Classification

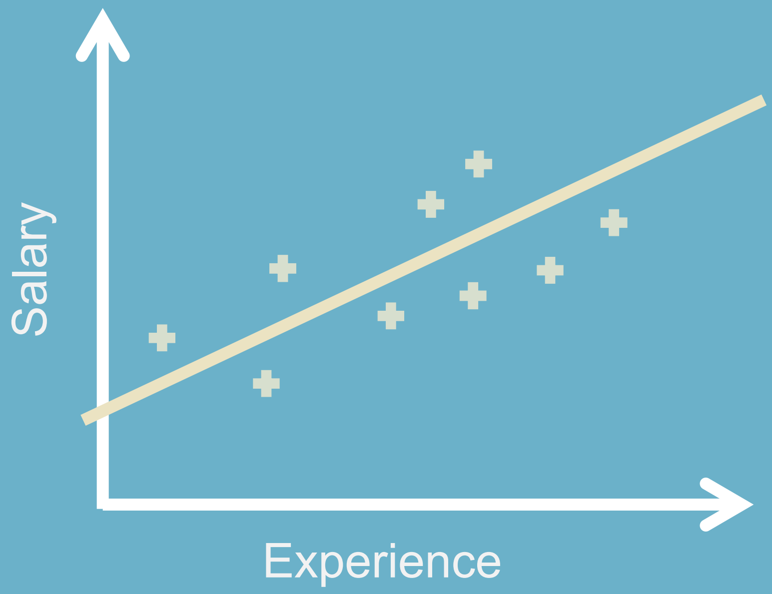
Introduction | Algorithms | Practice

Contents

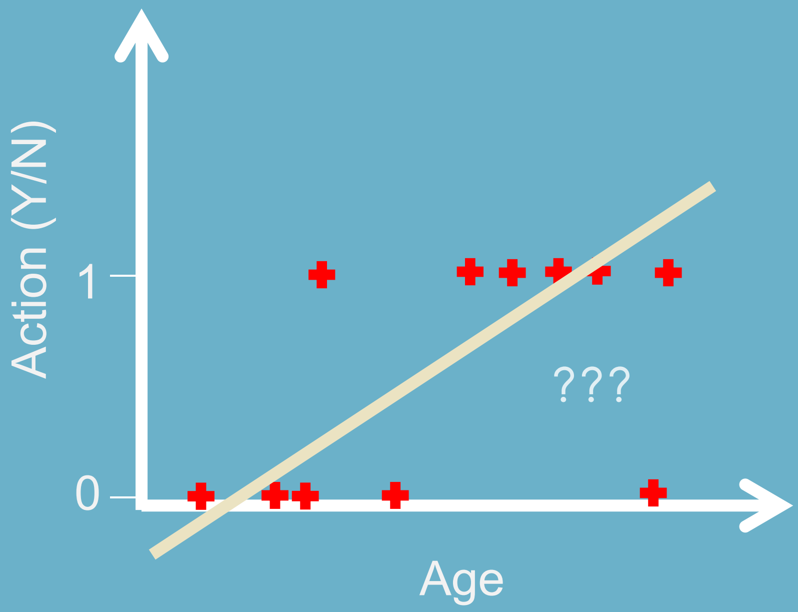
- Logistic Regression Intuition
- K Nearest Neighbor (KNN)

Logistic Regression Intuition

- We know this:



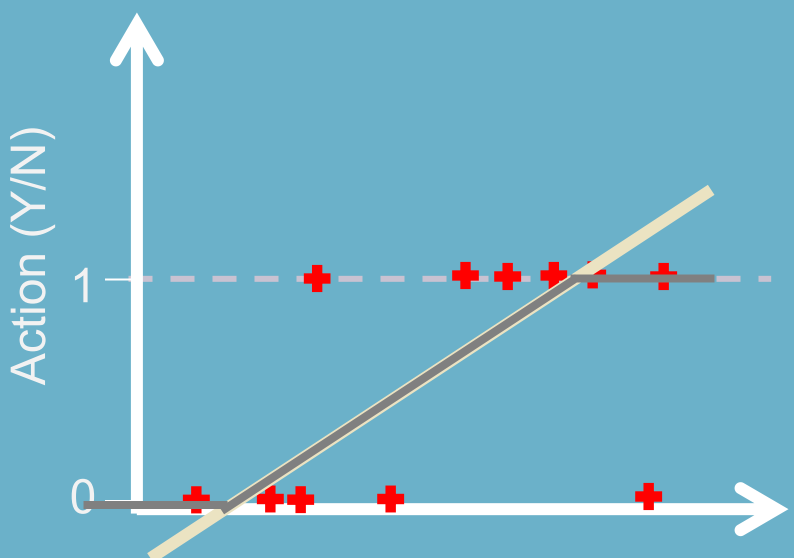
- This is New:



Logistic Regression Intuition

FORMULAE

- WHAT is the Best fit?



$$y = b_0 + b_1 * x_1$$

Sigmoid Function

$$p = \frac{1}{1 + e^{-y}}$$

$$\ln\left(\frac{p}{1 - p}\right) = b_0 + b_1 * x_1$$

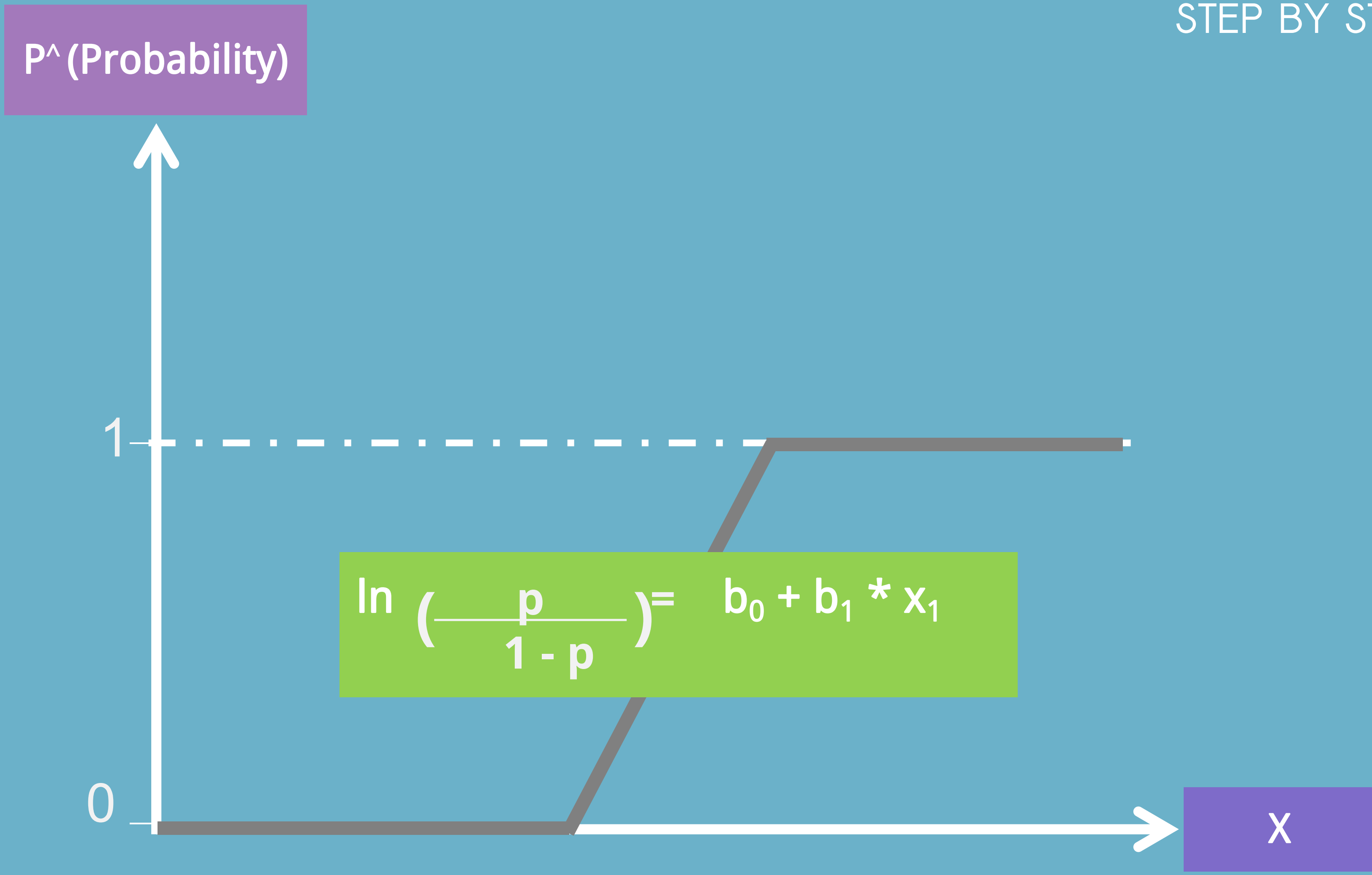
Logistic Regression Intuition

REALITY?

- WHAT JUST HAPPENED, Bro?

> Let's take it EASY

STEP BY STEP



Logistic Regression Intuition

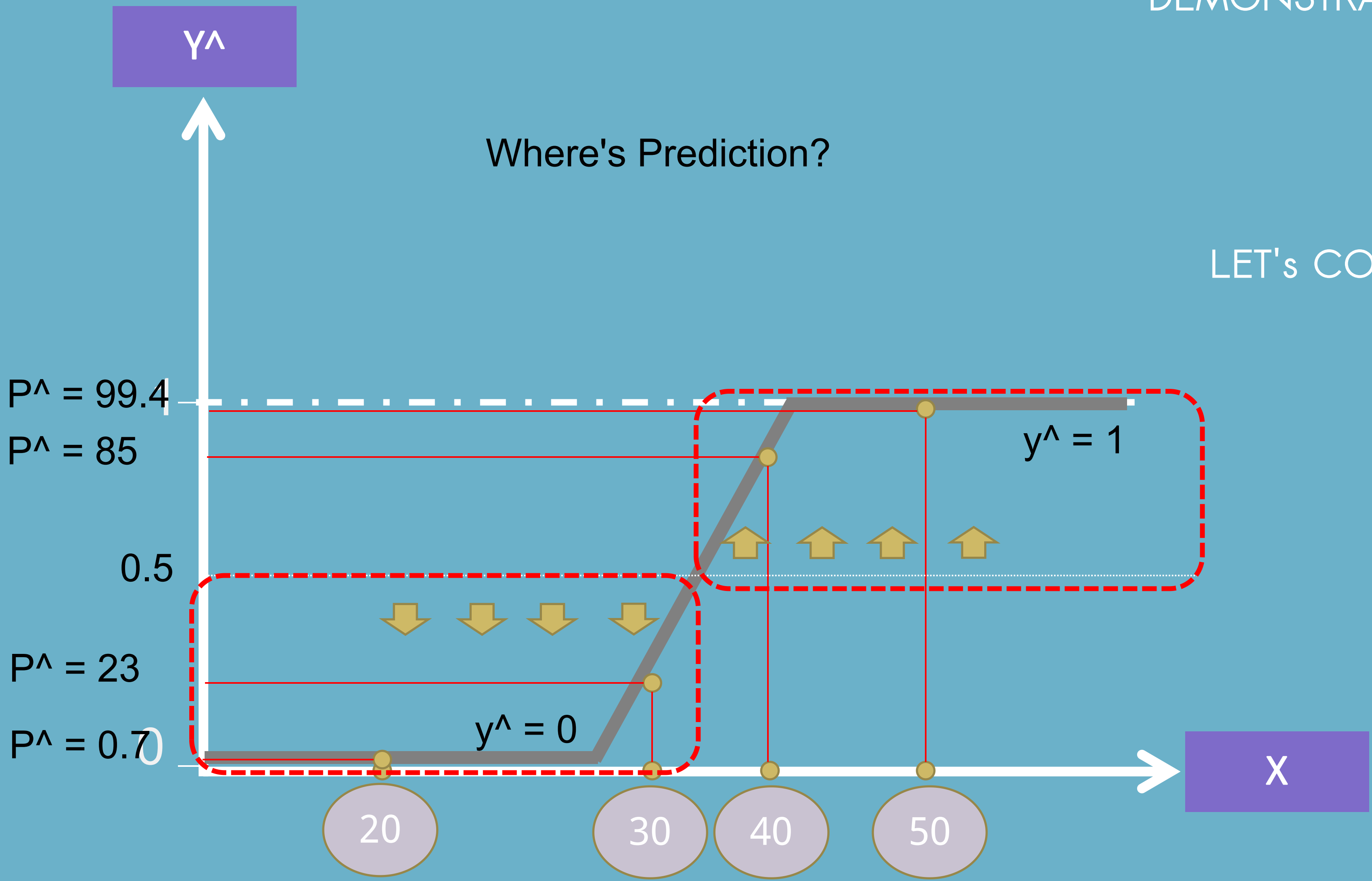
WHAT CAN WE DO

- Carry out probability
- It is not the 100% accurate or True
- We predict
- Probability here is P^ (P Hat)

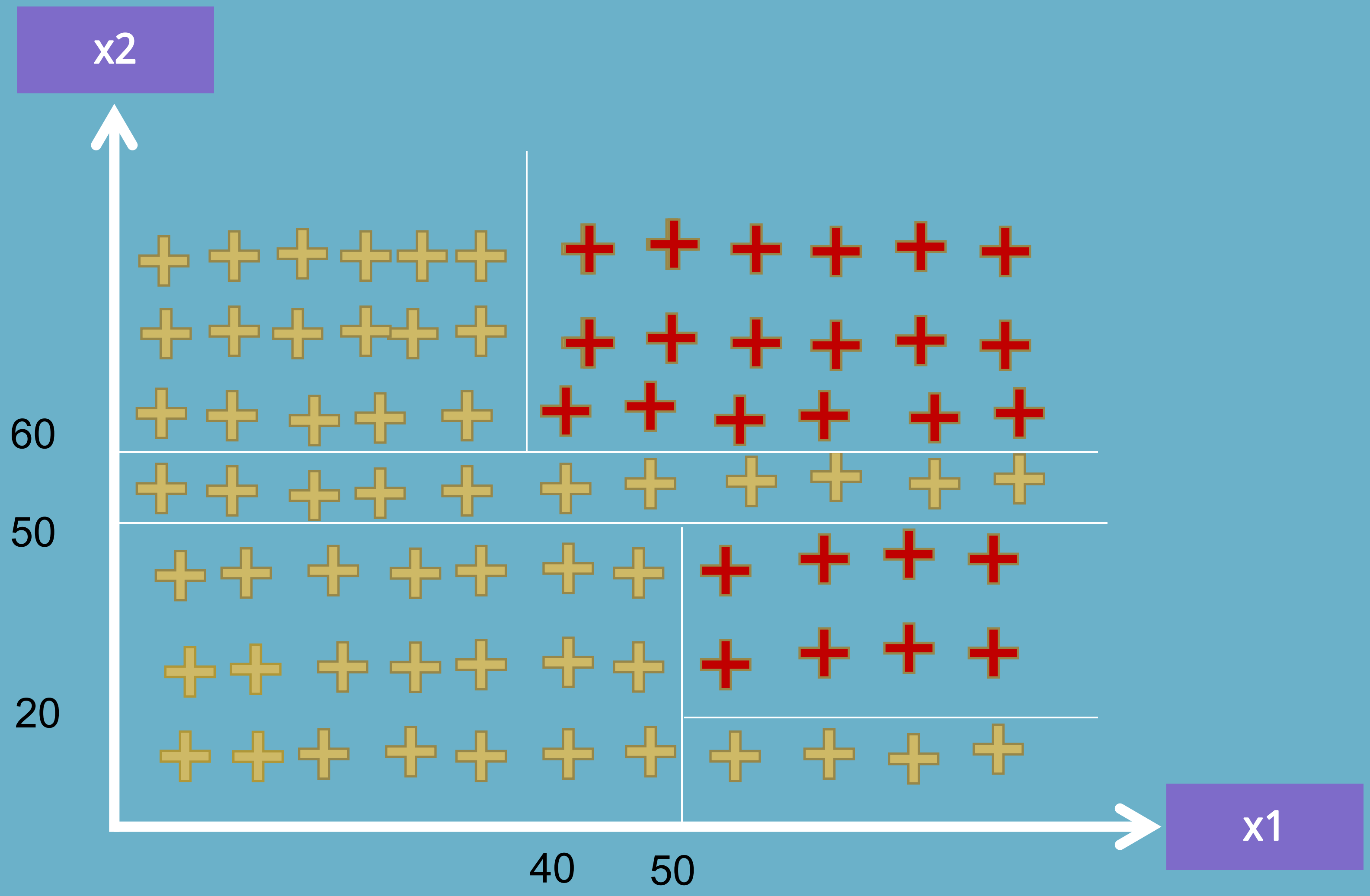
Logistic Regression Intuition

DEMONSTRATE

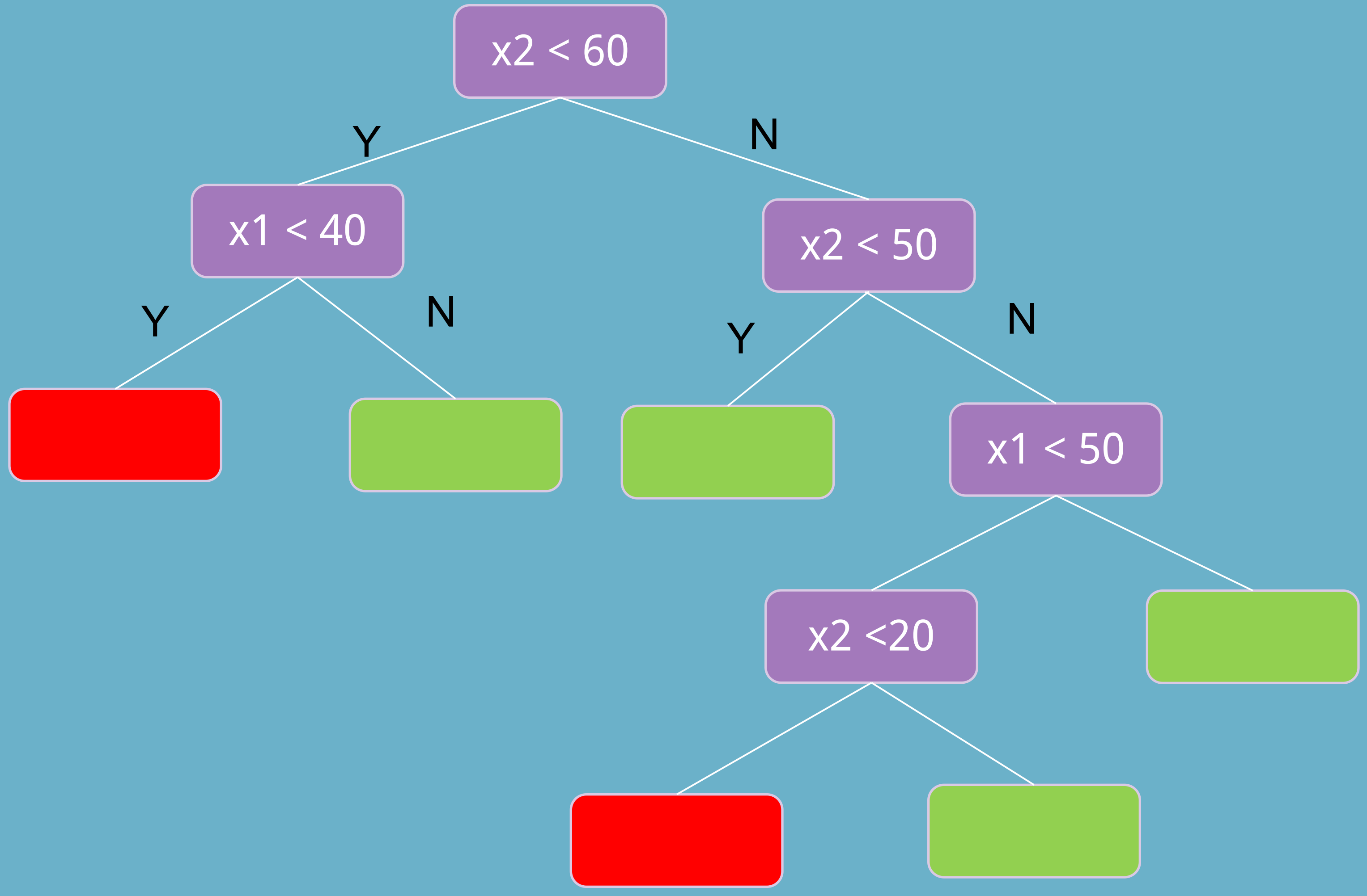
LET's CODE



Decision Tree Algorithm



Decision Tree Algorithm



Random Forest Generation

ENSEMBLE LEARNING

- Run an Algorithm multiple times
- More the recurring, better the results

PROBLEM STATEMENT

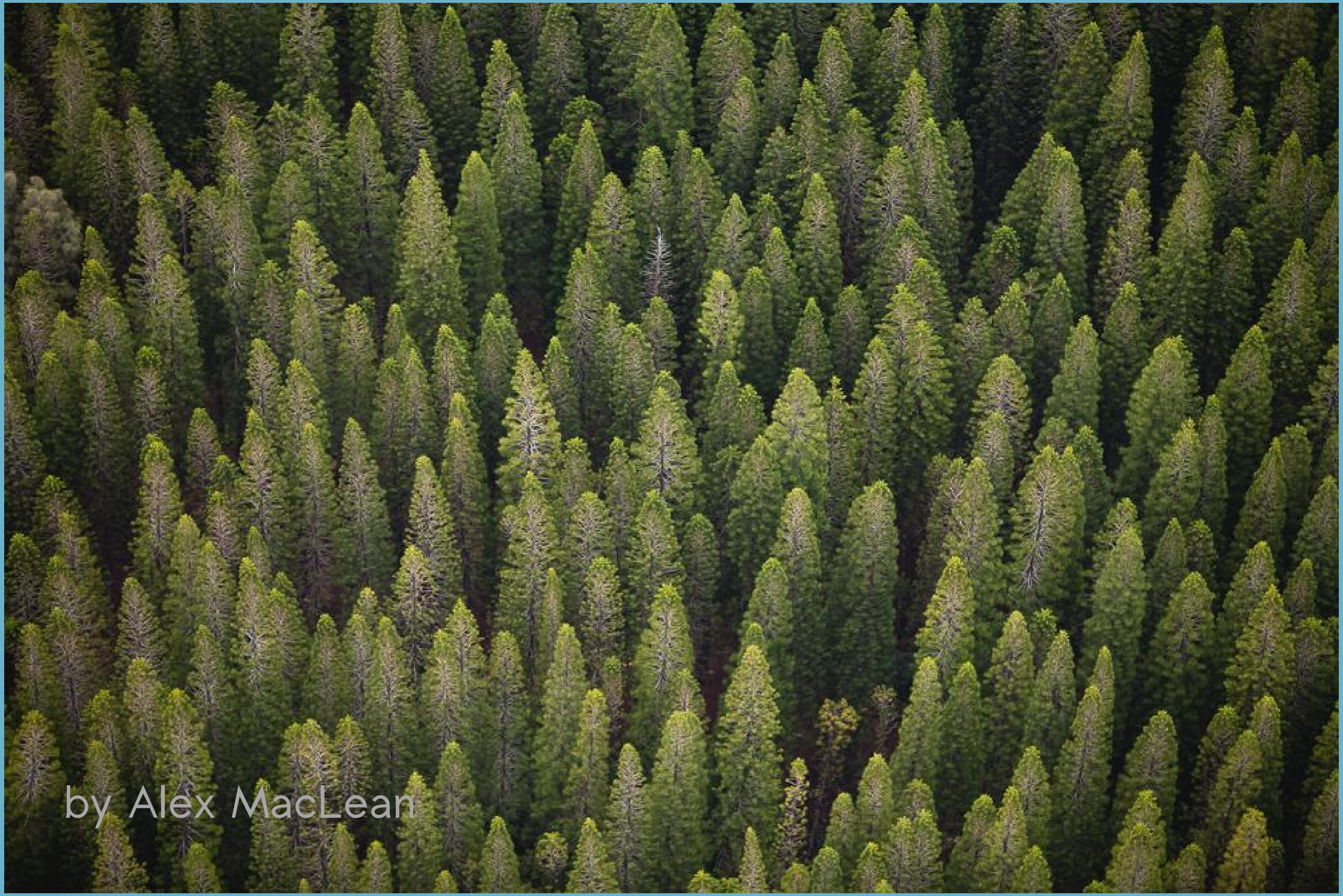
- The **truth** about the salary stays, since the advent of Decision Tree Algorithm has emerged now we can find a more precise value to catch **BLUFF**
- Apply **DTA** over the **Position_Salaries.csv** and find out the reality
- What is the **predicted salary**? is it **above** 160K or **around** 160K?
- Should the Job be given with **expected Offer**?

Random Forest Generation

STEP BY STEP

1. Pick a random K Data points from the Training Set
2. Build the Decision Tree associated to these K data points
3. Choose the no. of Ntree of trees you want to build and repeat step 1 & 2
4. For a new data point, make each one of your Ntree predict the category to which data points belongs, and assign the new data point to the category that wins the majority vote.

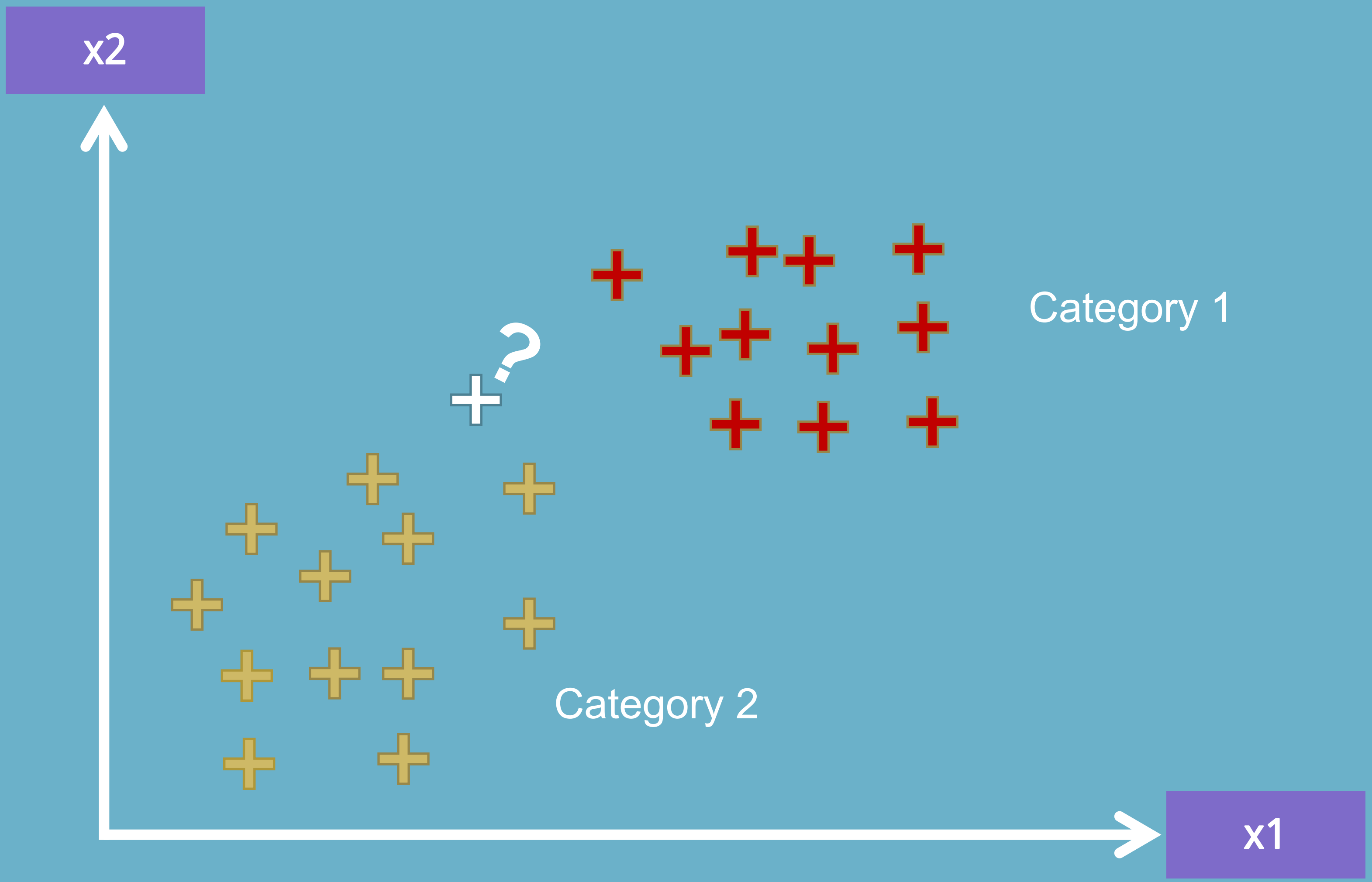
Random Forest Generation



Random Forest Generation

- [Research paper]

K Nearest Neighbor



K Nearest Neighbor

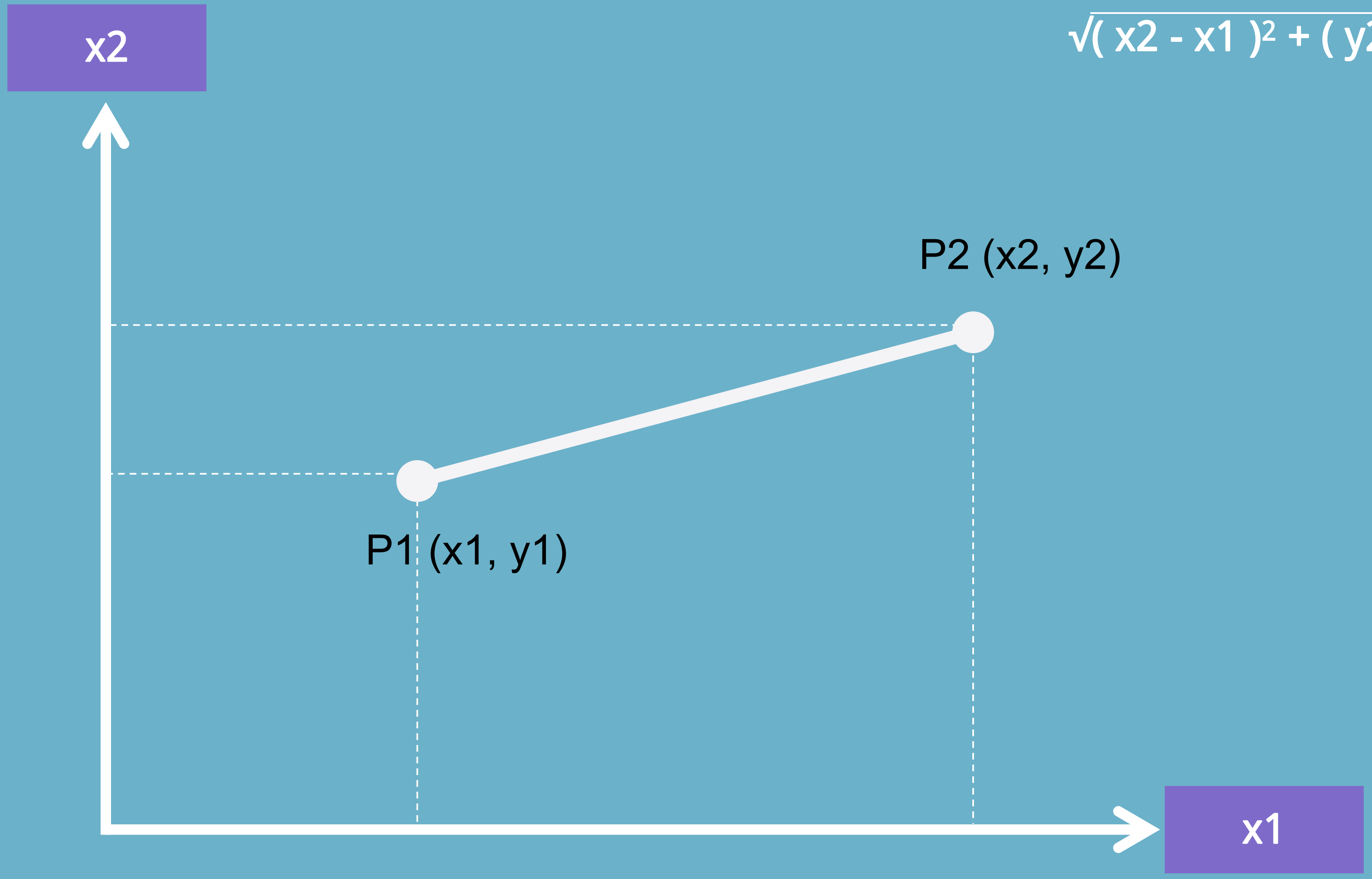
1. Choose the number of K (Default = 5)
2. Take the K nearest neighbor of the new Data point, according to Euclidean distance
3. Among these K neighbors, count the number of data points in each category
4. Assign the new data point to the category where you counted the most neighbors

YOUR MODEL IS READY TO ROLL!

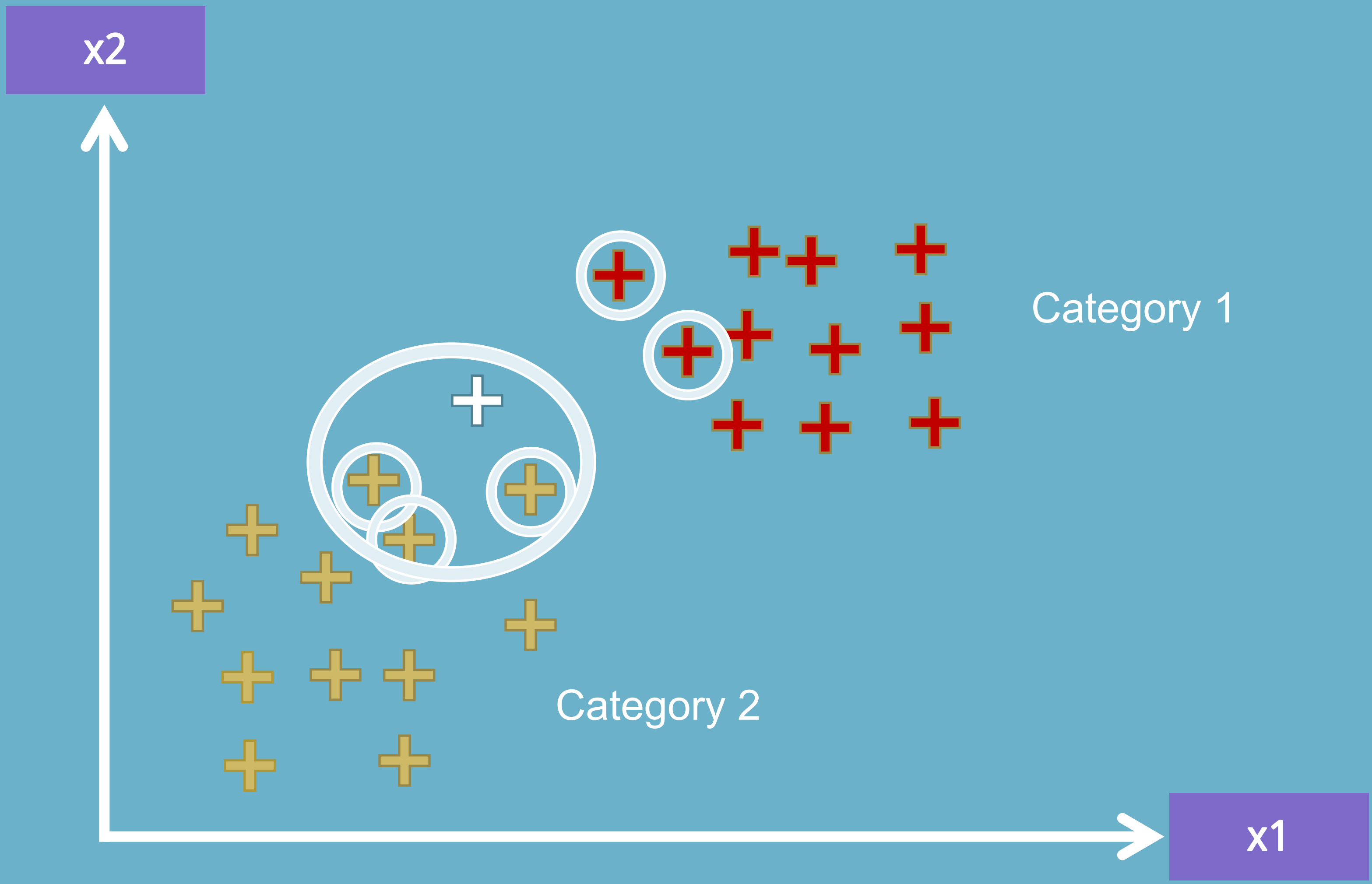
K Nearest Neighbor

EUCLIDEAN DISTANCE

$$\sqrt{(x2 - x1)^2 + (y2 - y1)^2}$$



K Nearest Neighbor

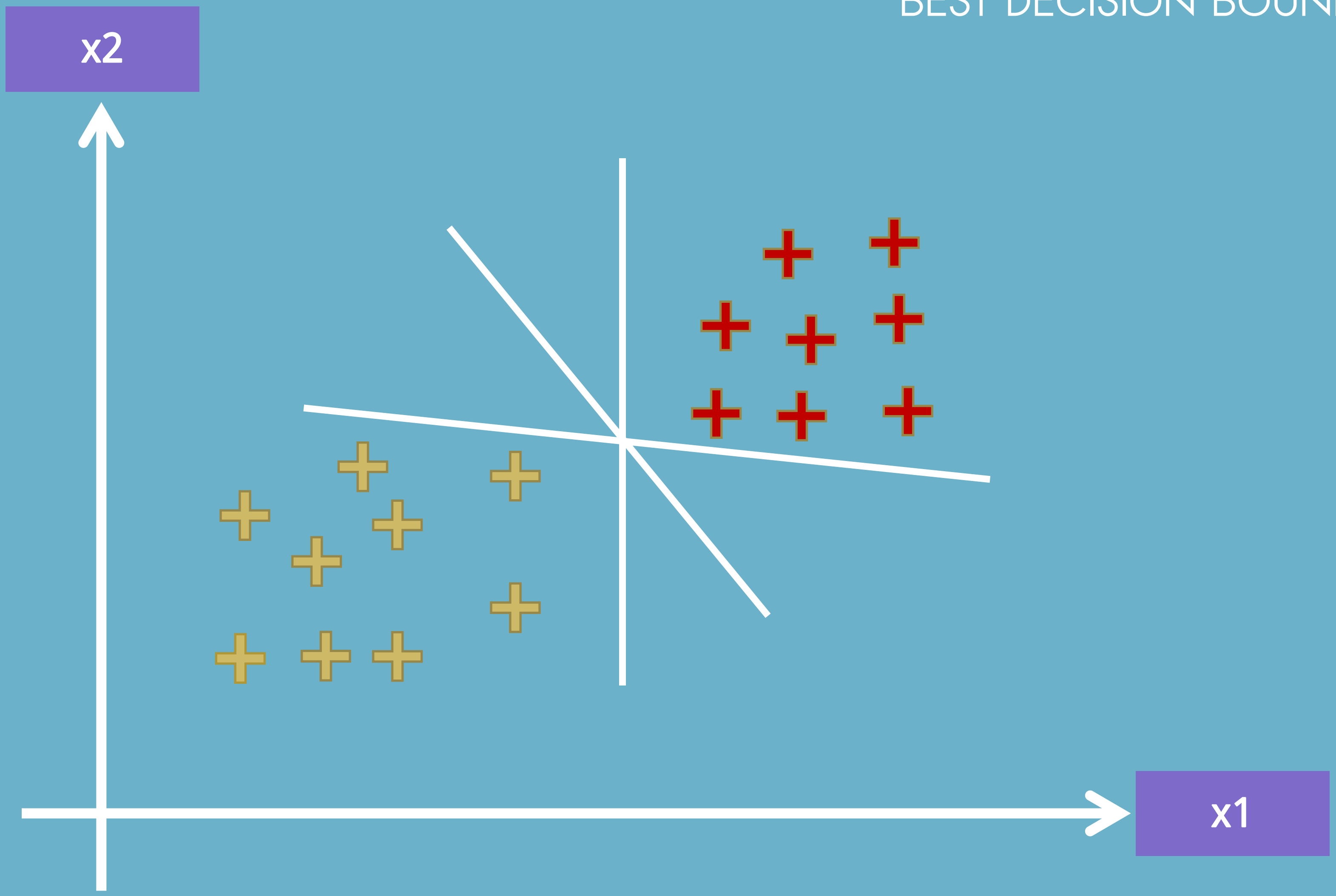


Support Vector Machine

- Developed in 1960
- Later focused in 1990
- Until now, when it's importance grew up

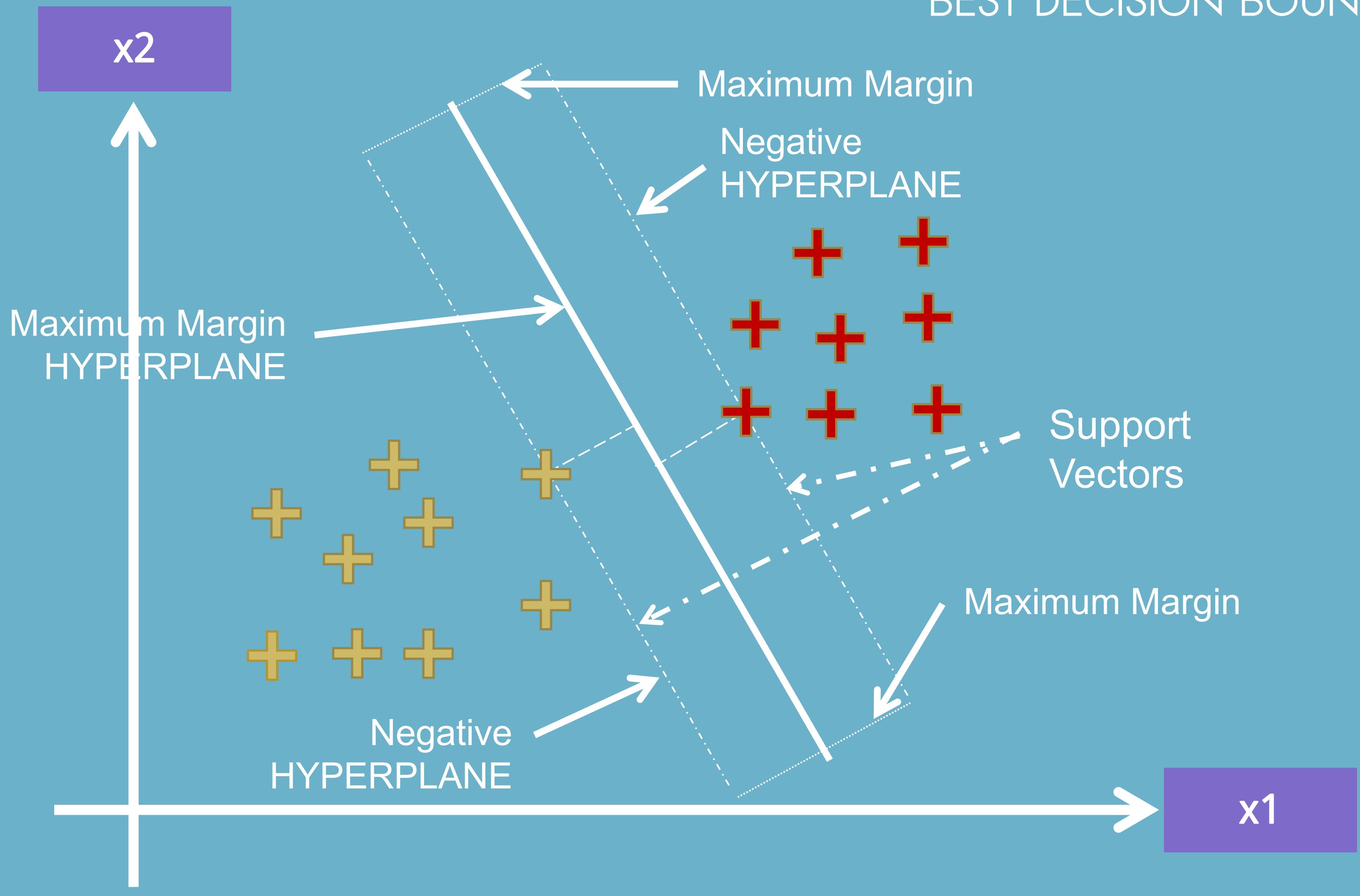
Support Vector Machine

BEST DECISION BOUNDRY

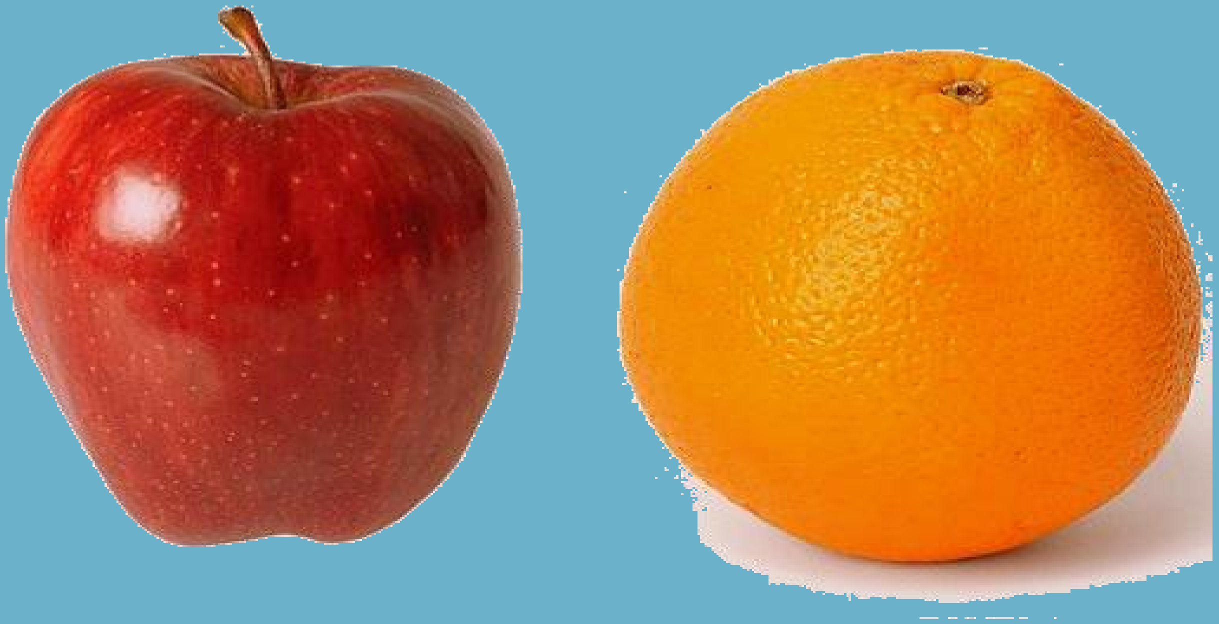


Support Vector Machine

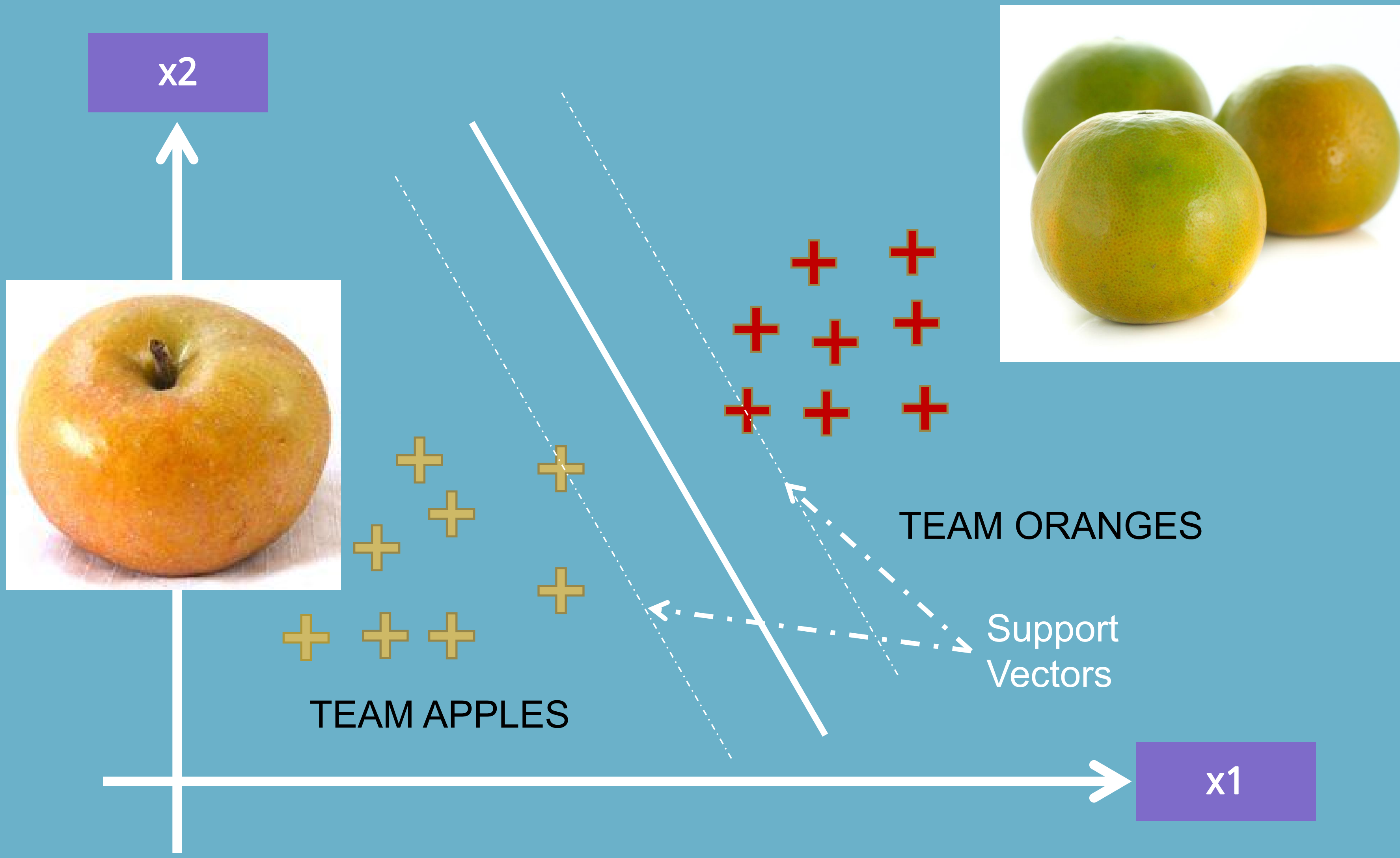
BEST DECISION BOUNDRY



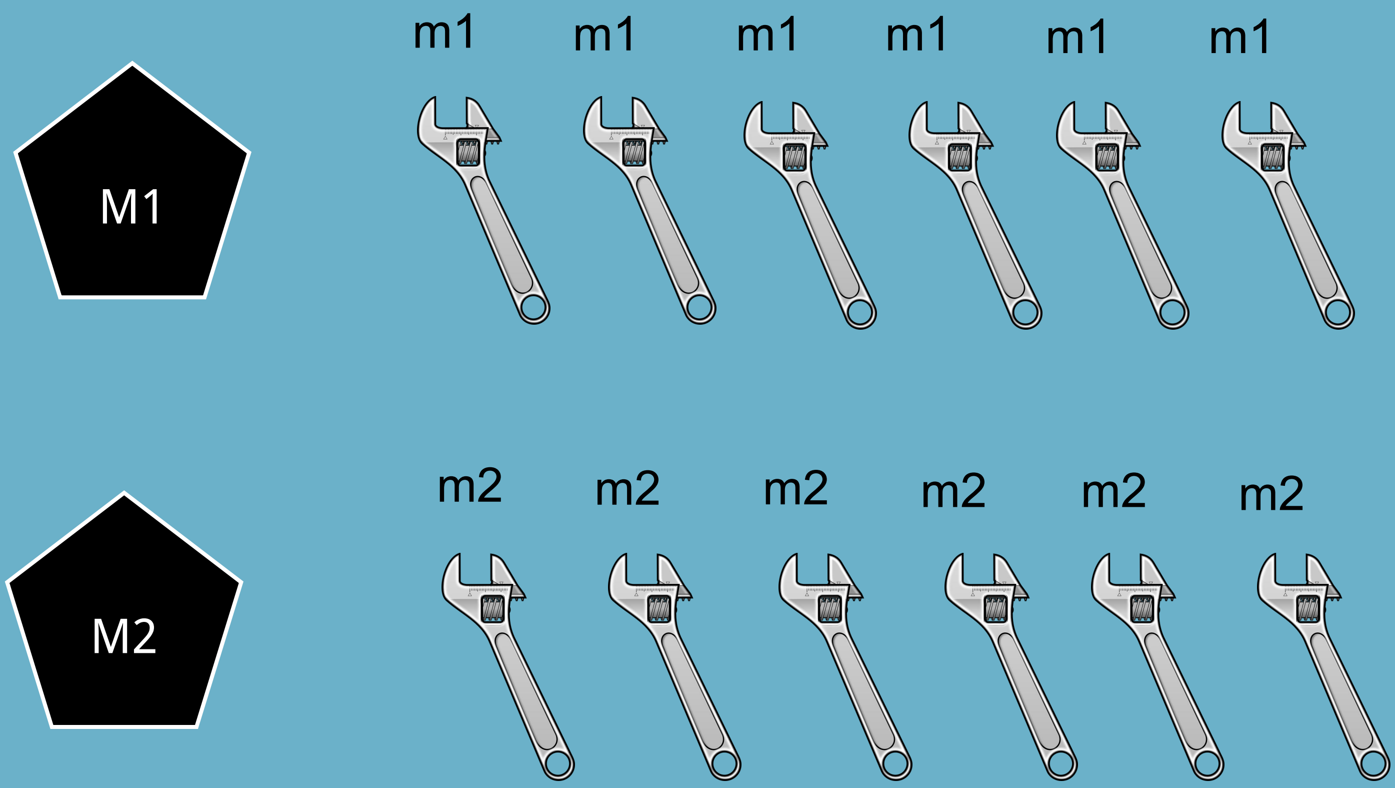
What's so special about SVMs?



Support Vector Machine

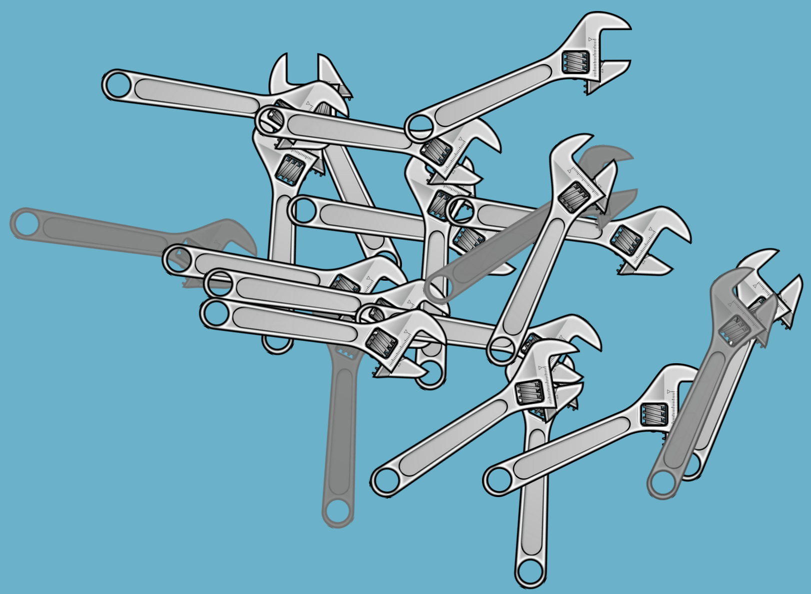


Bayes Theorem



Bayes Theorem

PROBABLITY OF DEFECT



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

Bayes Theorem

WHAT DO WE KNOW?

- Machine 1 = 30 spanners / Hr
- Machine 2 = 20 spanners / Hr

- Out of all produced parts:
  - 1% are defected

- Out of all defective parts:
  - 50% yeild from M1
  - 50% yeild from M2

QUESTION:

- What is the probablity of that a part produced by Machine 2 is Defective?

Bayes Theorem

WHAT HAVE WE LEARNT?

- Machine 1 = 30 spanners / Hr — P(M1) = 30/50 = 0.6
- Machine 2 = 20 spanners / Hr — P(M1) = 20/50 = 0.4

- Out of all produced parts:
  - 1% are defected — P(Defect) = 1%

- Out of all defective parts:
  - 50% yeild from M1 — P(M1 | Defect) = 50%
  - 50% yeild from M2 — P(M2 | Defect) = 50%

- QUESTION **P (Defect | M2) = ?**

Bayes Theorem

- Machine 1 = 30 spanners / Hr
- Machine 2 = 20 spanners / Hr
- Out of all produced parts:
  - 1% are defected
- Out of all defective parts:
  - 50% yeild from M1
  - 50% yeild from M2

P(M2) = 20/50 = 0.4
P(Defect) = 1%
P(M2 | Defect) = 50%
P (Defect | M2) = ?

$$P(\text{Defect} \mid M2) = \frac{P(M2 \mid \text{Defect}) * P(\text{Defect})}{P(M2)}$$

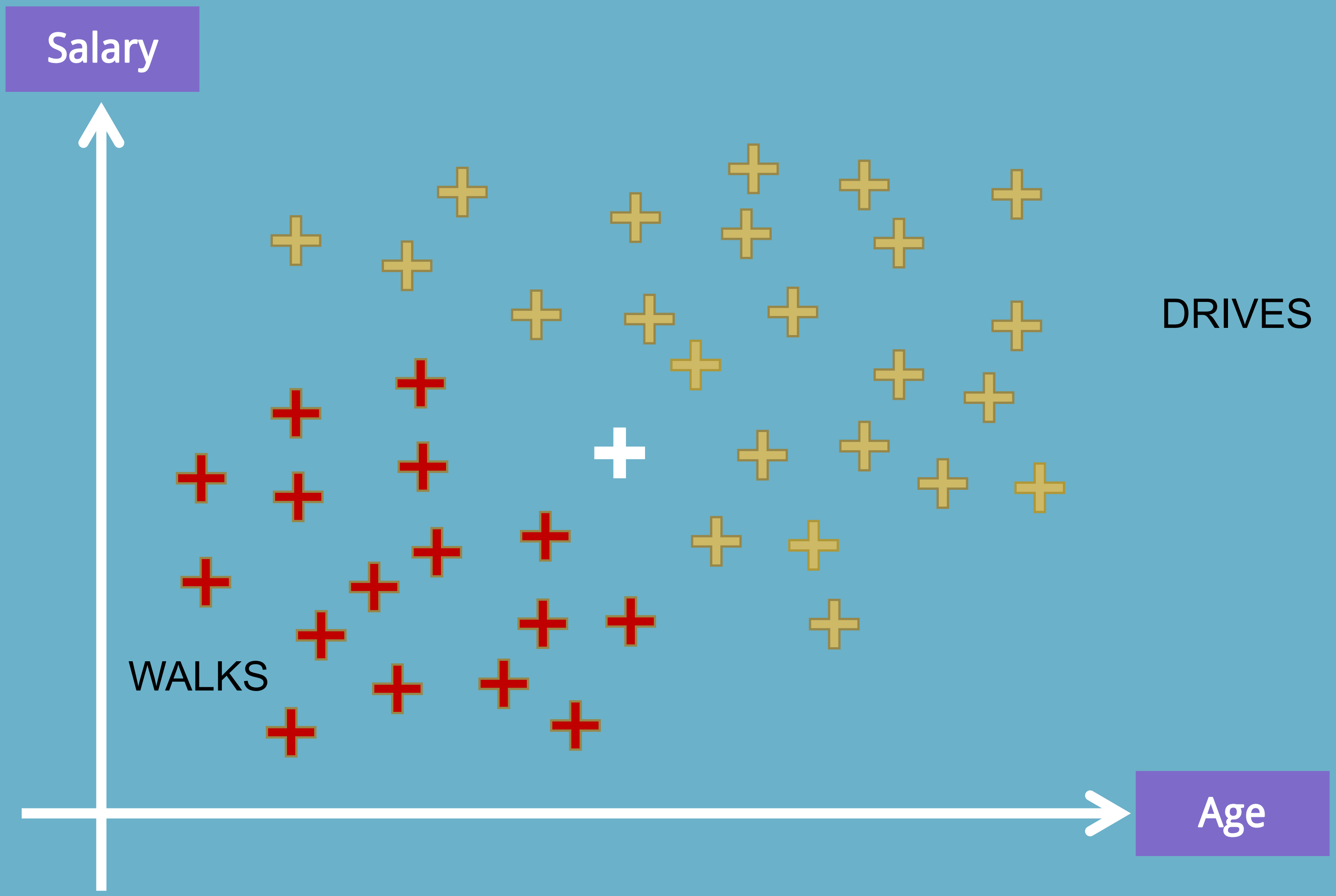
Bayes Theorem

SUBSTITUTION

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

NOT !!! AGAIN !!!

Naive Bayes



Naive Bayes

STEP 1

#3 LIKELIHOOD

#1 PRIOR PROBABILITY

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

#4 POSTIRIOR PROBABILITY

#2 MARGINAL LIKELIHOOD

Naive Bayes

STEP 2

#3 LIKELIHOOD

#1 PRIOR PROBABILITY

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

#4 POSTIRIOR PROBABILITY

#2 MARGINAL LIKELIHOOD

Naive Bayes

STEP 3

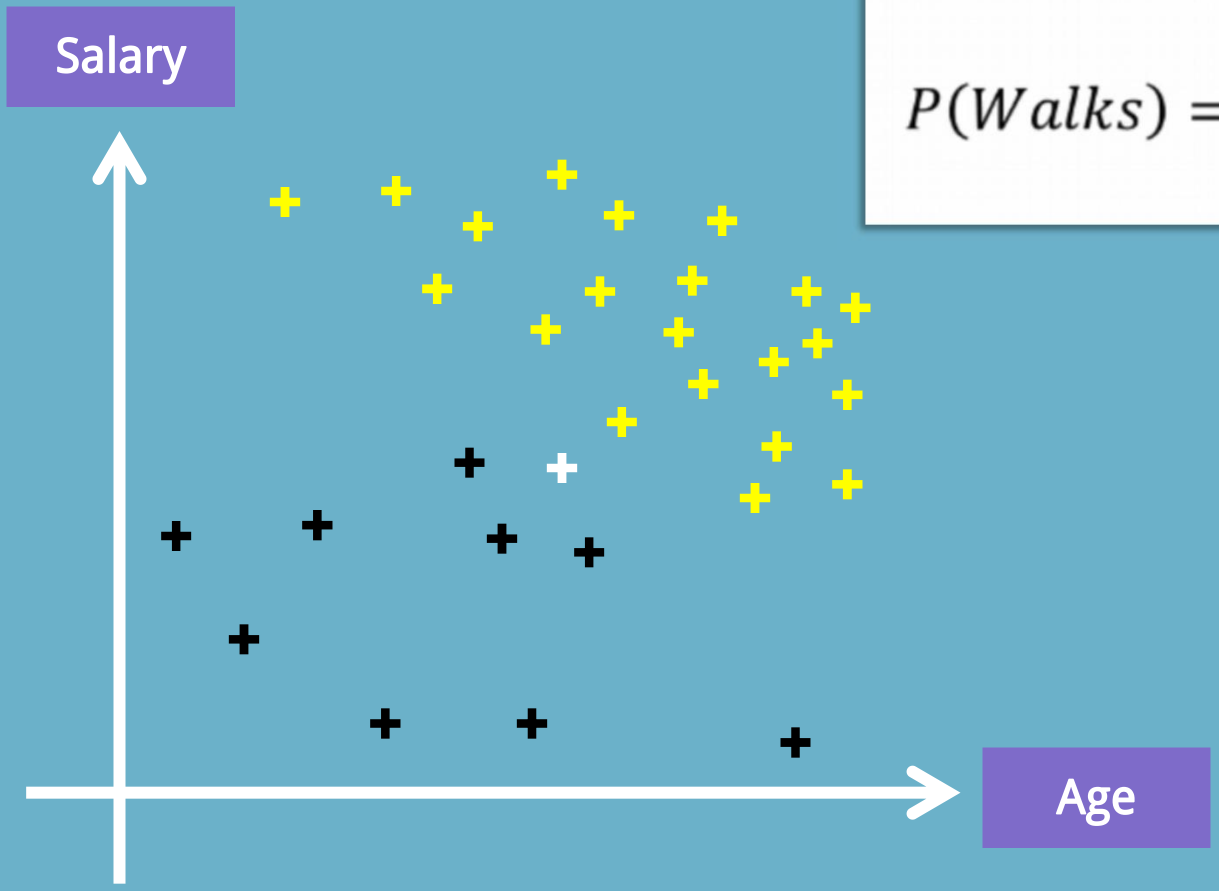
$$P(Walks|X) \ v.s. \ P(Drives|X)$$

Naive Bayes

STEP 1

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

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

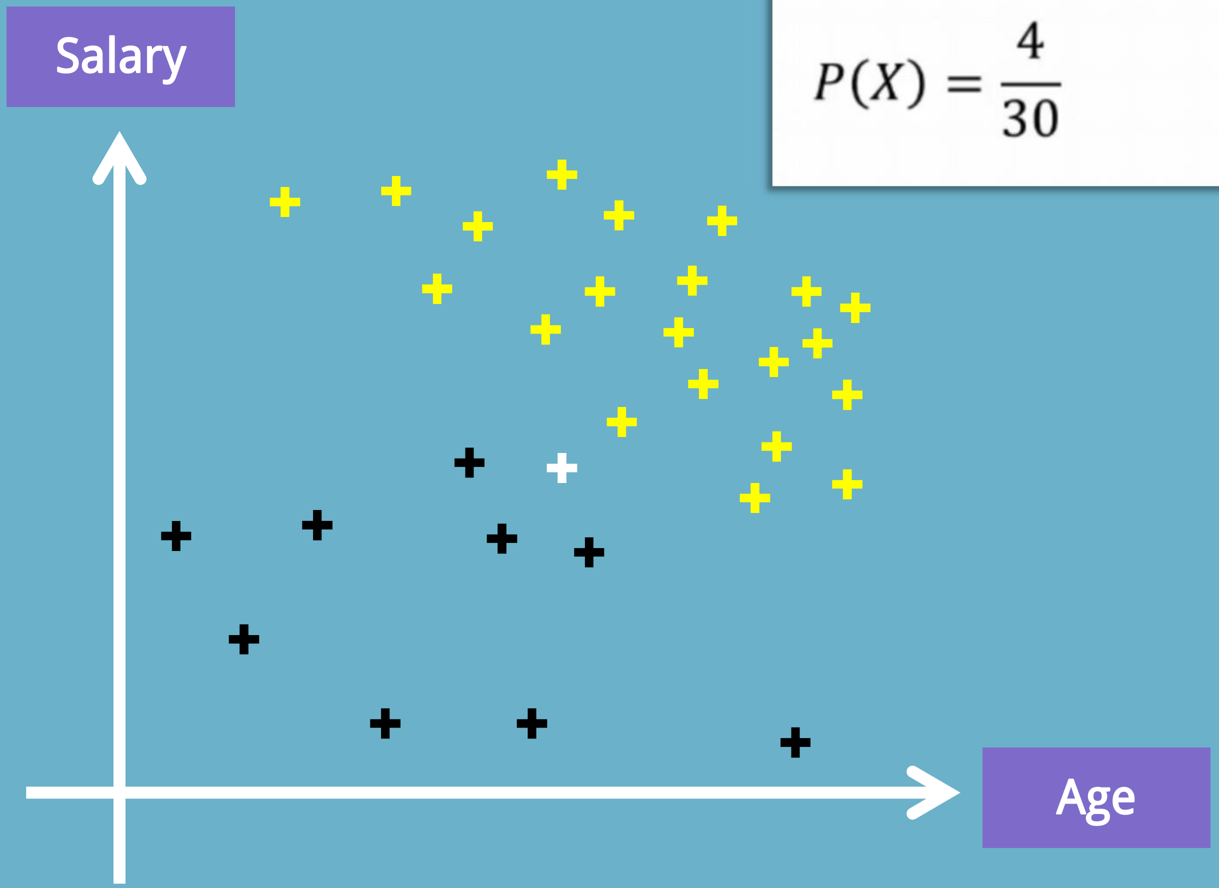


Naive Bayes

STEP 1

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

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

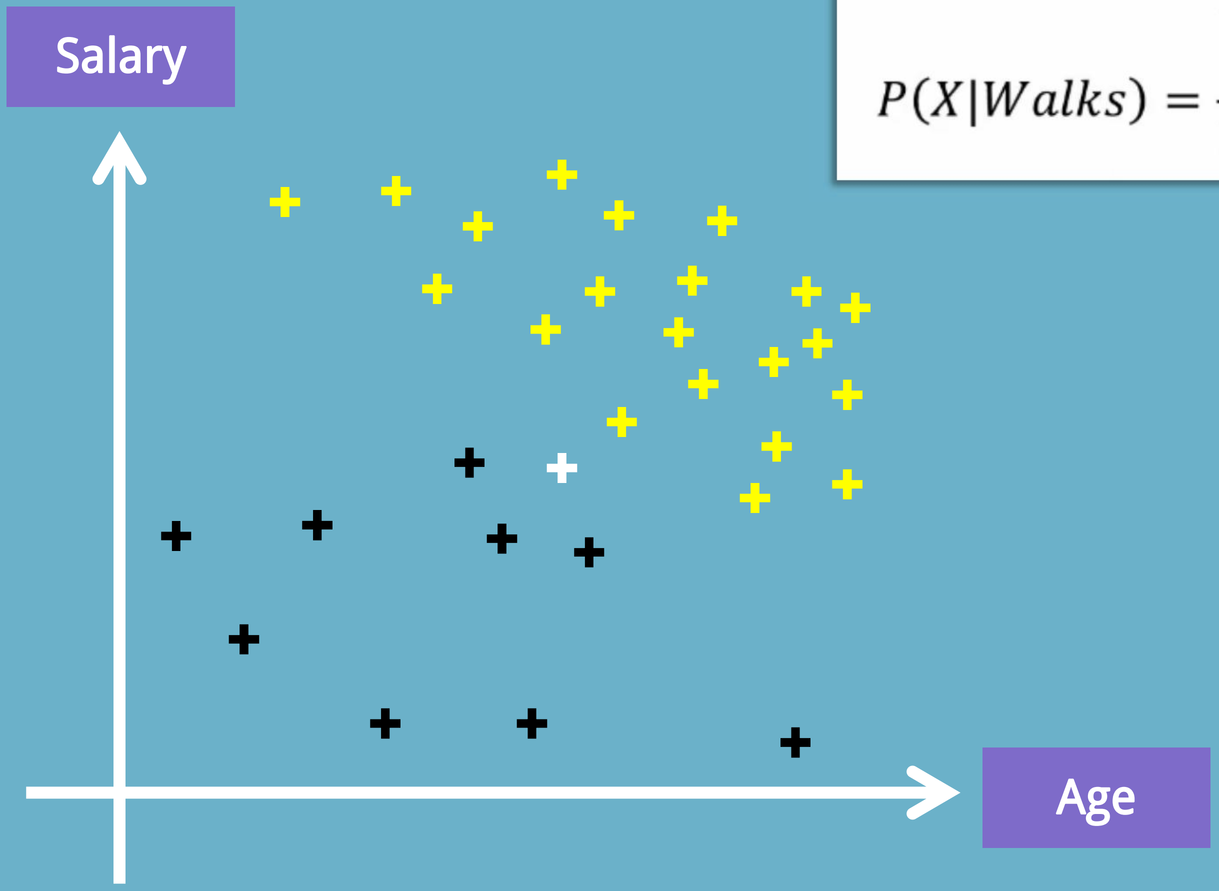


Naive Bayes

STEP 1

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

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



Naive Bayes

STEP 1

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

Naive Bayes

STEP 2

Perform Same for **P (Drives | X)**

STEP 3

Compare both results to assign new Data point

/mITuSkillologies

@mitu_skillologies

@mitu_group

miTu skillologies

/company/mitu-skillologies