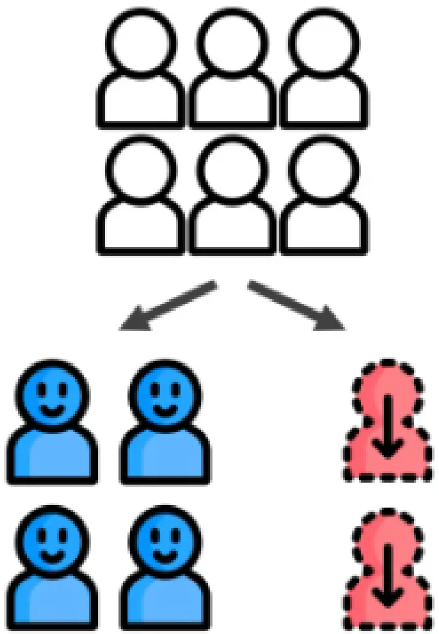
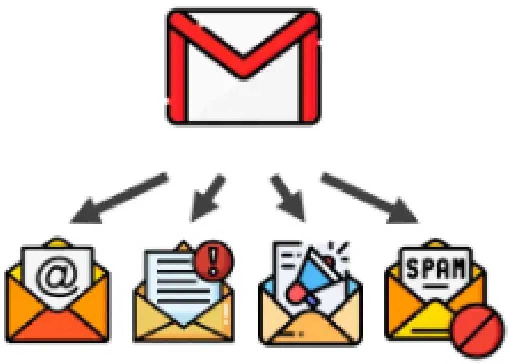
Classification

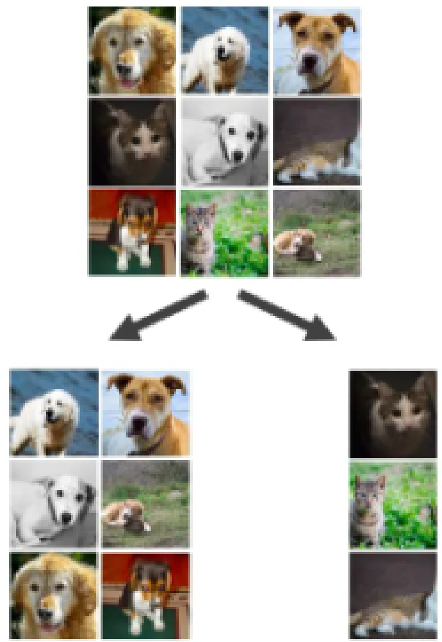
What is Classification?

Classification: a Machine Learning technique to identify the <u>category</u> of new observations based on training data.



Likely to stay *Likely to leave*





Dogs *Cats*

Logistic Regression

Logistic Regression

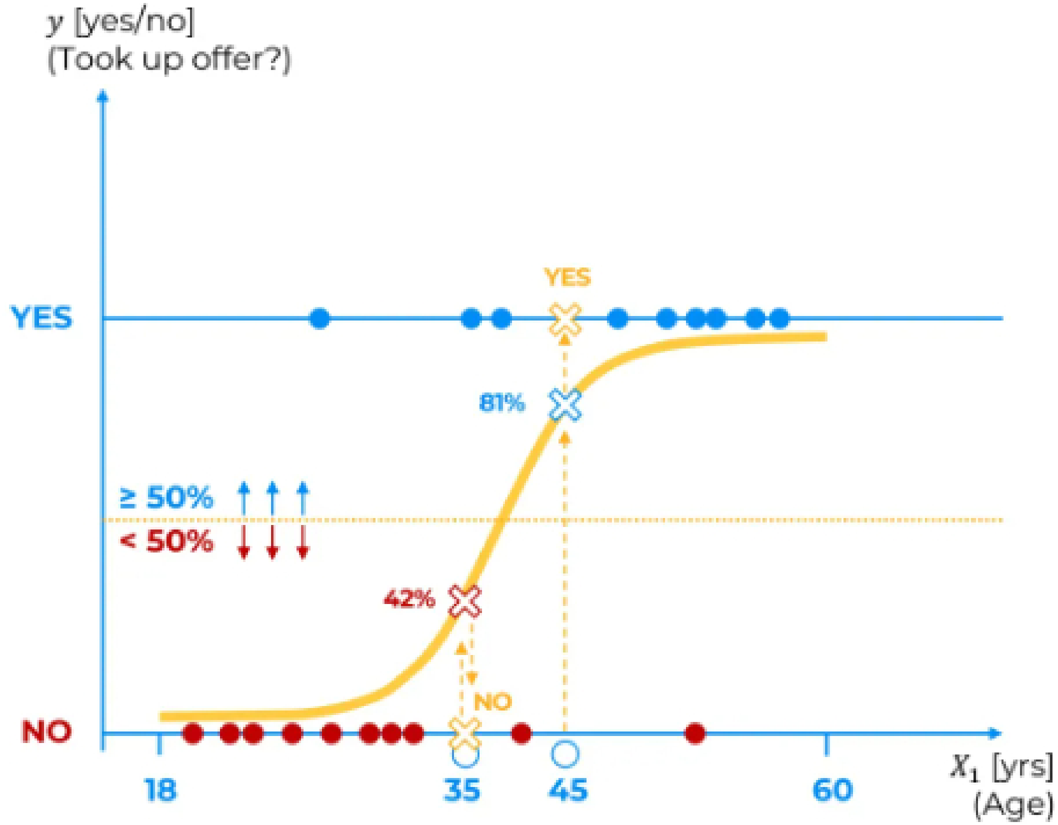
Logistic regression: predict a categorical dependent variable from a number of independent variables.

Will purchase health insurance: Yes / No

~

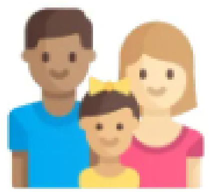
Age

$$\ln \frac{p}{1-p} = b_0 + b_1 X_1$$



Logistic Regression

$\sim$



Will purchase health insurance: Yes / No

Age

Income

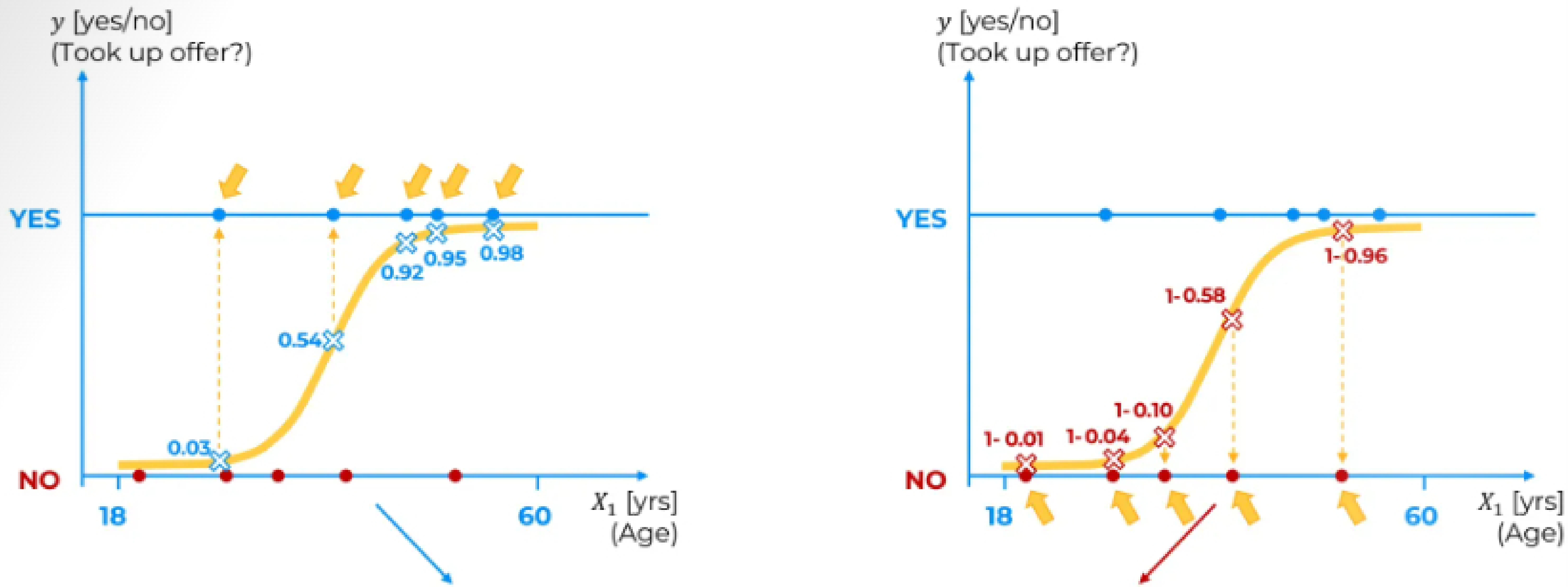
Level of Education

Family or Single

$$\ln \frac{p}{1-p} = b_0 + b_1 X_1 + b_2 X_2 + b_3 X_3 + b_4 X_4$$

Maximum Likelihood

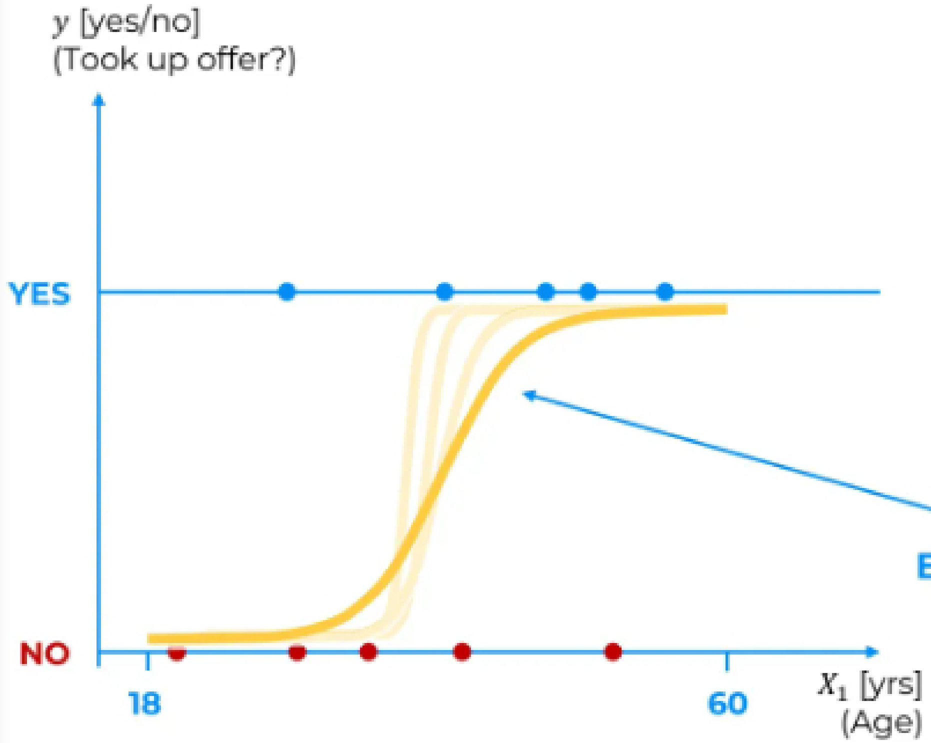
Maximum Likelihood



Likelihood = 0.03 x 0.54 x 0.92 x 0.95 x 0.98 x (1 – 0.01) x (1 – 0.04) x (1 – 0.10) x (1 – 0.58) x (1 – 0.96)

Likelihood = **0.00019939**

Maximum Likelihood



Likelihood = 0.00007418

Likelihood = 0.00012845

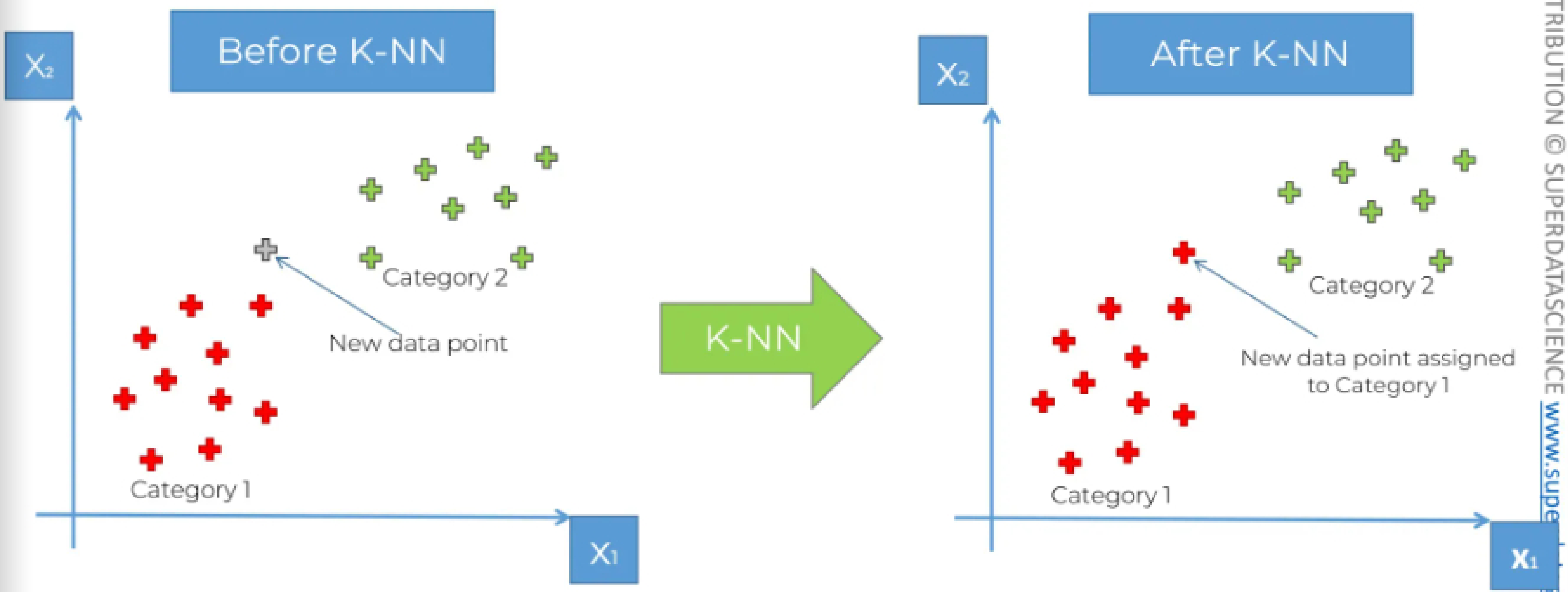
Likelihood = 0.00016553

$\boxed{\text{Likelihood = 0.00019939}}$

Best Curve <= Maximum Likelihood

K-NN Intuition

What K-NN does for you



How did it do that ?

STEP 1: Choose the number K of neighbors

STEP 2: Take the K nearest neighbors of the new data point, according to the Euclidean distance

STEP 3: Among these K neighbors, count the number of data points in each category

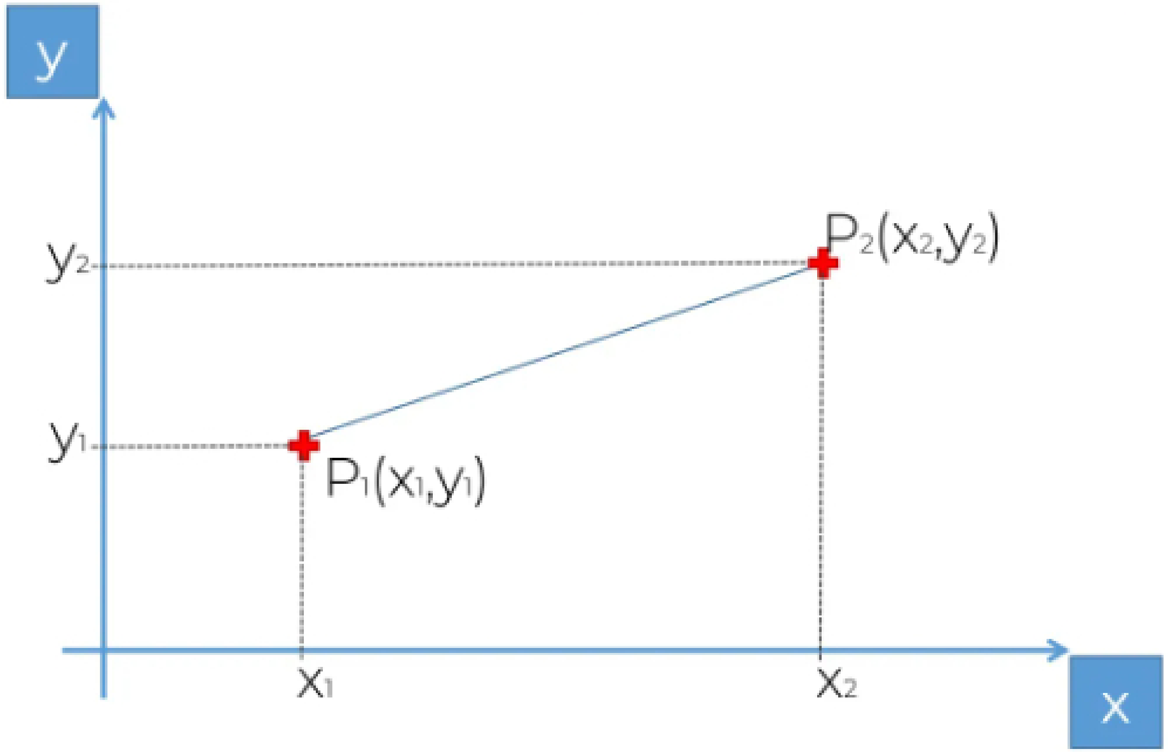
STEP 4: Assign the new data point to the category where you counted the most neighbors

Your Model is Ready

K-NN algorithm

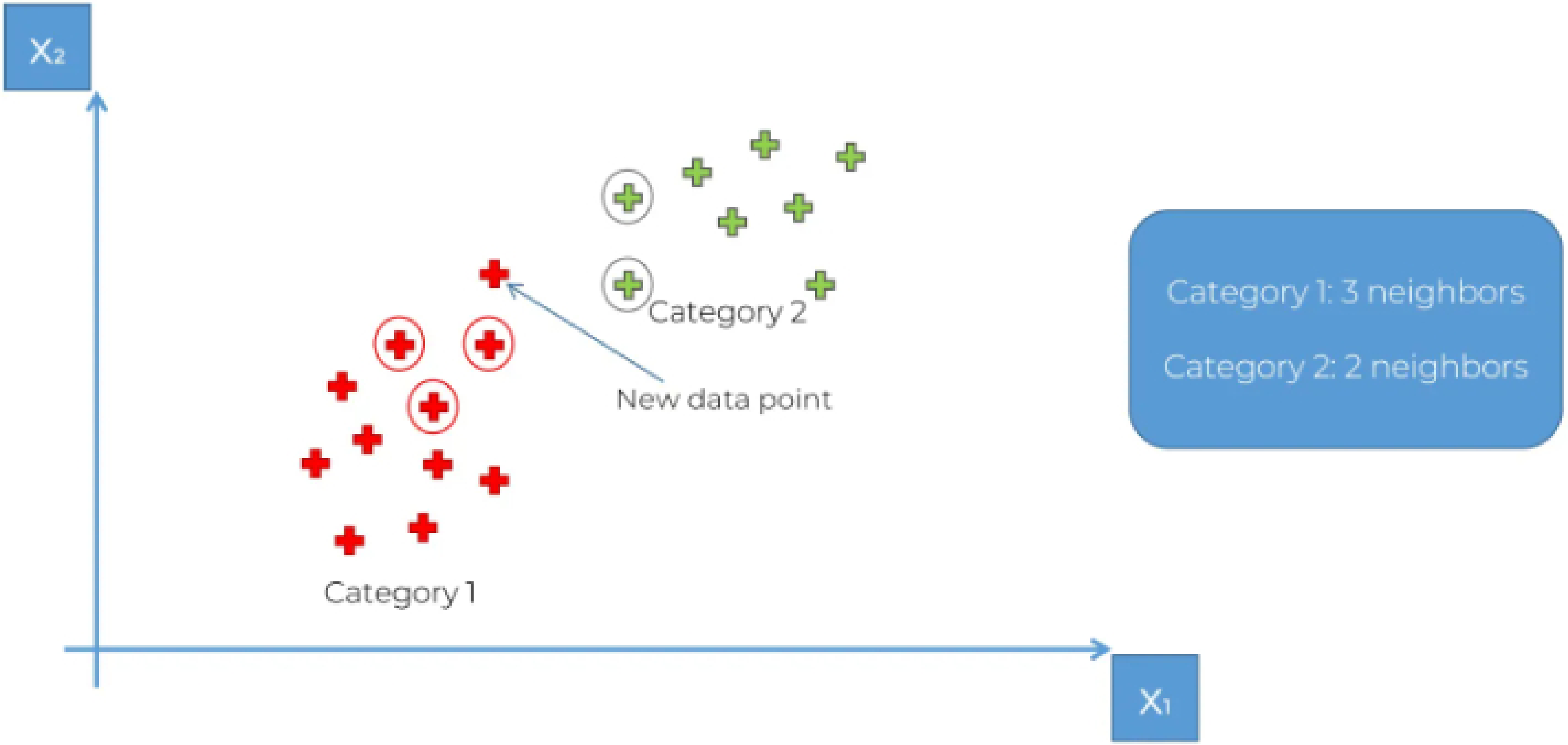
STEP 1: Choose the number K of neighbors: K = 5

Euclidean Distance



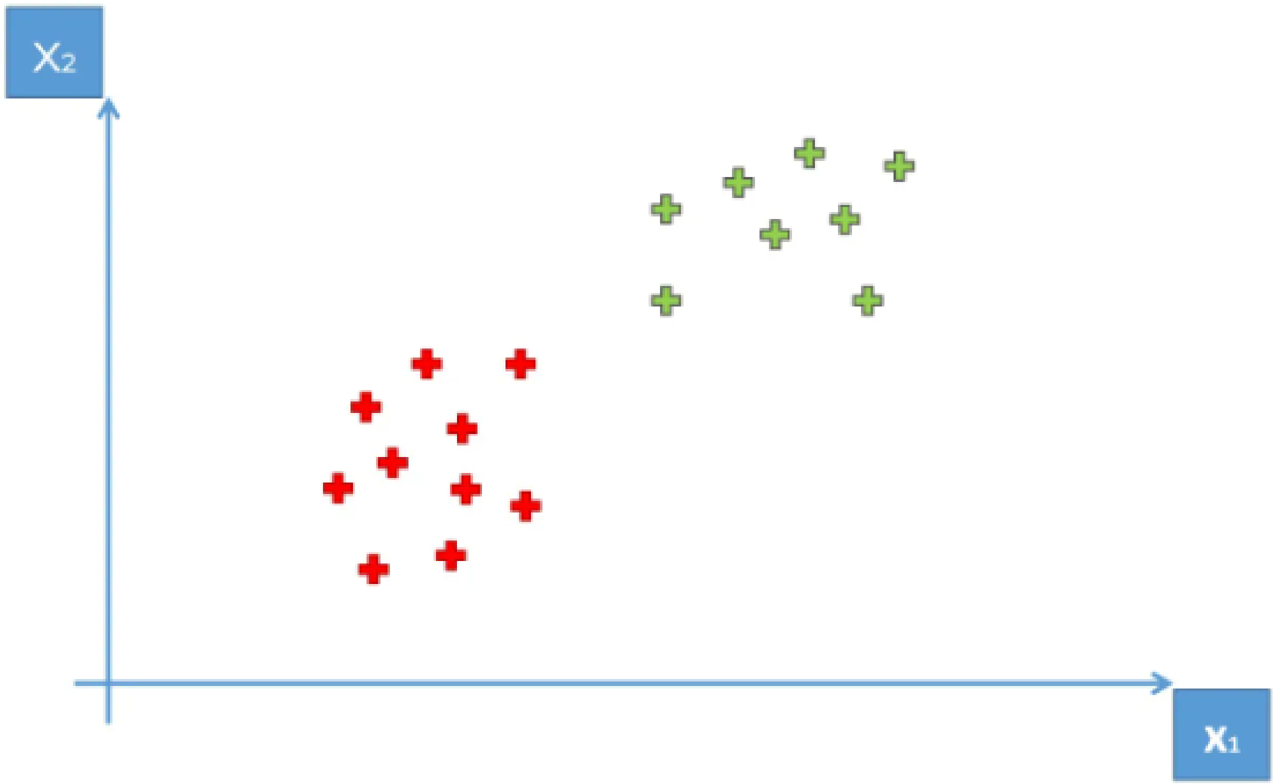
$$\text{Euclidean Distance between } P_1 \text{ and } P_2 = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$$

K-NN algorithm

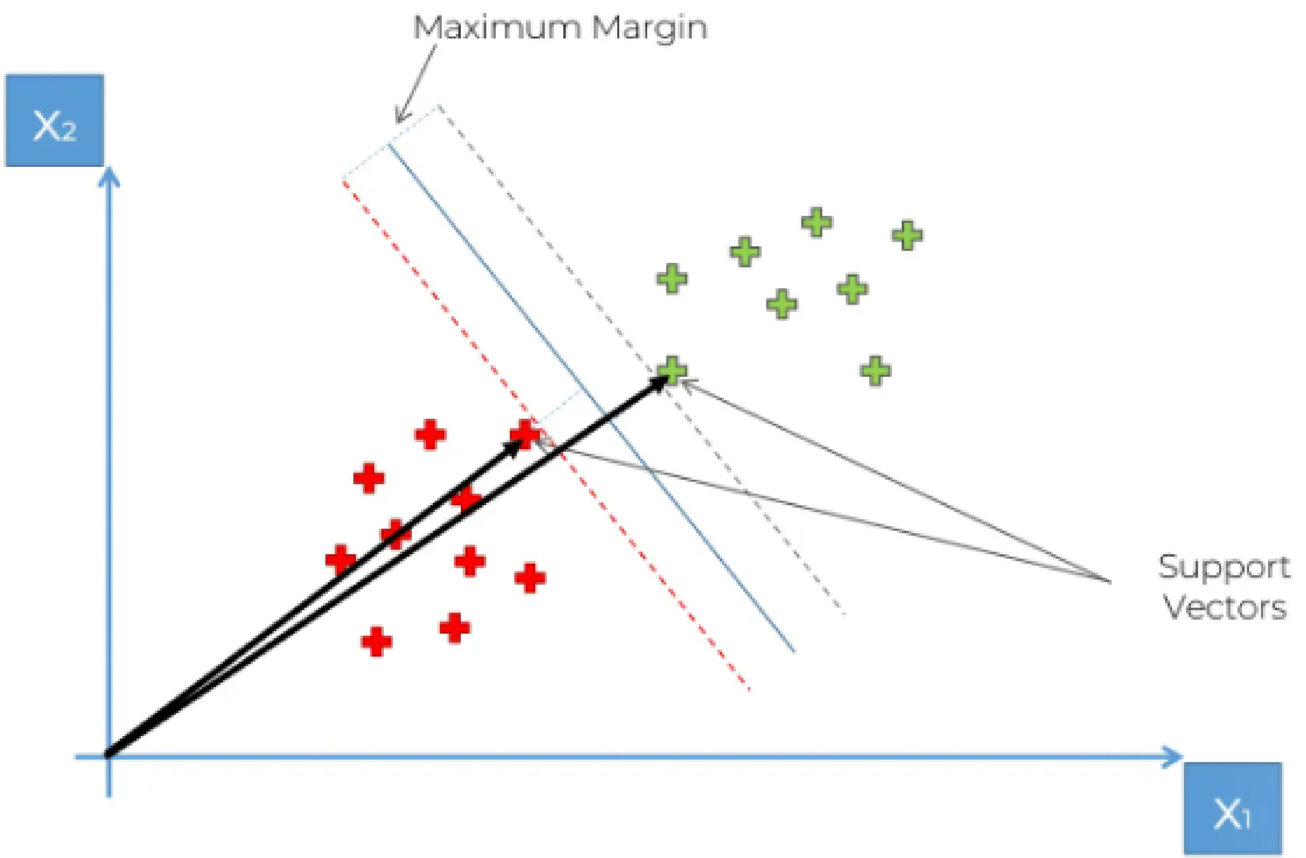


SVM Intuition

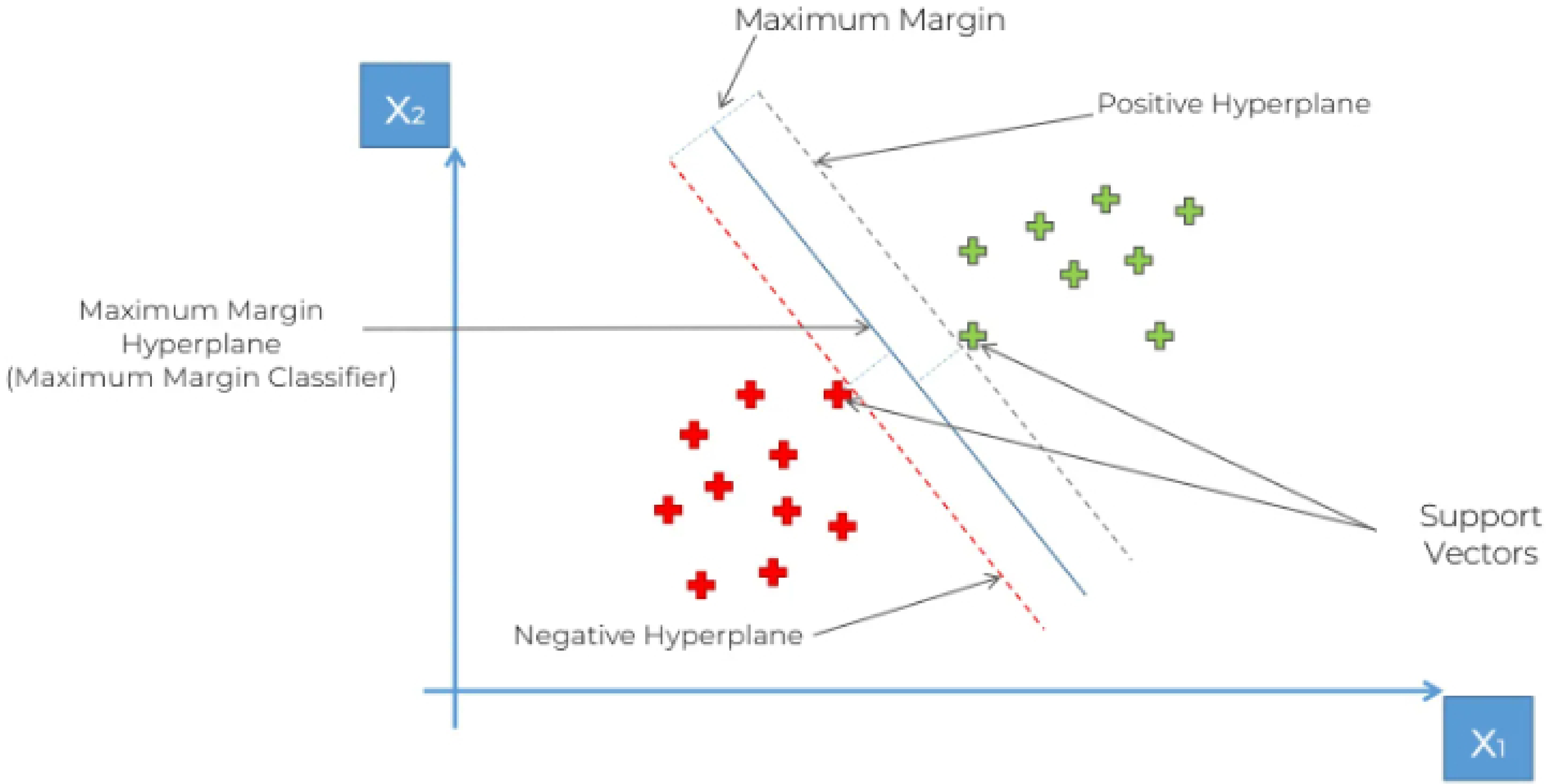
How to separate these points ?



Support Vectors

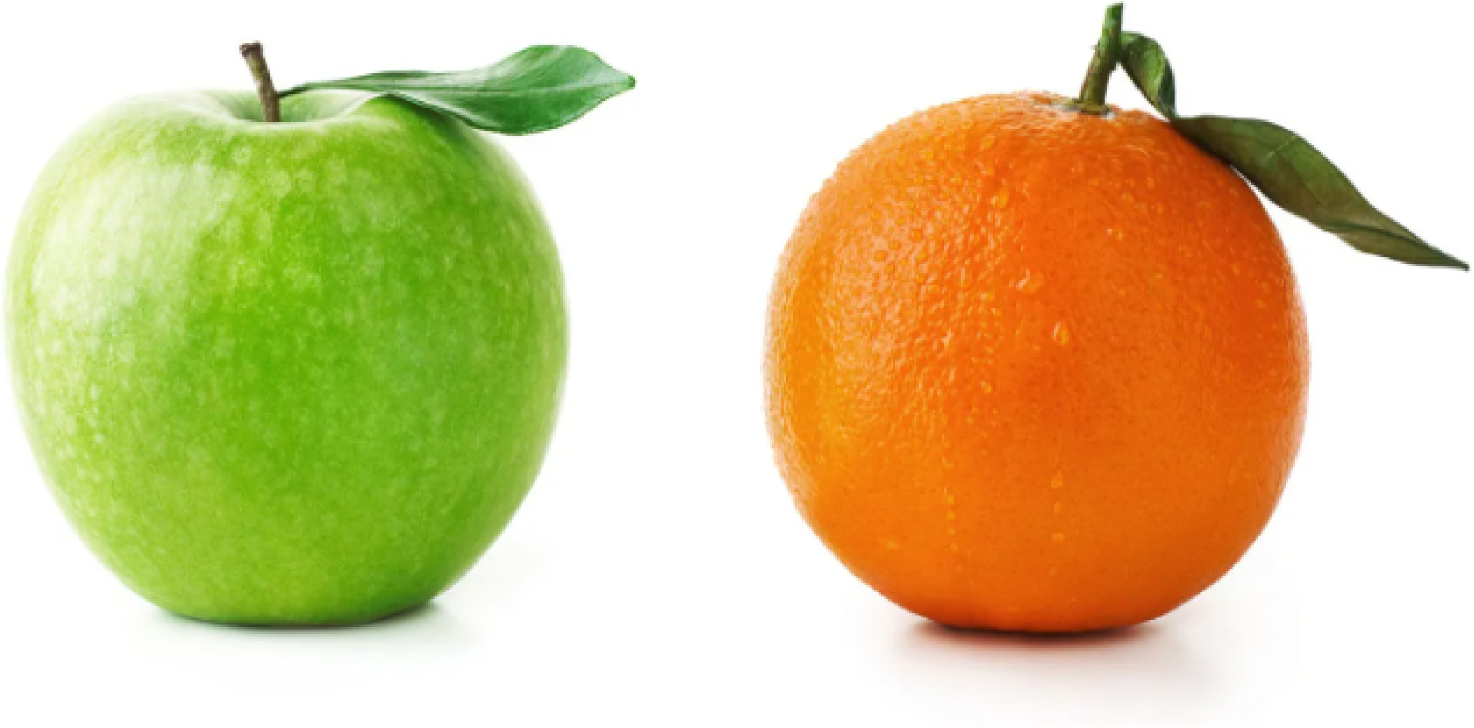


Hyperplanes

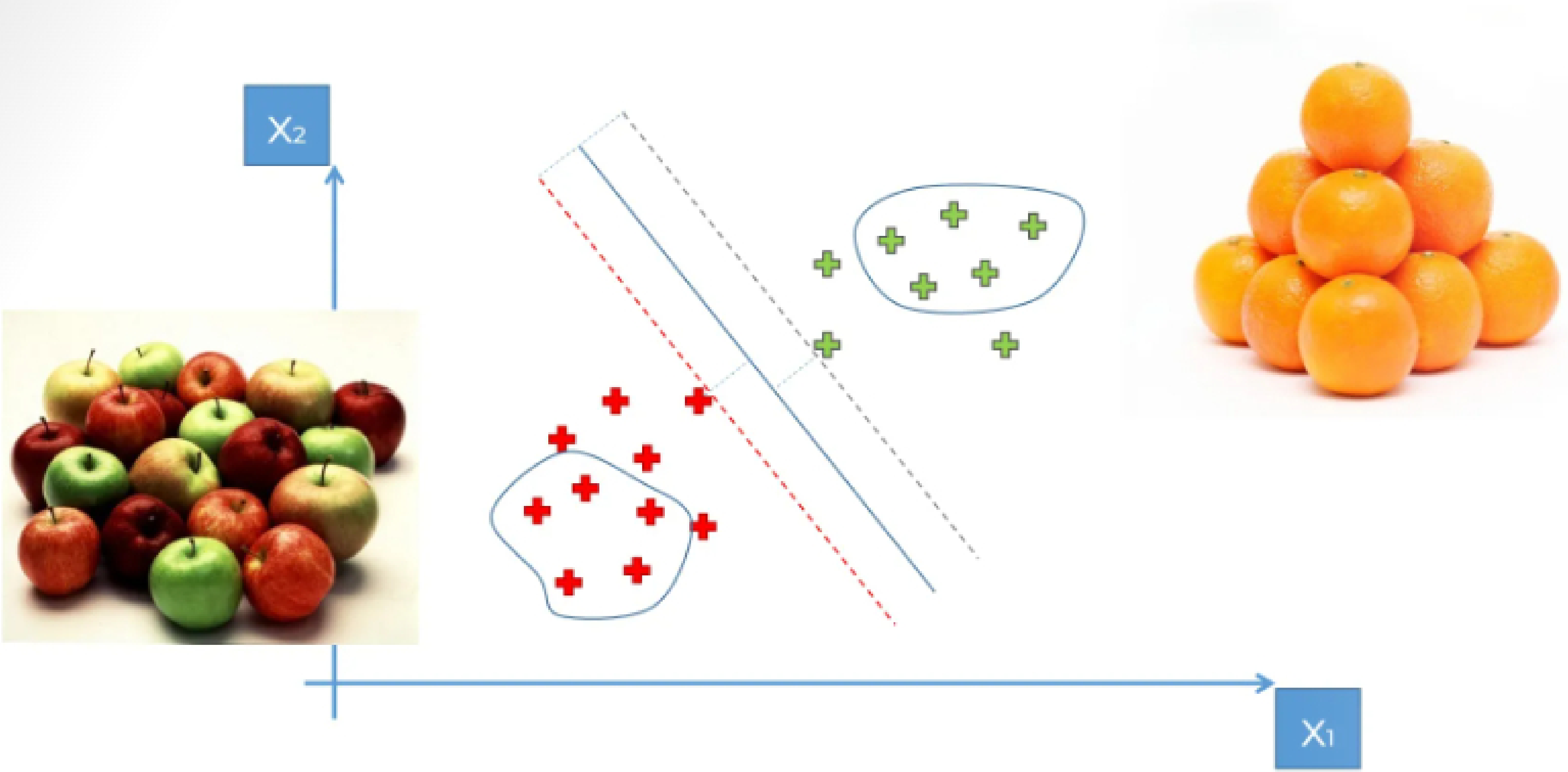


What's So Special About SVMs?

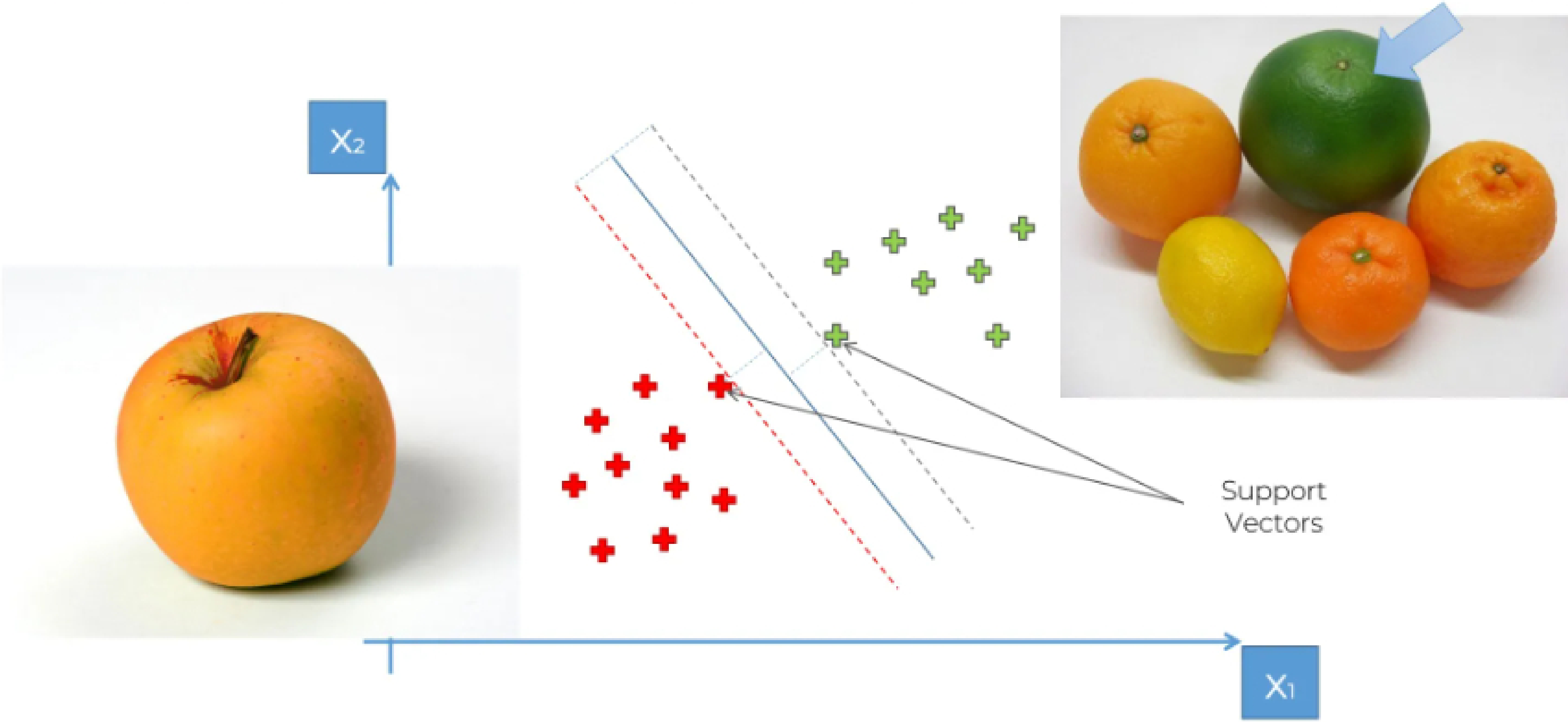
What's So Special About SVMs?



What's So Special About SVMs?

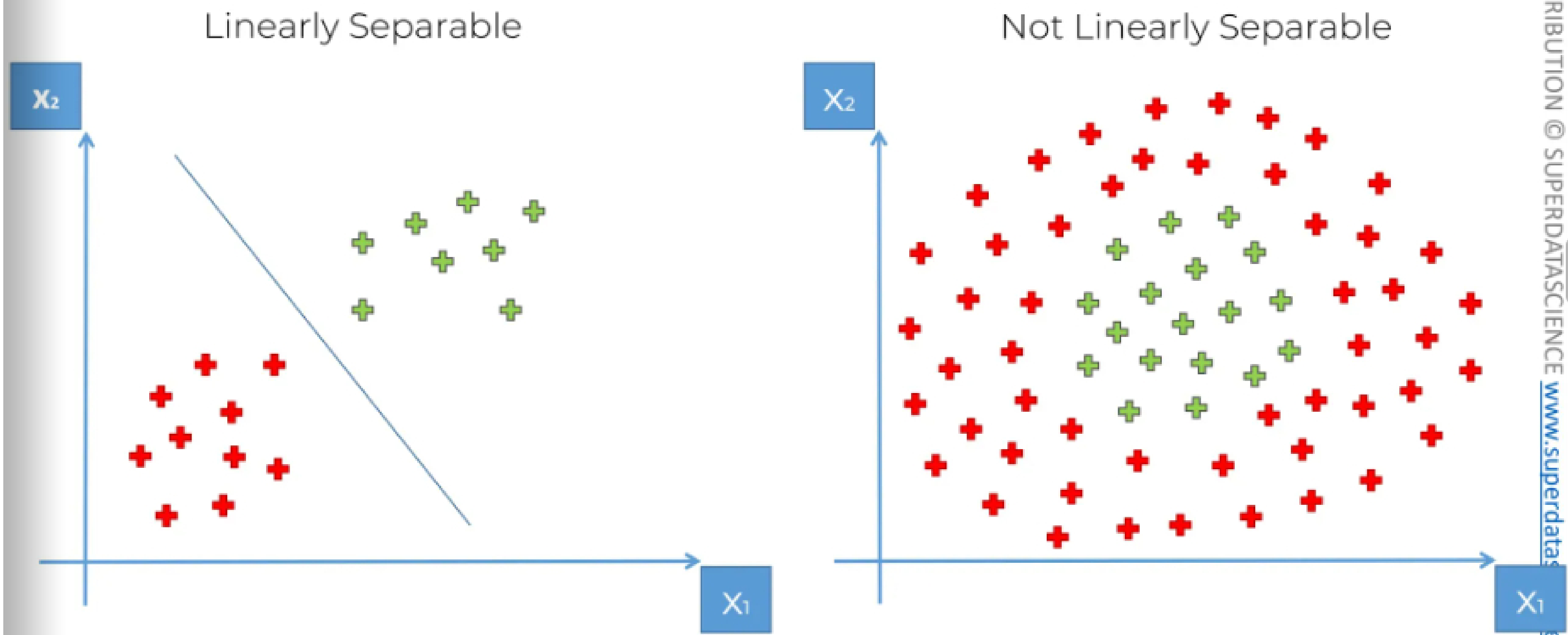


What’s So Special About SVMs?



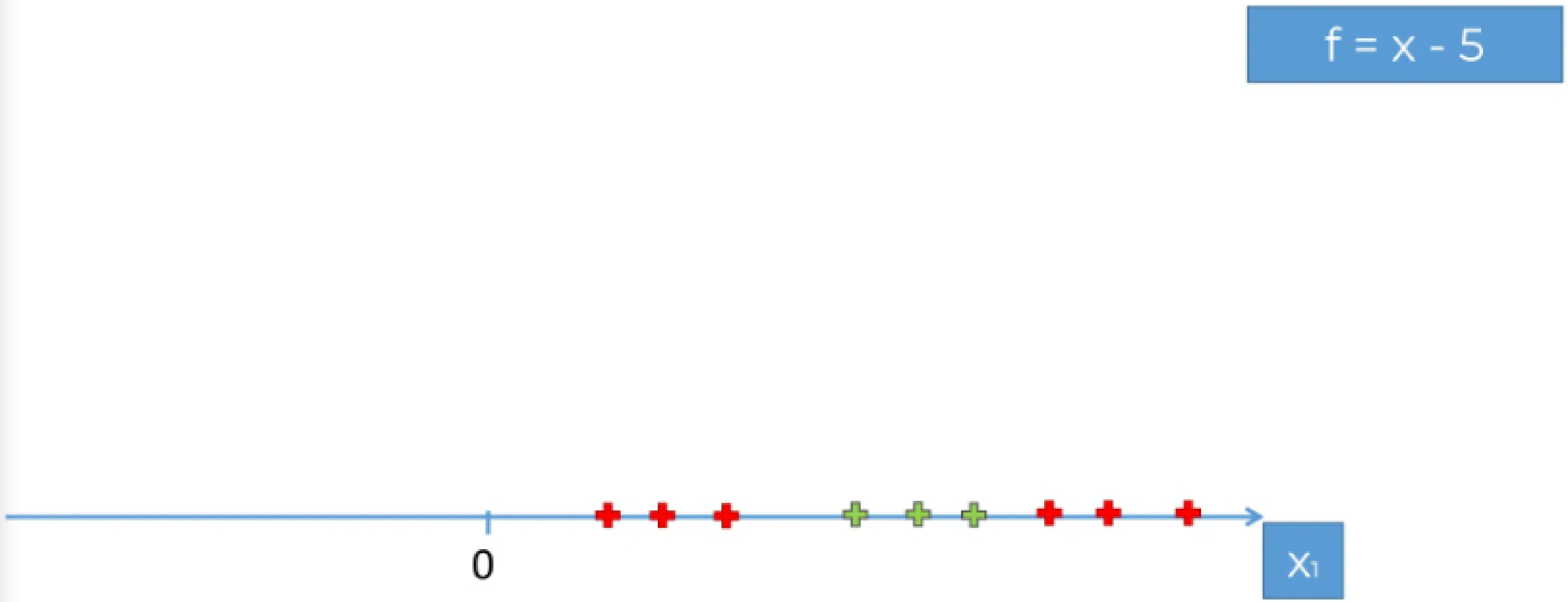
Kernel SVM Intuition

Linear Separability

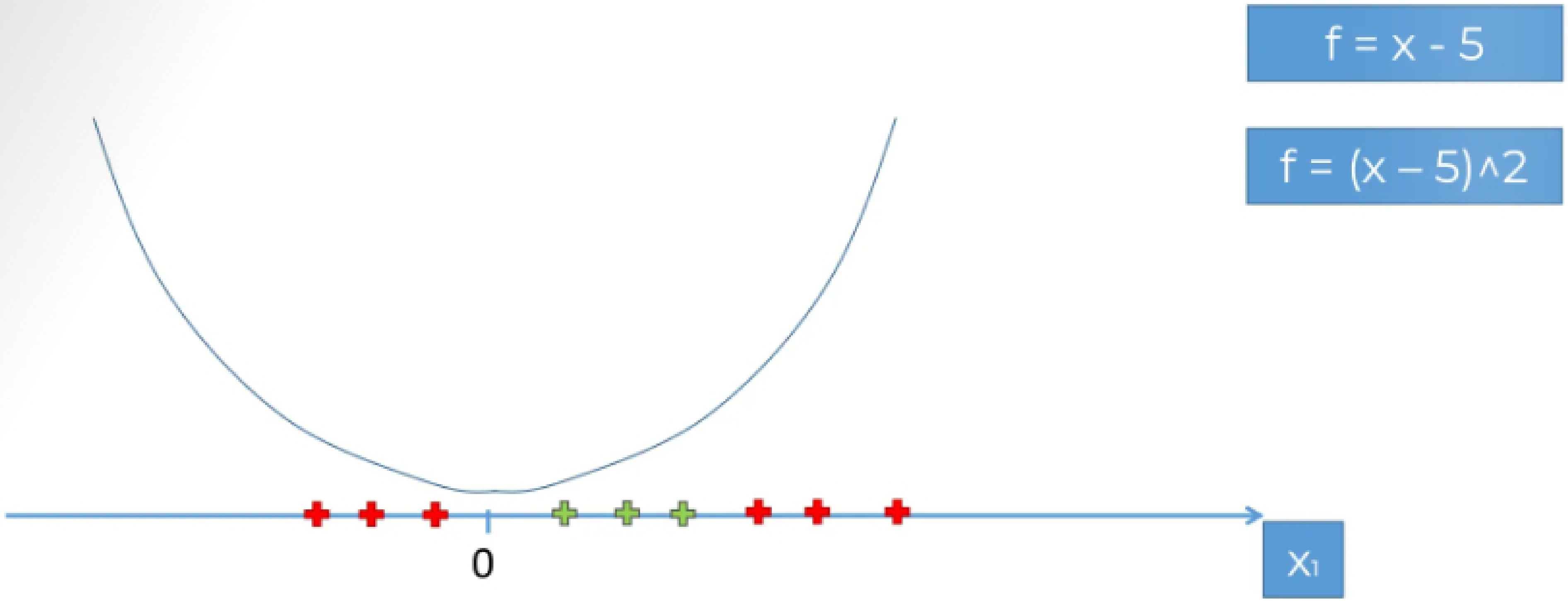


A Higher-Dimensional Space

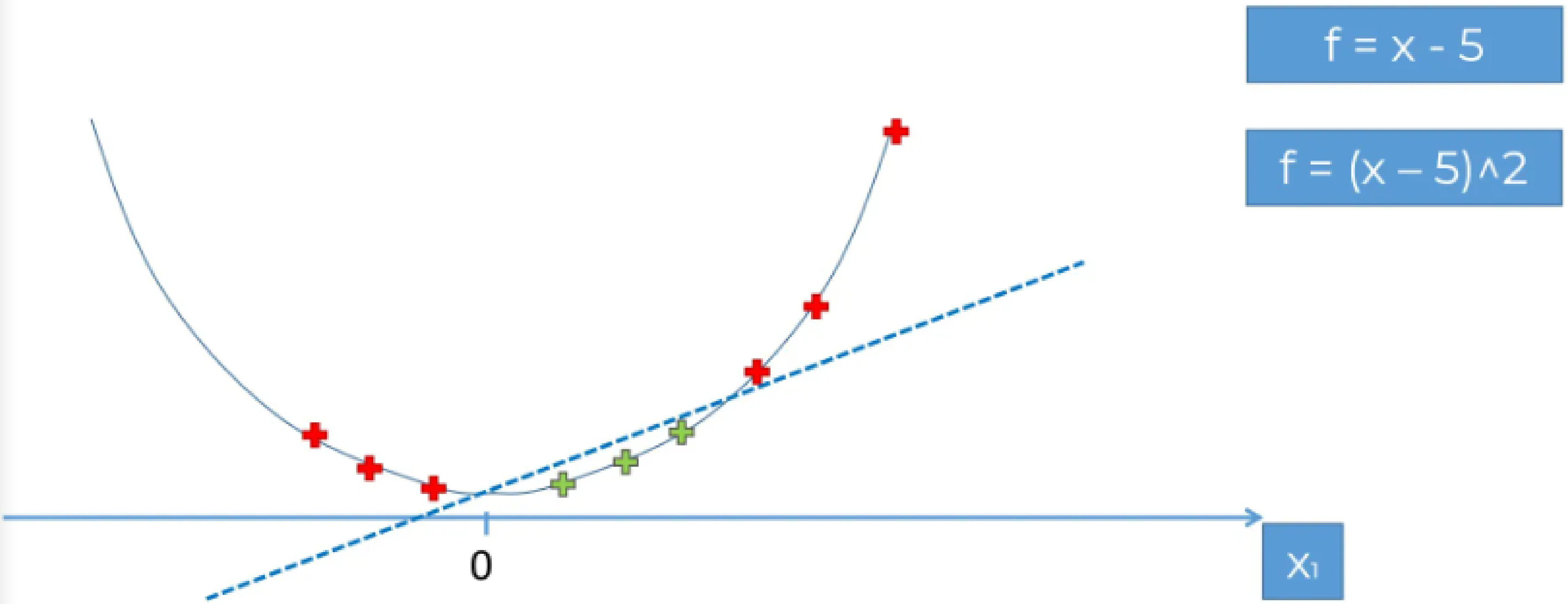
Mapping to a Higher Dimension

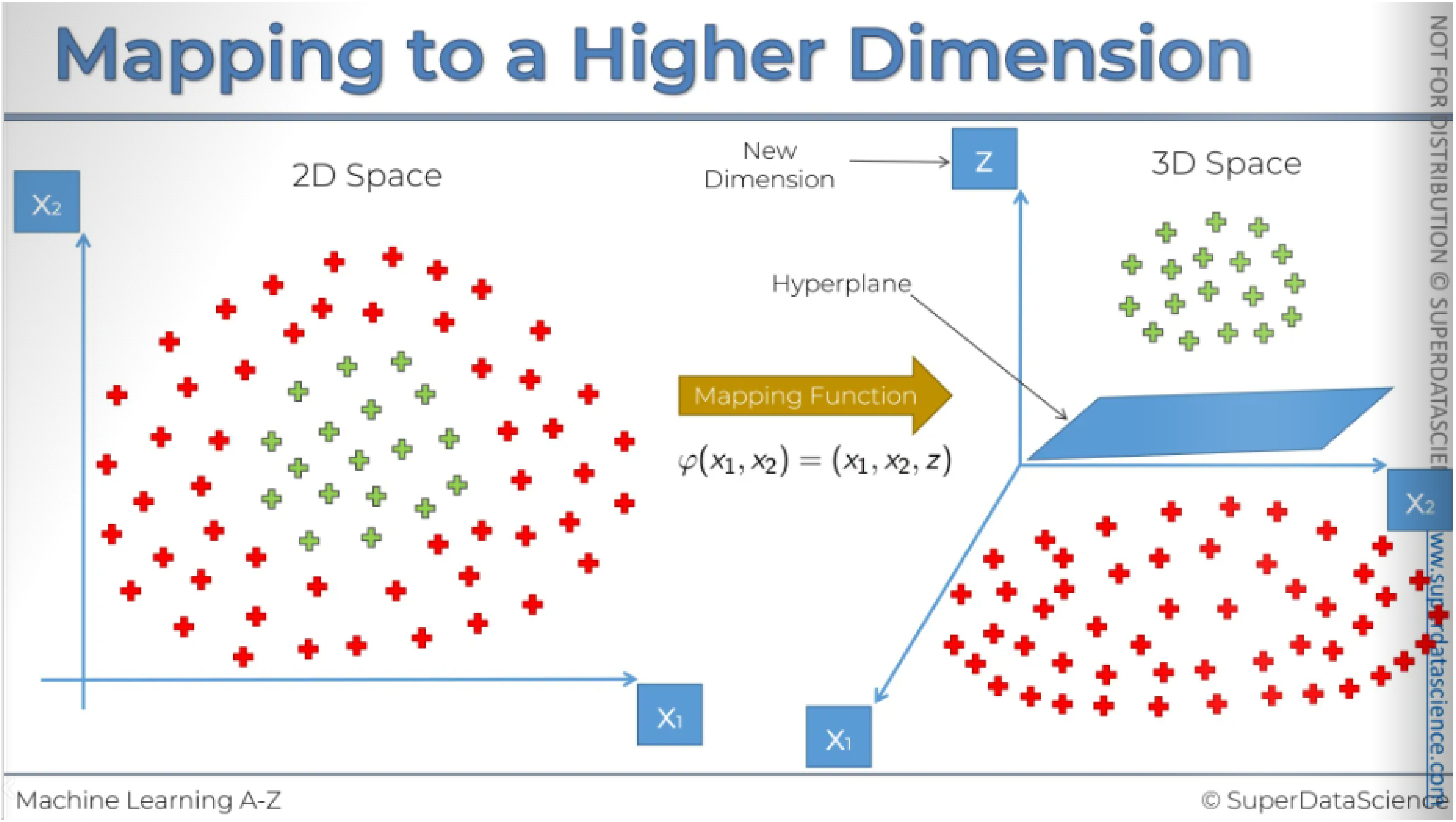


Mapping to a Higher Dimension



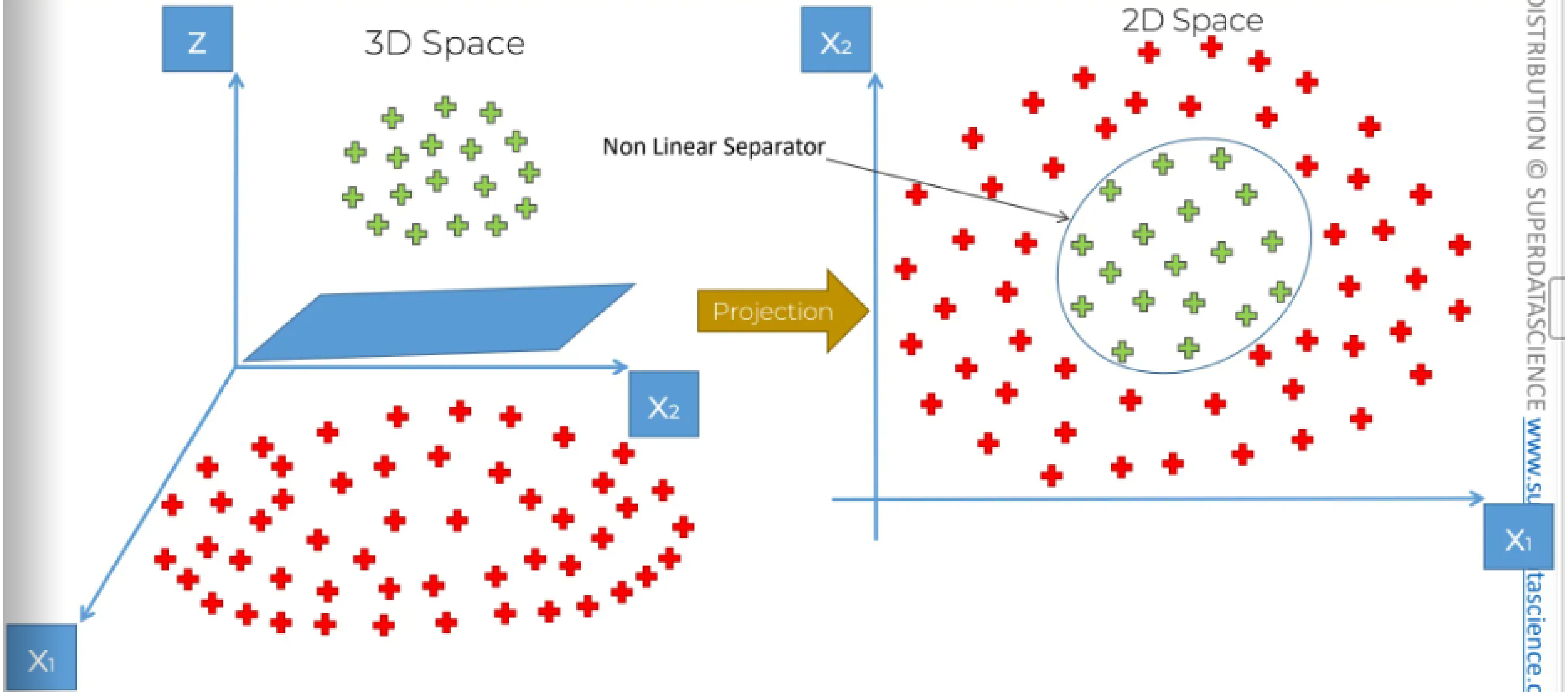
Mapping to a Higher Dimension





Mapping to a Higher Dimension
2D Space
X_2
X_1
New Dimension
Z
3D Space
Hyperplane
Mapping Function
$\varphi(x_1, x_2) = (x_1, x_2, z)$
X_2
X_1
NOT FOR DISTRIBUTION © SUPERDATASCIE
www.superdatascience.com
Machine Learning A-Z
© SuperDataScience

Projecting back to 2D Space

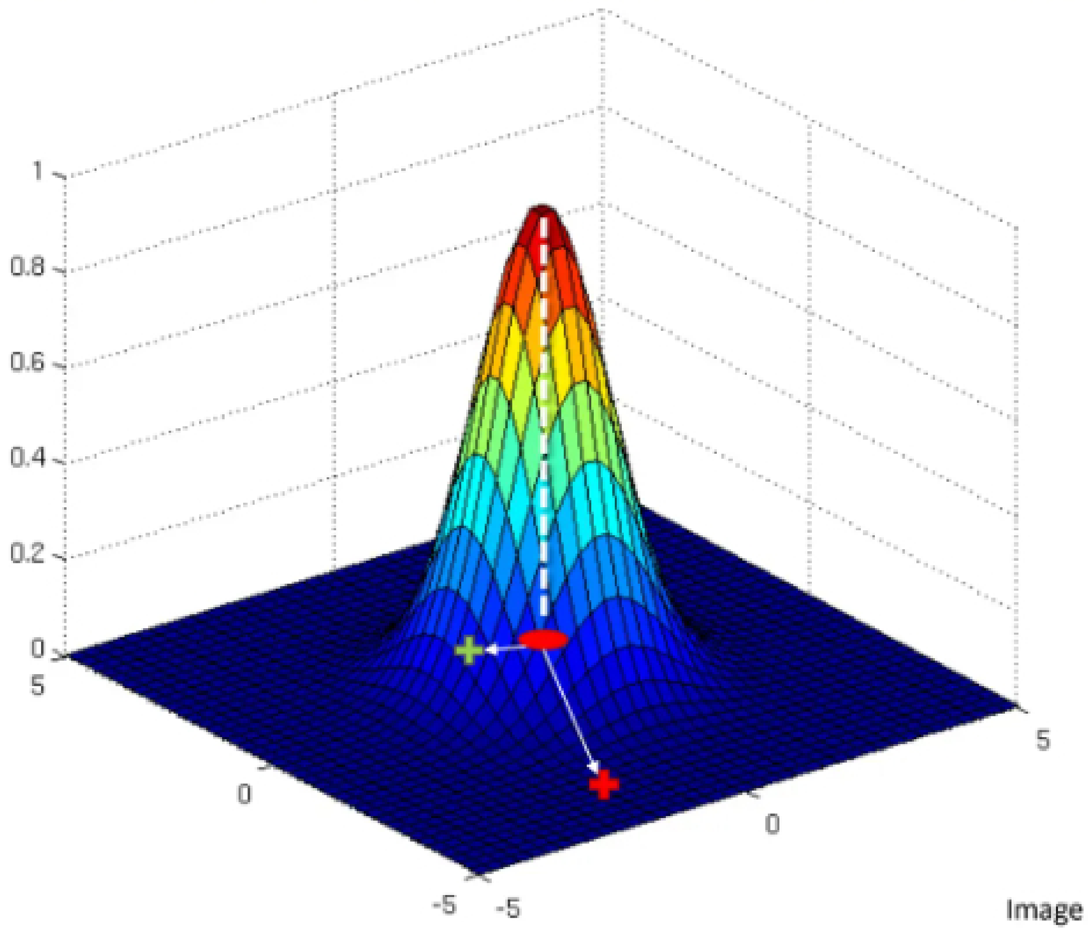


But there is a catch...

Mapping to a Higher Dimensional Space can be highly compute-intensive

The Kernel Trick

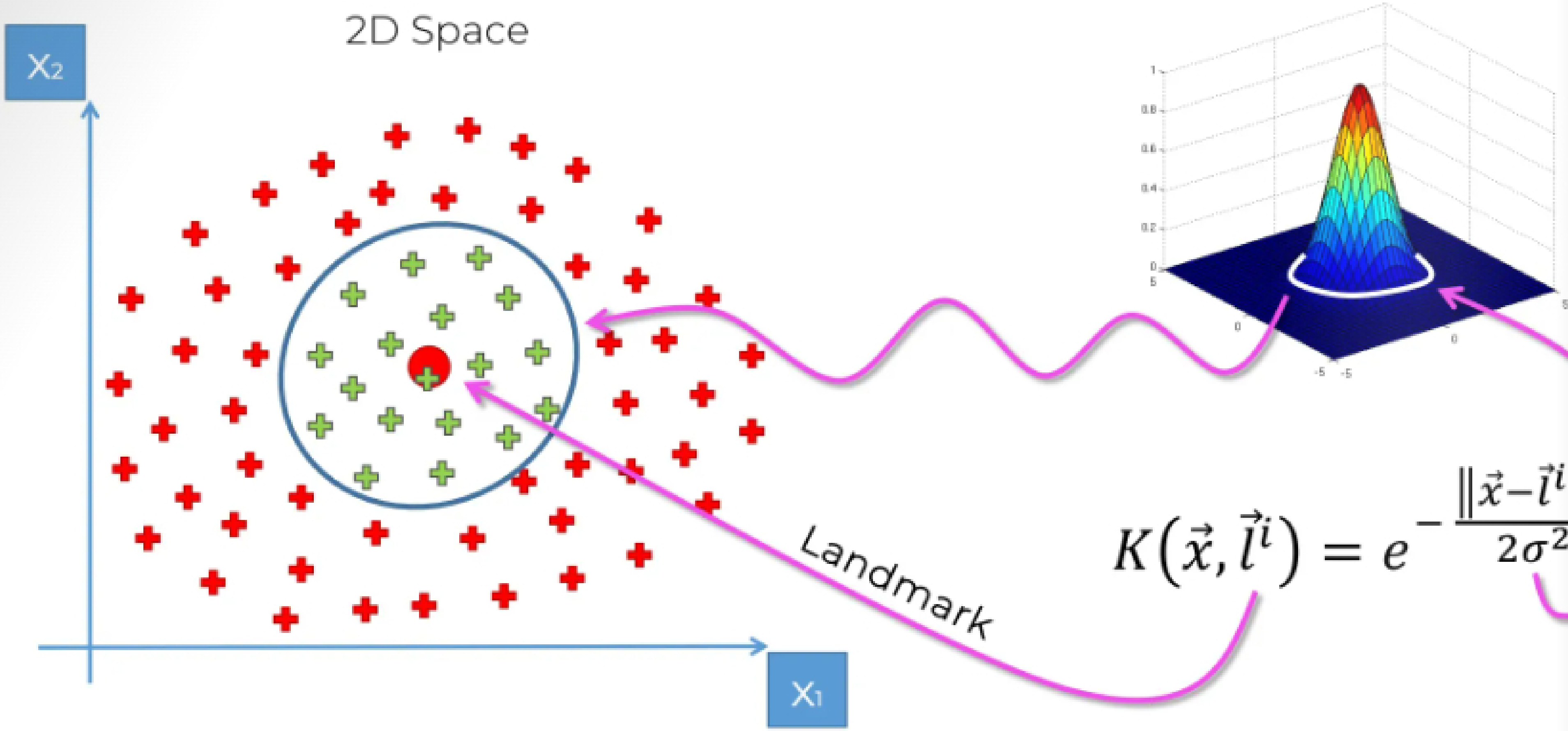
The Gaussian RBF Kernel



$$K(\vec{x}, \vec{l}^i) = e^{-\frac{\|\vec{x}-\vec{l}^i\|^2}{2\sigma^2}}$$

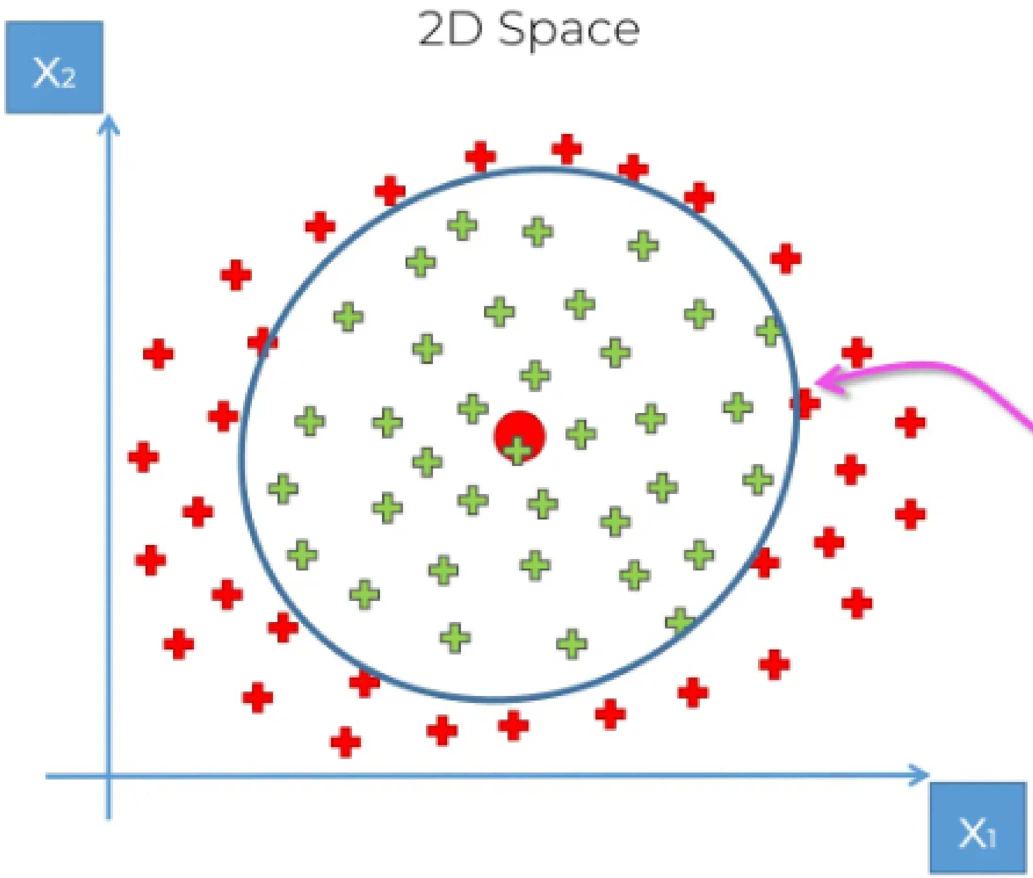
Image source: http://www.cs.toronto.edu/~duvenaud/cookbook/index.html

The Gaussian RBF Kernel

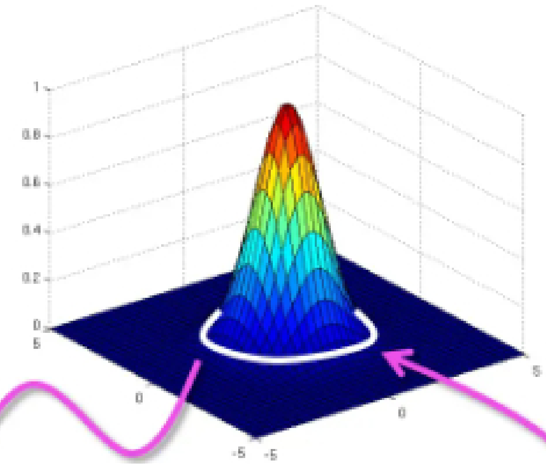


$$K(\vec{x},\vec{l}^i) = e^{-\frac{\|\vec{x}-\vec{l}^i\|^2}{2\sigma^2}}$$

The Gaussian RBF Kernel

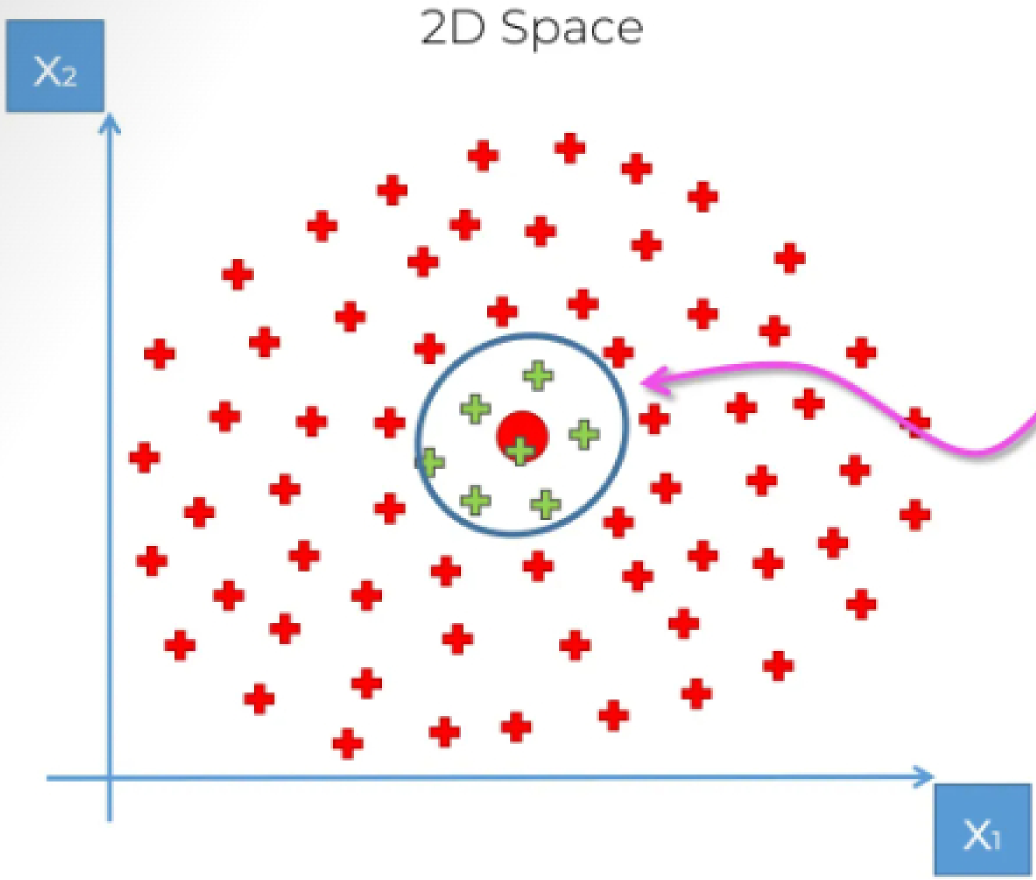


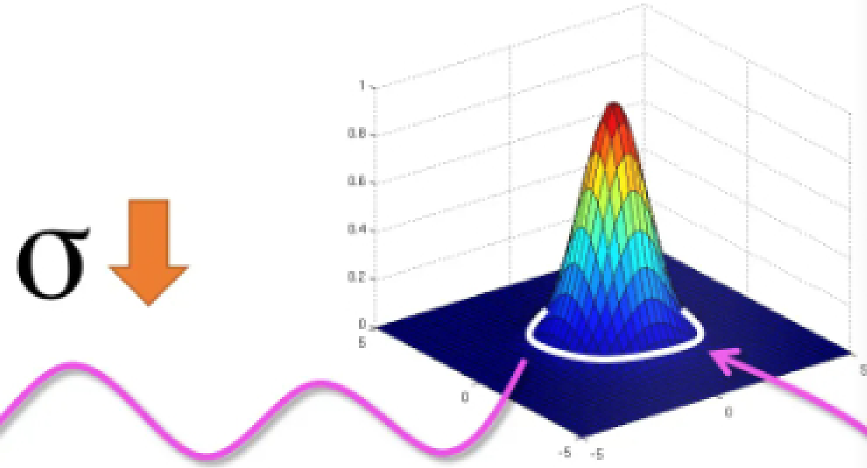
σ



$$K(\vec{x},\vec{l}^i) = e^{-\frac{\left\|\vec{x}-\vec{l}^i\right\|^2}{2\sigma^2}}$$

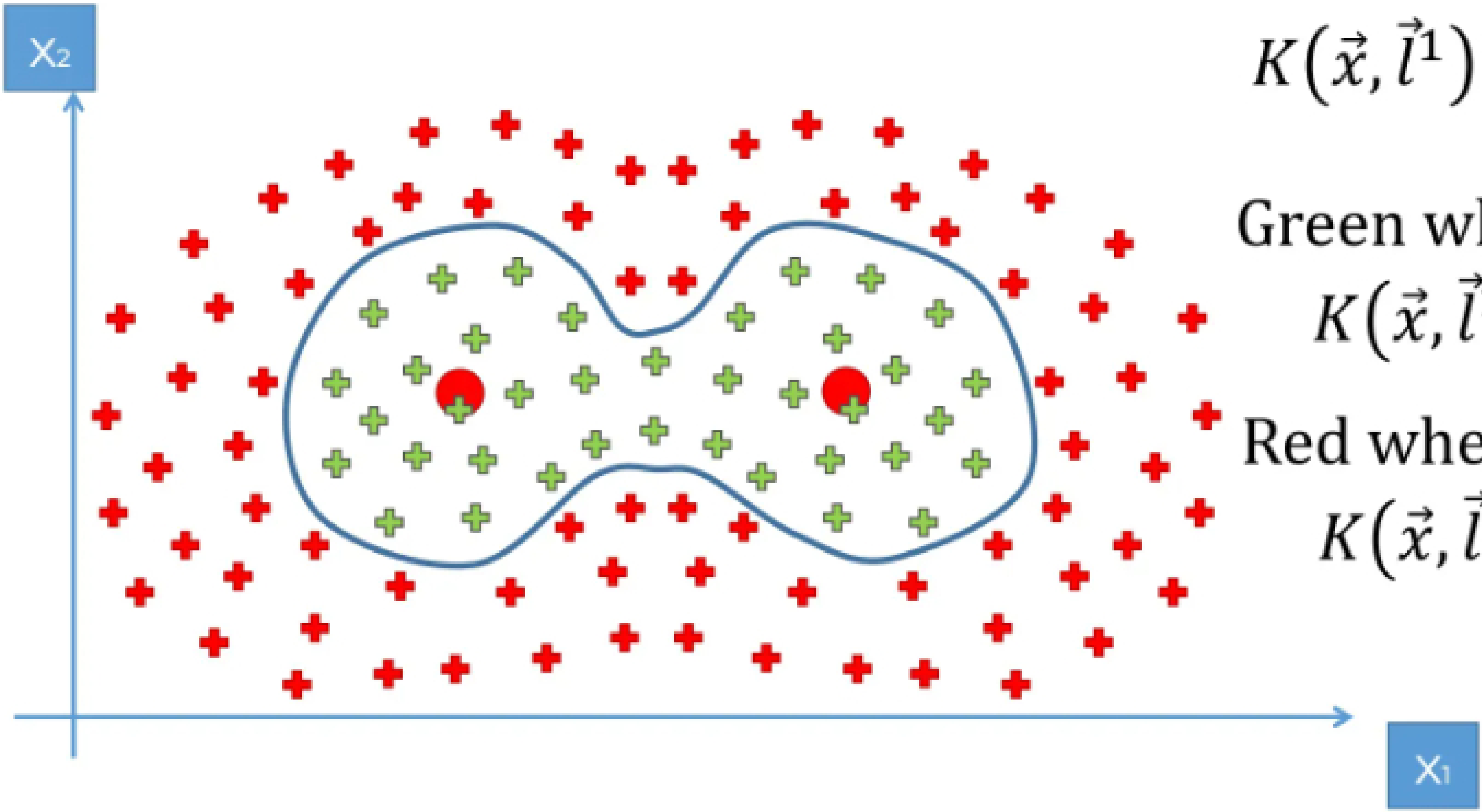
The Gaussian RBF Kernel





$$K(\vec{x}, \vec{l}^i) = e^{-\frac{\|\vec{x}-\vec{l}^i\|^2}{2\sigma^2}}$$

The Gaussian RBF Kernel



$$K(\vec{x},\vec{l}^1) + K(\vec{x},\vec{l}^2)$$

(Simplified Formula)

Green when:

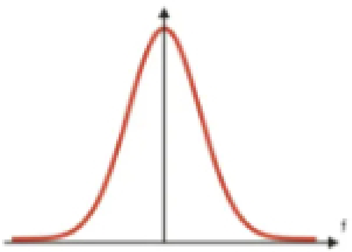
$$K(\vec{x},\vec{l}^1) + K(\vec{x},\vec{l}^2) >$$

Red when:

$$K(\vec{x},\vec{l}^1) + K(\vec{x},\vec{l}^2) =$$

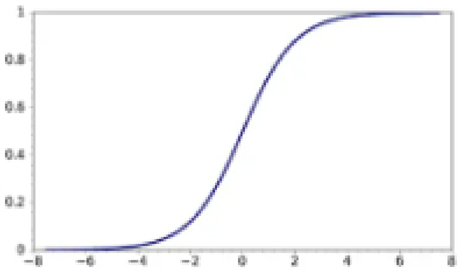
Types of Kernel Functions

Types of Kernel Functions



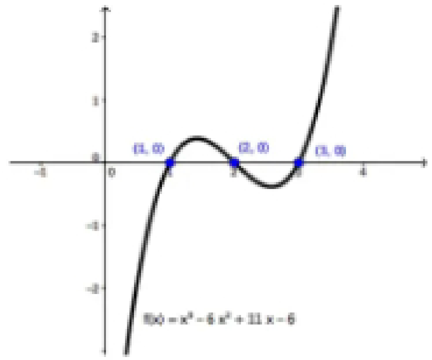
Gaussian RBF Kernel

$$K(\vec{x}, \vec{l}^i) = e^{-\frac{\|\vec{x}-\vec{l}^i\|^2}{2\sigma^2}}$$



Sigmoid Kernel

$$K(X,Y) = \tanh(\gamma \cdot X^T Y + r)$$



Polynomial Kernel

$$K(X,Y) = (\gamma \cdot X^T Y + r)^d, \gamma > 0$$

Non-Linear SVR (Advanced)

Heads-up about Non-Linear SVR

Section on SVR:

- SVR Intuition ✓

Section on SVM:

- SVM Intuition ✓

Section on Kernel SVM:

- Kernel SVM Intuition ✓
- Mapping to a higher dimension ✓
- The Kernel Trick ✓
- Types of Kernel Functions ✓
- Non-linear Kernel SVR

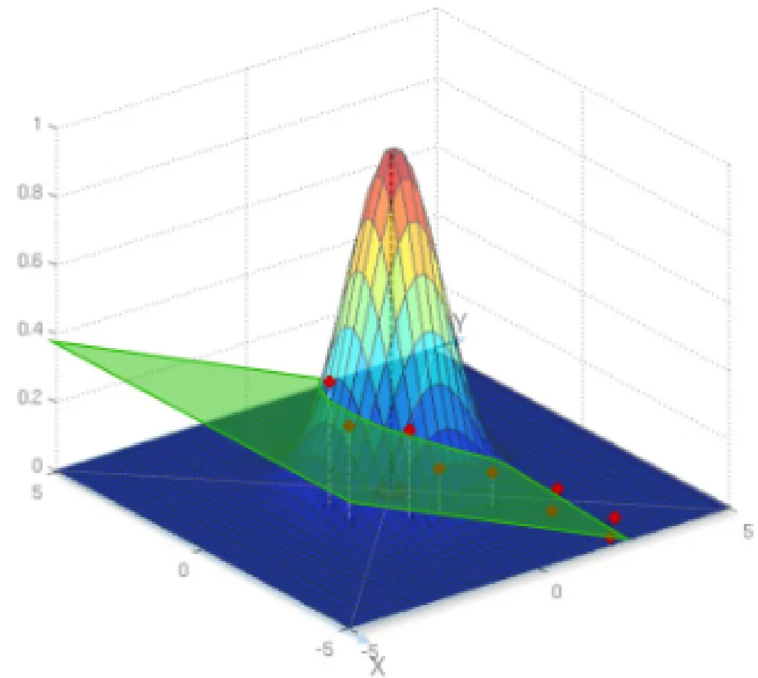
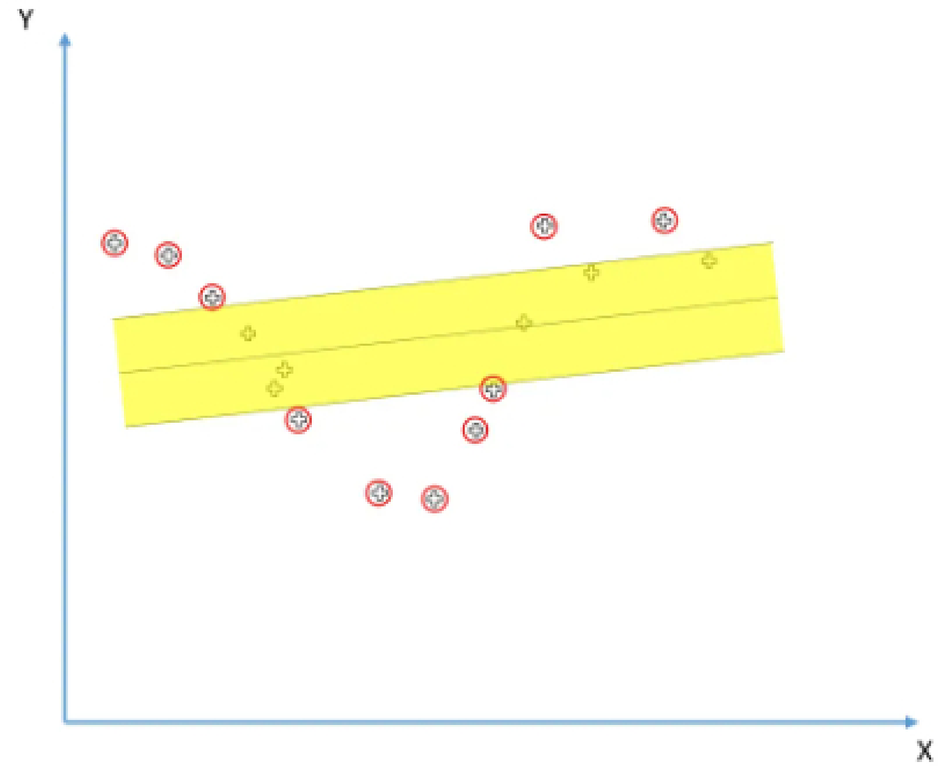


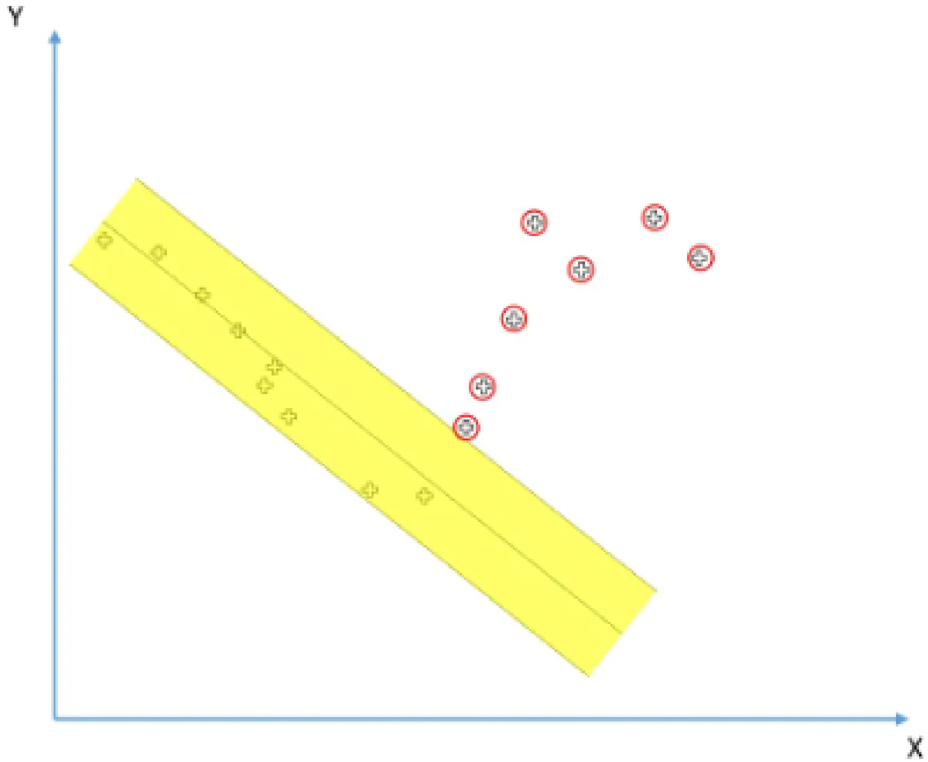
Image source: http://www.cs.toronto.edu/~duvenaud/cookbook/index.html

Non-Linear SVR

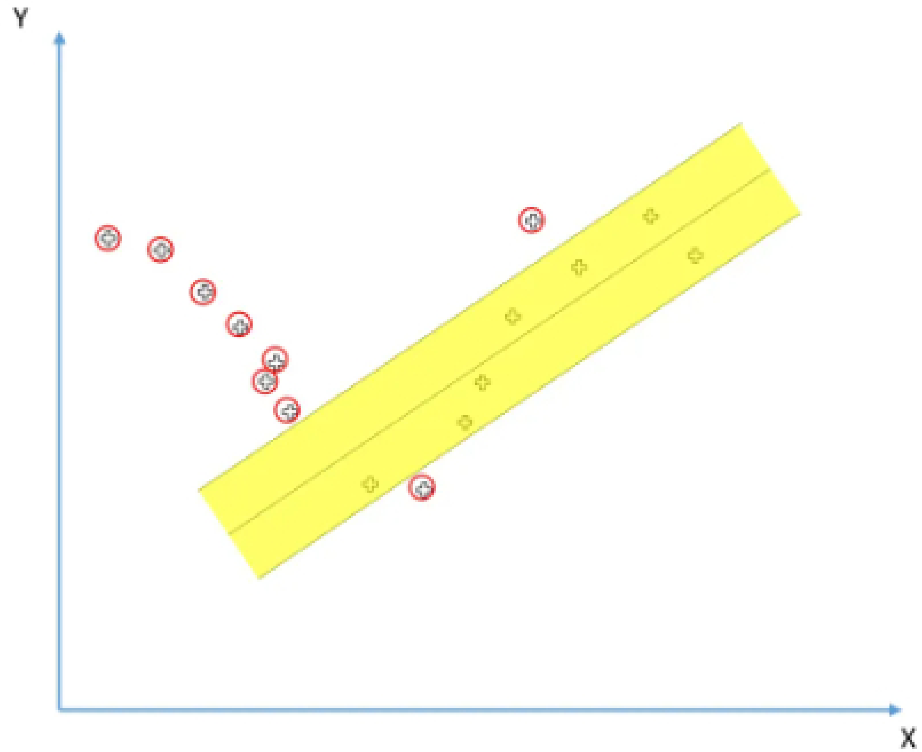
Non-Linear SVR



Non-Linear SVR

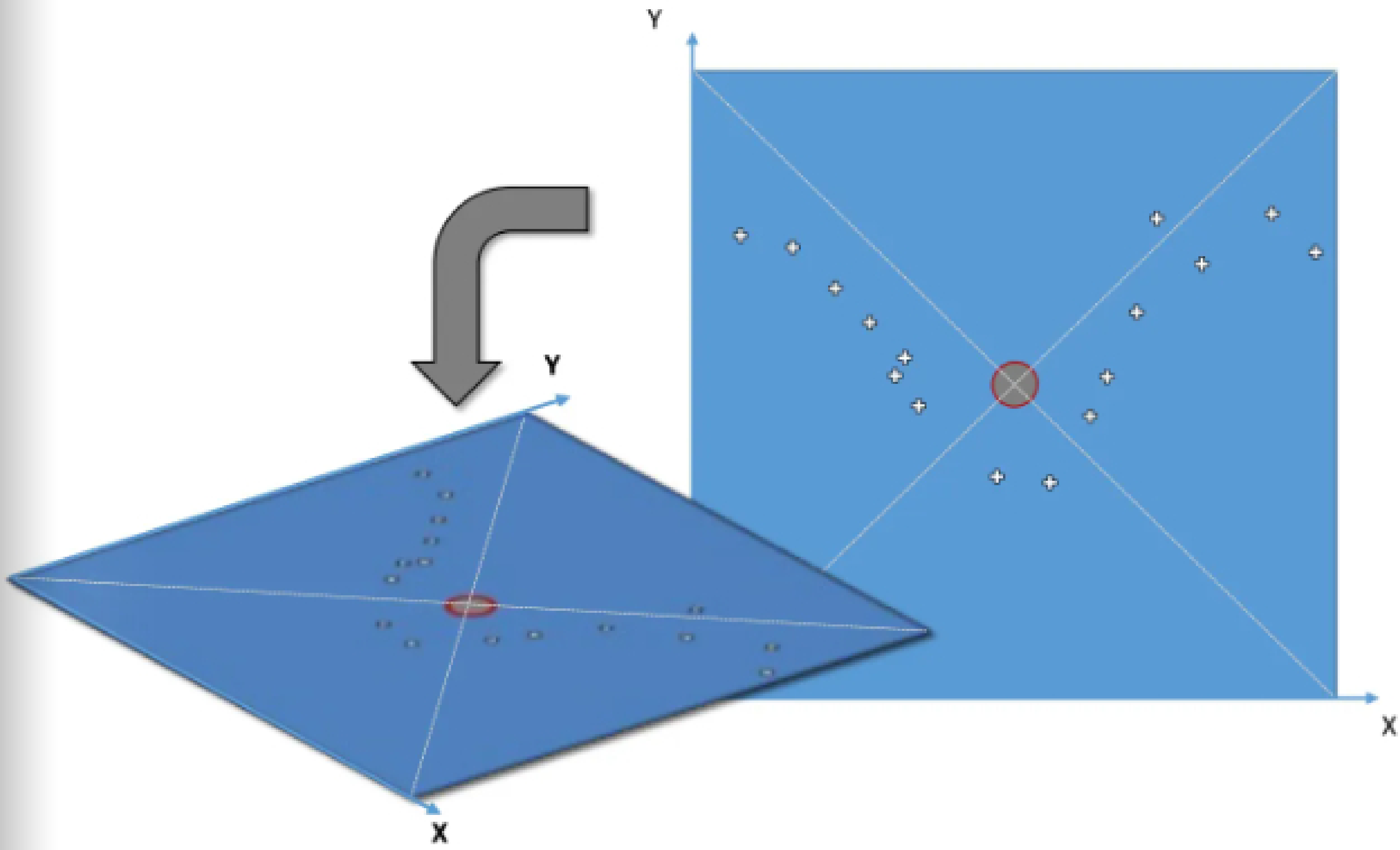


Non-Linear SVR

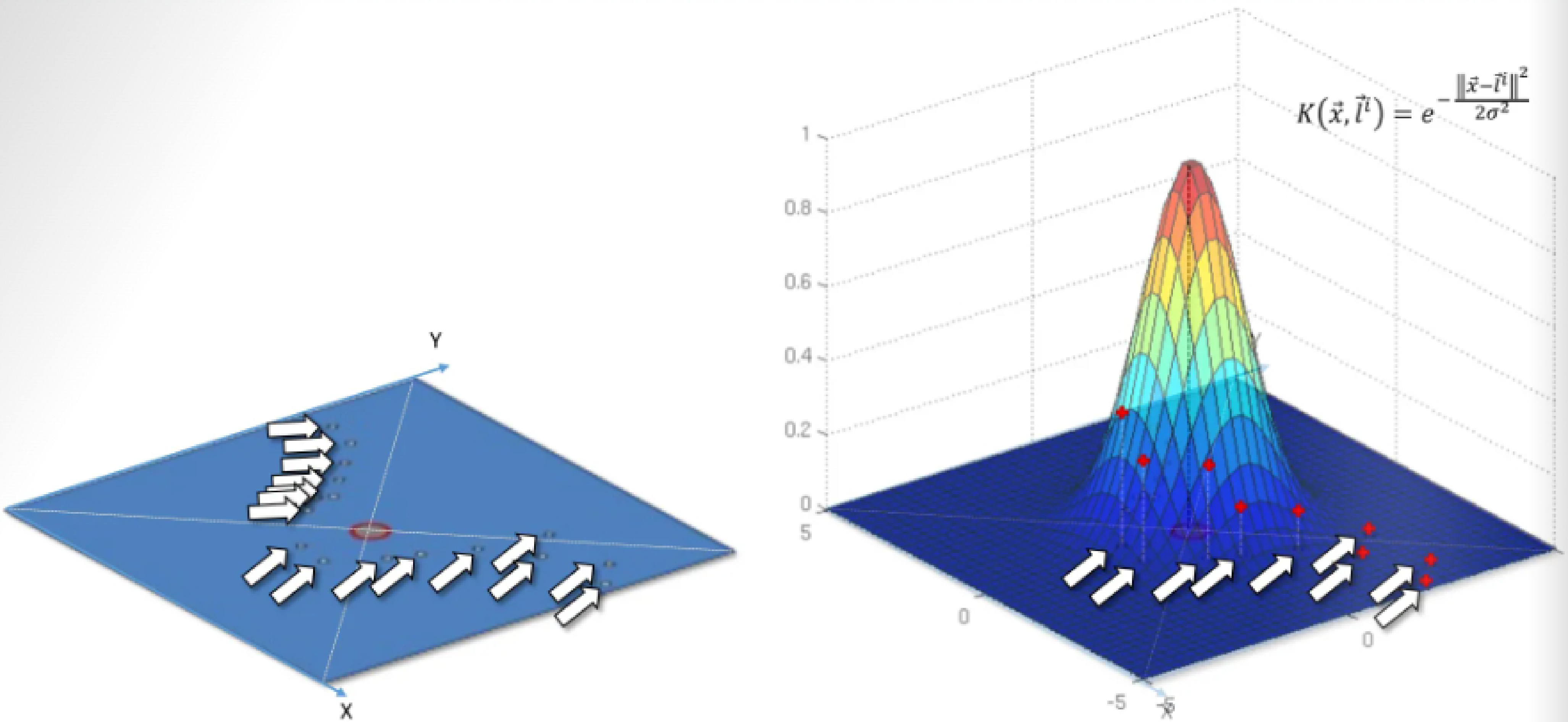


Non-Linear SVR

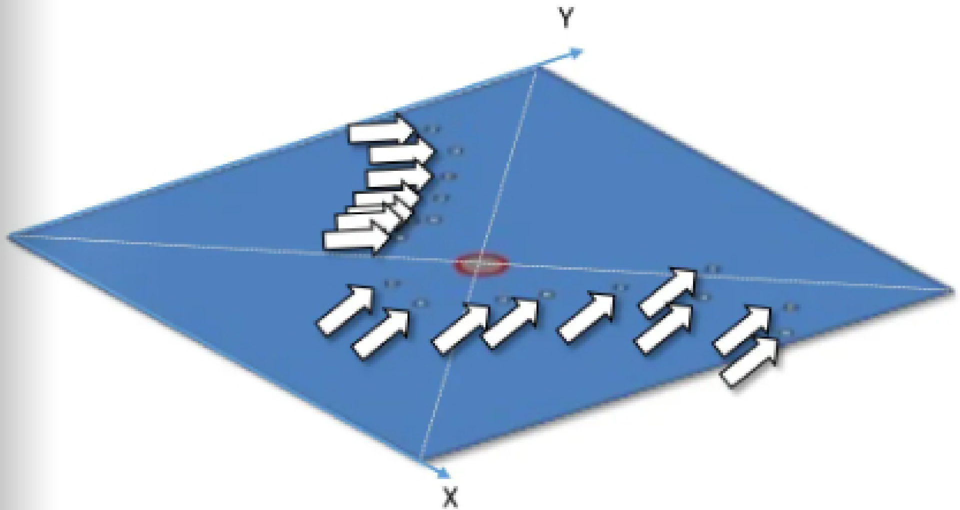
Non-Linear SVR

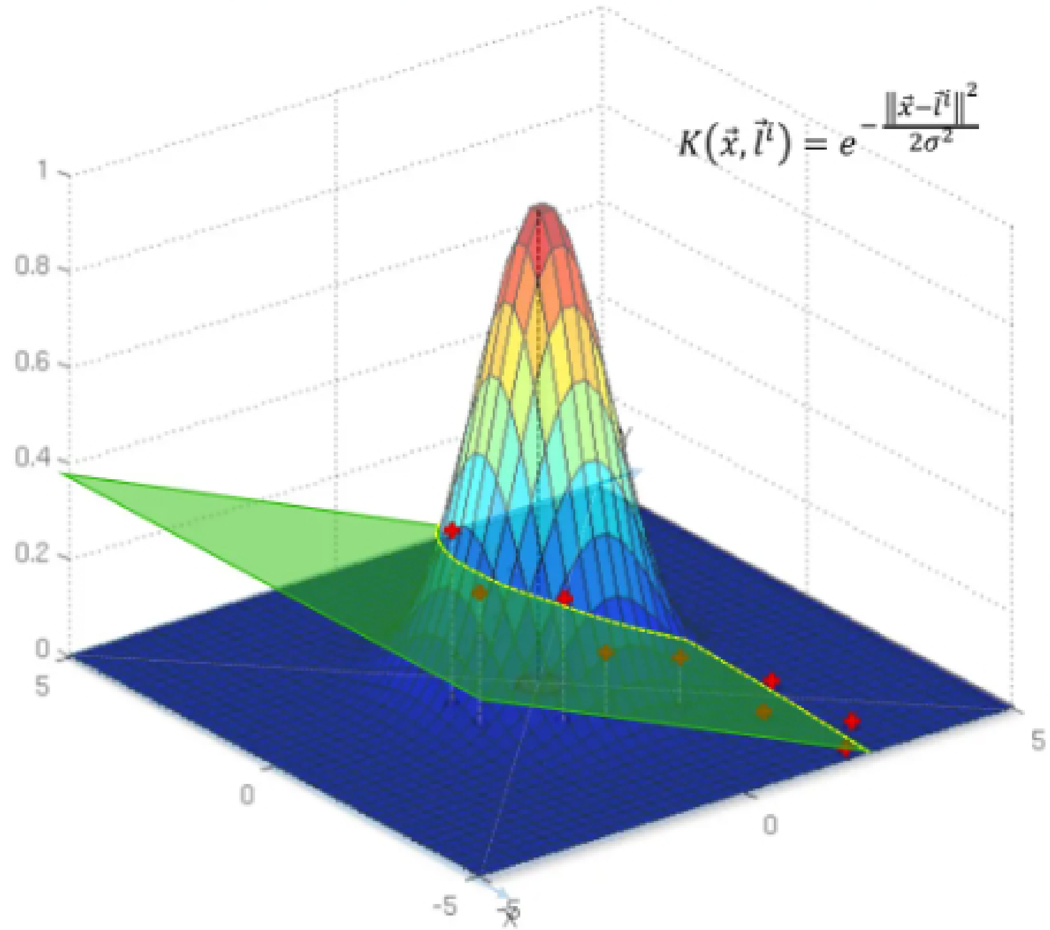


Non-Linear SVR

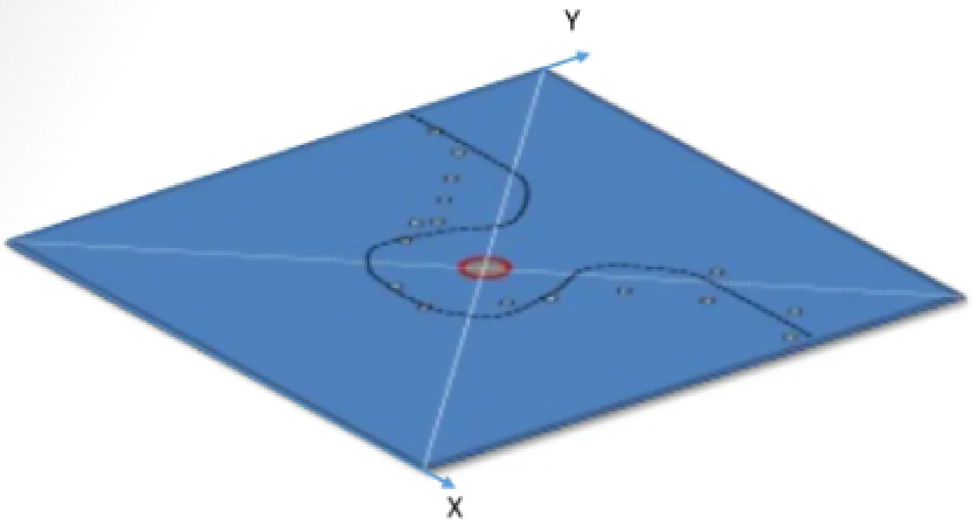


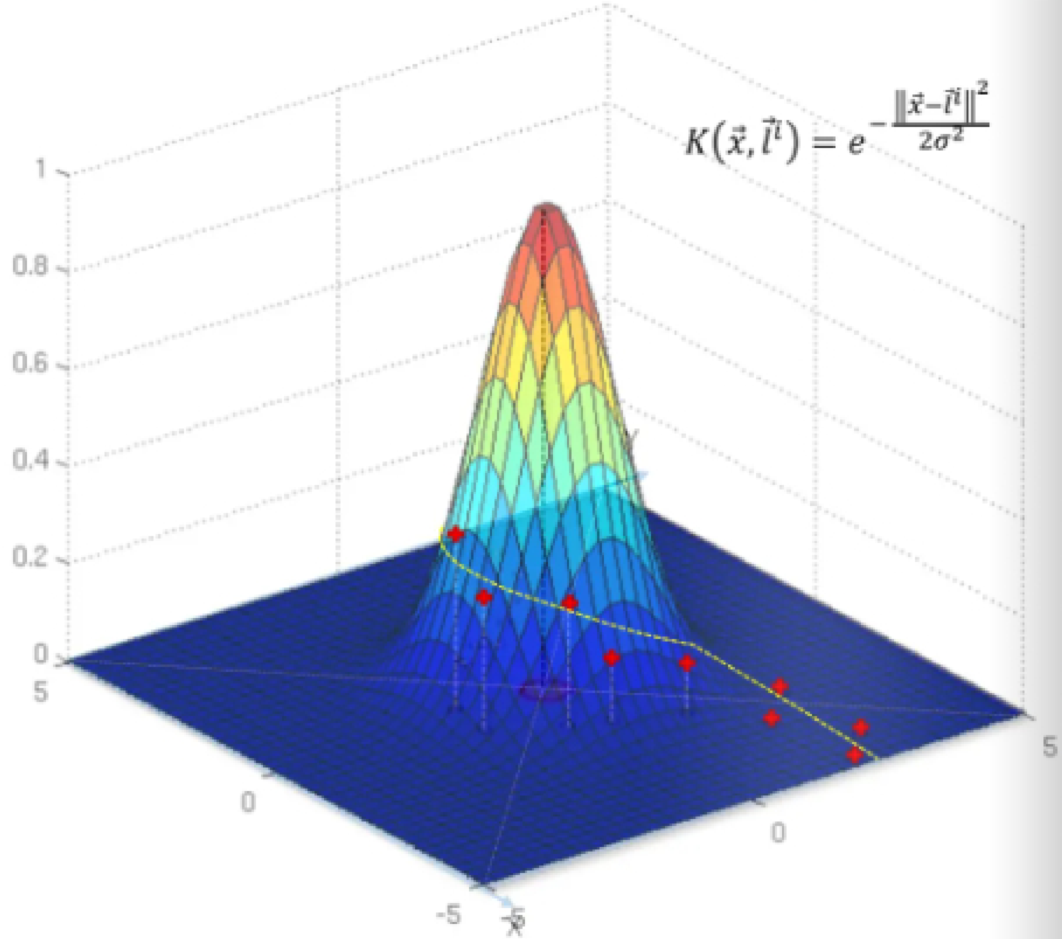
Non-Linear SVR



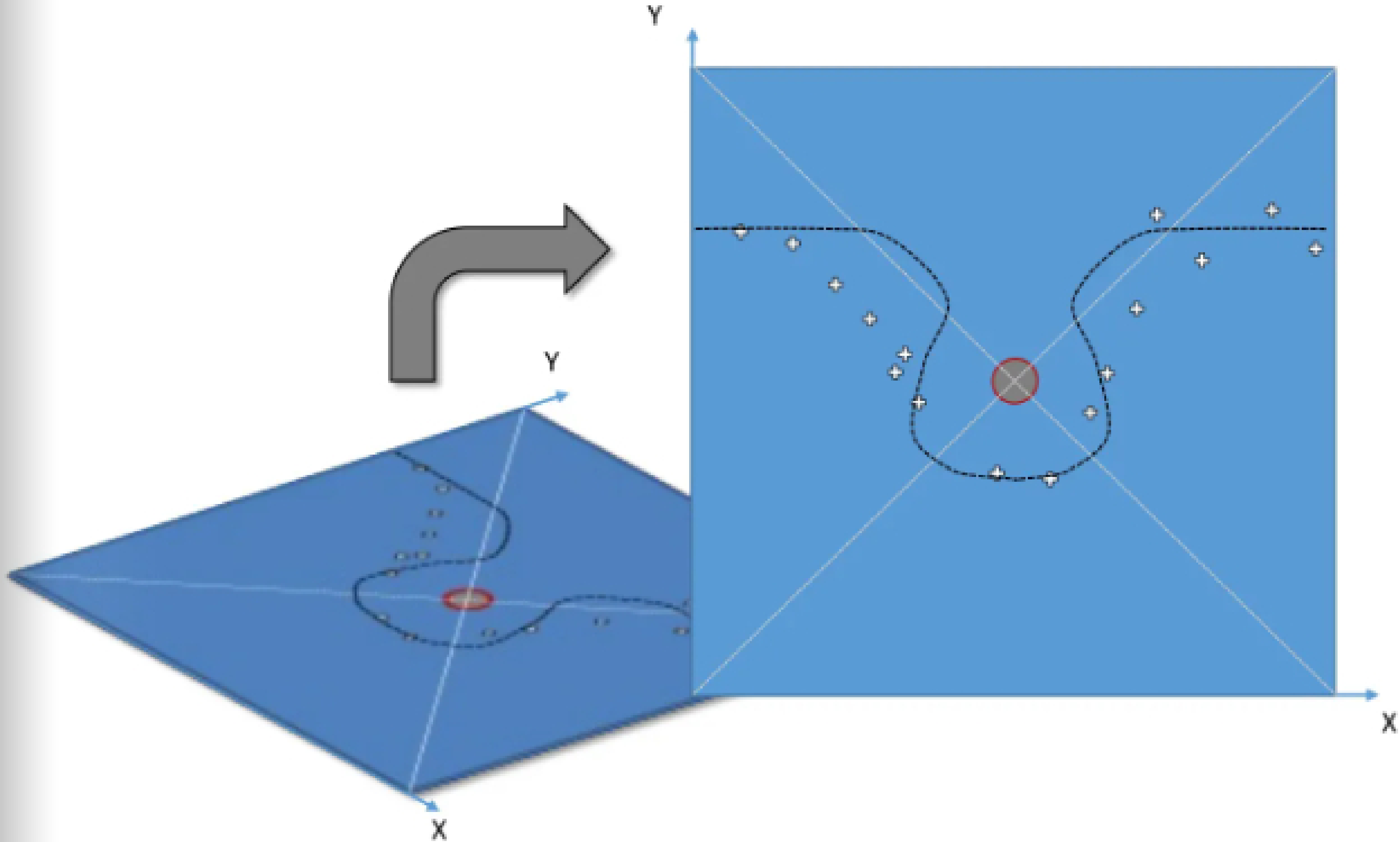


Non-Linear SVR

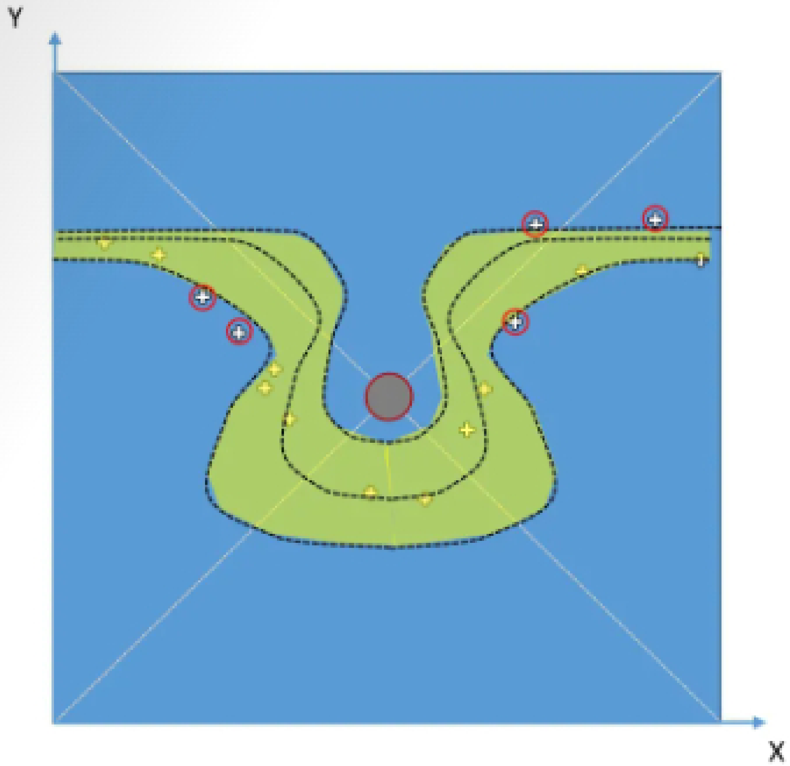


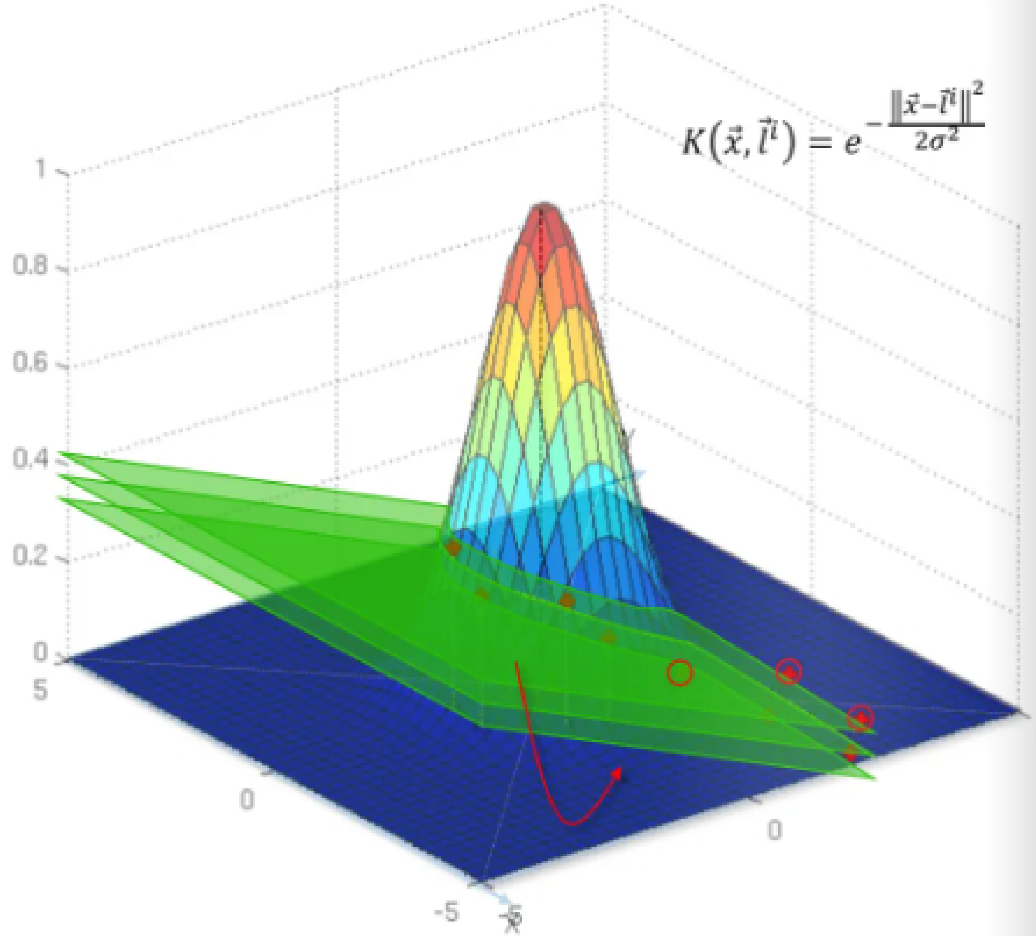


Non-Linear SVR



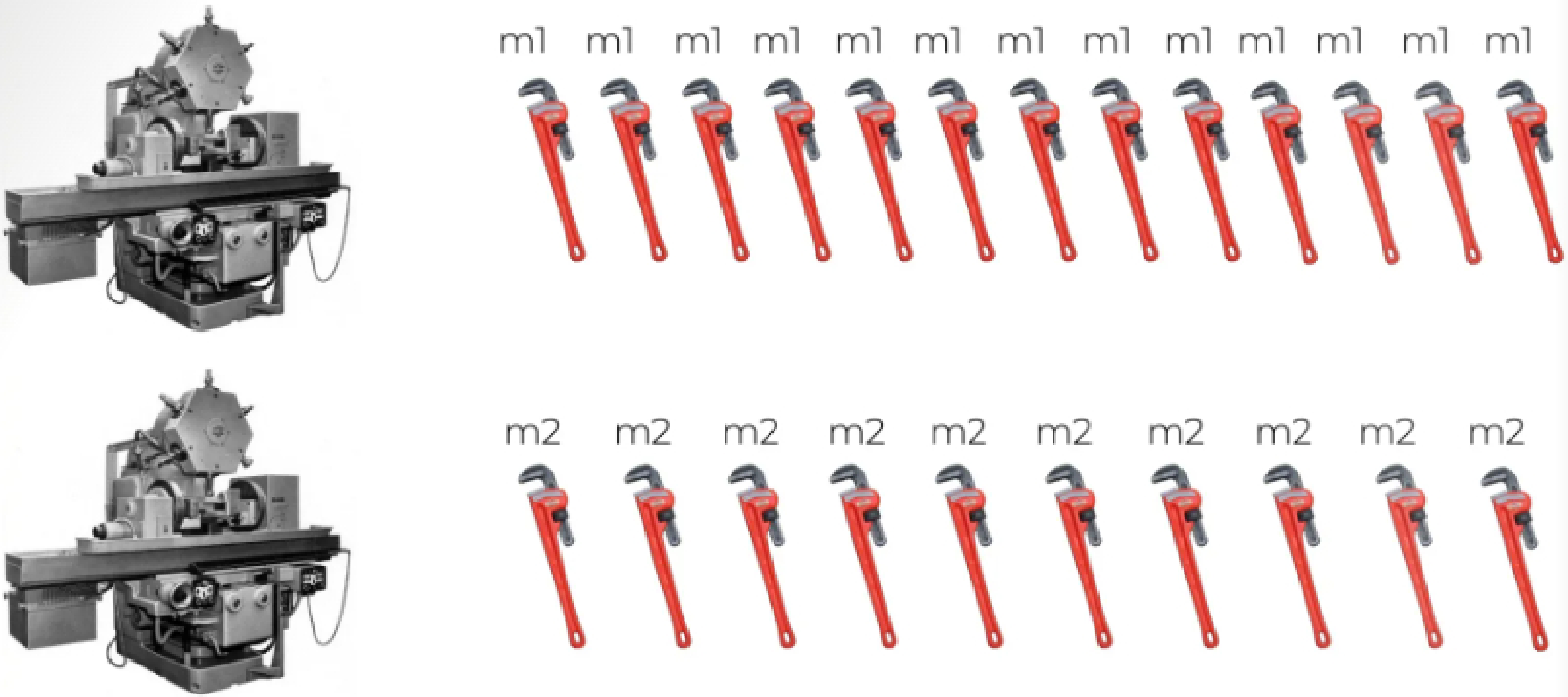
Non-Linear SVR



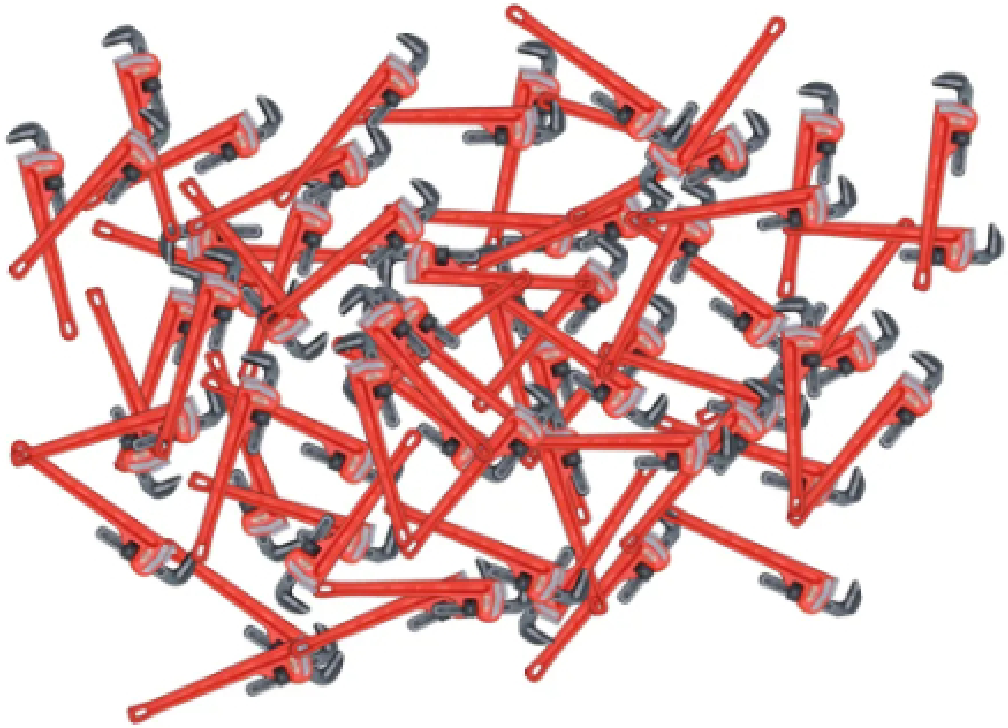


Bayes' Theorem

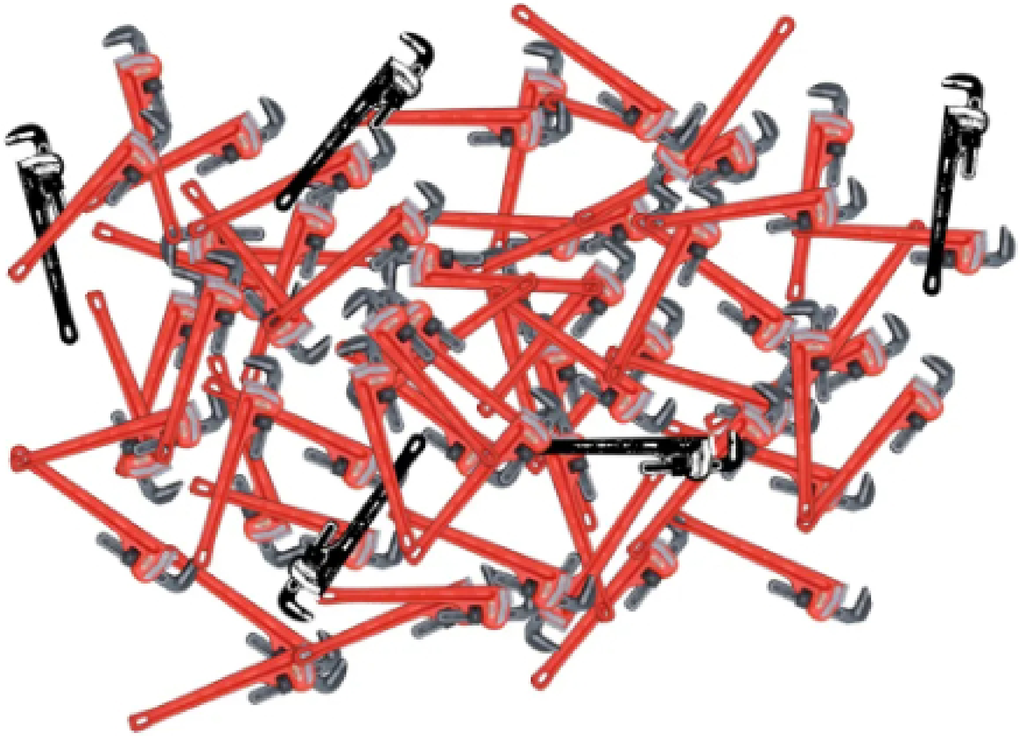
Bayes Theorem



Bayes Theorem

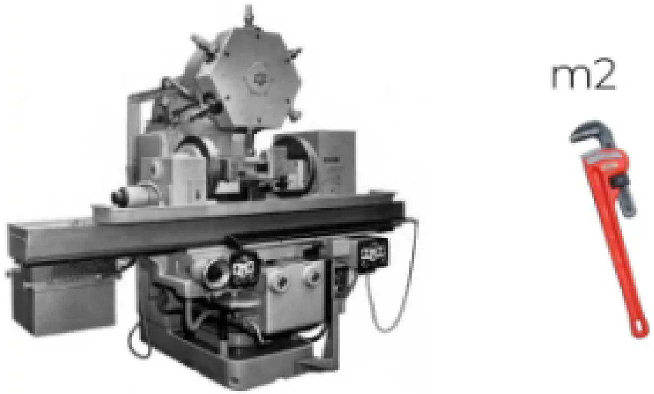


Bayes Theorem



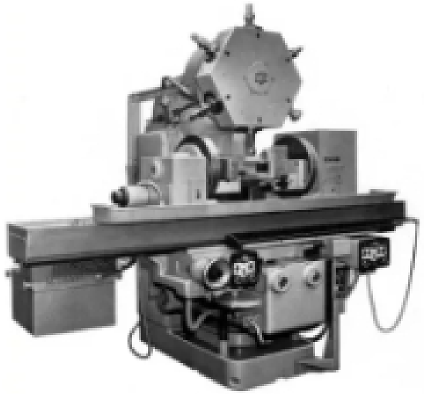
Bayes Theorem

What's the probability?



Bayes Theorem

What's the probability?



Bayes Theorem

$$P(A|B) = \frac{P(B|A) * P(A)}{P(B)}$$

Bayes Theorem

Mach1: 30 wrenches / hr
Mach2: 20 wrenches / hr

Out of all produced parts:
We can SEE that 1% are defective

Out of all defective parts:
We can SEE that 50% came from mach1
And 50% came from mach2

Question:
What is the probability that a part produced by mach2 is defective = ?

Bayes Theorem

Mach1: 30 wrenches / hr -> P(Mach1) = 30/50 = 0.6
Mach2: 20 wrenches / hr -> P(Mach2) = 20/50 = 0.4

Out of all produced parts:
We can SEE that 1% are defective -> P(Defect) = 1%

Out of all defective parts:
We can SEE that 50% came from mach1 -> P(Mach1 | Defect) = 50%
And 50% came from mach2 -> P(Mach2 | Defect) = 50%

Question:
What is the probability that a part produced by mach2 is defective = ? -> P(Defect | Mach2) = ?

Bayes Theorem

Mach1: 30 wrenches / hr
Mach2: 20 wrenches / hr

~~-> P(Mach1) = 30/50 = 0.6~~
-> P(Mach2) = 20/50 = 0.4

Out of all produced parts:
We can SEE that 1% are defective

-> P(Defect) = 1%

Out of all defective parts:
We can SEE that 50% came from mach1
And 50% came from mach2

~~-> P(Mach1 | Defect) = 50%~~
-> P(Mach2 | Defect) = 50%

Question:
What is the probability that a part produced by mach2 is defective = ?

-> P(Defect | Mach2) = ?

Bayes Theorem

Mach1: 30 wrenches / hr
Mach2: 20 wrenches / hr
Out of all produced parts:
We can SEE that 1% are defective
Out of all defective parts:
We can SEE that 50% came from mach1
And 50% came from mach2
Question:
What is the probability that a part produced by mach2 is defective = ?

-> P(Mach2) = 20/50 = 0.4
-> P(Defect) = 1%
-> P(Mach2 | Defect) = 50%
-> P(Defect | Mach2) = ?

Bayes Theorem

Mach1: 30 wrenches / hr
Mach2: 20 wrenches / hr
Out of all produced parts:
We can SEE that 1% are defective
Out of all defective parts:
We can SEE that 50% came from mach1
And 50% came from mach2
Question:
What is the probability that a part
produced by mach2 is defective = ?

-> P(Mach2) = 20/50 = 0.4
-> P(Defect) = 1%
-> P(Mach2 | Defect) = 50%
-> P(Defect | Mach2) = ?

$$P(\text{Defect} \mid \text{Mach2}) = \frac{P(\text{Mach2} \mid \text{Defect}) \ * \ P(\text{Defect})}{P(\text{Mach2})}$$

Bayes Theorem

Mach1: 30 wrenches / hr
Mach2: 20 wrenches / hr
Out of all produced parts:
We can SEE that 1% are defective
Out of all defective parts:
We can SEE that 50% came from mach1
And 50% came from mach2
Question:
What is the probability that a part produced by mach2 is defective = ?

-> P(Mach2) = 20/50 = 0.4
-> P(Defect) = 1%
-> P(Mach2 | Defect) = 50%
-> P(Defect | Mach2) = ?

$$P(\text{Defect} \mid \text{Mach2}) = \frac{0.5 \quad * \quad 0.01}{0.4}$$

Bayes Theorem

Mach1: 30 wrenches / hr
Mach2: 20 wrenches / hr
Out of all produced parts:
We can SEE that 1% are defective
Out of all defective parts:
We can SEE that 50% came from mach1
And 50% came from mach2
Question:
What is the probability that a part
produced by mach2 is defective = ?

-> P(Mach2) = 20/50 = 0.4
-> P(Defect) = 1%
-> P(Mach2 | Defect) = 50%
-> P(Defect | Mach2) = ?

$$P(\text{Defect} \mid \text{Mach2}) = \frac{0.5 \quad * \quad 0.01}{0.4} = 0.0125 = 1.25\%$$

It's intuitive!

$$P(\text{Defect} \mid \text{Mach2}) = \frac{P(\text{Mach2} \mid \text{Defect}) * P(\text{Defect})}{P(\text{Mach2})} = 1.25\%$$

Let's look at an example:

- 1000 wrenches
- 400 came from Mach2
- 1% have a defect = 10
- of them 50% came from Mach2 = 5
- % defective parts from Mach2 = 5/400 = 1.25%

It's intuitive!

Obvious question:
If the items are labeled, why couldn't we just count the number of defective wrenches that came from Mach2 and divide by the total number that came from Mach2?

Bayes Theorem

Quick exercise:

P(Defect | Mach1) = ?

Bayes Theorem

Mach1: 30 wrenches / hr -> P(Mach1) = 30/50 = 0.6
Mach2: 20 wrenches / hr -> P(Mach2) = 20/50 = 0.4

Out of all produced parts:
We can SEE that 1% are defective -> P(Defect) = 1%

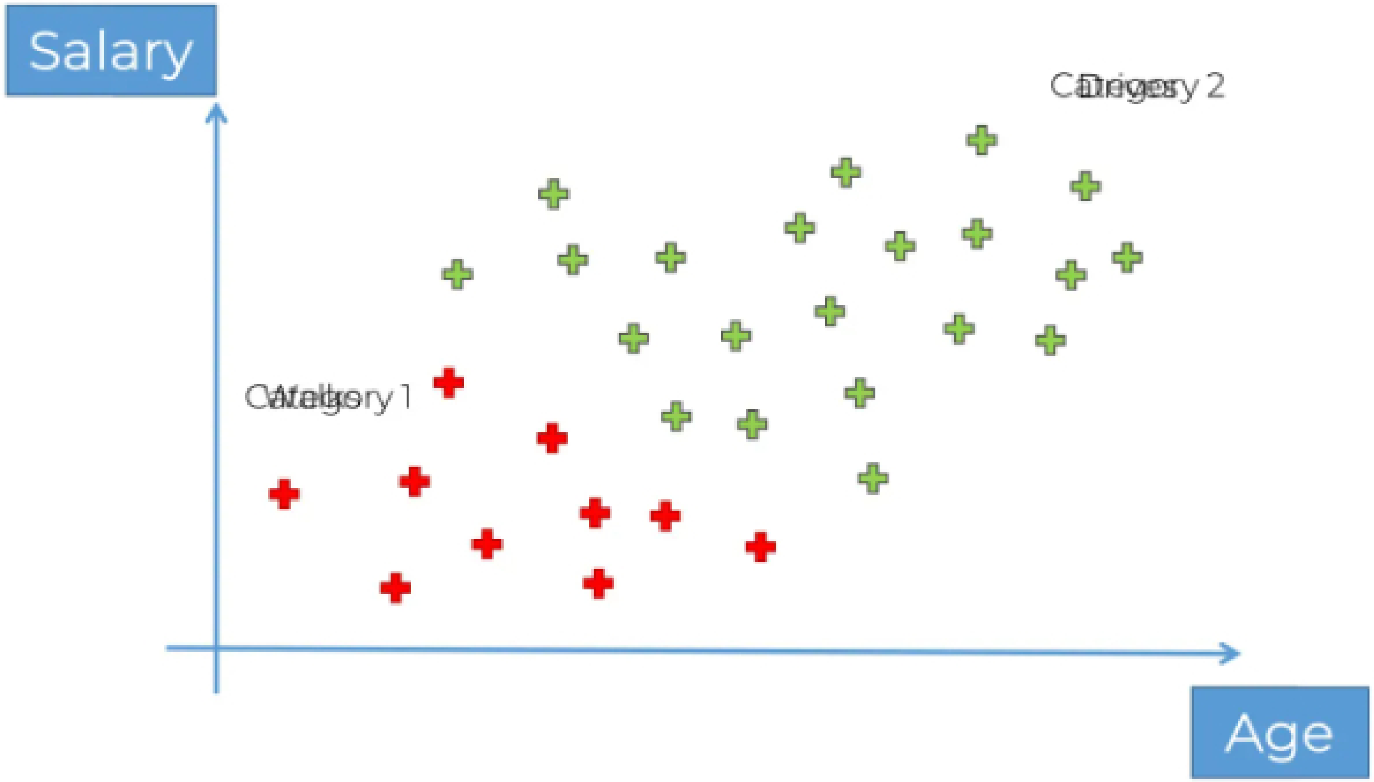
Out of all defective parts:
We can SEE that 50% came from mach1 -> P(Mach1 | Defect) = 50%
And 50% came from mach2 -> P(Mach2 | Defect) = 50%

Naïve Bayes Classifier Intuition

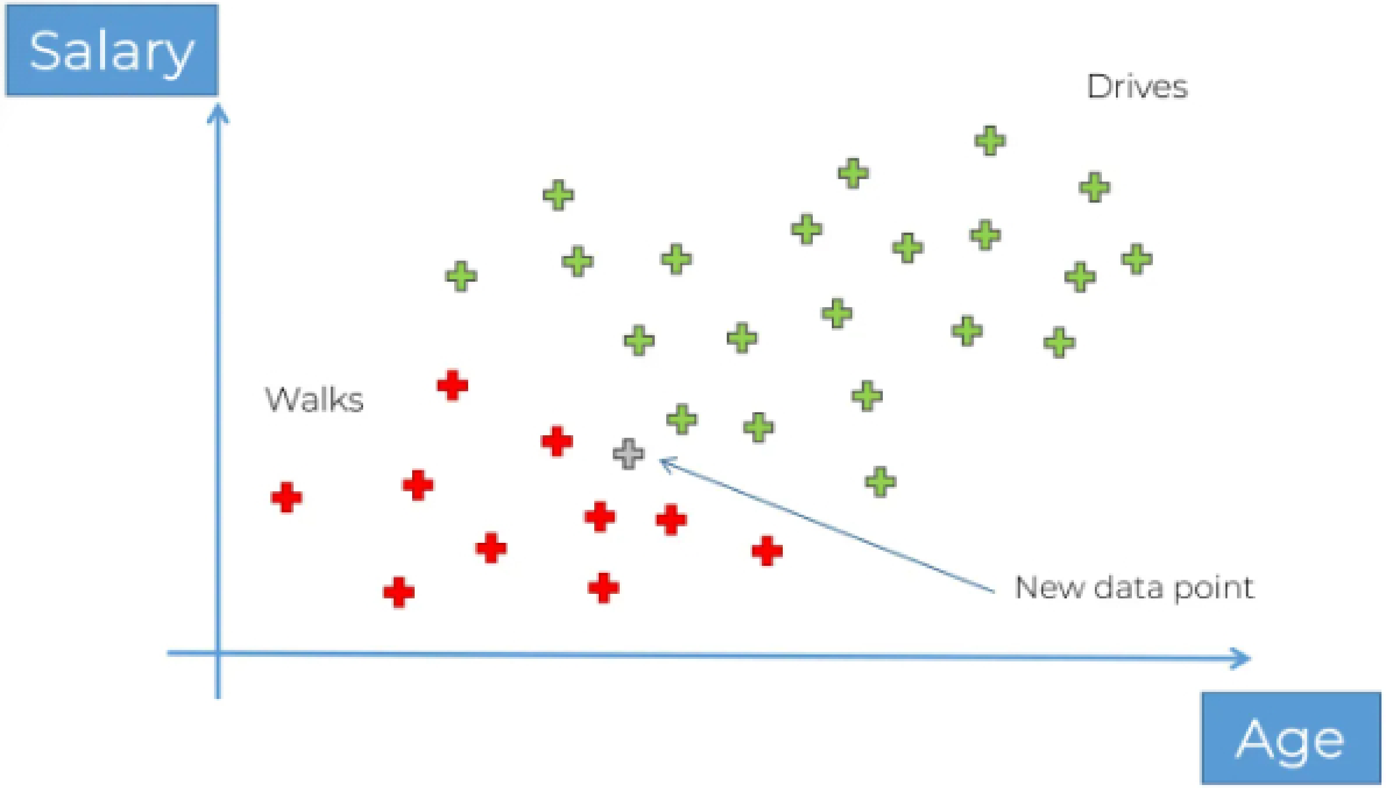
Naïve Bayes

$$P(A|B) = \frac{P(B|A) * P(A)}{P(B)}$$

Naïve Bayes



Naïve Bayes



Naïve Bayes

Plan of Attack

Naïve Bayes

$$P(A|B) = \frac{P(B|A) * P(A)}{P(B)}$$

Step 1

#4 Posterior Probability

#3 Likelihood

#1 Prior Probability

$$P(Walks|X) = \frac{P(X|Walks) * P(Walks)}{P(X)}$$

#2 Marginal Likelihood

Step 2

#4 Posterior Probability

#3 Likelihood

#1 Prior Probability

$$P(Drives|X) = \frac{P(X|Drives) * P(Drives)}{P(X)}$$

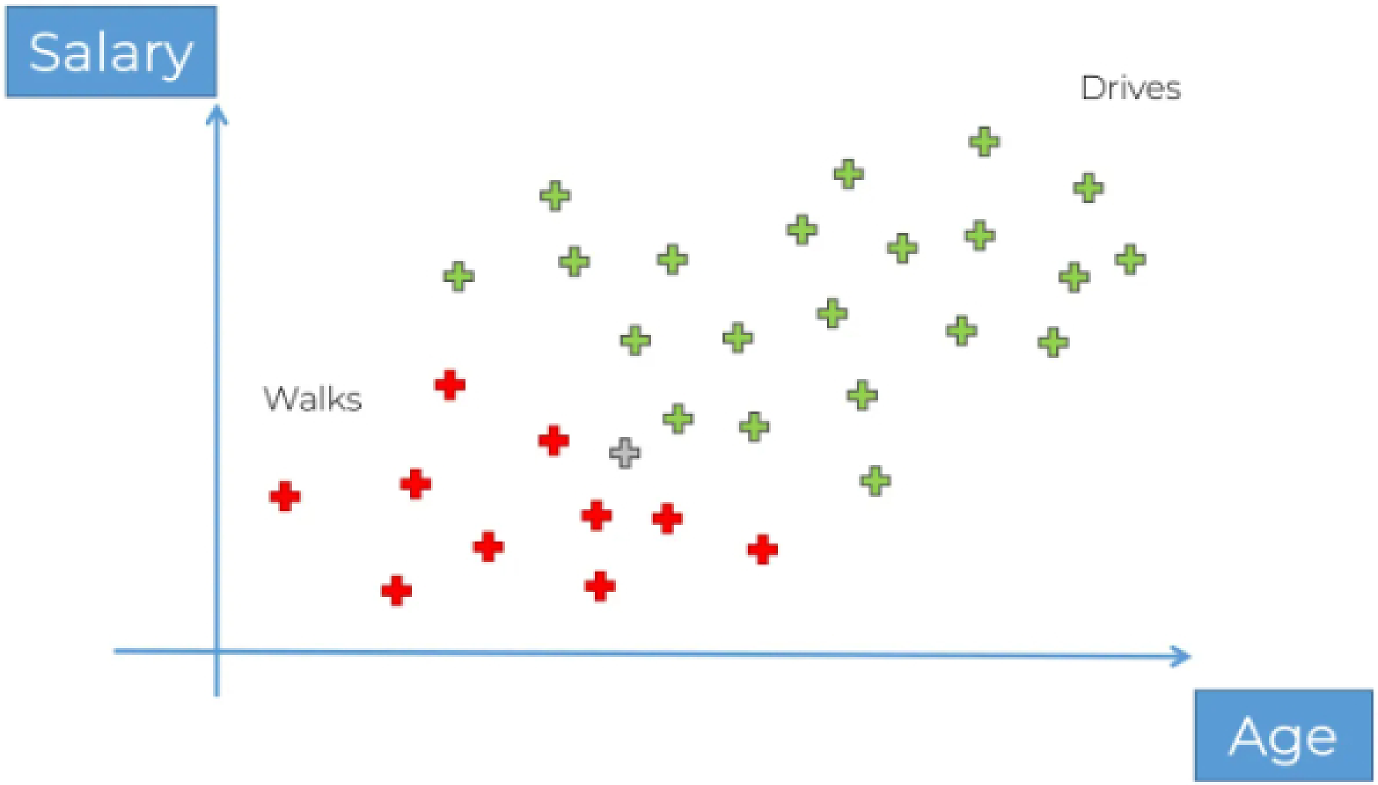
#2 Marginal Likelihood

Step 3

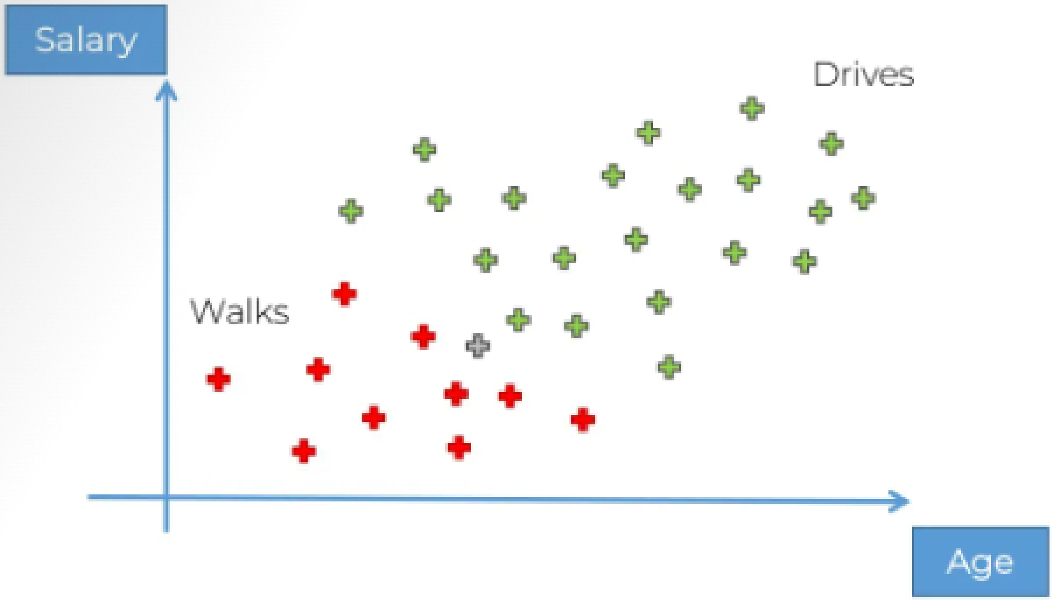
$$P(Walks|X)\ v.s.\ P(Drives|X)$$

Ready?

Naïve Bayes: Step 1



Naïve Bayes: Step 1



#1. P(Walks)

$$P(Walks) = \frac{Number\ of\ Walkers}{Total\ Observations}$$

$$P(Walks) = \frac{10}{30}$$

Naïve Bayes: Step 1

#4 Posterior Probability

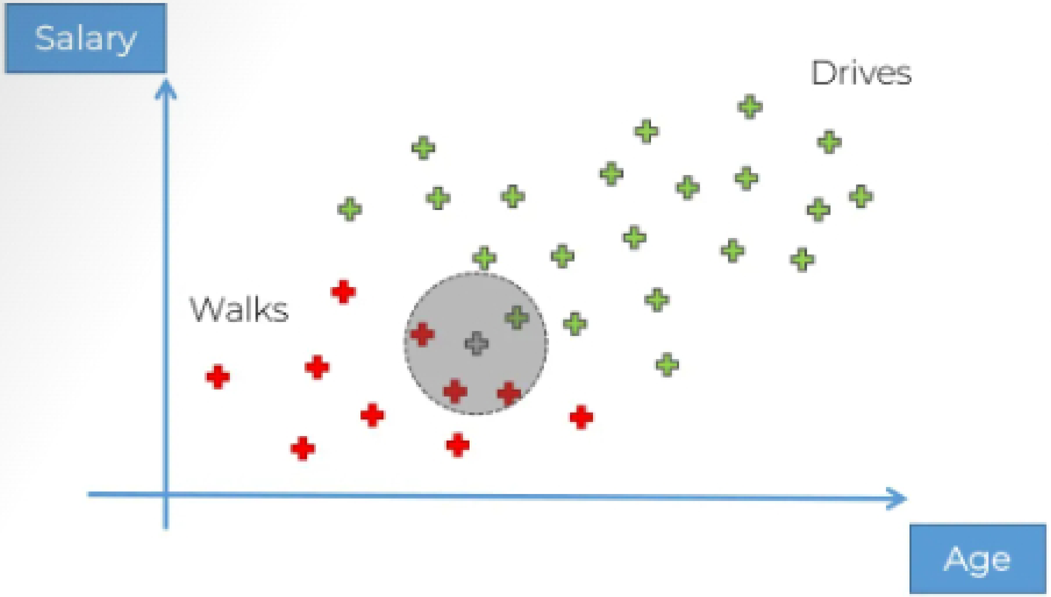
#3 Likelihood

#1 Prior Probability

$$P(Walks|X) = \frac{P(X|Walks) * P(Walks)}{P(X)}$$

#2 Marginal Likelihood

Naïve Bayes: Step 1



#2. P(X)

$$P(X) = \frac{Number\ of\ Similar\ Observations}{Total\ Observations}$$

$$P(X) = \frac{4}{30}$$

Naïve Bayes: Step 1

#4 Posterior Probability

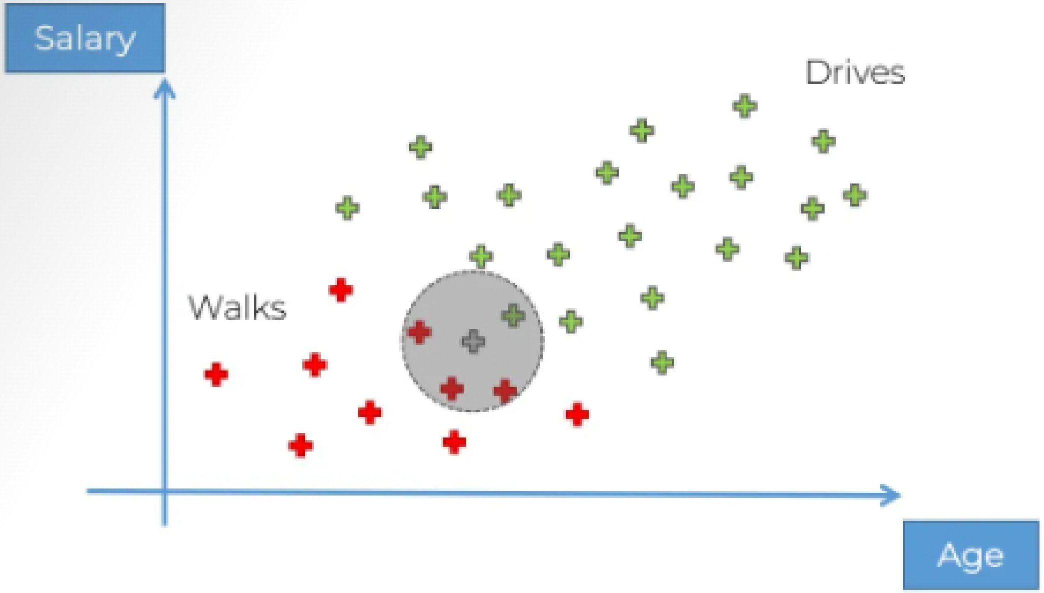
#3 Likelihood

#1 Prior Probability

$$P(Walks|X) = \frac{P(X|Walks) * P(Walks)}{P(X)}$$

#2 Marginal Likelihood

Naïve Bayes: Step 1



#3. P(X|Walks)

$$P(X|Walks) = \frac{\text{Number of Similar Observations Among those who Walk}}{\text{Total number of Walkers}}$$

$$P(X|Walks) = \frac{3}{10}$$

Naïve Bayes: Step 1

#4 Posterior Probability

#3 Likelihood

#1 Prior Probability

$$P(Walks|X) = \frac{P(X|Walks) * P(Walks)}{P(X)}$$

#2 Marginal Likelihood

Naïve Bayes: Step 1

#4 Posterior Probability

#3 Likelihood

#1 Prior Probability

$$P(Walks|X) = \frac{\frac{3}{10} * \frac{10}{30}}{\frac{4}{30}} = 0.75$$

#2 Marginal Likelihood

Naïve Bayes

Step 1 – Done.

Step 2

#4 Posterior Probability

#3 Likelihood

#1 Prior Probability

$$P(Drives|X) = \frac{P(X|Drives) * P(Drives)}{P(X)}$$

#2 Marginal Likelihood

Naïve Bayes: Step 2

#4 Posterior Probability

#3 Likelihood

#1 Prior Probability

$$P(Drives|X) = \frac{\frac{1}{20} * \frac{20}{30}}{\frac{4}{30}} = 0.25$$

#2 Marginal Likelihood

Step 2 – Done.

Step 3

$$P(Walks|X)\ v.s.\ P(Drives|X)$$

Step 3

$$0.75 \; v.s. \; 0.25$$

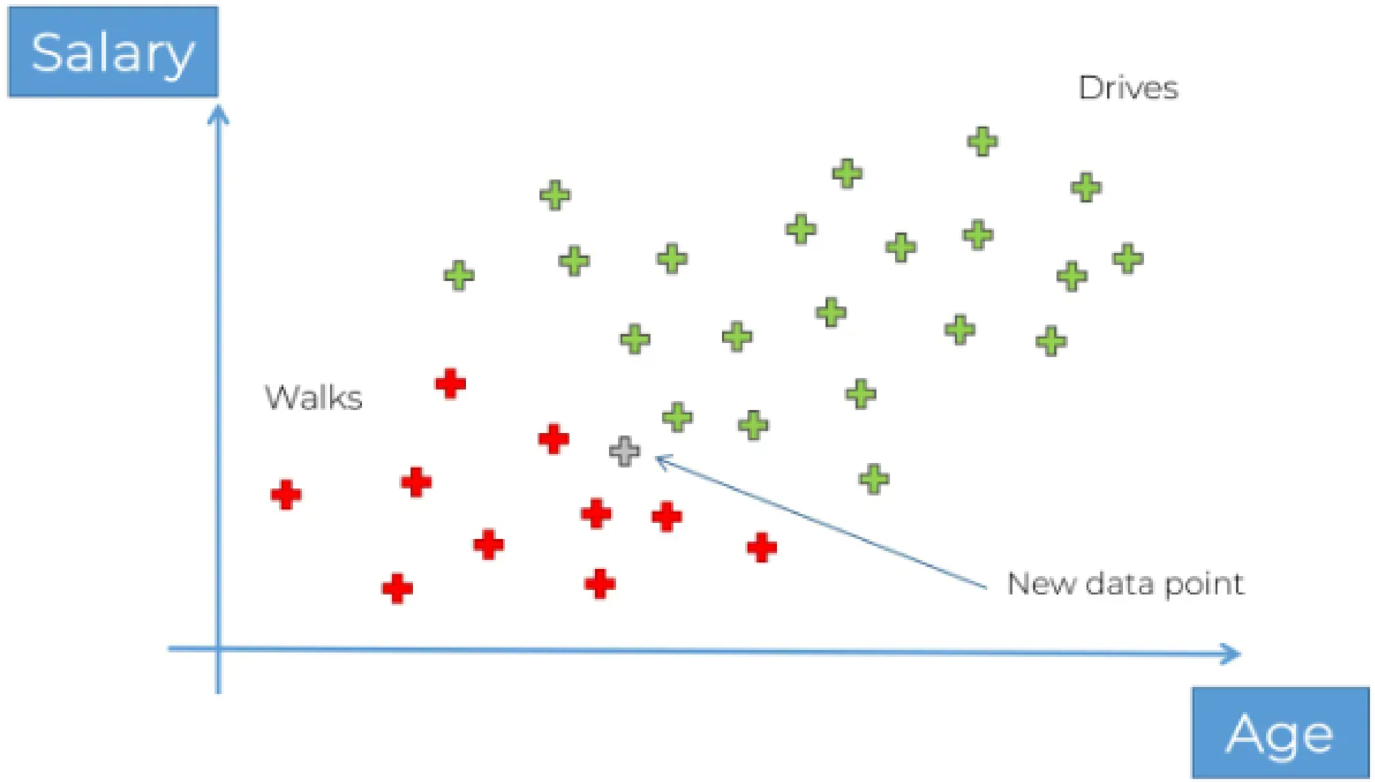
Step 3

$$0.75 > 0.25$$

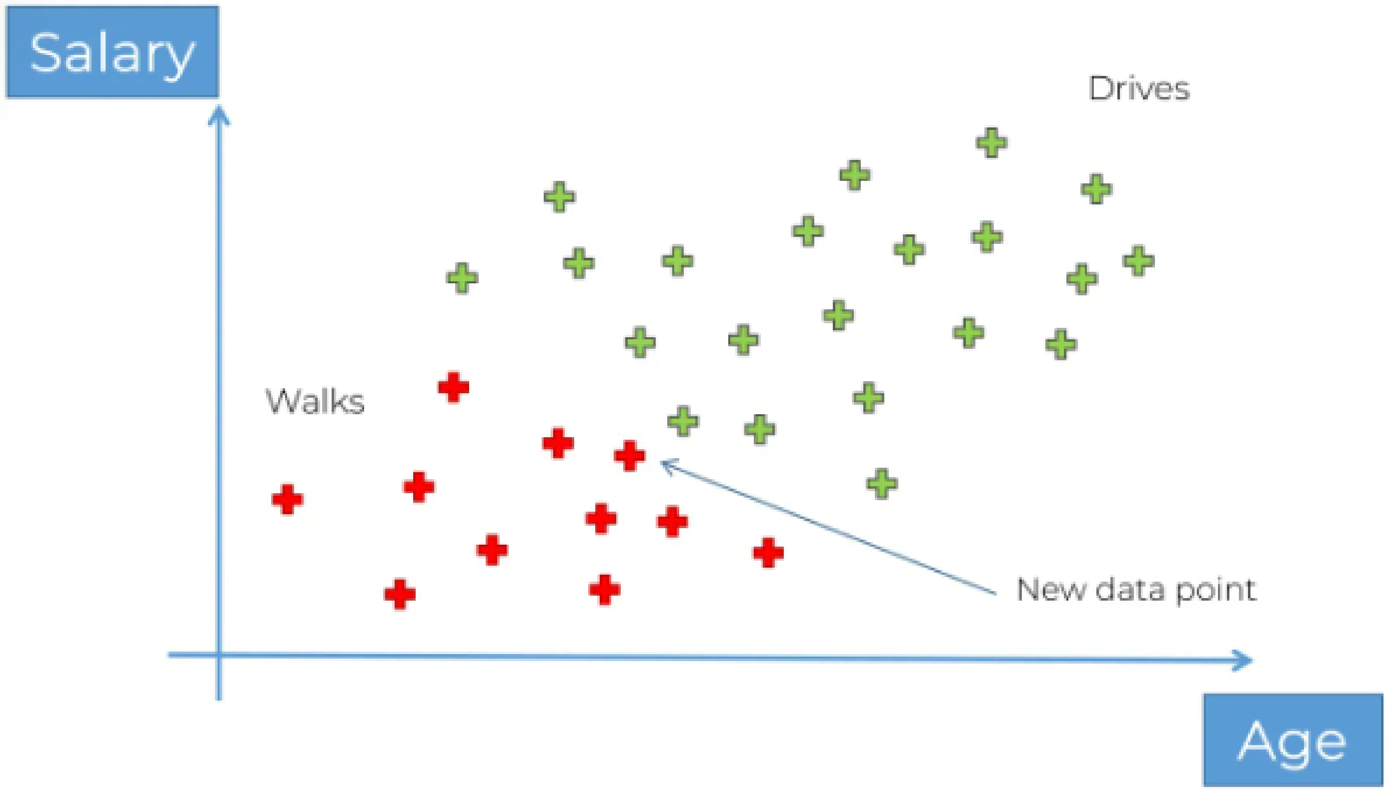
Step 3

$$P(Walks|X) > P(Drives|X)$$

Naïve Bayes



Naïve Bayes



Naïve Bayes Classifier Intuition (Challenge Reveal)

Step 2

#4 Posterior Probability

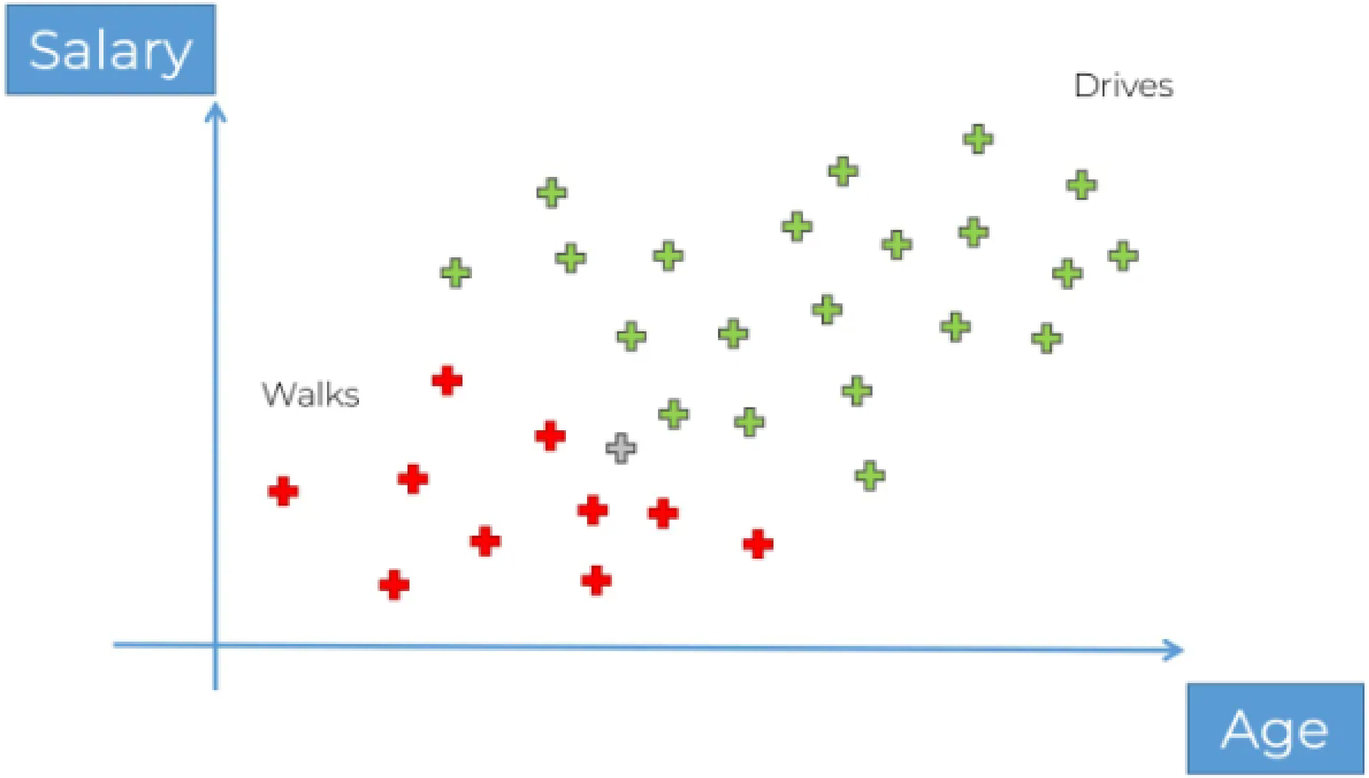
#3 Likelihood

#1 Prior Probability

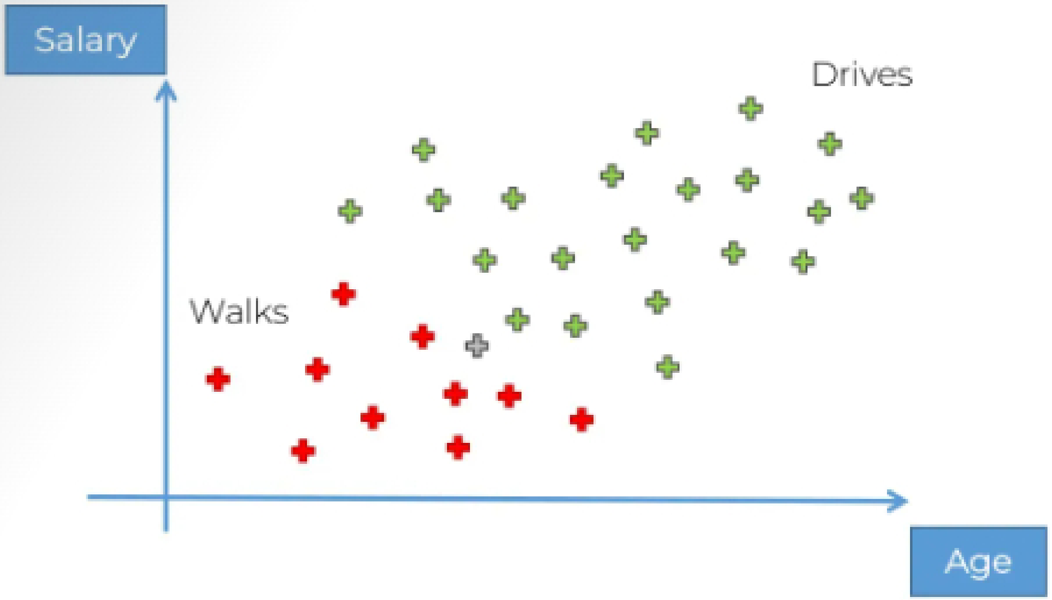
$$P(Drives|X) = \frac{P(X|Drives) * P(Drives)}{P(X)}$$

#2 Marginal Likelihood

Naïve Bayes: Step 2



Naïve Bayes: Step 2



#1. P(Drives)

$$P(Drives) = \frac{Number\ of\ Drivers}{Total\ Observations}$$

$$P(Drives) = \frac{20}{30}$$

Naïve Bayes: Step 2

#4 Posterior Probability

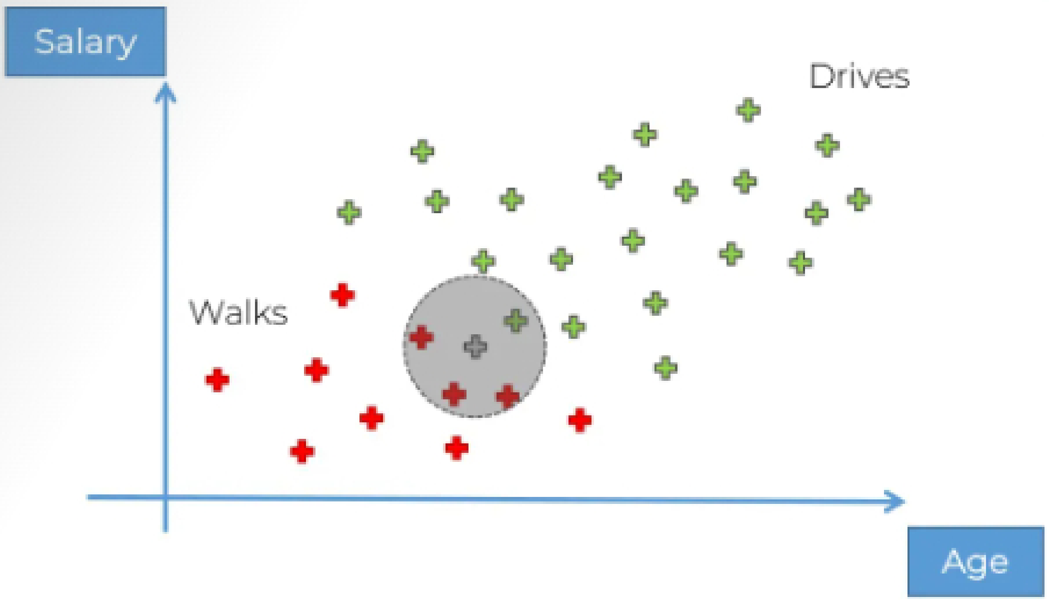
#3 Likelihood

#1 Prior Probability

$$P(Drives|X) = \frac{P(X|Drives) * P(Drives)}{P(X)}$$

#2 Marginal Likelihood

Naïve Bayes: Step 2



#2. P(X)

$$P(X) = \frac{Number\ of\ Similar\ Observations}{Total\ Observations}$$

$$P(X) = \frac{4}{30}$$

Naïve Bayes: Step 2

#4 Posterior Probability

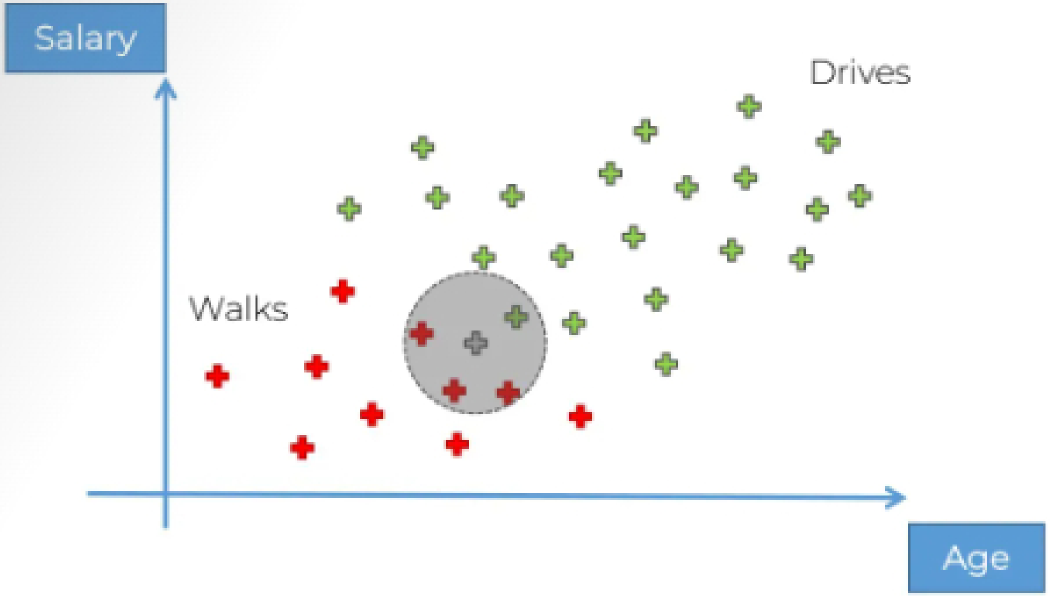
#3 Likelihood

#1 Prior Probability

$$P(Drives|X) = \frac{P(X|Drives) * P(Drives)}{P(X)}$$

#2 Marginal Likelihood

Naïve Bayes: Step 2



#3. P(X|Drives)

$$P(X|Drives) = \frac{\begin{array}{c} Number\ of\ Similar \\ Observations \\ Among\ those\ who\ Walk \end{array}}{Total\ number\ of\ Walkers}$$

$$P(X|Drives) = \frac{1}{20}$$

Naïve Bayes: Step 2

#4 Posterior Probability

#3 Likelihood

#1 Prior Probability

$$P(Drives|X) = \frac{P(X|Drives) * P(Drives)}{P(X)}$$

#2 Marginal Likelihood

Naïve Bayes: Step 2

#4 Posterior Probability

#3 Likelihood

#1 Prior Probability

$$P(Drives|X) = \frac{\frac{1}{20} * \frac{20}{30}}{\frac{4}{30}} = 0.25$$

#2 Marginal Likelihood

Step 2 – Done.

Naïve Bayes Classifier Additional Comments

Naïve Bayes

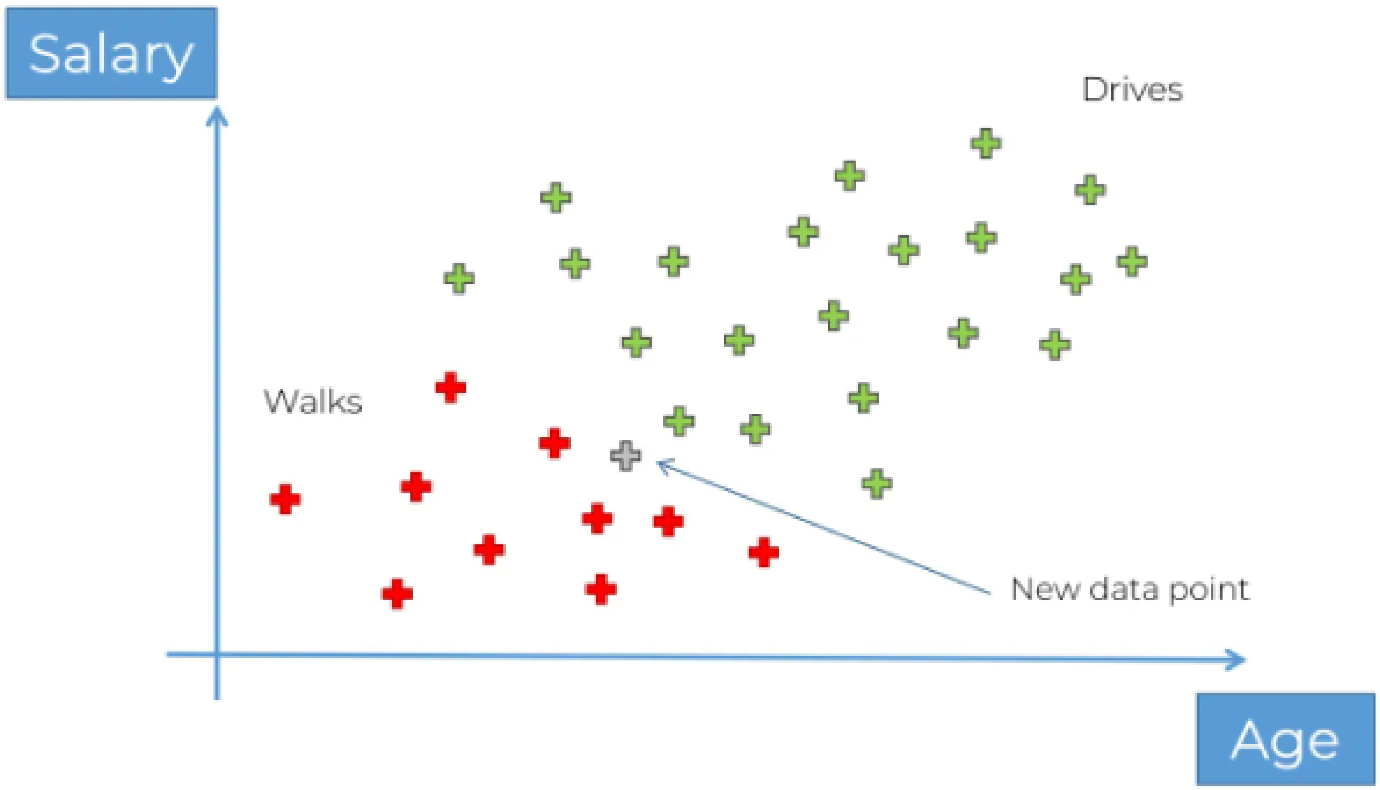
1. Q: Why "Naïve"?
2. P(X)
3. More than 2 features

Naïve Bayes

Q: Why "Naïve"?

A: Independence assumption

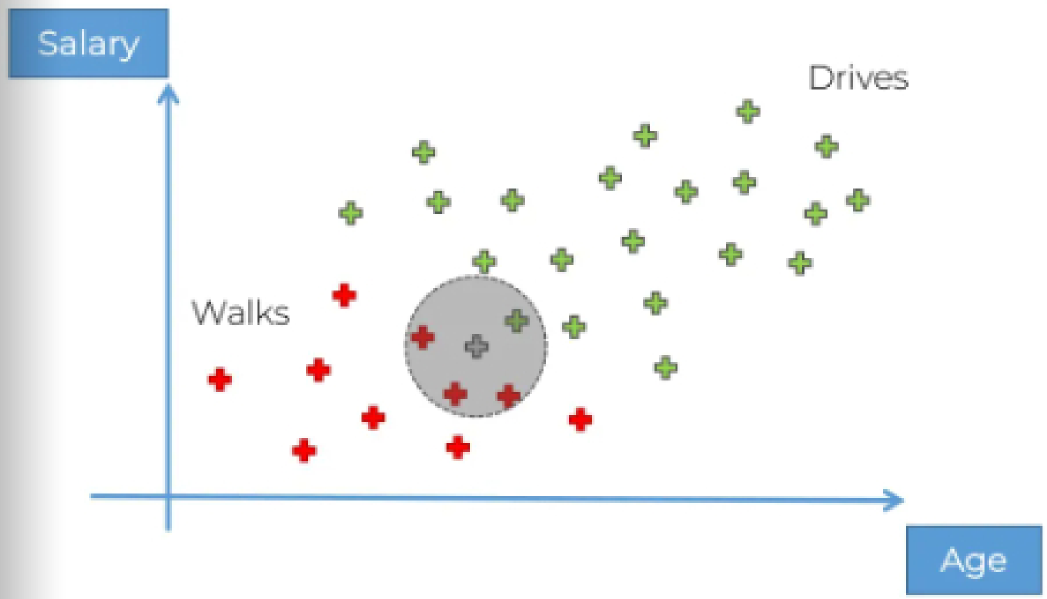
Naïve Bayes



Naïve Bayes

$$P(X)$$

Naïve Bayes: Step 2



#2. P(X)

$$P(X) = \frac{Number\ of\ Similar\ Observations}{Total\ Observations}$$

$$P(X) = \frac{4}{30}$$

NOTE: Same both times

Step 1

#4 Posterior Probability

#3 Likelihood

#1 Prior Probability

$$P(Walks|X) = \frac{P(X|Walks) * P(Walks)}{P(X)}$$

#2 Marginal Likelihood

Step 2

#4 Posterior Probability

#3 Likelihood

#1 Prior Probability

$$P(Drives|X) = \frac{P(X|Drives) * P(Drives)}{P(X)}$$

#2 Marginal Likelihood

Step 3

$$P(Walks|X) \quad v.s. \quad P(Drives|X)$$

Step 3

$$\frac{P(X|Walks) * P(Walks)}{\cancel{P(X)}} \quad v.s. \quad \frac{P(X|Drives) * P(Drives)}{\cancel{P(X)}}$$

Naïve Bayes

More than 2 classes

Step 3

$$P(Walks|X)\ v.s.\ P(Drives|X)$$

Step 3

$$0.75 \; v.s. \; 0.25$$

Step 3

$$0.75 > 0.25$$

Step 3

$$P(Walks|X) > P(Drives|X)$$

Decision Tree Intuition

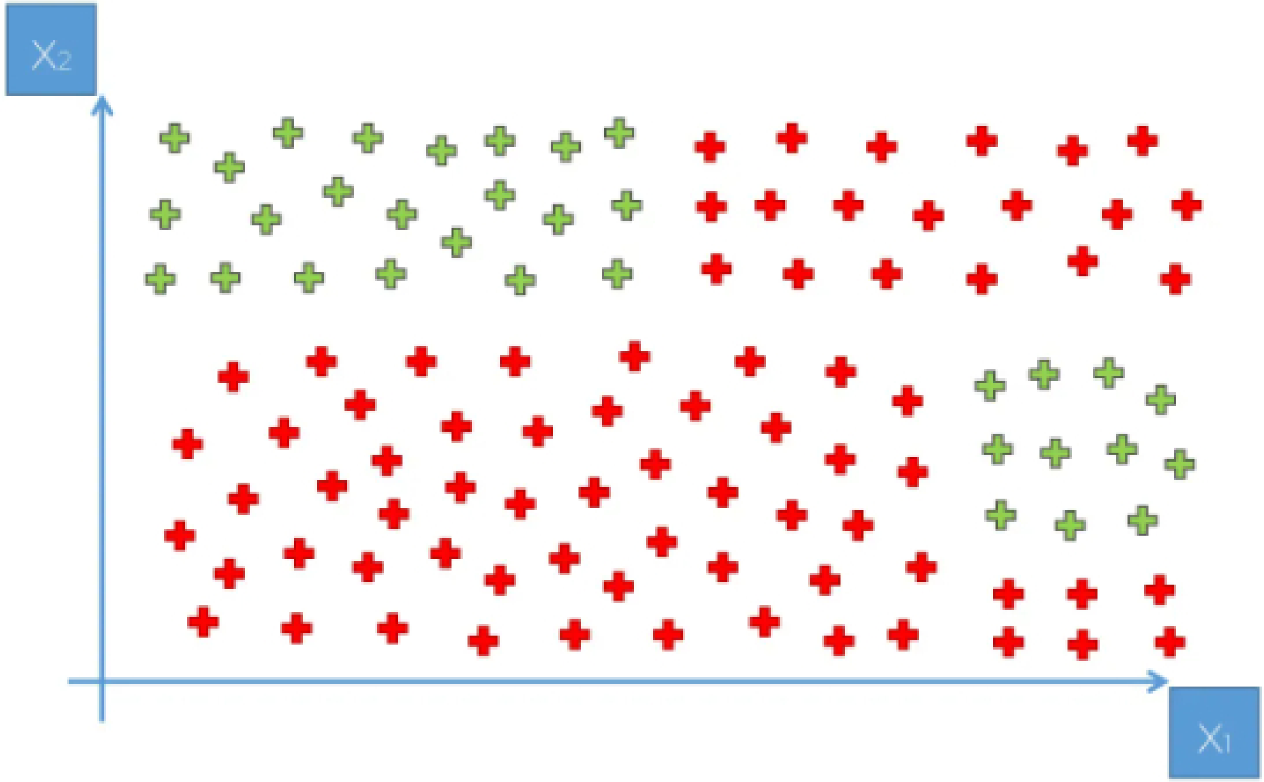
Decision Tree Intuition

CART

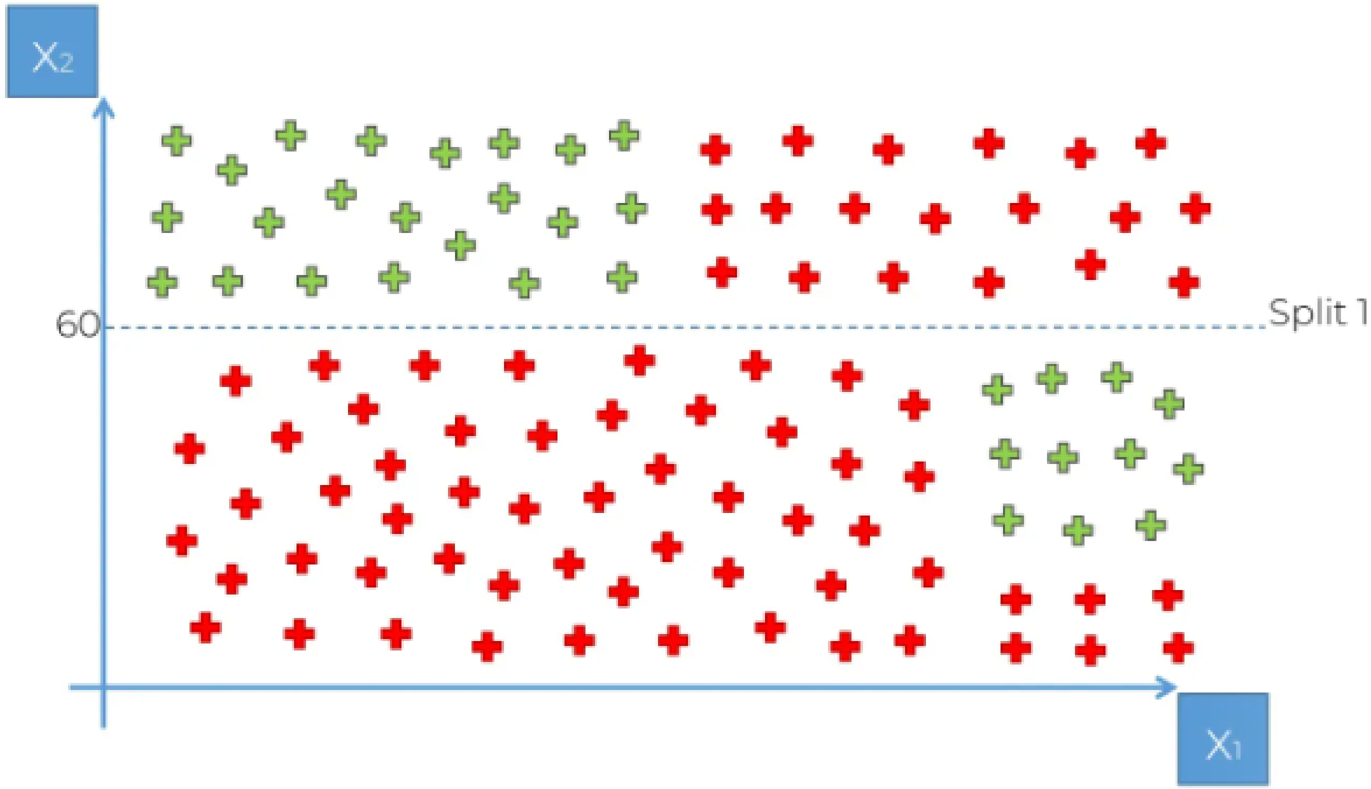
Classification Trees

Regression Trees

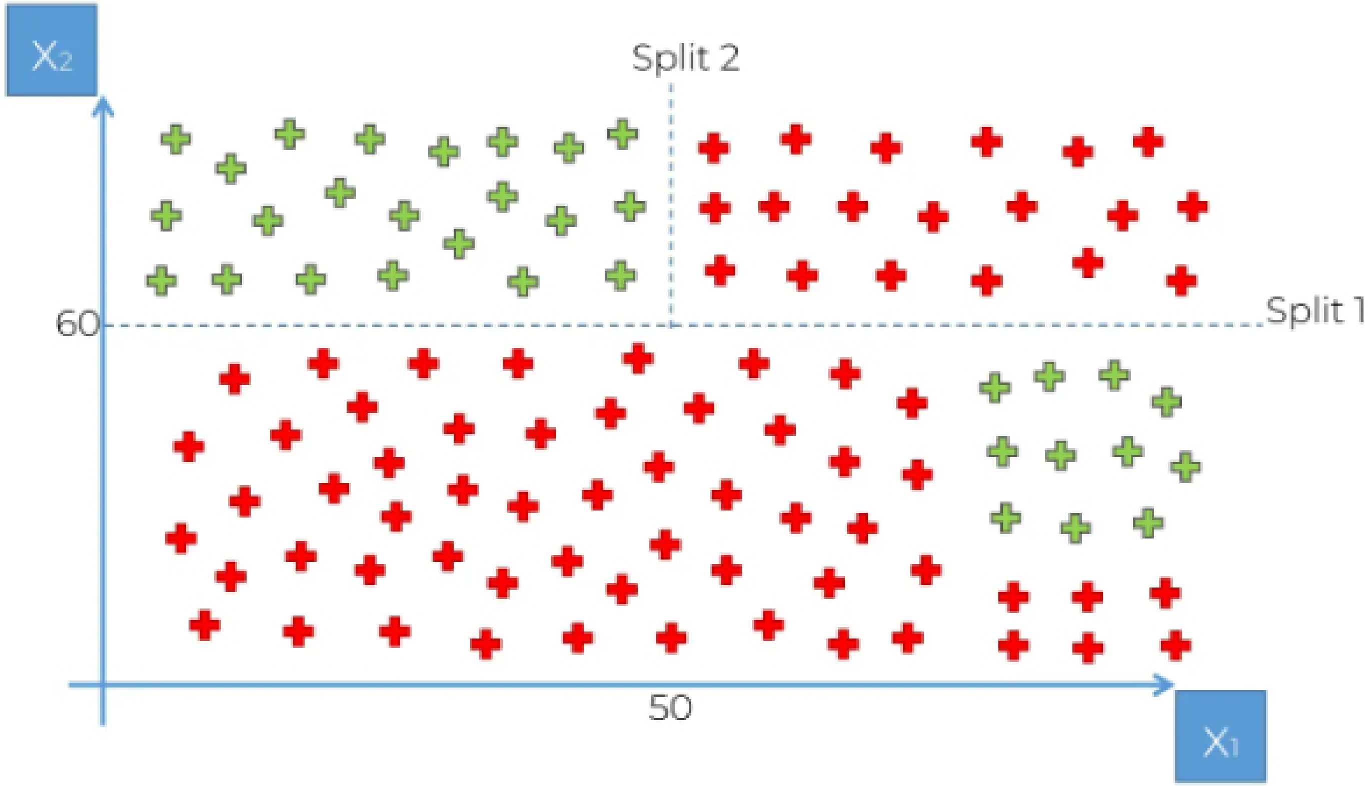
Decision Tree Intuition



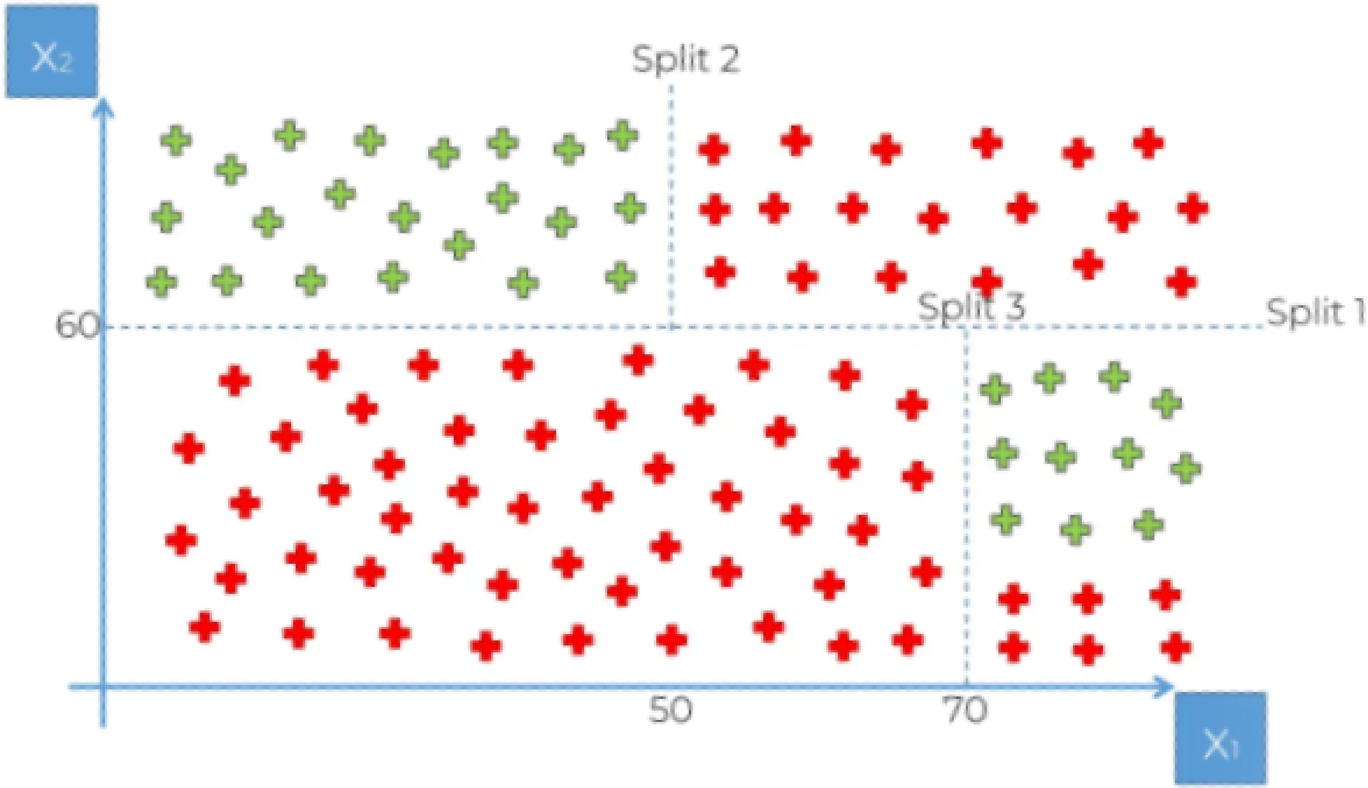
Decision Tree Intuition



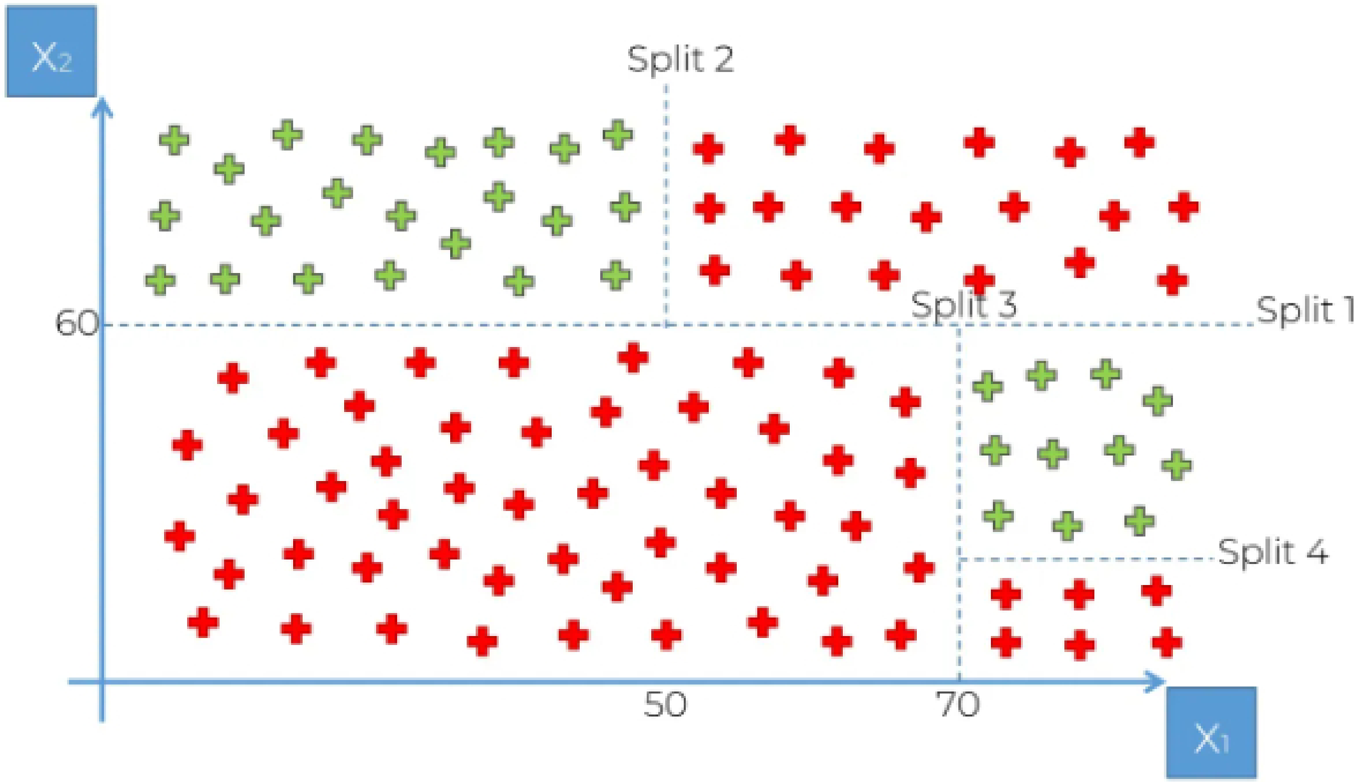
Decision Tree Intuition



Decision Tree Intuition



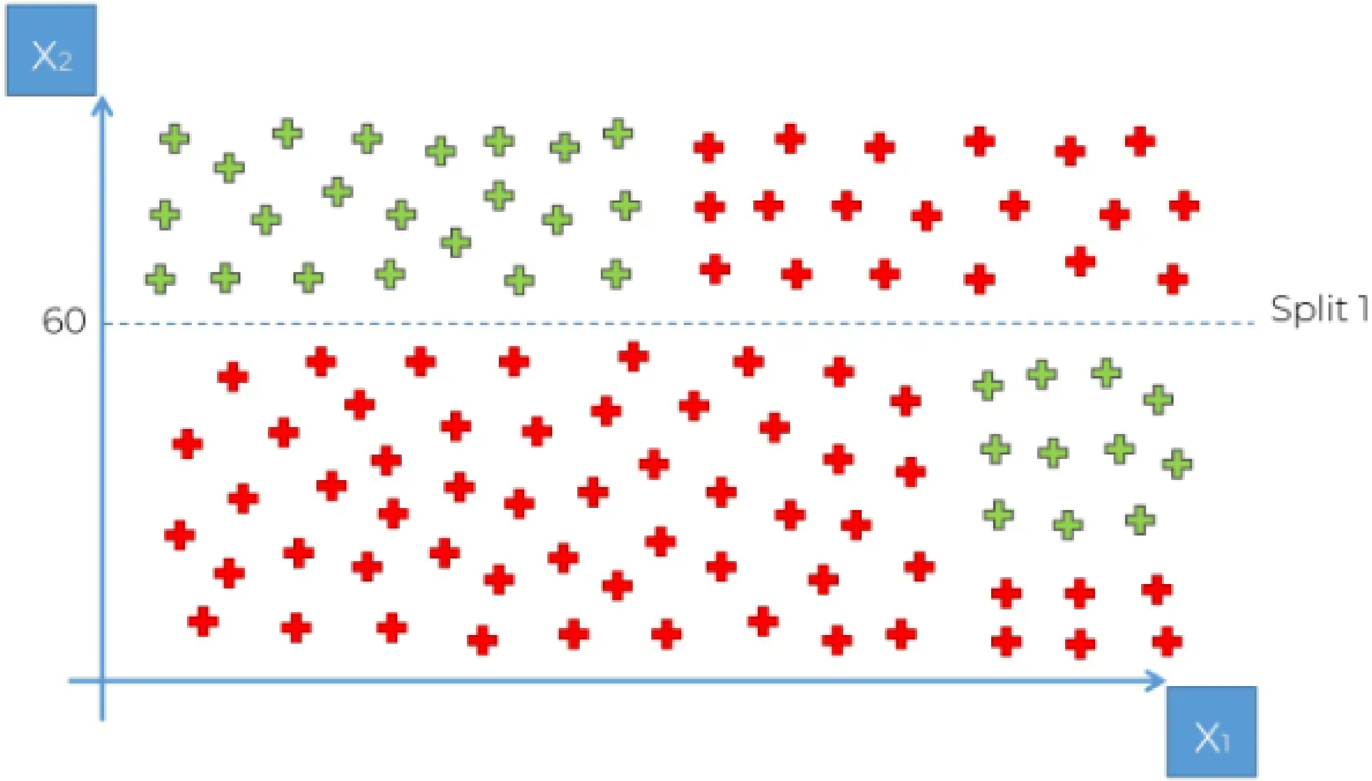
Decision Tree Intuition



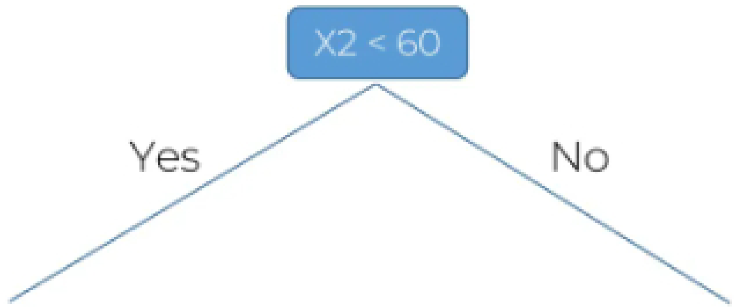
Decision Tree Intuition

Rewind...

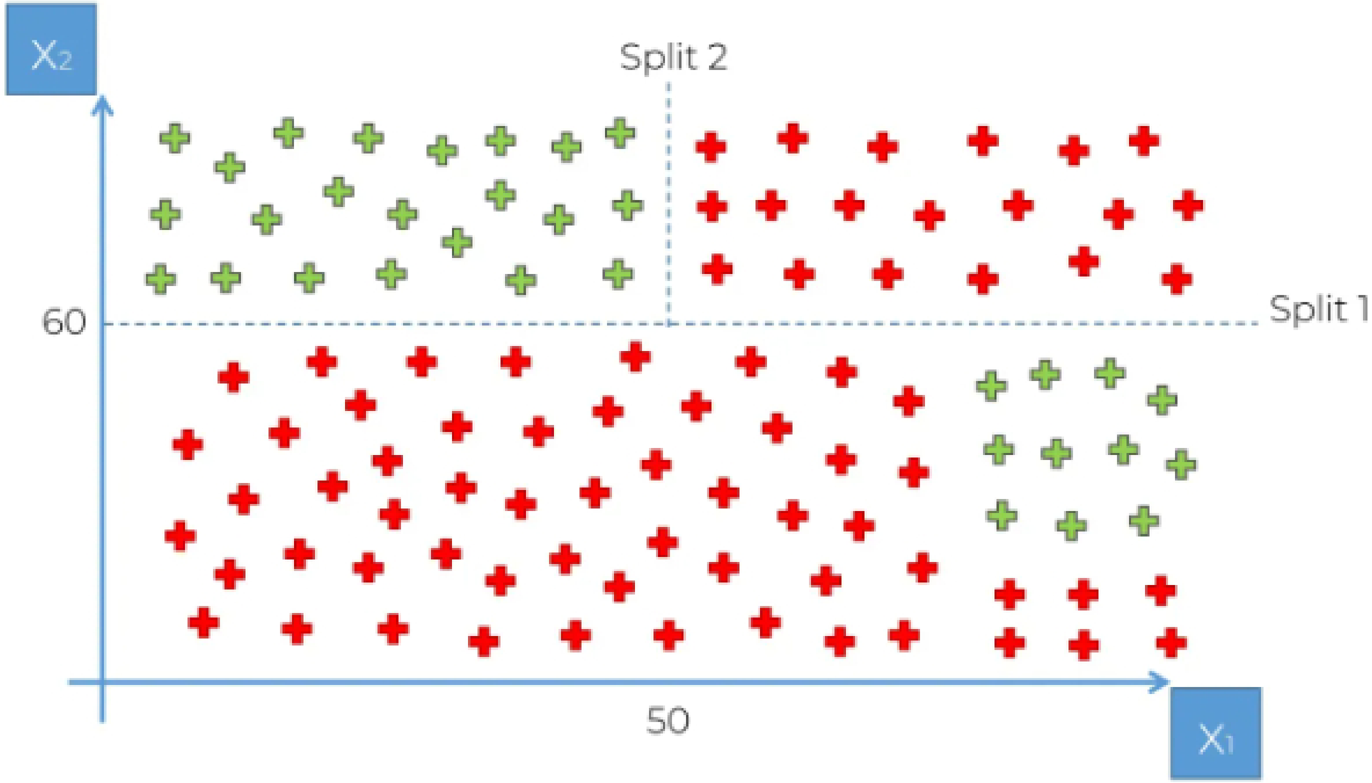
Split 1



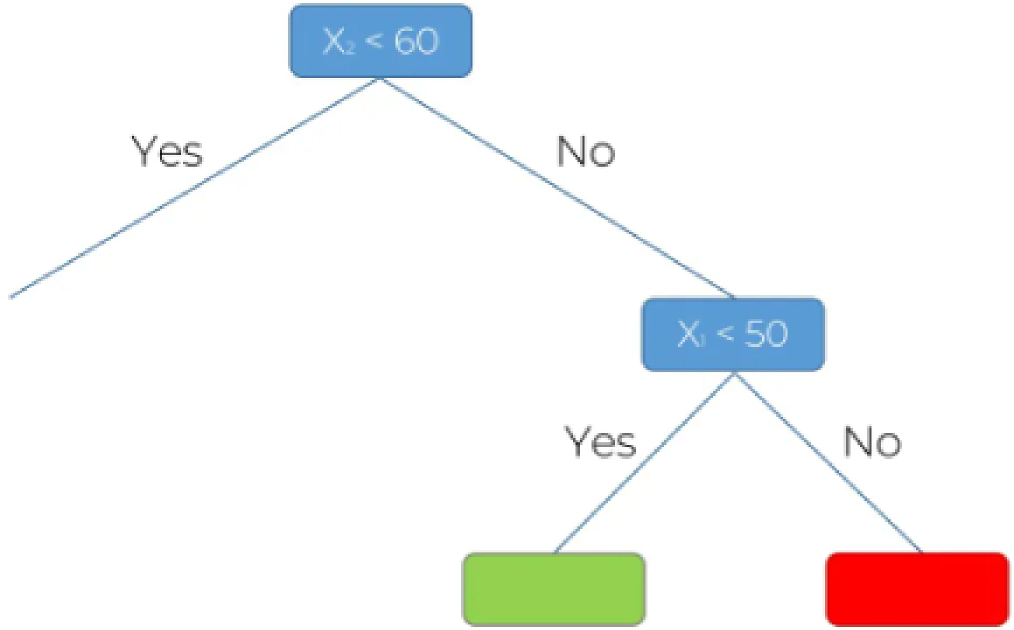
Split 1



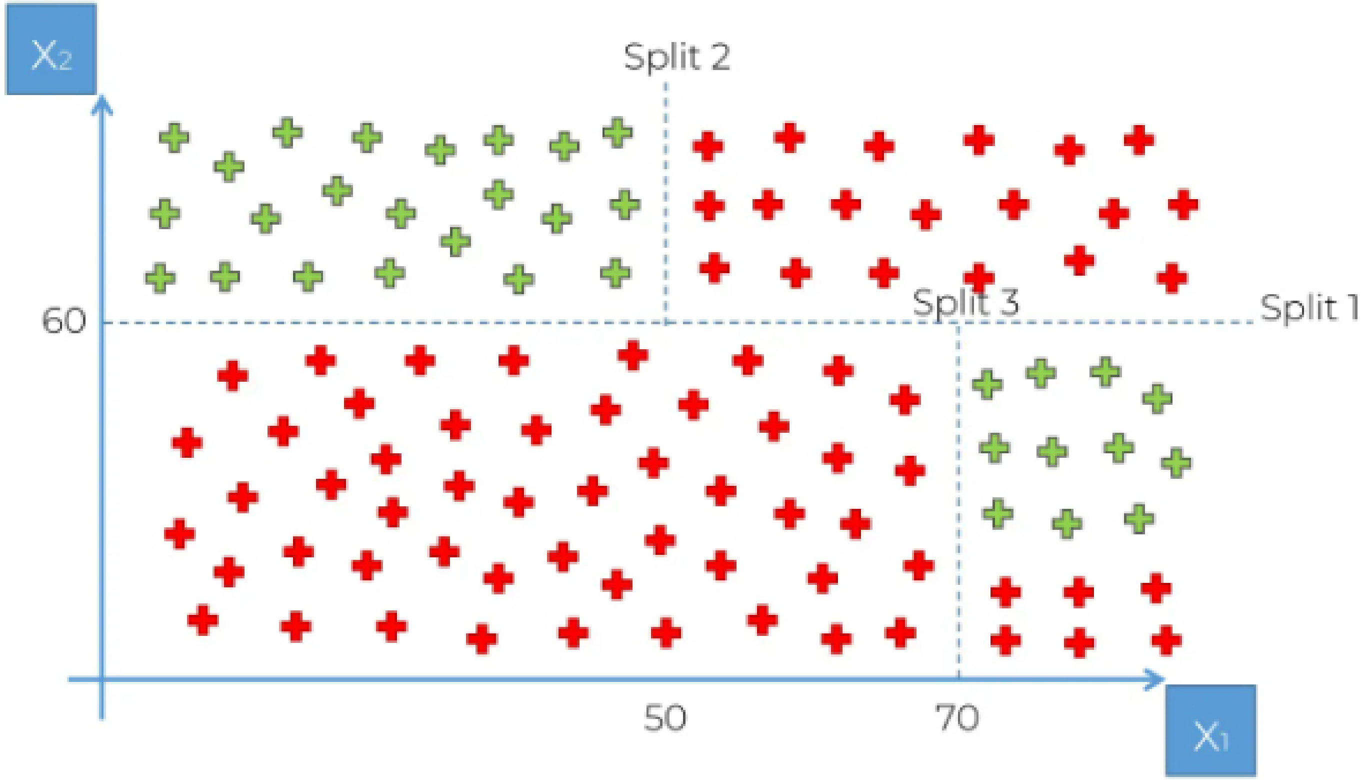
Split 2



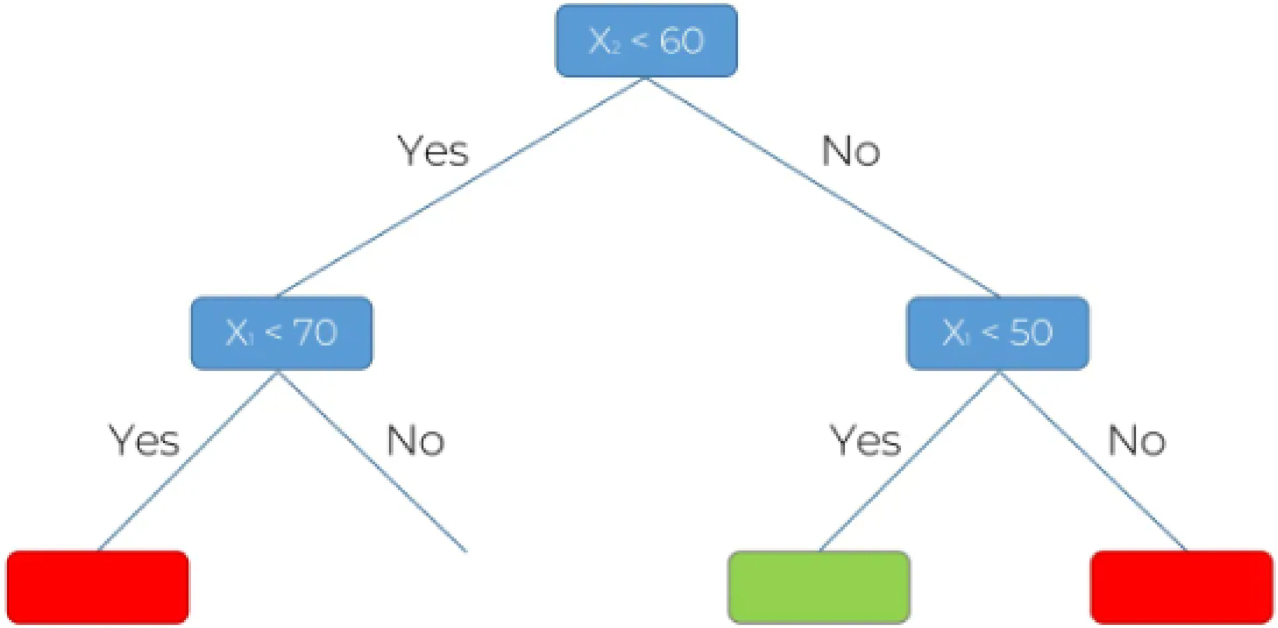
Split 2



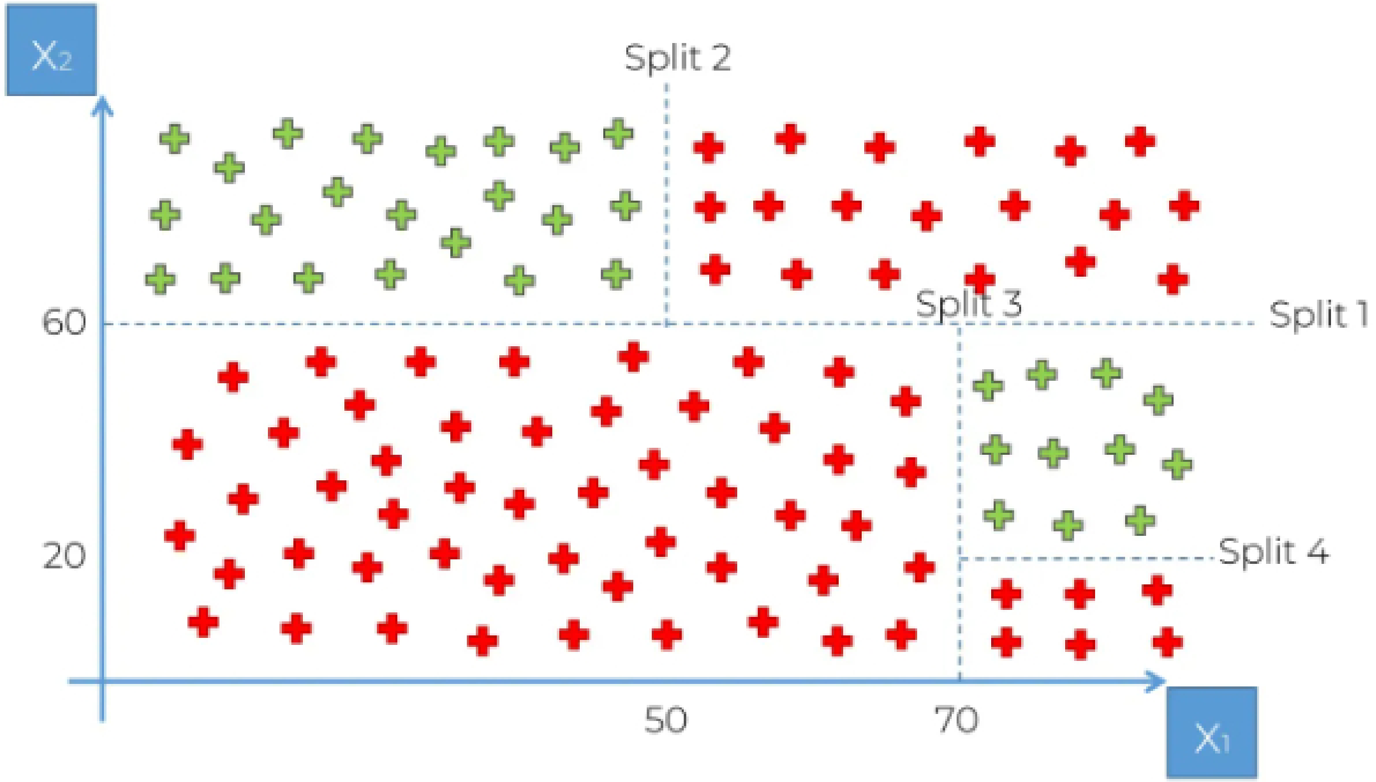
Split 3



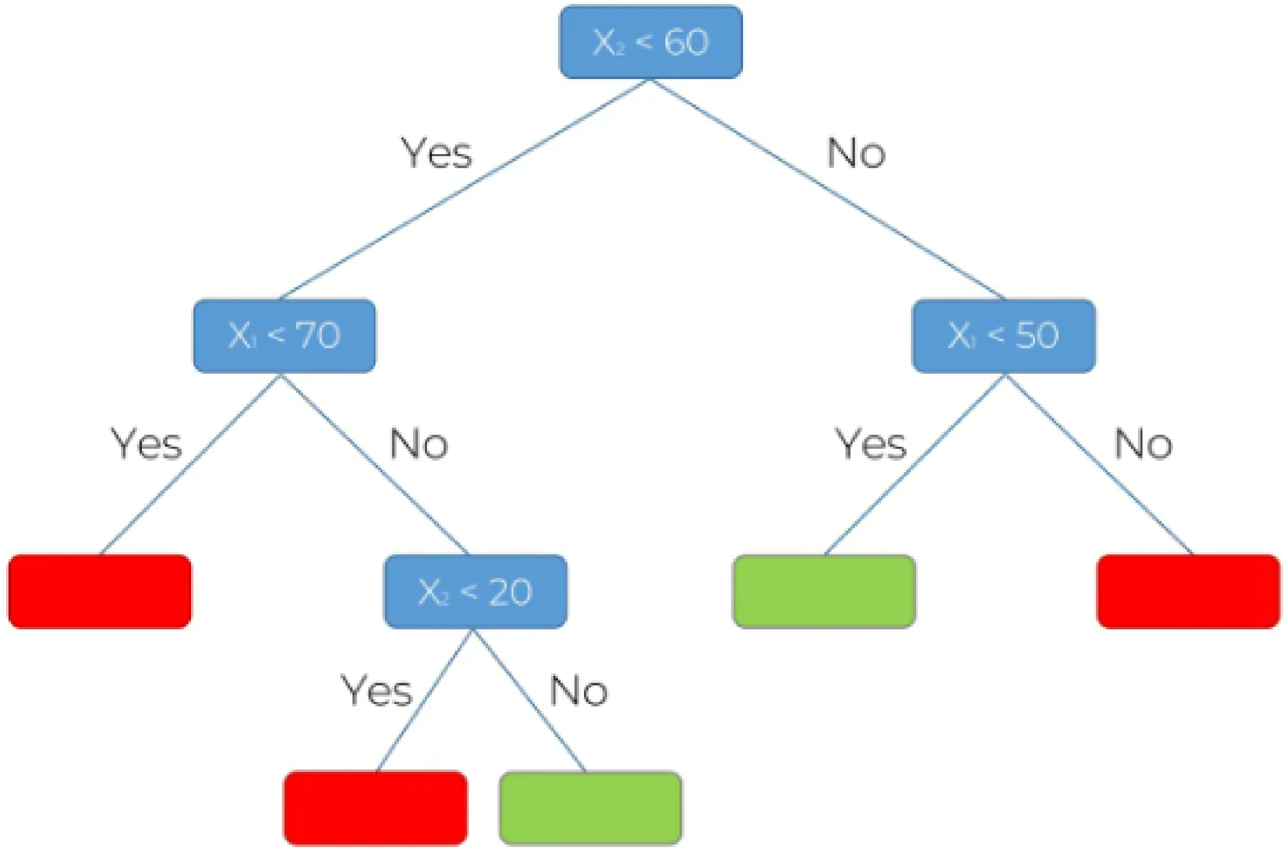
Split 3



Split 4



Split 4



Decision Trees

- Old Method
- Reborn with upgrades
- Random Forest
- Gradient Boosting
- etc.

Random Forest Intuition

Random Forest Intuition

Ensemble Learning

Random Forest Intuition

STEP 1: Pick at random K data points from the Training set.

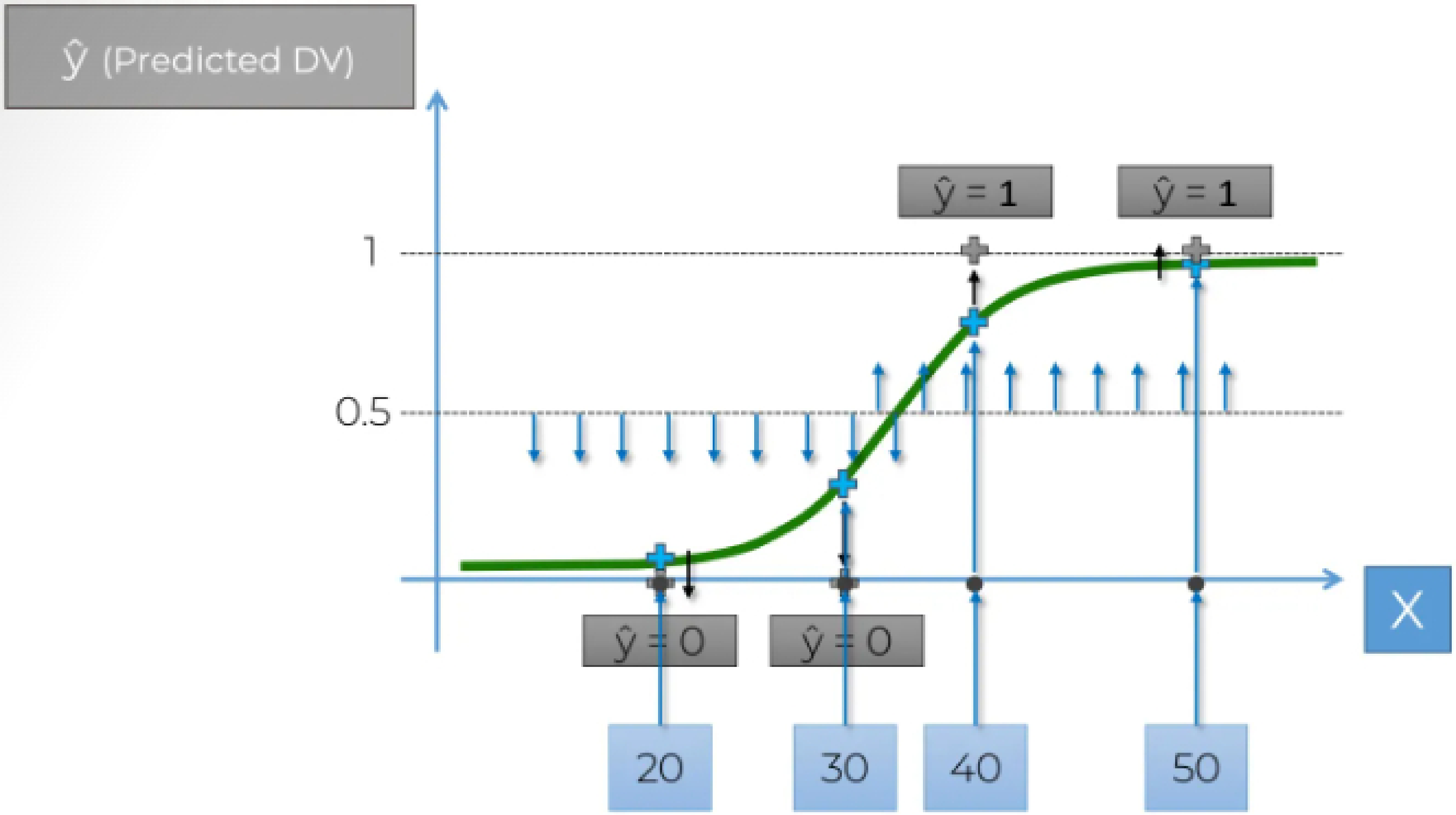
STEP 2: Build the Decision Tree associated to these K data points.

STEP 3: Choose the number Ntree of trees you want to build and repeat STEPS 1 & 2

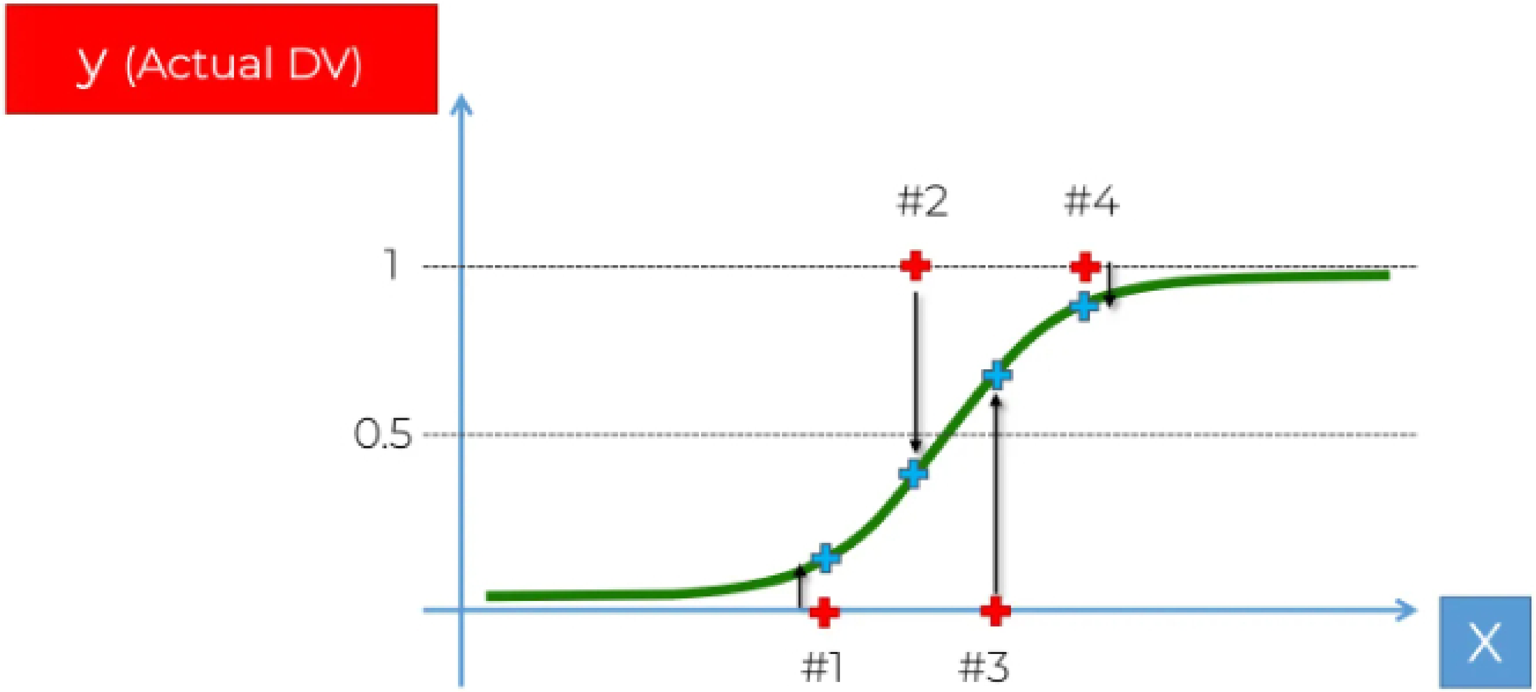
STEP 4: For a new data point, make each one of your Ntree trees predict the category to which the data points belongs, and assign the new data point to the category that wins the majority vote.

False Positives
False Negatives

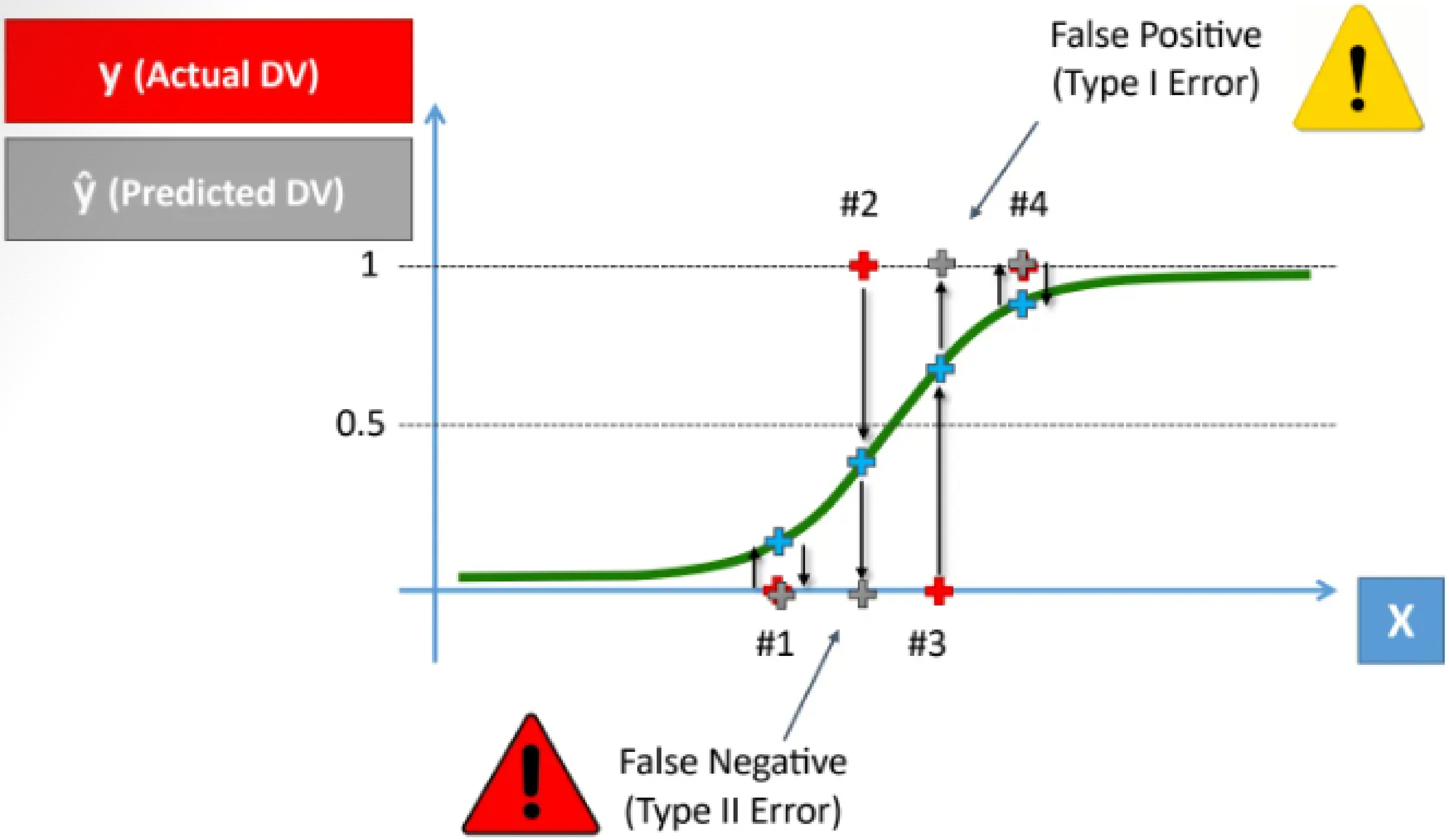
False Positives & Negatives



False Positives & Negatives



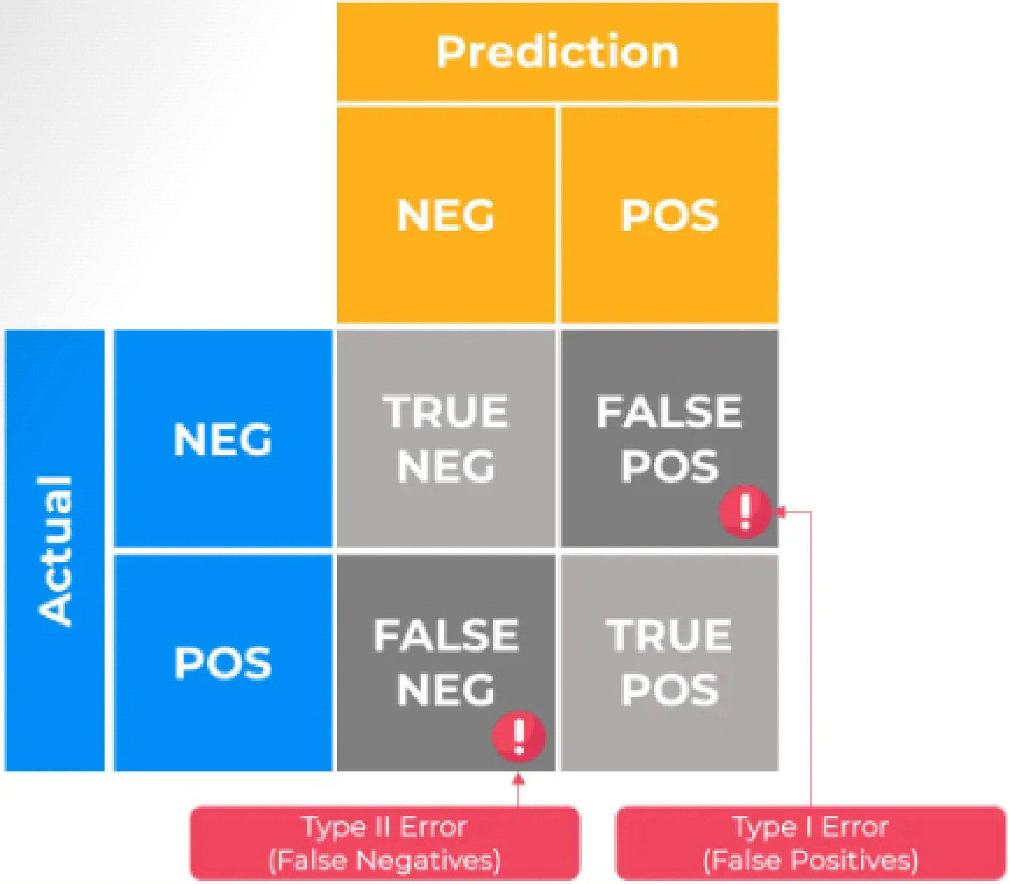
False Positives & Negatives



Fin.

Confusion Matrix & Accuracy

Confusion Matrix & Accuracy



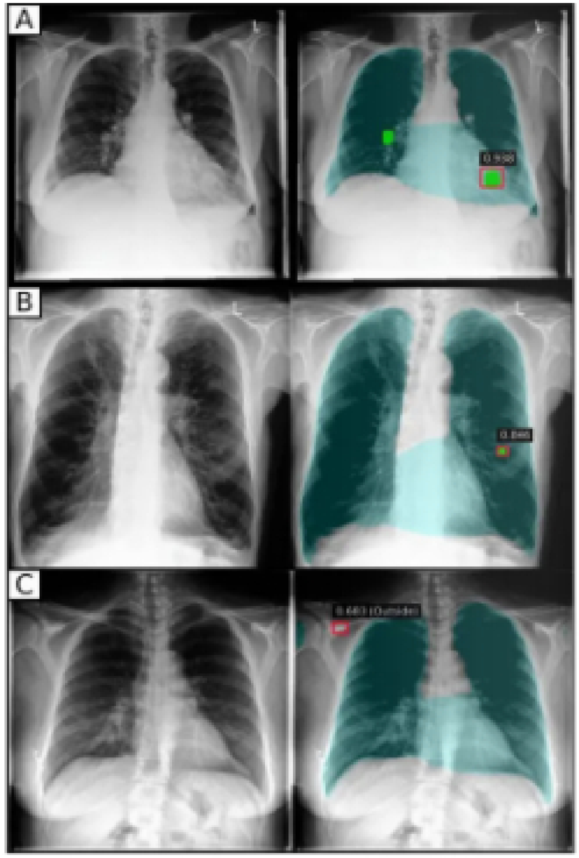


Image source: nature.com

Confusion Matrix & Accuracy

		Prediction	
		NEG	POS
Actual	NEG	43	12
	POS	4	41

Type II Error
(False Negatives)

Type I Error
(False Positives)

Accuracy Rate and Error Rate:

$$AR = \frac{Correct}{Total} = \frac{TN + TP}{Total} = \frac{84}{100} = 84\%$$

$$ER = \frac{Incorrect}{Total} = \frac{FP + FN}{Total} = \frac{16}{100} = 16\%$$

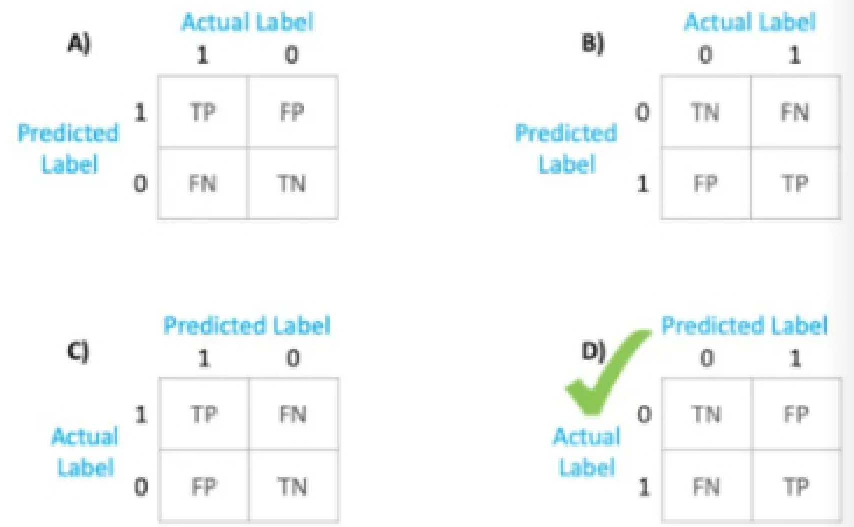
Additional Reading

Understanding the Confusion Matrix from Scikit learn

Samarth Agrawal (2021)

Link:

https://towardsdatascience.com/understanding-the-confusion-matrix-from-scikit-learn-c51d88929c79



Referring back to original image. Option D is the default output. Image by Author

Accuracy Paradox

Accuracy Paradox

		ŷ (Predicted DV)	
		0	1
y (Actual DV)	0	9,700	150
	1	50	100

<u>Scenario 1:</u>

Accuracy Rate = Correct / Total
AR = 9,800/10,000 = 98%

Accuracy Paradox

		ŷ (Predicted DV)	
		0	1
y (Actual DV)	0	9,850	0
	1	150	0

<u>Scenario 1:</u>

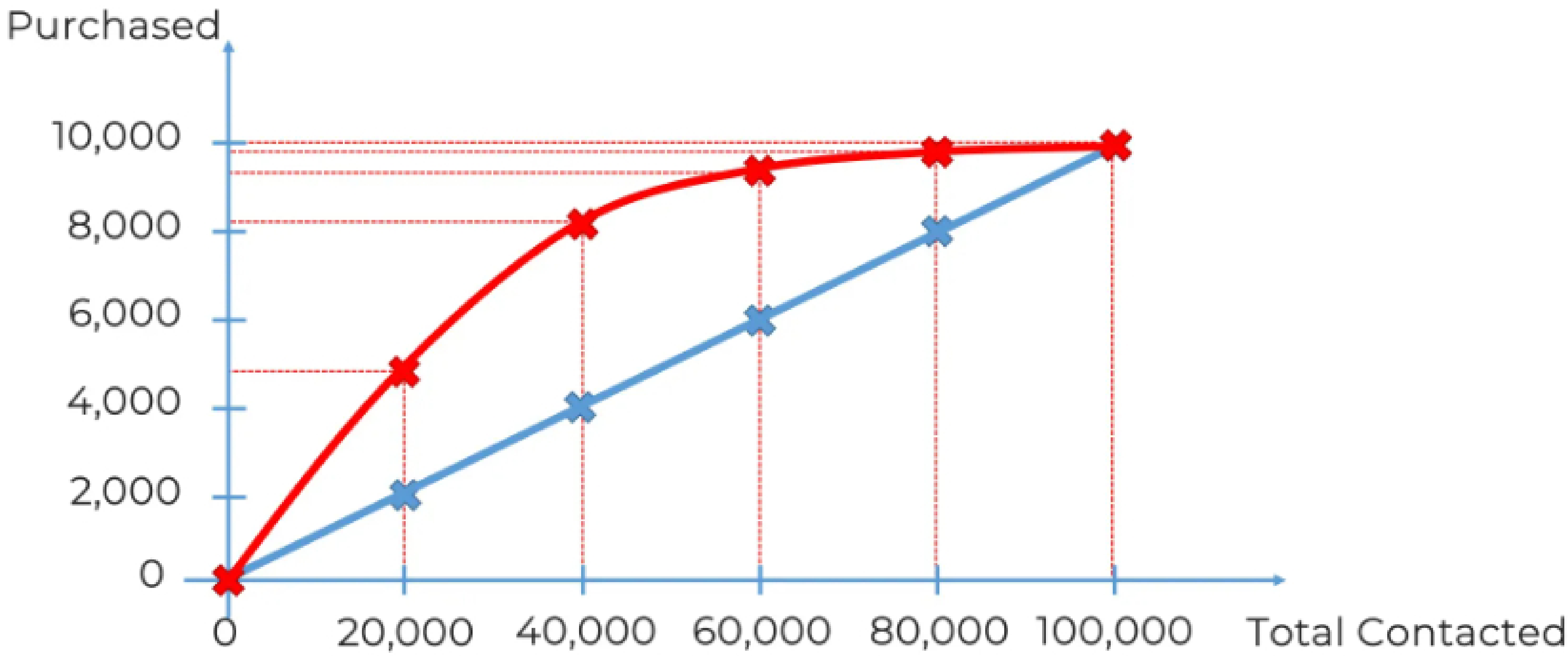
Accuracy Rate = Correct / Total
AR = 9,800/10,000 = 98%

<u>Scenario 2:</u>

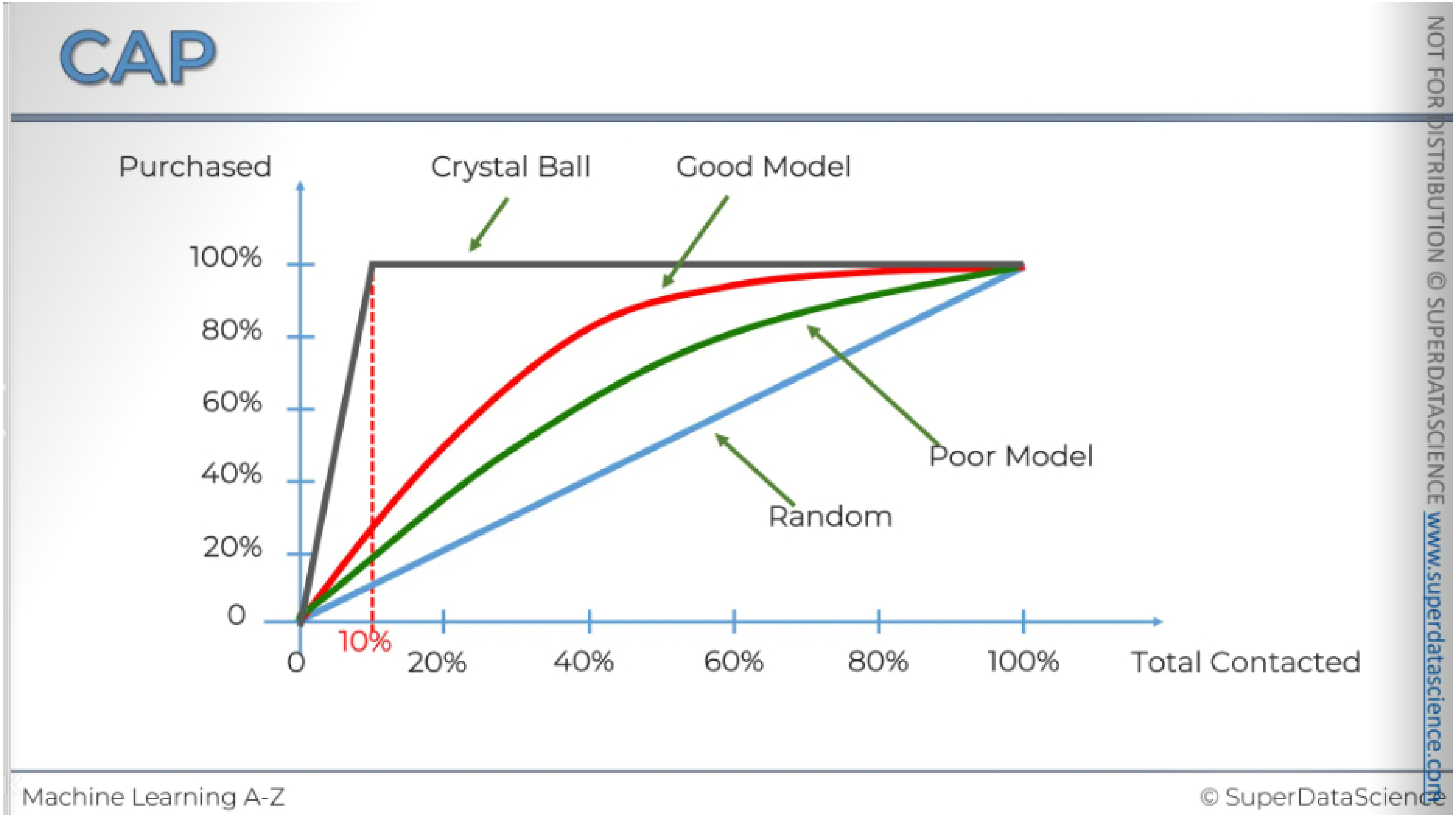
Accuracy Rate = Correct / Total
AR = 9,850/10,000 = 98.5%

Cumulative Accuracy Profile

CAP



Purchased
10,000
8,000
6,000
4,000
2,000
0
0
20,000
40,000
60,000
80,000
100,000
Total Contacted


CAP
Purchased
Crystal Ball
Good Model
100%
80%
60%
40%
20%
0
Poor Model
Random
10%
0
20%
40%
60%
80%
100%
Total Contacted
Machine Learning A-Z
© SuperDataScience
NOT FOR DISTRIBUTION © SUPERDATASCIENCE www.superdatascience.com

CAP

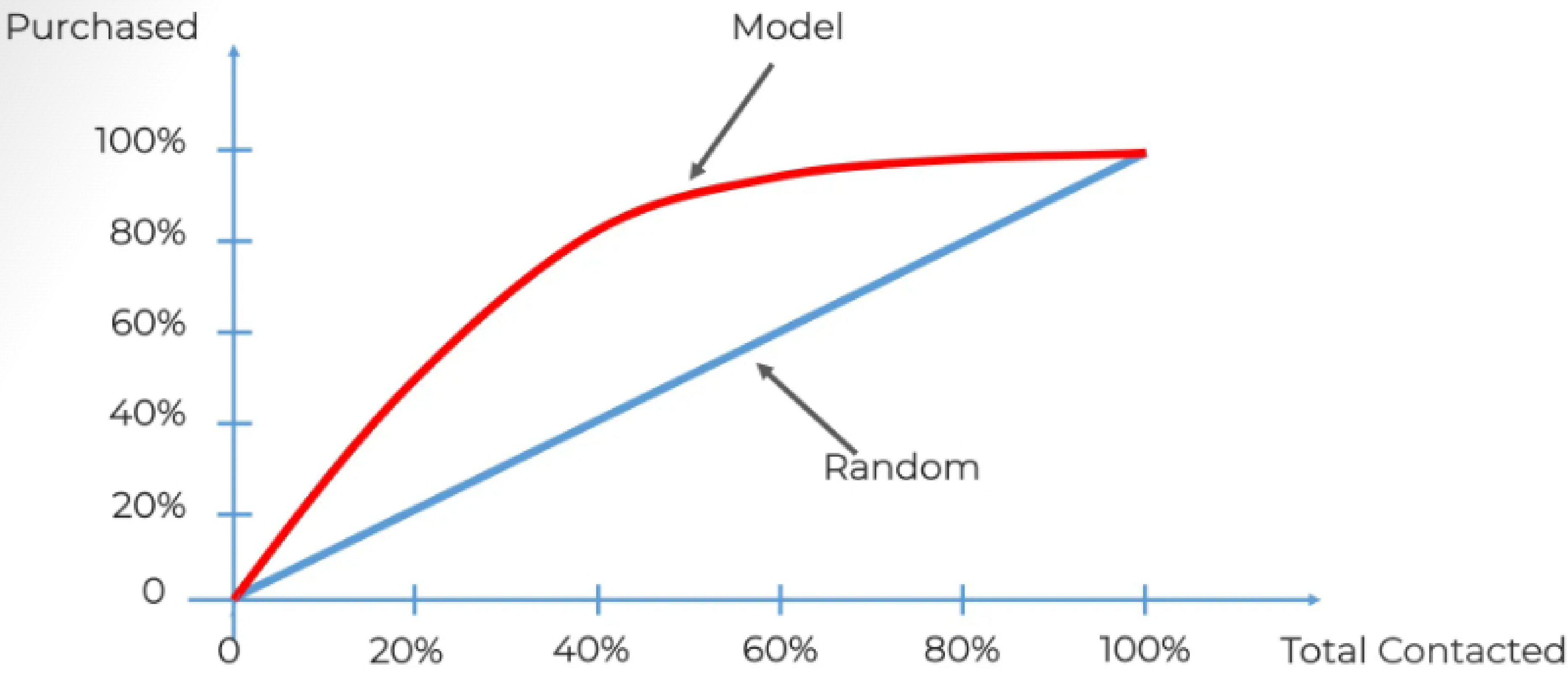
Note:

CAP = Cumulative Accuracy Profile

$\neq$

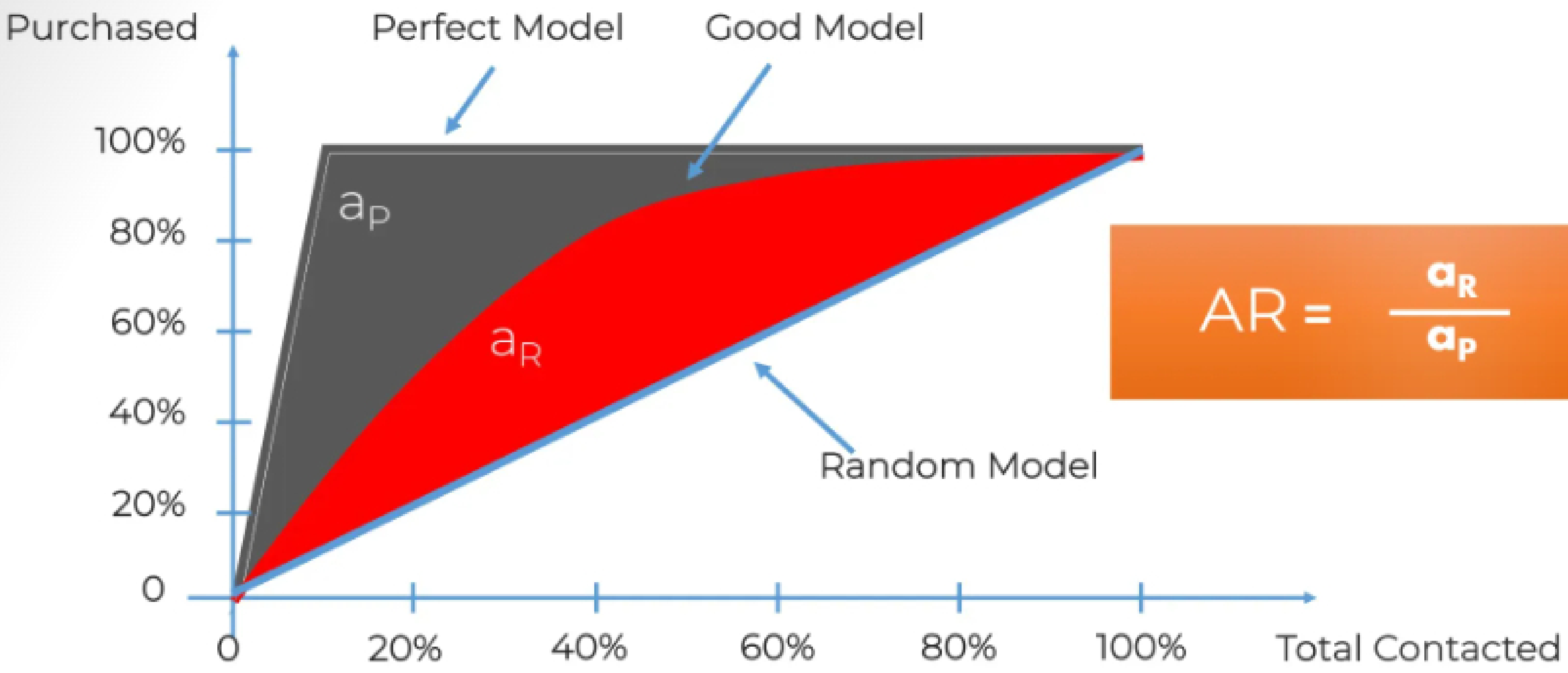
ROC = Receiver Operating Characteristic

CAP


Purchased
Model
100%
80%
60%
40%
20%
0
Random
0
20%
40%
60%
80%
100%
Total Contacted

CAP Analysis

CAP Analysis



$$AR = \frac{a_R}{a_P}$$

CAP Analysis