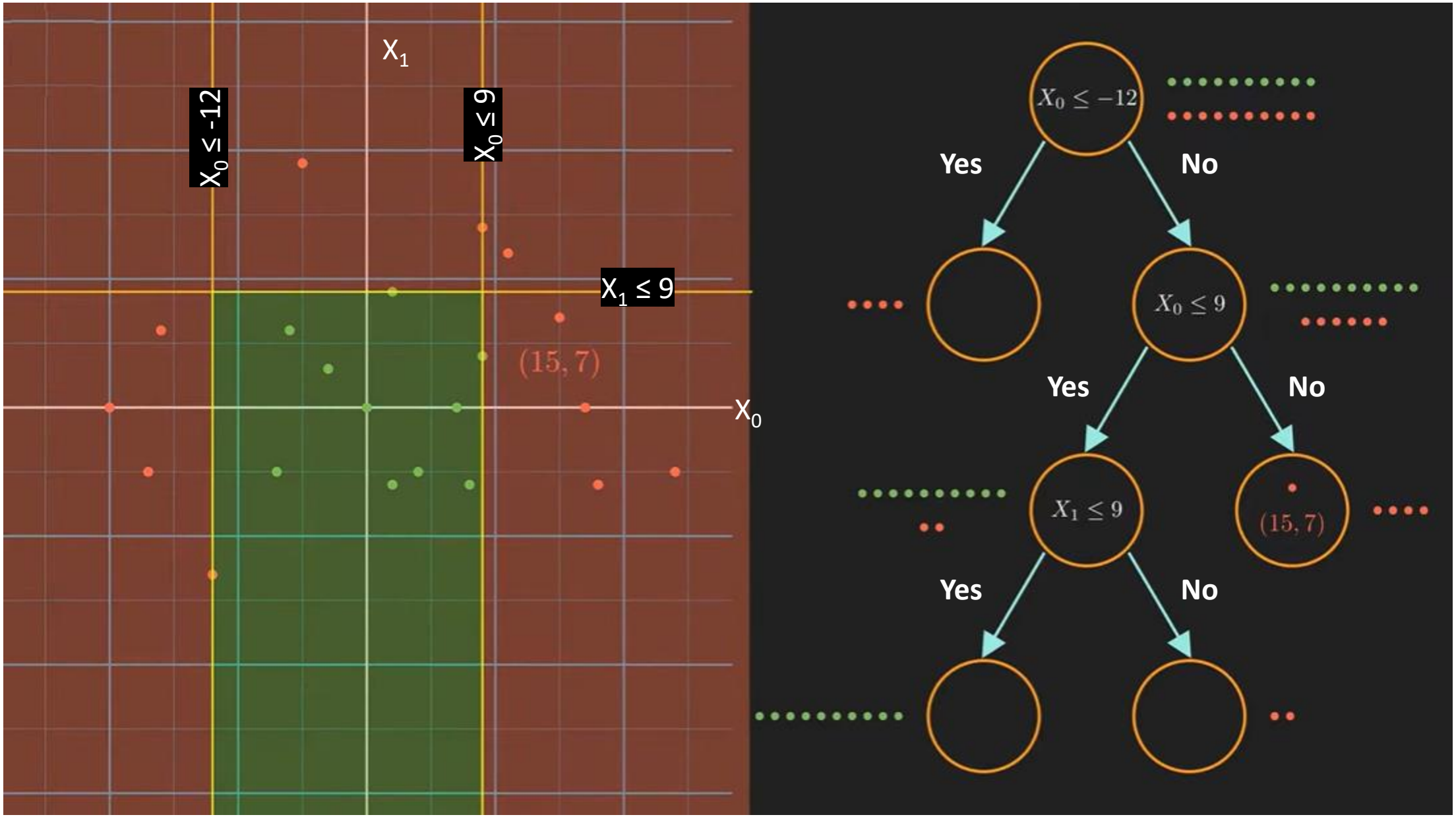
Classification

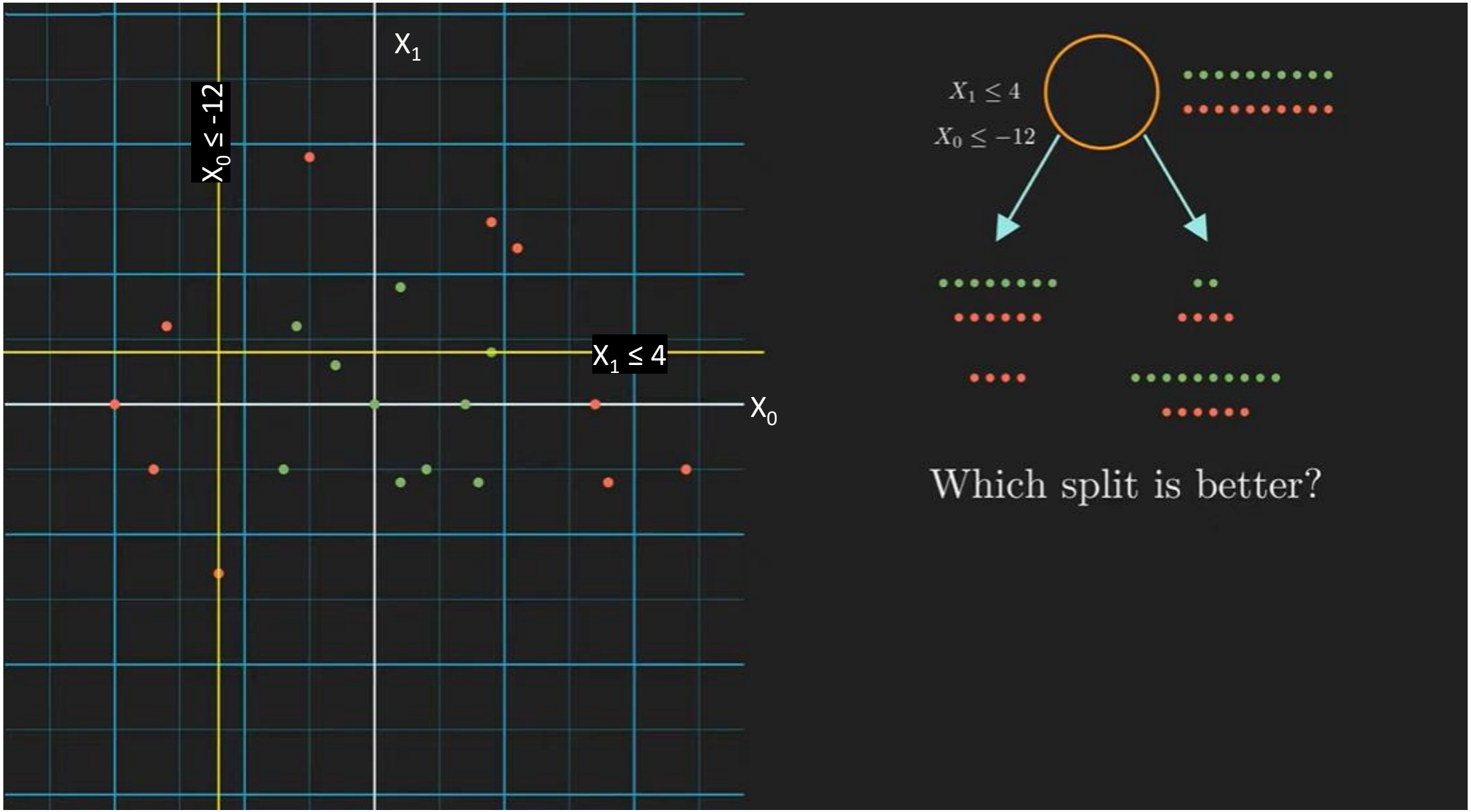

X_1
X_0
$X_0 \leq -12$
$X_0 \leq 9$
$X_1 \leq 9$

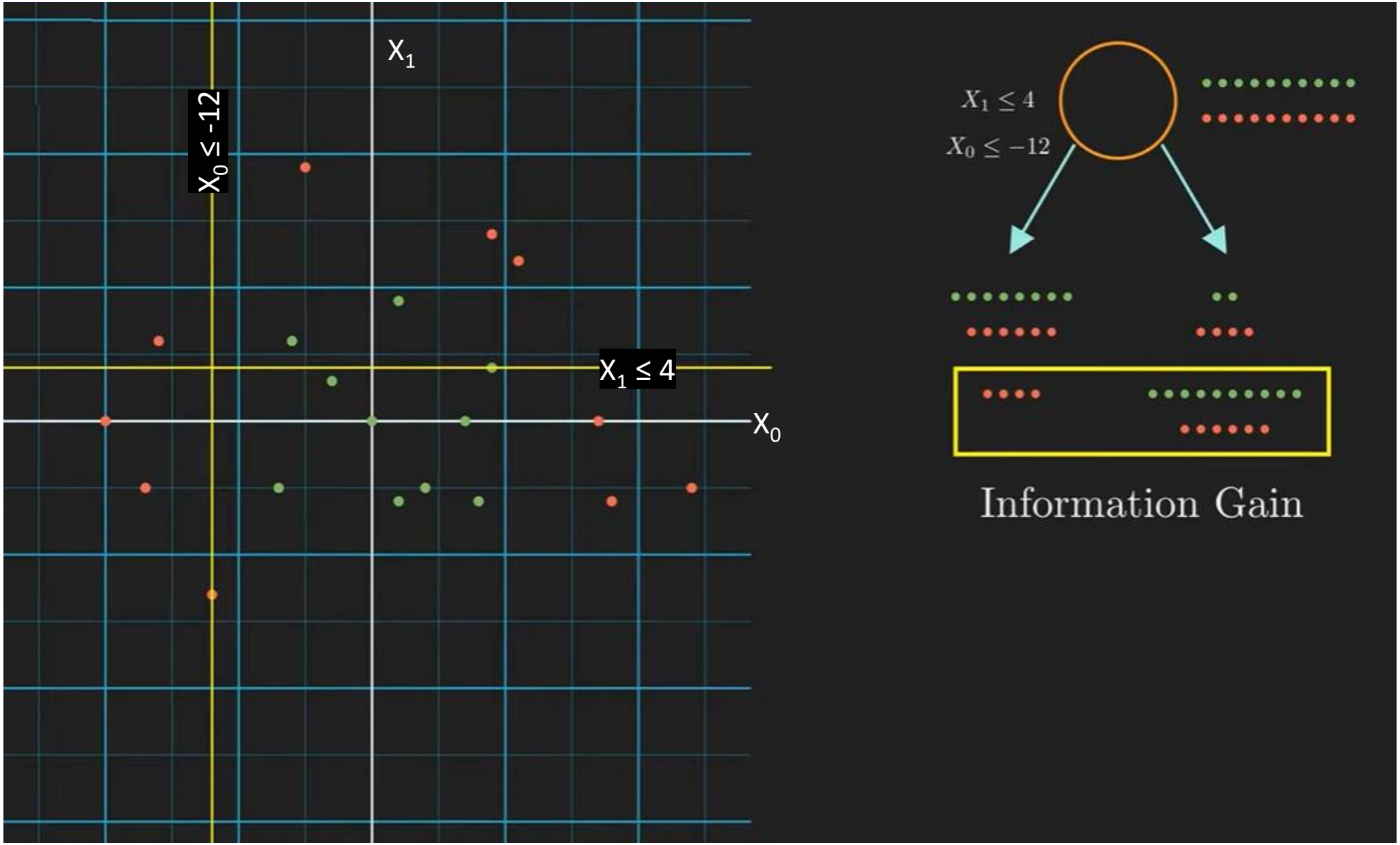

X_1
$X_0 \leq -12$
$X_0 \leq 9$
$X_1 \leq 9$
X_0
$X_0 \leq -12$
Yes
No
$X_0 \leq 9$
Yes
No
$X_1 \leq 9$
Yes
No

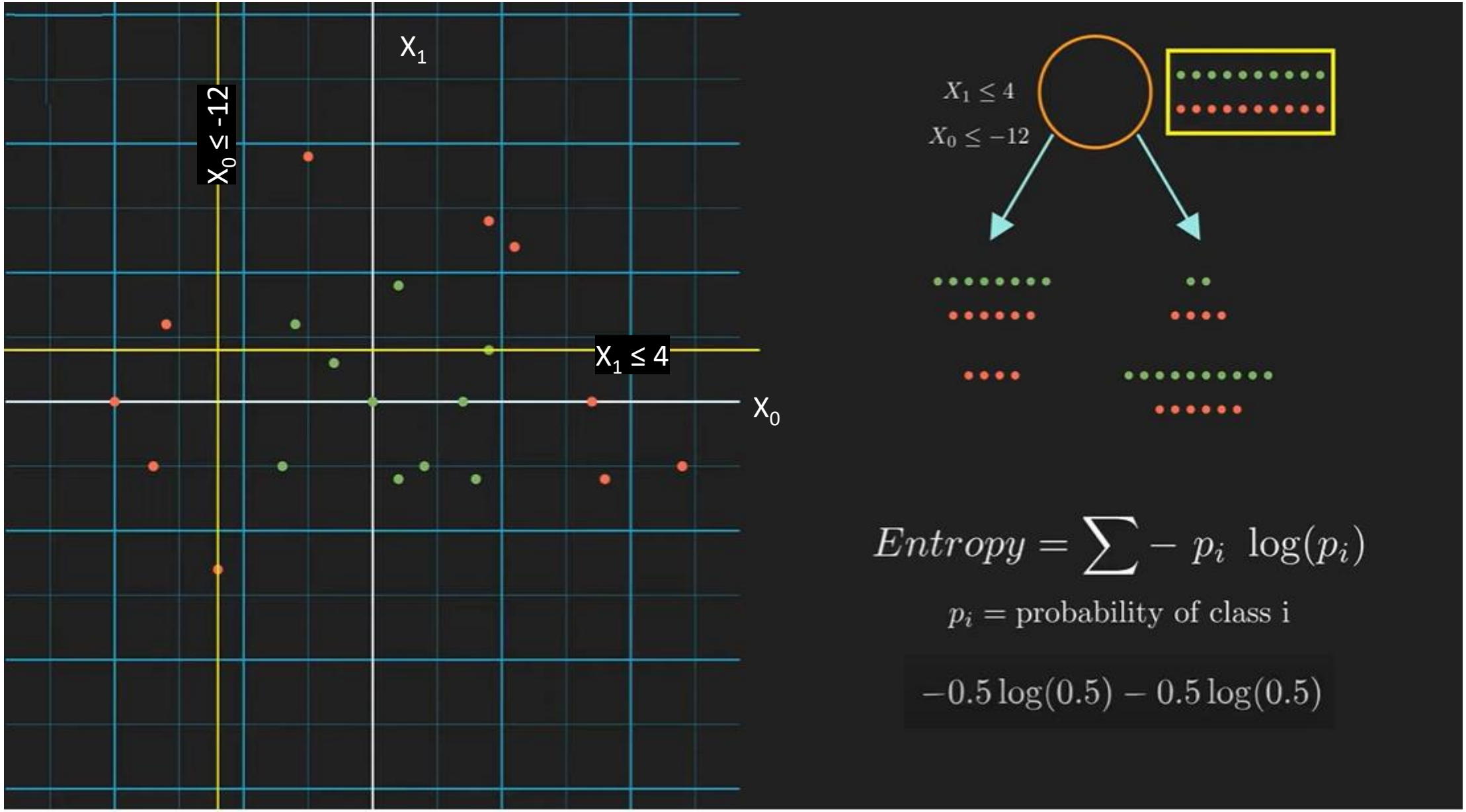

X_1
$X_0 \leq -12$
$X_0 \leq 9$
$X_1 \leq 9$
(15, 7)
X_0
$X_0 \leq -12$
Yes
No
(15, 7)
$X_0 \leq 9$
Yes
No
$X_1 \leq 9$
Yes
No

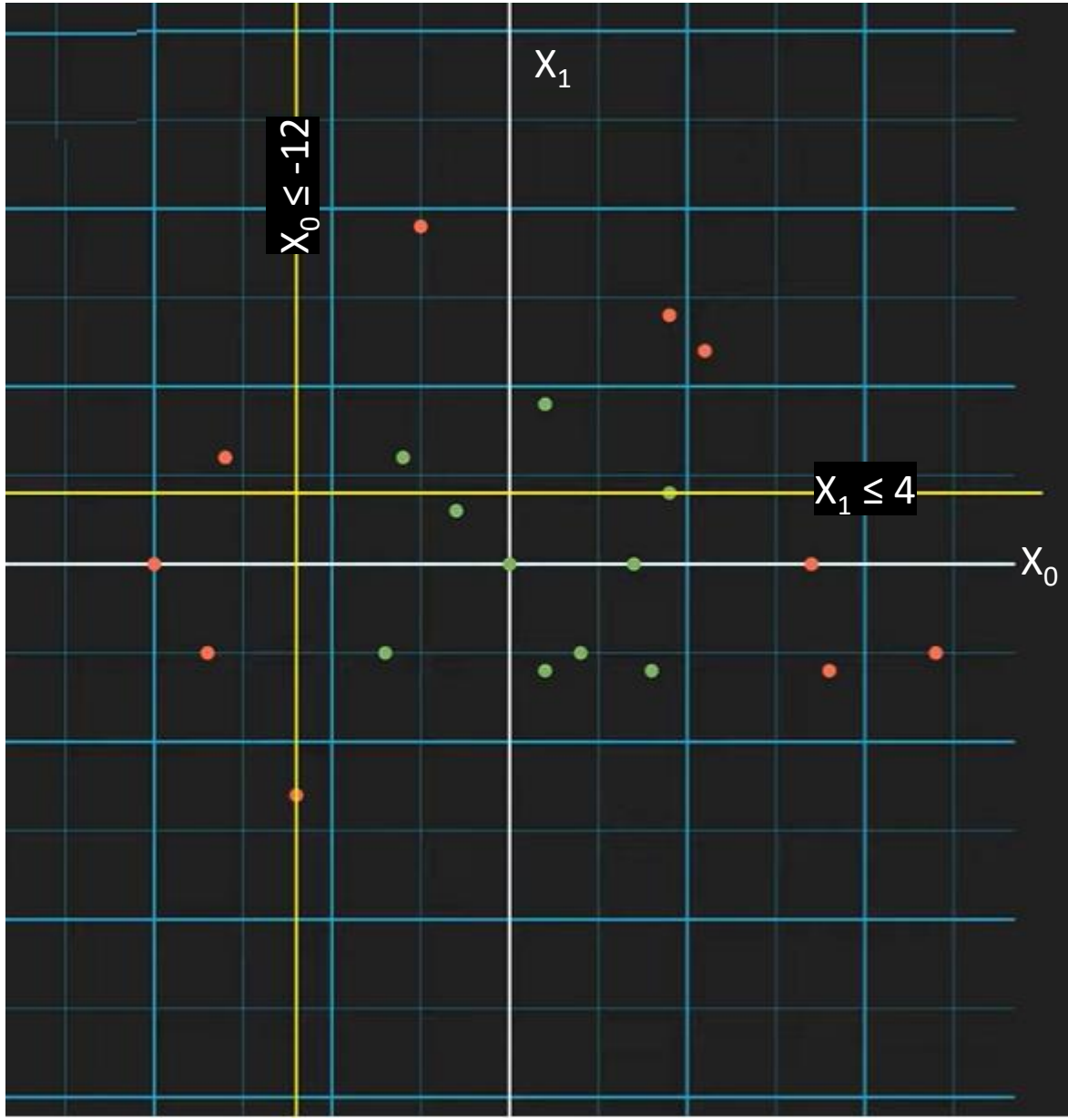

X_1
$X_0 \leq -12$
$X_0 \leq 9$
$X_1 \leq 9$
(15, 7)
X_0
$X_0 \leq -12$
Yes
No
$X_0 \leq 9$
Yes
No
$X_1 \leq 9$
(15, 7)
Yes
No

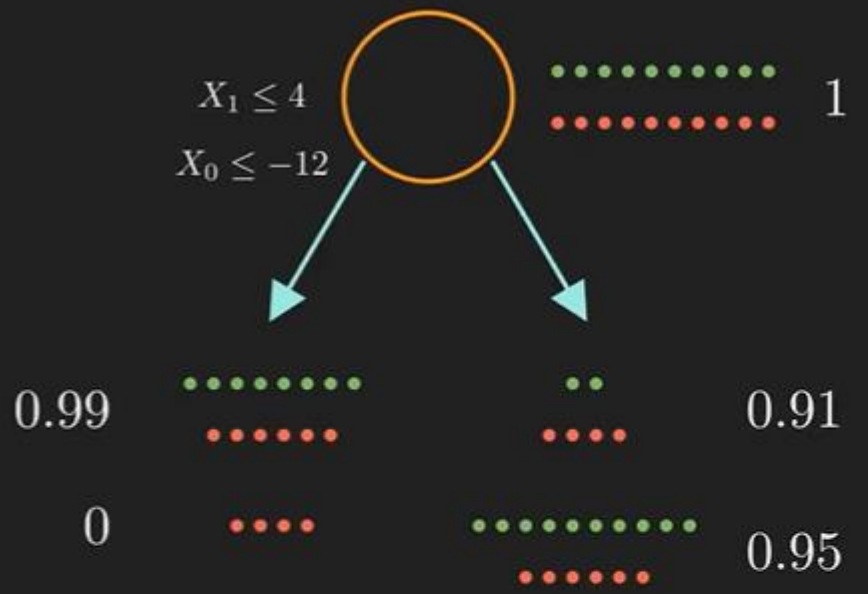

X_1
$X_0 \leq -12$
$X_0 \leq 9$
$X_1 \leq 9$
(15, 7)
X_0
$X_0 \leq -12$
Yes
No
$X_0 \leq 9$
Yes
No
$X_1 \leq 9$
(15, 7)
Yes
No


X_1
$X_0 \leq -12$
$X_1 \leq 4$
X_0
$X_1 \leq 4$
$X_0 \leq -12$
Which split is better?


X_1
$X_0 \leq -12$
$X_1 \leq 4$
X_0
$X_1 \leq 4$
$X_0 \leq -12$
Information Gain


X_1
$X_0 \leq -12$
$X_1 \leq 4$
X_0
$X_1 \leq 4$
$X_0 \leq -12$
$Entropy = \sum - p_i \ \log(p_i)$
p_i = probability of class i
$-0.5\log(0.5) - 0.5\log(0.5)$

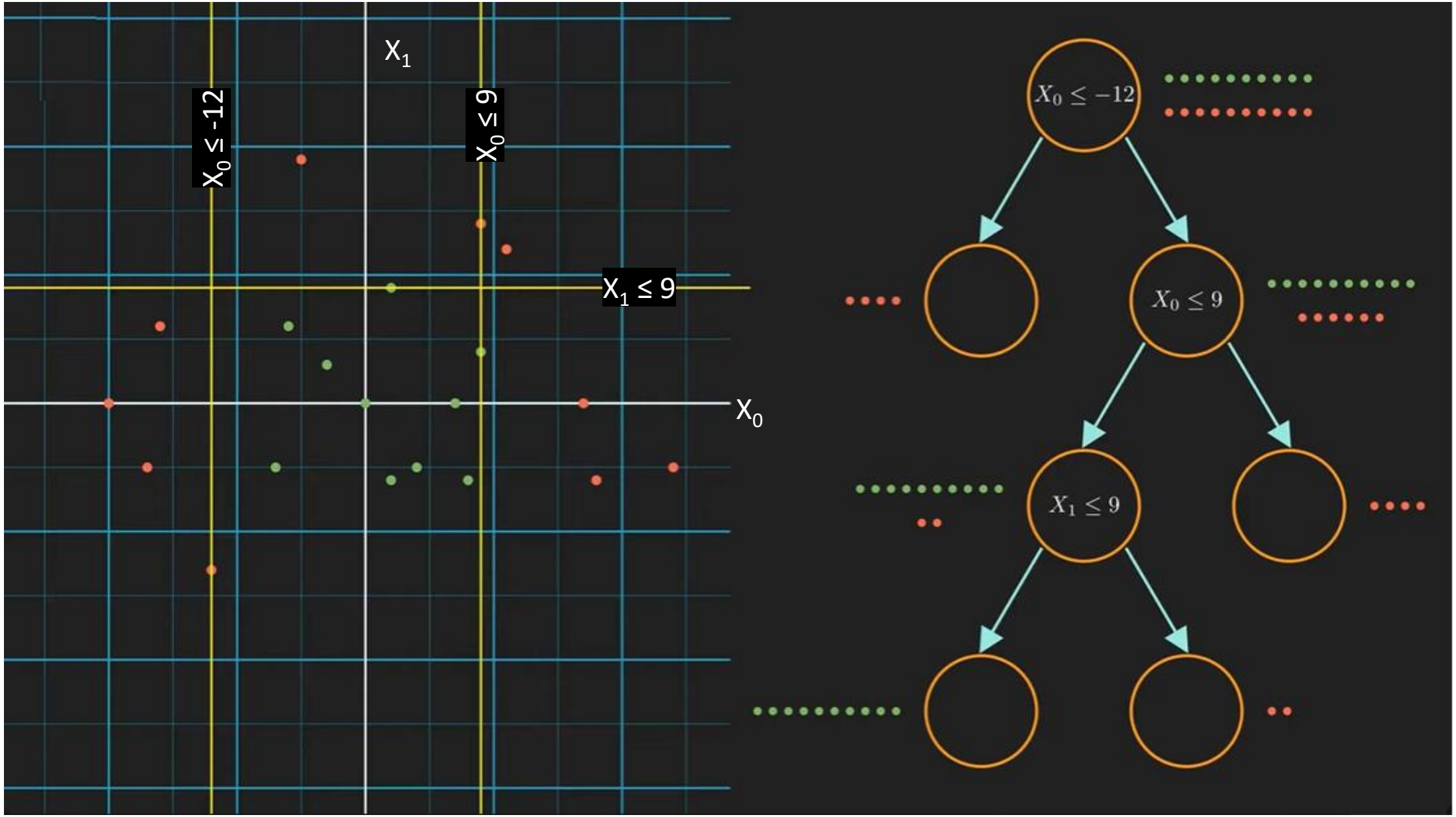




$$\mathrm{IG} = E(parent) - \sum w_i \; E(child_i)$$

$$\mathrm{IG}_1 = 1 - \frac{14}{20} \times .99 - \frac{6}{20} \times .91 = 0.034$$

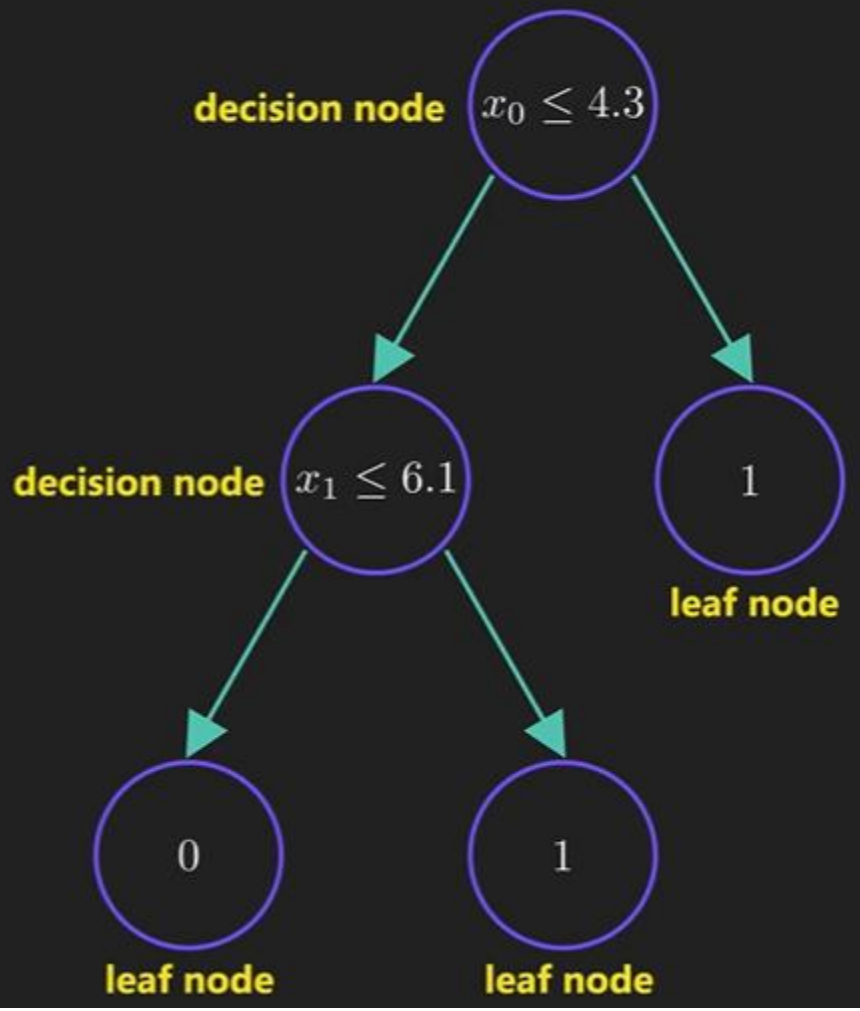
$$\mathrm{IG}_2 = 1 - \frac{4}{20} \times 0 - \frac{16}{20} \times .95 = 0.24$$


X_1
$X_0 \leq -12$
$X_0 \leq 9$
$X_1 \leq 9$
X_0
$X_0 \leq -12$
$X_0 \leq 9$
$X_1 \leq 9$

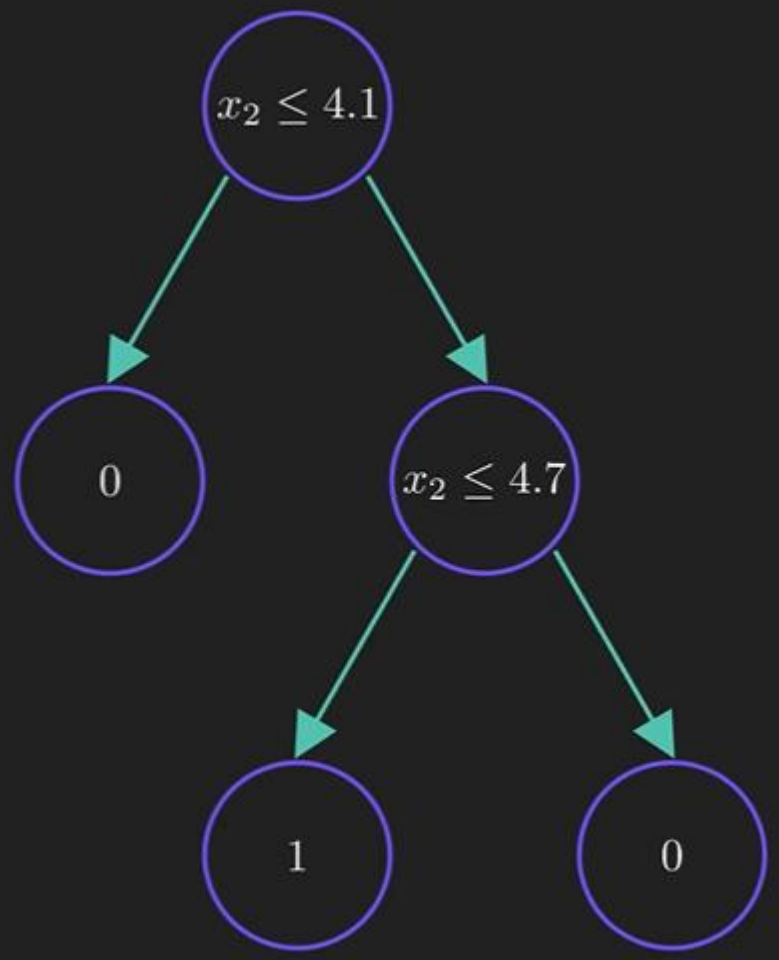
Random Forest

Classification

id	x_0	x_1	x_2	x_3	x_4	y
0	4.3	4.9	4.1	4.7	5.5	0
1	3.9	6.1	5.9	5.5	5.9	0
2	2.7	4.8	4.1	5.0	5.6	0
3	6.6	4.4	4.5	3.9	5.9	1
4	6.5	2.9	4.7	4.6	6.1	1
5	2.7	6.7	4.2	5.3	4.8	1



id	x_0	x_1	x_2	x_3	x_4	y
0	4.3	4.9	4.1	4.7	5.5	0
1	6.5	4.1	5.9	5.5	5.9	0
2	2.7	4.8	4.1	5.0	5.6	0
3	6.6	4.4	4.5	3.9	5.9	1
4	6.5	2.9	4.7	4.6	6.1	1
5	2.7	6.7	4.2	5.3	4.8	1



id	x_0	x_1	x_2	x_3	x_4	y
0	4.3	4.9	4.1	4.7	5.5	0
1	3.9	6.1	5.9	5.5	5.9	0
2	2.7	4.8	4.1	5.0	5.6	0
3	6.6	4.4	4.5	3.9	5.9	1
4	6.5	2.9	4.7	4.6	6.1	1
5	2.7	6.7	4.2	5.3	4.8	1

id	id	id	id
2	2	4	3
0	1	1	3
2	3	3	2
4	1	0	5
5	4	0	1
5	4	2	2

Bootstrapped Datasets

id	x_0	x_1	x_2	x_3	x_4	y
0	4.3	4.9	4.1	4.7	5.5	0
1	3.9	6.1	5.9	5.5	5.9	0
2	2.7	4.8	4.1	5.0	5.6	0
3	6.6	4.4	4.5	3.9	5.9	1
4	6.5	2.9	4.7	4.6	6.1	1
5	2.7	6.7	4.2	5.3	4.8	1

id
2
0
2
4
5
5

x_0, x_1

id
2
1
3
1
4
4

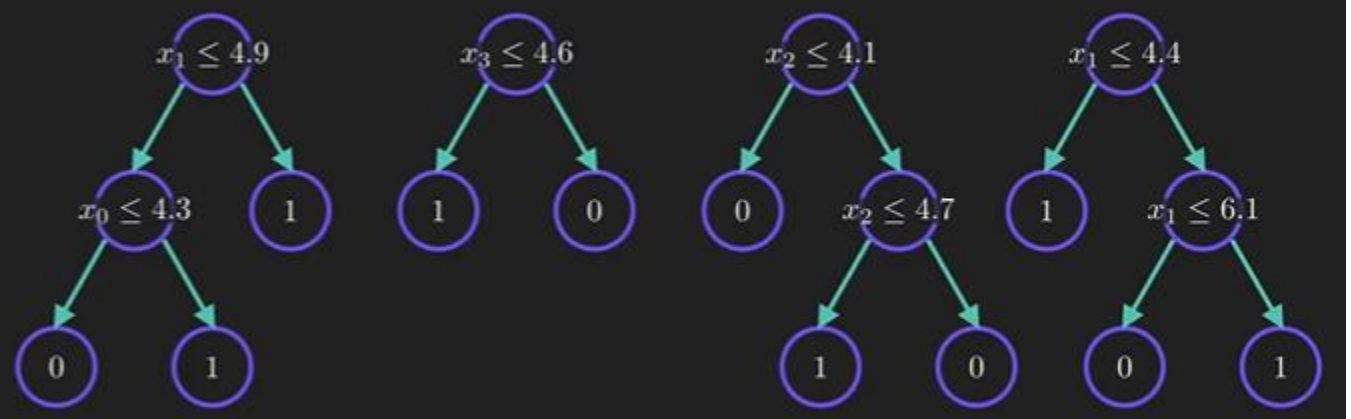
x_2, x_3

id
4
1
3
0
0
2

x_2, x_4

id
3
3
2
5
1
2

x_1, x_3



id	x_0	x_1	x_2	x_3	x_4	y
0	4.3	4.9	4.1	4.7	5.5	0
1	3.9	6.1	5.9	5.5	5.9	0
2	2.7	4.8	4.1	5.0	5.6	0
3	6.6	4.4	4.5	3.9	5.9	1
4	6.5	2.9	4.7	4.6	6.1	1
5	2.7	6.7	4.2	5.3	4.8	1

id
2
0
2
4
5
5

x_0, x_1

id
2
1
3
1
4
4

x_2, x_3

id
4
1
3
0
0
2

x_2, x_4

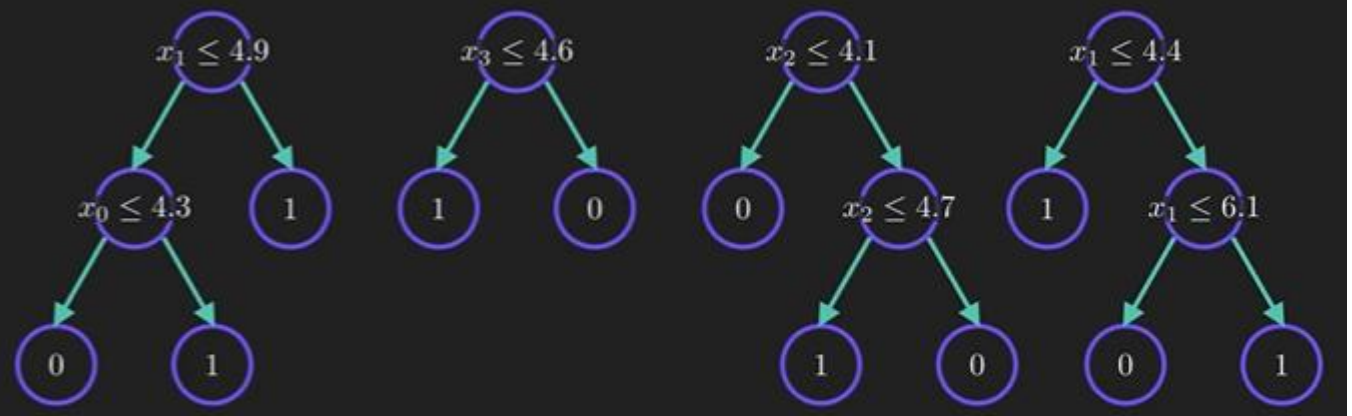
id
3
3
2
5
1
2

x_1, x_3

2.8	6.2	4.3	5.3	5.5

Bootstrap + Aggregating

(Bagging)



1 0 1 1

Why is it called random?

Why Bootstrapping and Feature Selection?

How many features to consider?

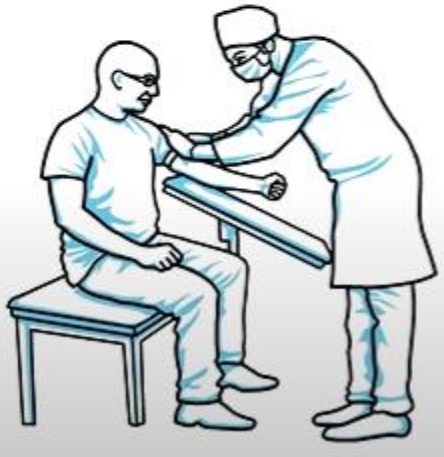
How to use this for regression?

Sensitivity and Specificity

SENSITIVITY AND SPECIFICITY

What are they?

How are they calculated?



TESTS → not perfect

Presence or absence of disease

How good is the test?

DISEASE

TEST	Disease	No disease
+	✓ True positive (TP)	✗ False positive (FP)
-	✗ False negative (FN)	✓ True negative (TN)

not always perfect

False positive (FP) → more testing
treatment
psychological effects
Cost
Risk

False negative (FN) → delays in diagnosis and treatment
false sense of security
risky behaviour
legal consequences

$$\uparrow \text{SENSITIVITY} = \frac{TP \uparrow}{TP+FN \downarrow}$$

$$\uparrow \text{SPECIFICITY} = \frac{TN \uparrow}{FP+TN \downarrow}$$

DISEASE

TEST	Disease	No disease
+	90 True positive (TP)	100 False positive (FP)
−	10 False negative (FN)	400 True negative (TN)
600 people:	100	500

$$\text{SENSITIVITY} = \frac{TP}{TP+FN} = \frac{90}{100} = 0.9\ (90\%)$$

$$\text{SPECIFICITY} = \frac{TN}{FP+TN} = \frac{400}{500} = 0.8\ (80\%)$$

High sensitivity → screening tests (low false negatives)

High specificity → confirmatory tests (low false positives)

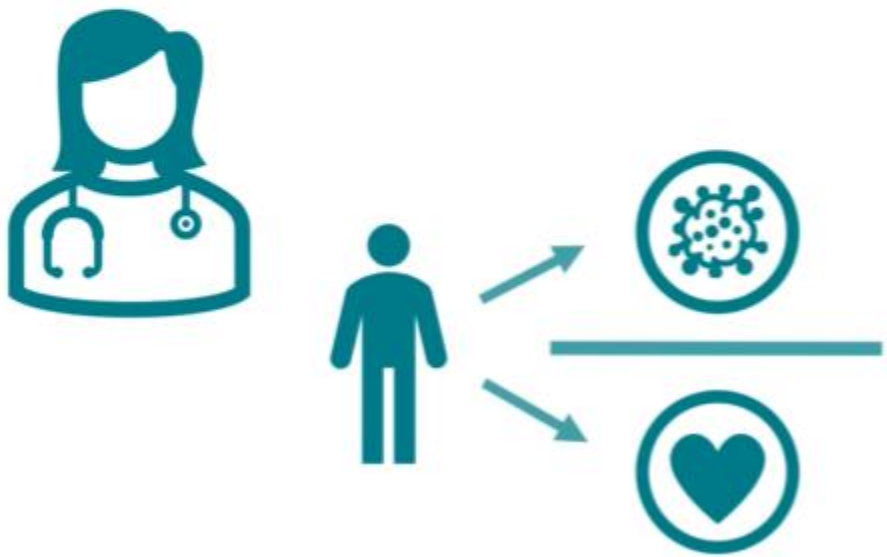
PERFECT TEST

SENSITIVITY 100%

SPECIFICITY 100%

ROC and AUC

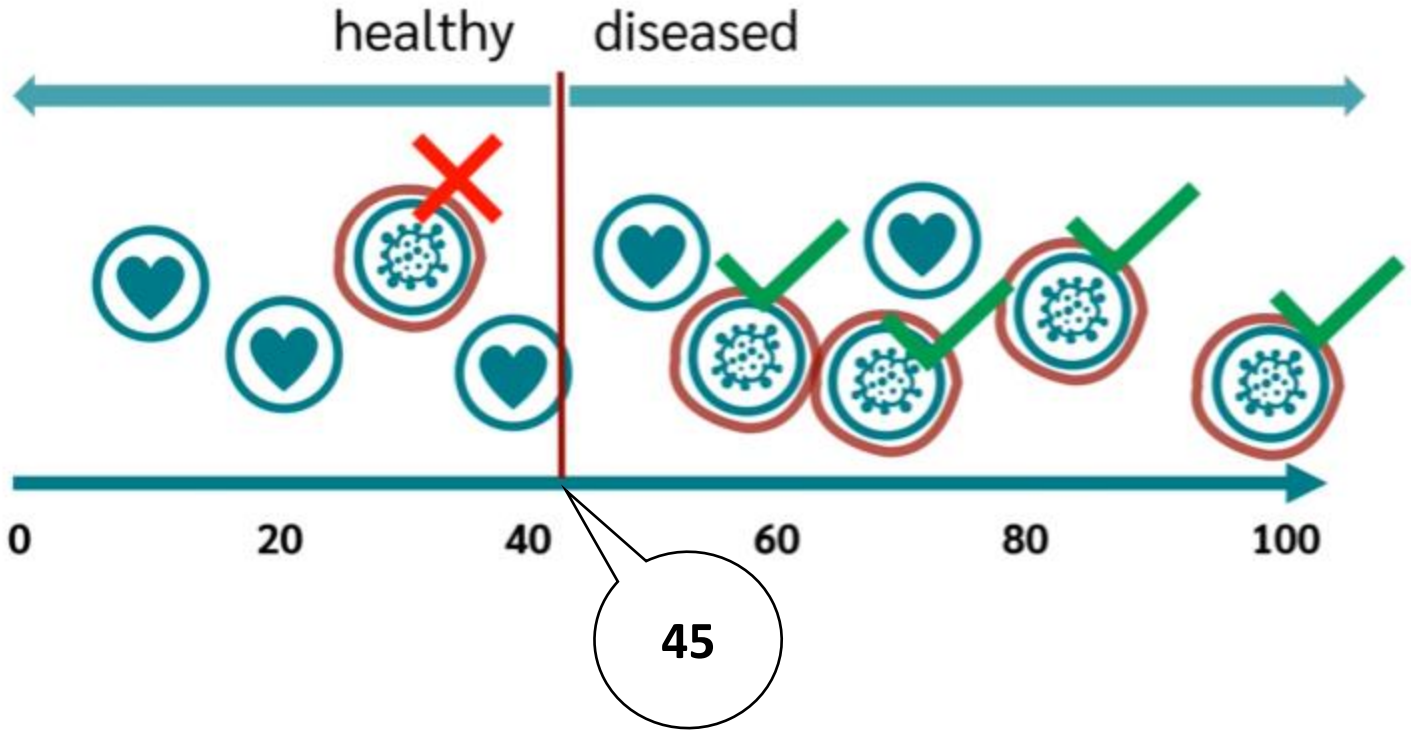
We would like to **classify** based on a **screening,** whether a person **has cancer** or **not.**



For this, we get **data from 10 people** about how high the **blood level** is and whether there is a **disease or not**.

This value is called the **True Positive Rate (TPR)** and is equal to the **sensitivity.**

So, we **correctly** classified **4 out of 5 as "diseased".**

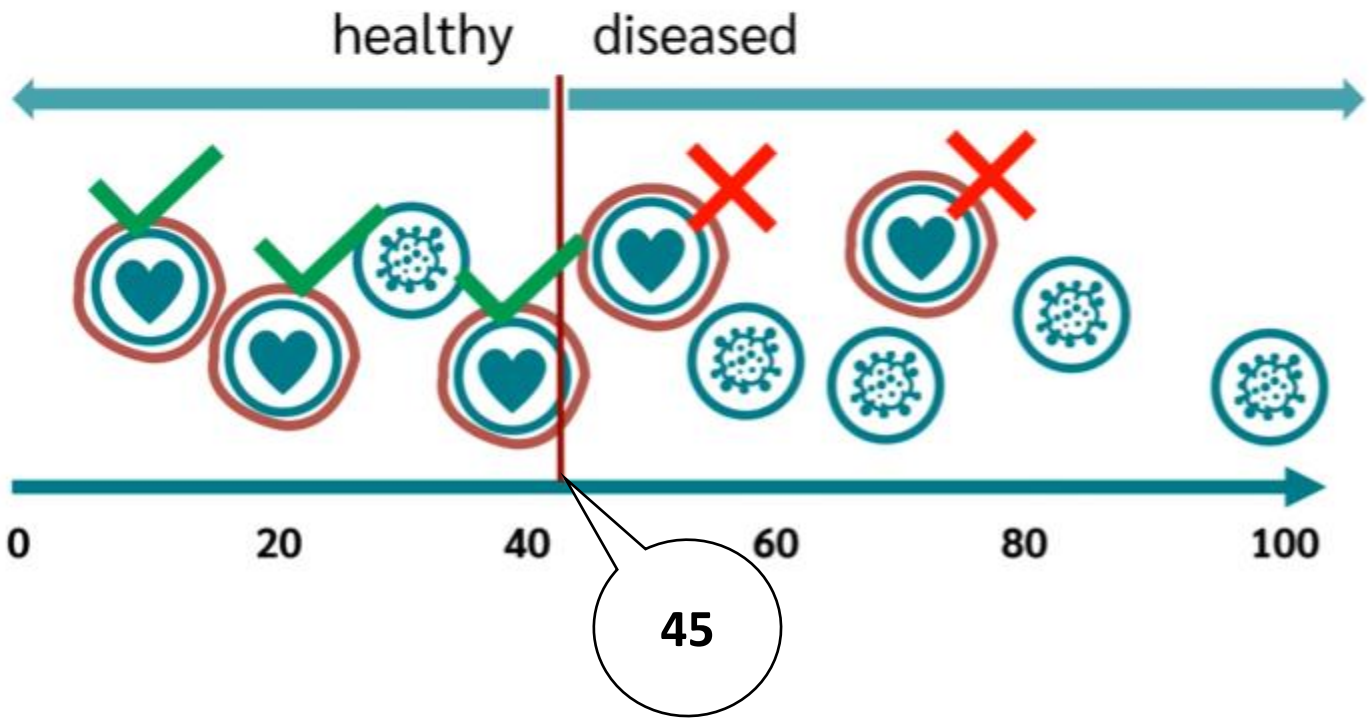


And it is precisely these **two values** that are then plotted on the **ROC Curve**.

So, for a **threshold of 45**, we get a **True Positive Rate** of **4/5** i.e. **0.8** and a **False Positive Rate** of **3/5** i.e. **0.6**.

So, we misclassified **2 out of 5 as "diseased“.** This value is called the **False Positive Rate (FPR)** and is 1 - specificity.

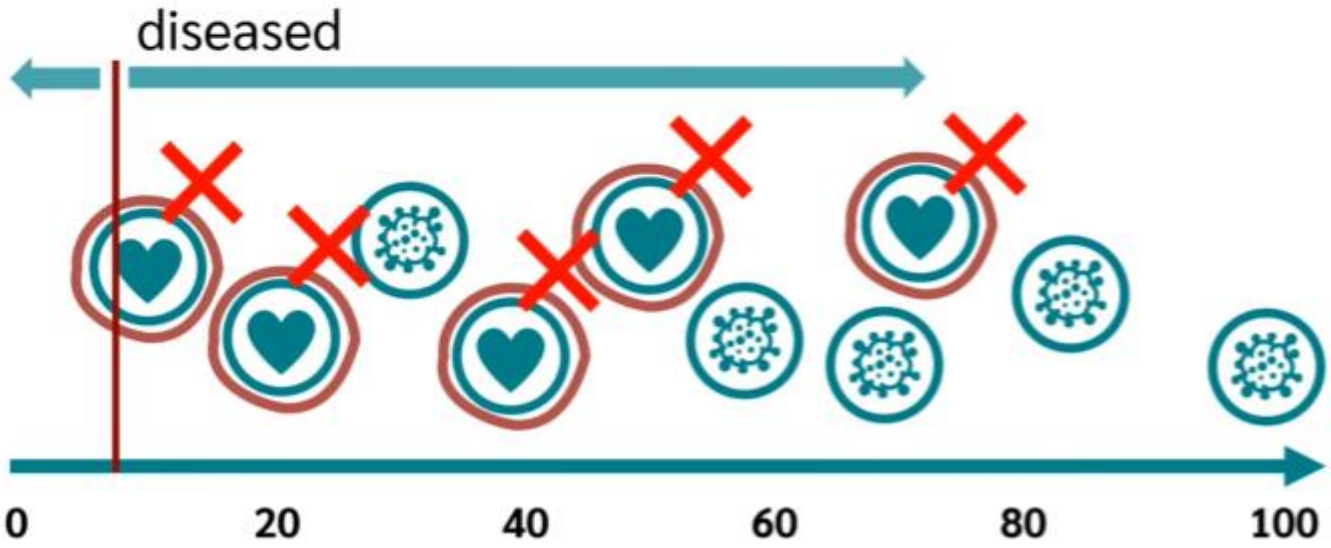
We can now calculate for **each threshold**, the **True Positive Rate** and the **False Positive Rate**.

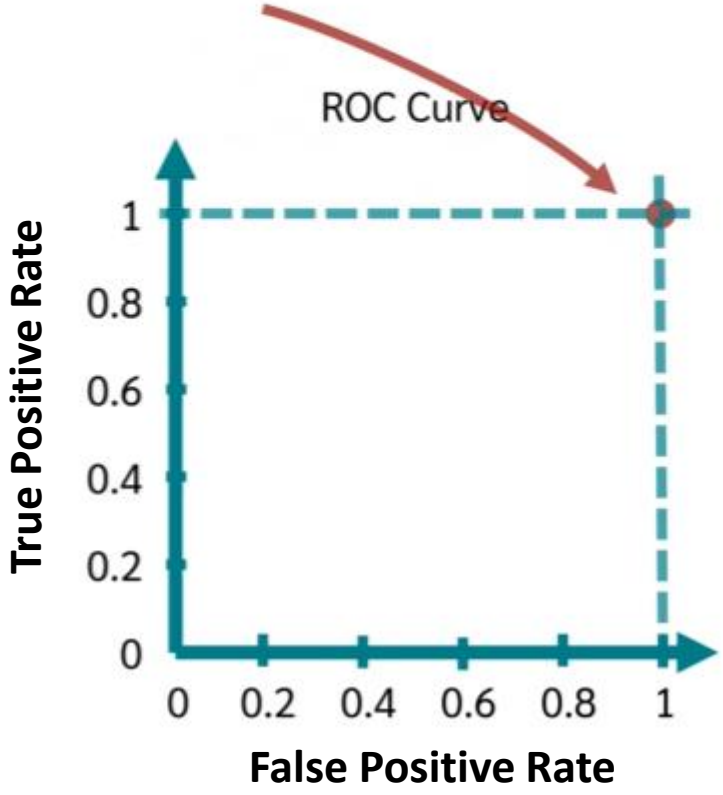


If we choose the **threshold value** to be very small,

i.e., pushed all the way to the **left**,

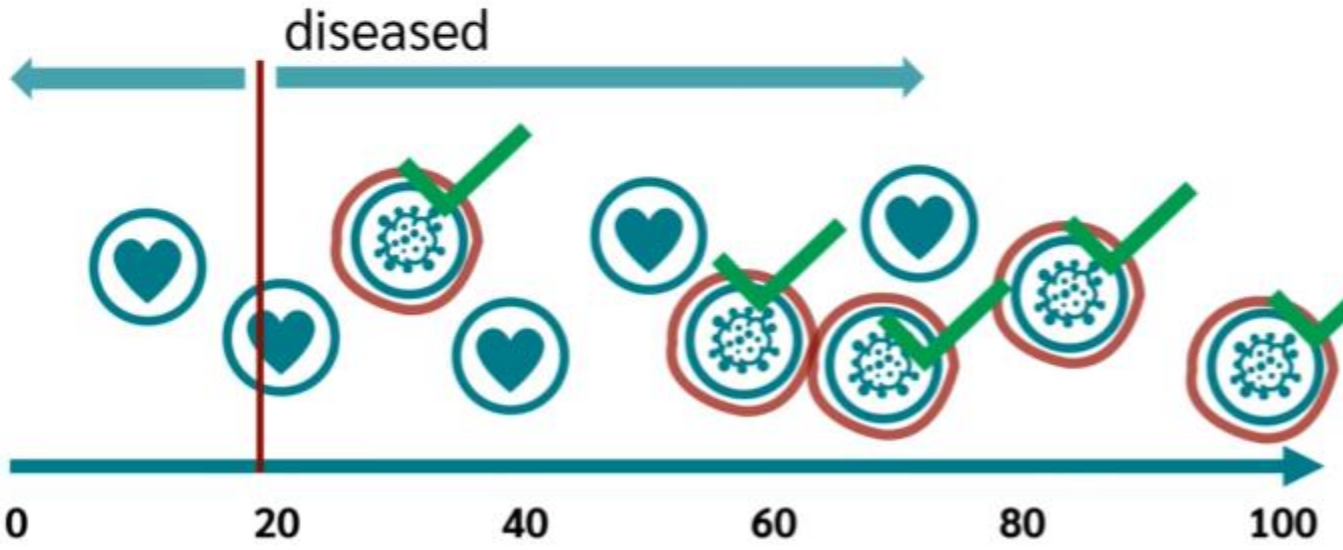
With that, we have the **first point**.

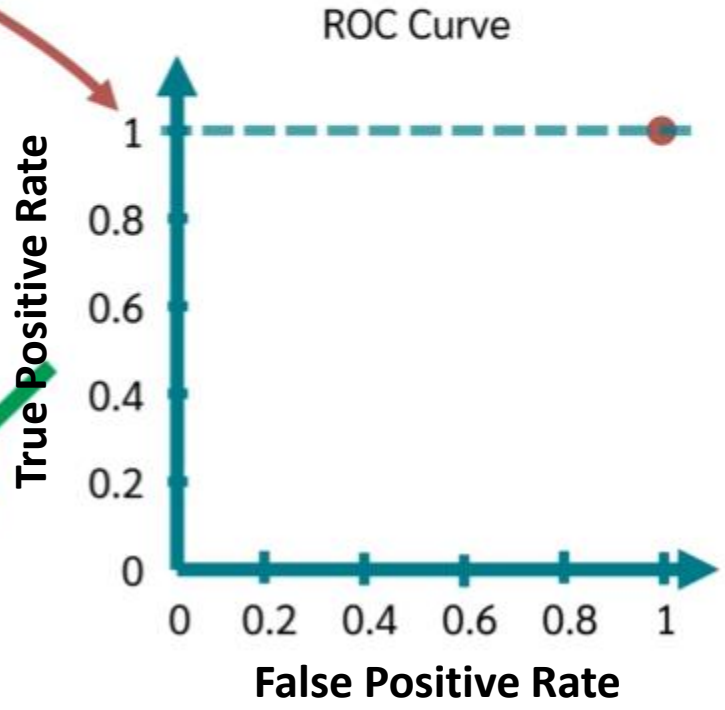




Here we still **correctly classify** all **5 diseased** people as **"diseased"**.

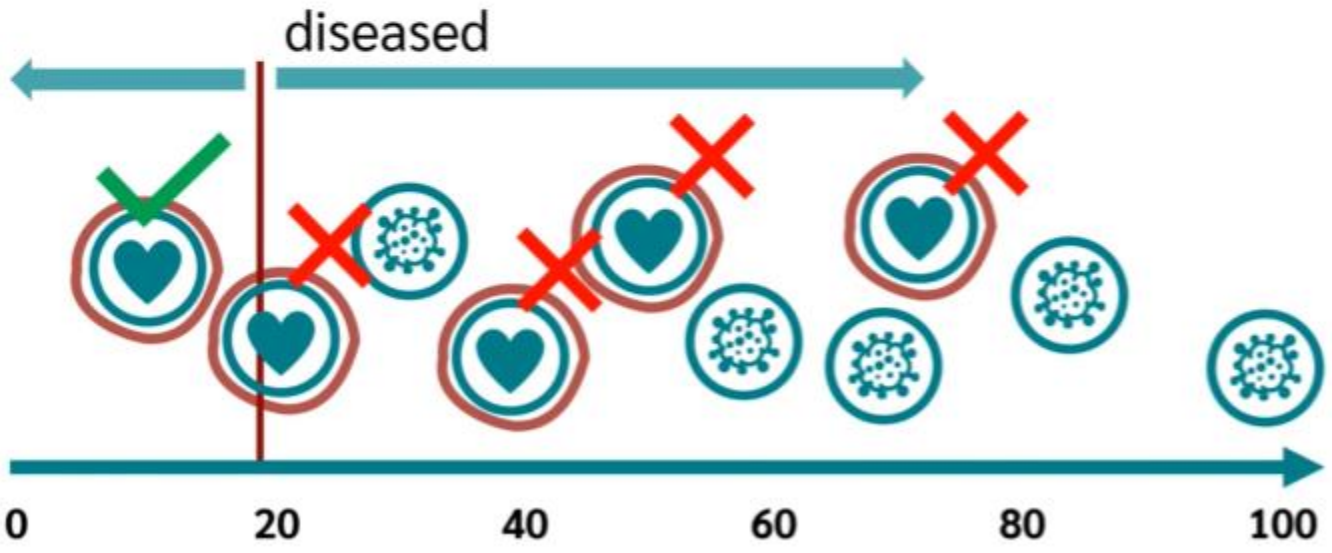
So, we continue to have a **True Positive Rate 5/5**.

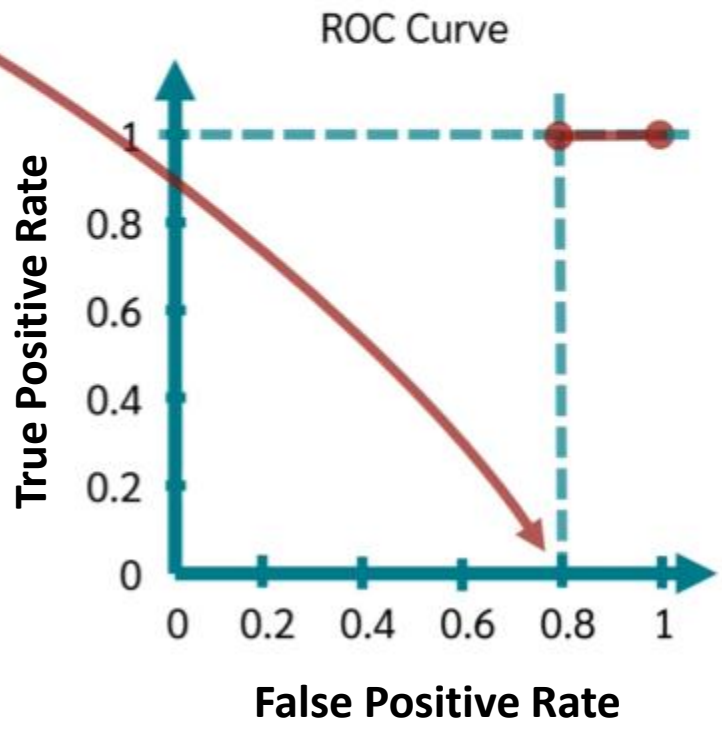




However, of the **5 healthy individuals**, we now **misclassify only 4 out of 5** as "diseased."

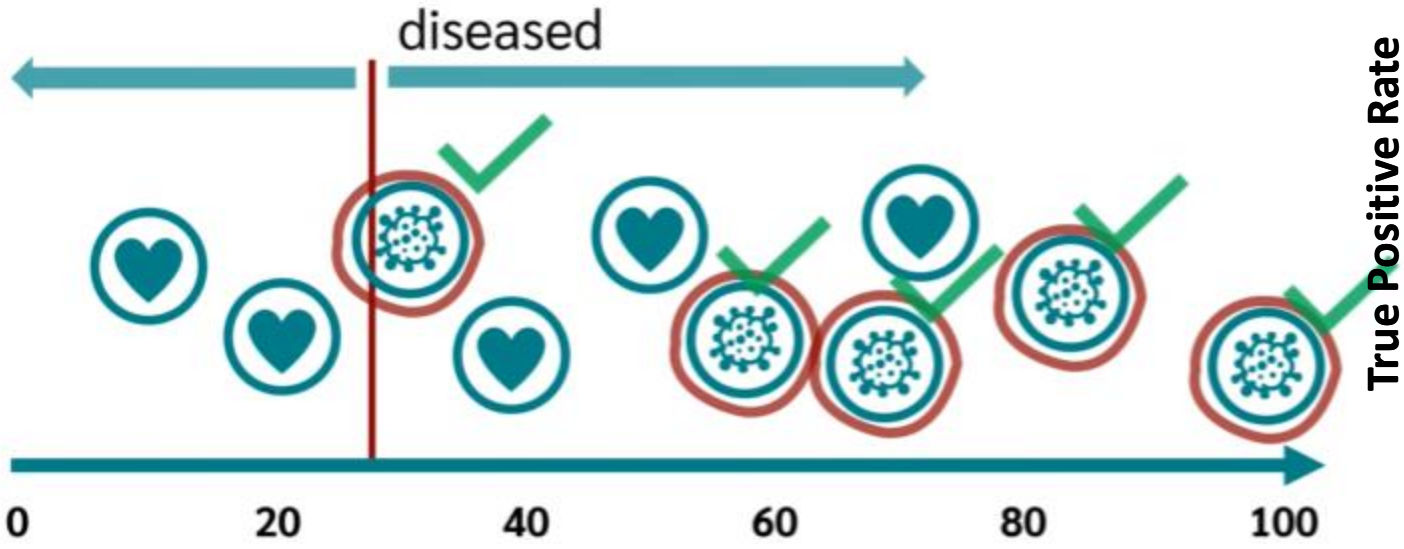
So, we have **4** out of **5** or **0.8**.

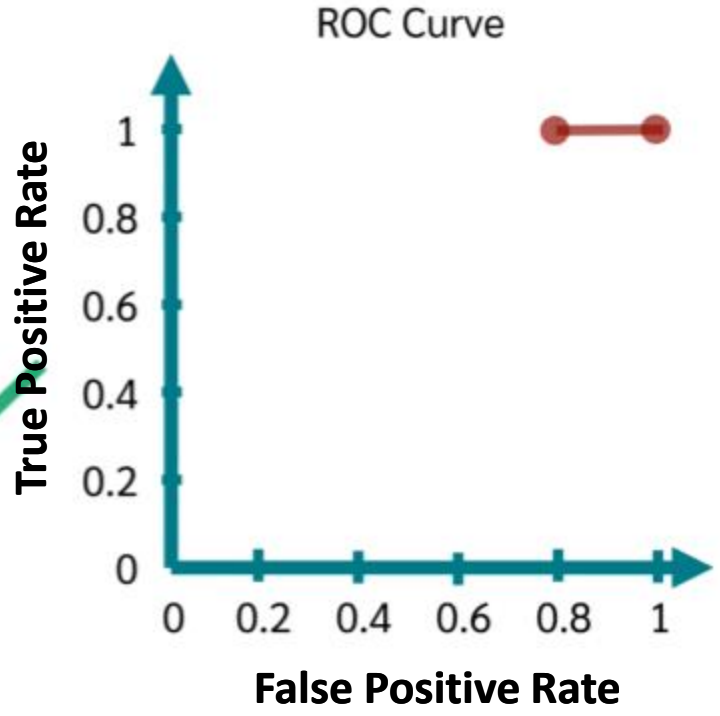




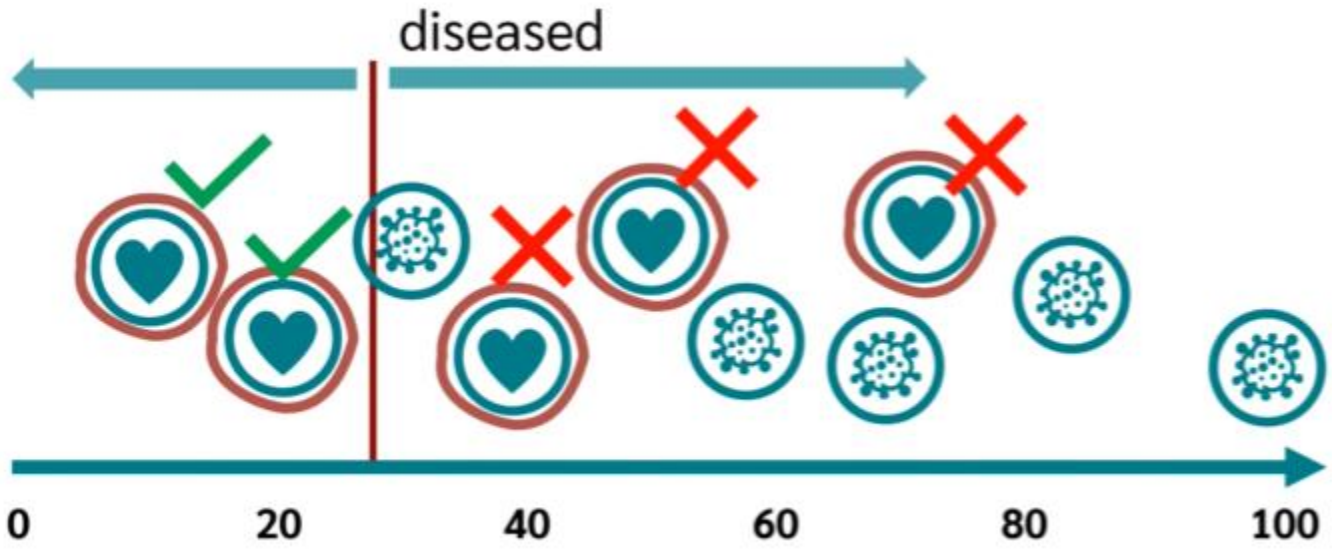
At the next **threshold,** we still have a **True Positive Rate of 1**.

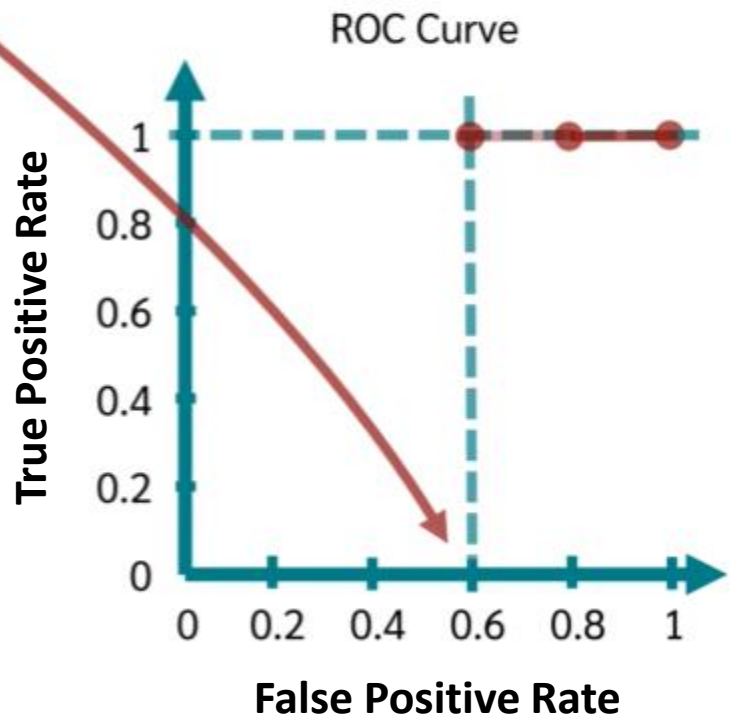
All **5 diseased** people are correctly classified.



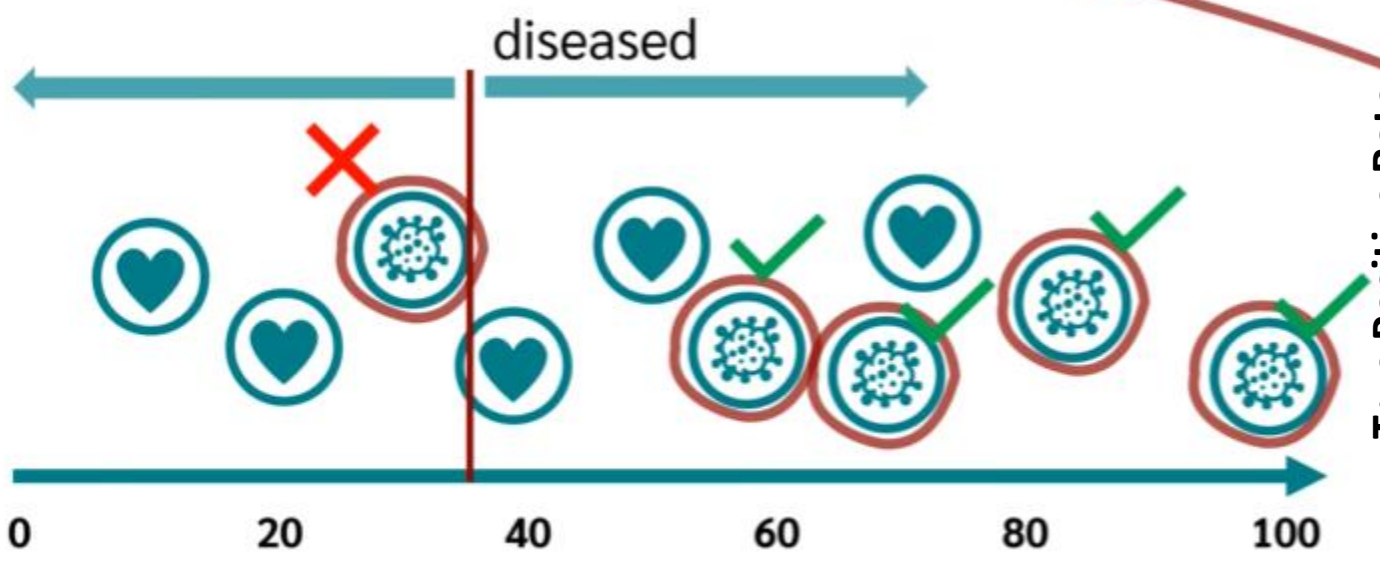


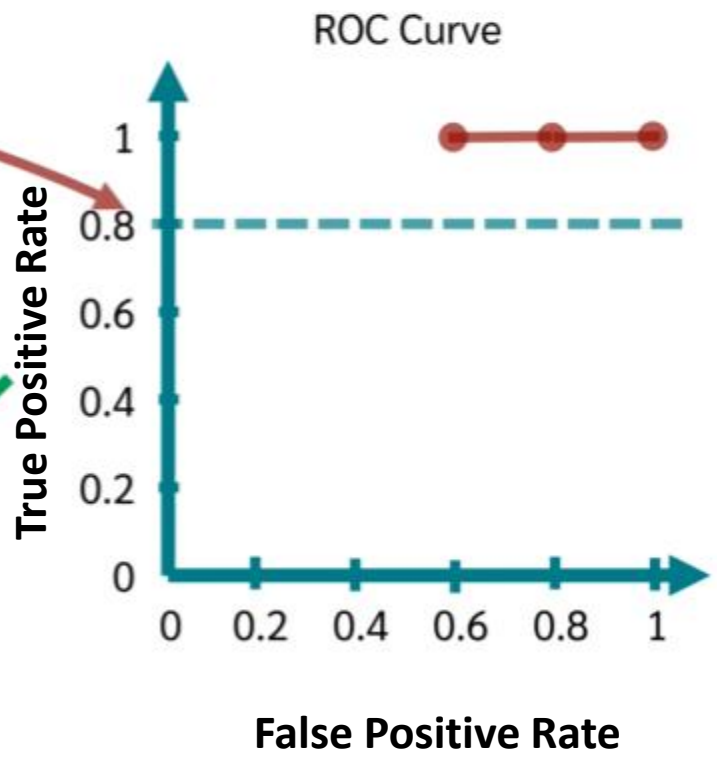
And a **False Positive Rate** of **3/5**, i.e. **0.6**



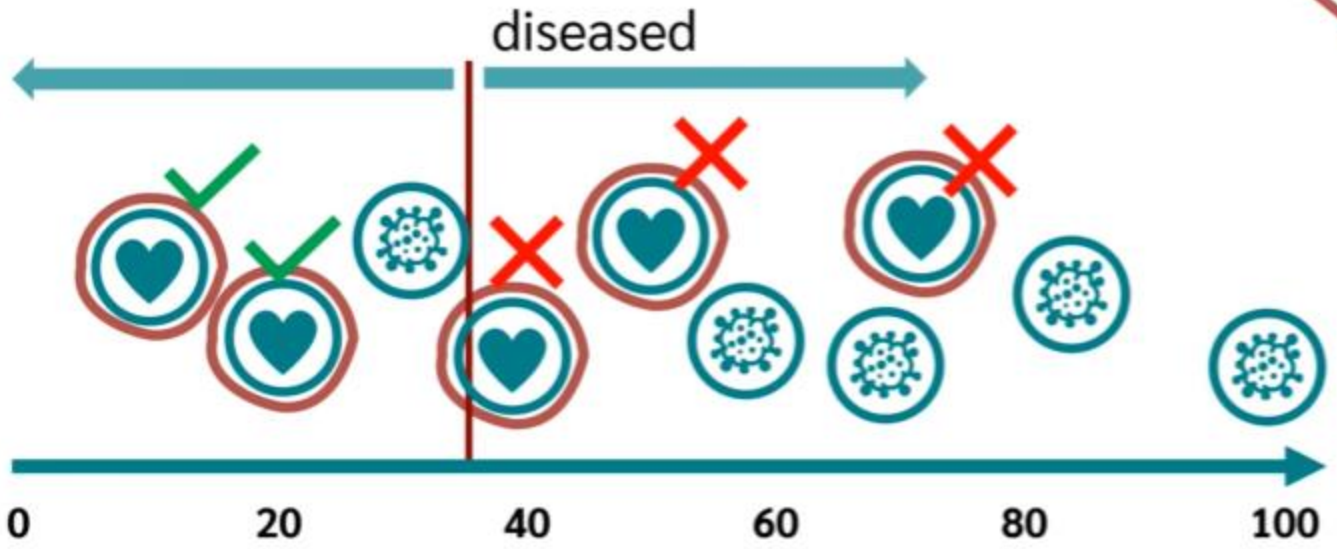


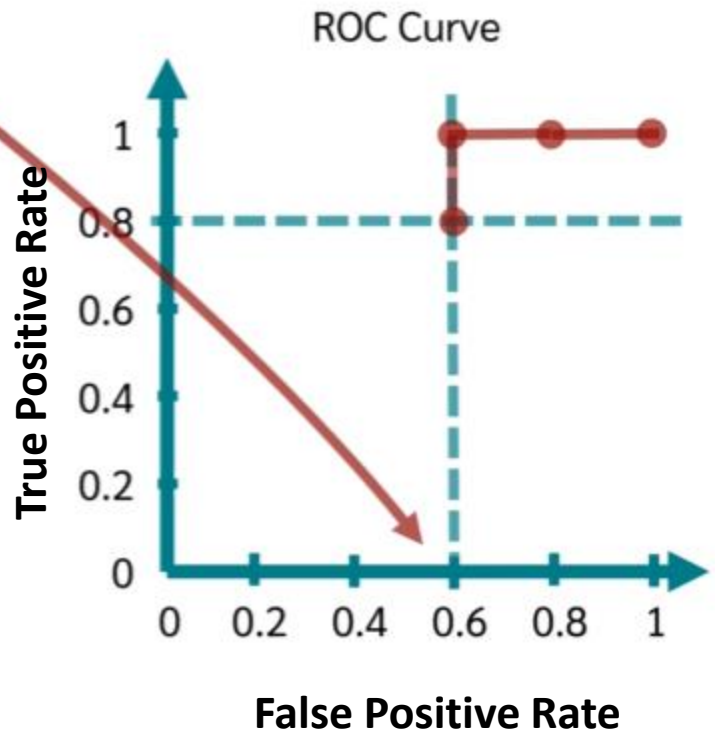
We therefore obtain a **True Positive Rate** of **4/5**, i.e. **0.8**.





And a **False Positive Rate** of **3/5**, i.e. **0.6**.

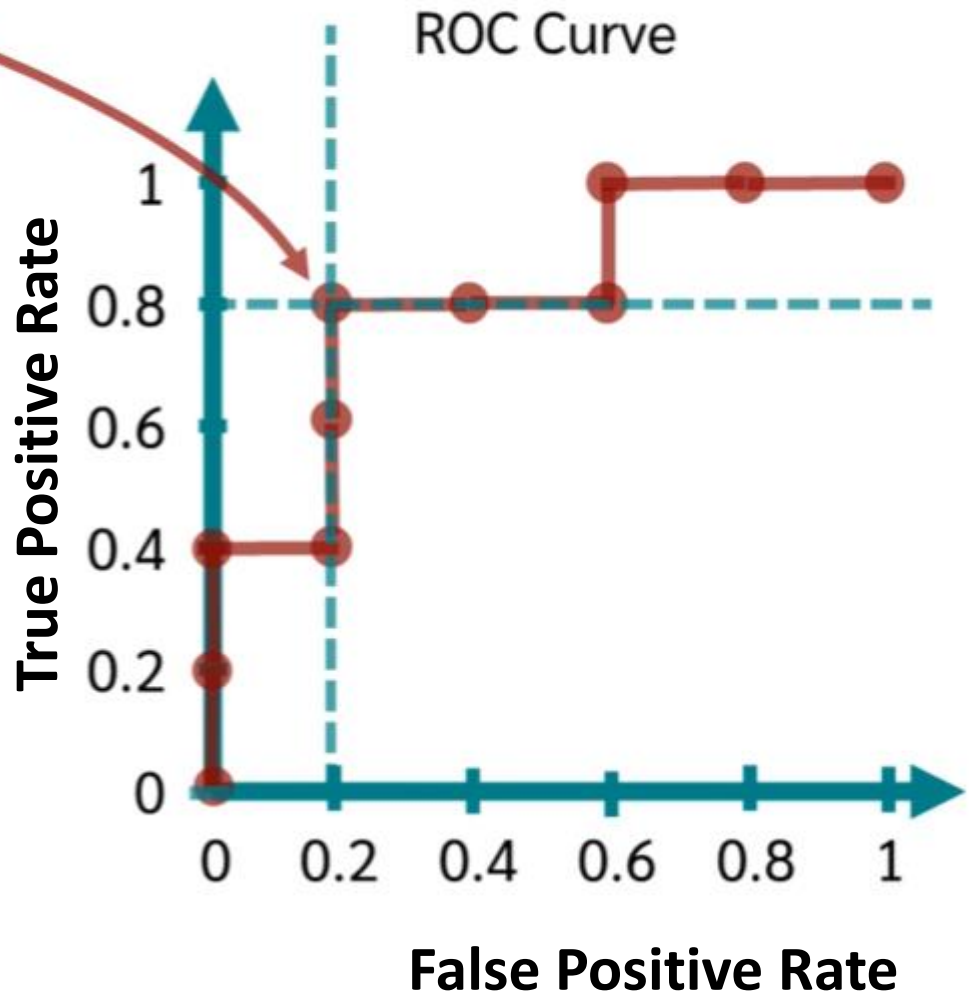




At this point, for example

80% of the diseased individuals were correctly classified as “diseased“

and **20%** of the healthy individuals were incorrectly assigned “diseased“.



A **classification model** is better the **higher** the **curve**.

Hyperparameter Tuning

GridSearchCV

RandomSearchCV

Types of Parameters

Parameters

Model Parameters

These are the parameters of the model that can be determined by training with training data. These can be considered as internal Parameters.

- **Weights**
- **Bias**

Y = w*X + b

Hyperparameters

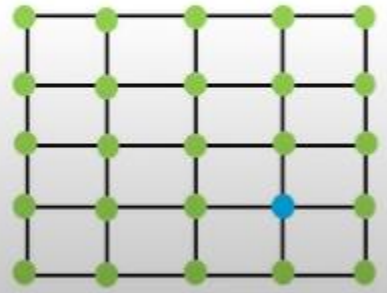
Hyperparameters are parameters whose values control the learning process. These are adjustable parameters used to obtain an optimal model. External Parameters.

- **Learning rate**
- **Number of Epochs**
- **n_estimators**

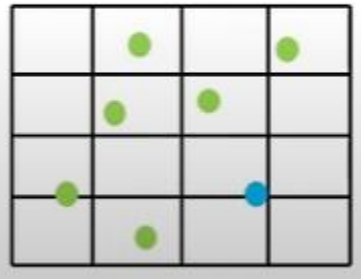
Hyperparameter Tuning

Hyperparameter Tuning refers to the process of choosing the optimum set of hyperparameters for a Machine Learning model. This process is also called **Hyperparameter Optimization.**

Hyperparameter Tuning Types:



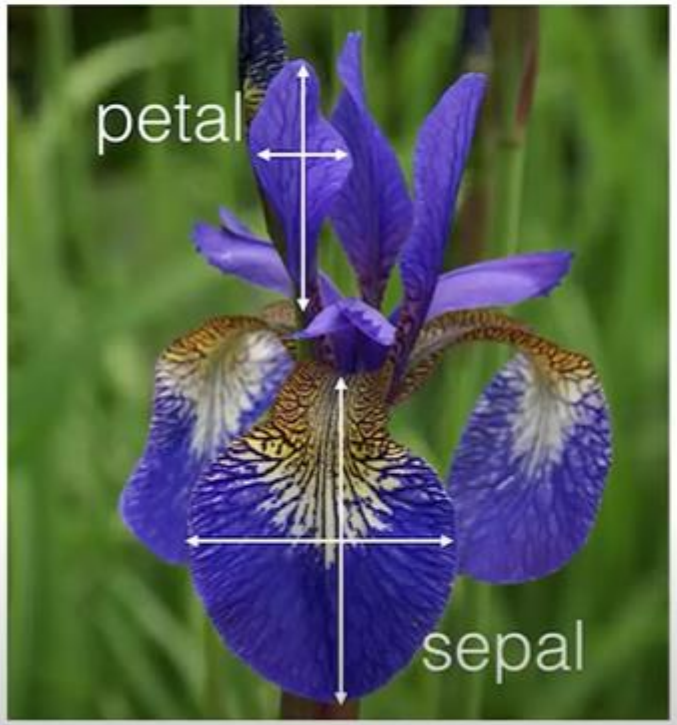
GridSearchCV



RandomizedSeaechCV

Support Vector Classifier:
C: [1,5,10]
kernel: ('linear', 'poly', 'rbf', 'sigmoid')

sklearn iris flower dataset



features

target label

sepal length (cm)	sepal width (cm)	petal length (cm)	petal width (cm)	flower
4.6	3.2	1.4	0.2	setosa
5.3	3.7	1.5	0.2	setosa
5.0	3.3	1.4	0.2	setosa
7.0	3.2	4.7	1.4	versicolor
6.4	3.2	4.5	1.5	versicolor

Model selection

Random Forest

SVM

Decision Tree

Logistic Regression

Naïve Bayes

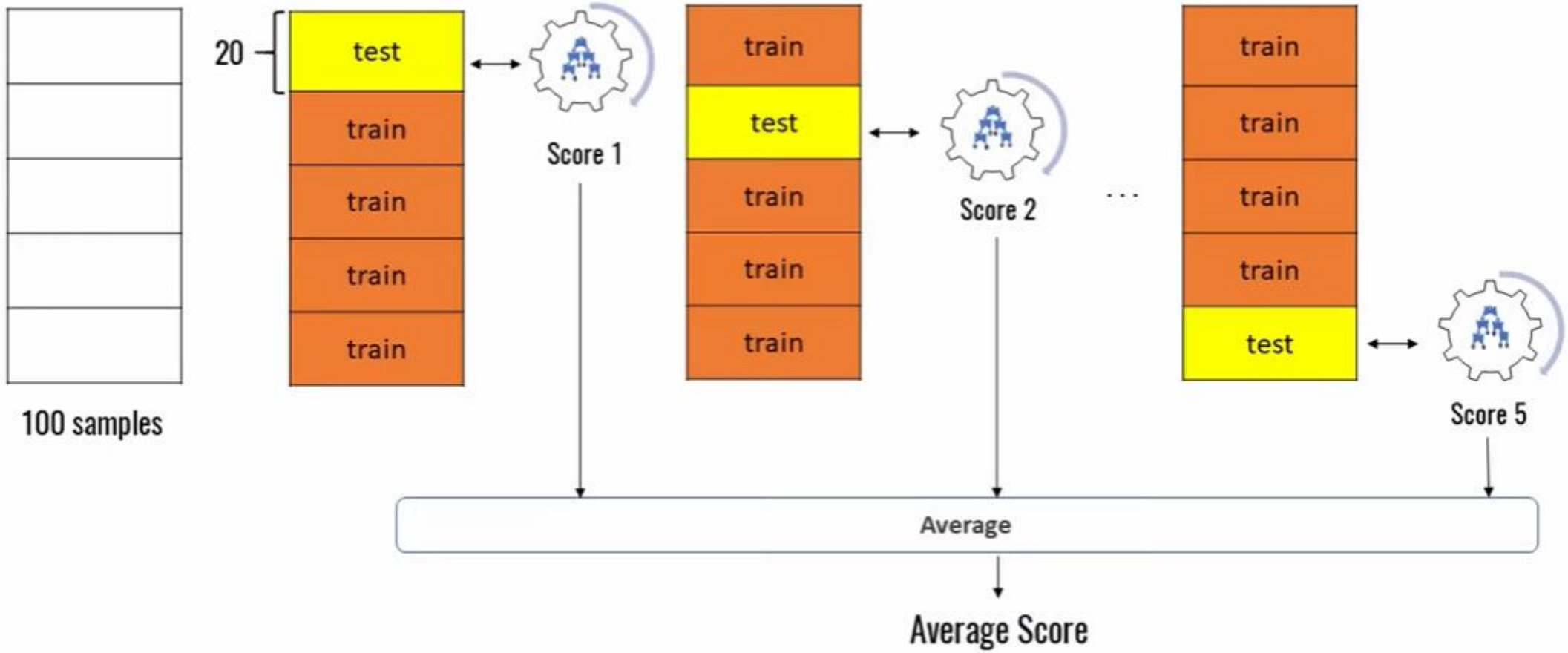
Hyperparameter Tuning

SVM

example

```
model = svm.SVC(kernel='rbf',C=30,gamma='auto')
```

Parameter	Values
Kernel	'rbf', 'linear', 'poly'
C	Integer
Gamma	float



5 fold cross validation