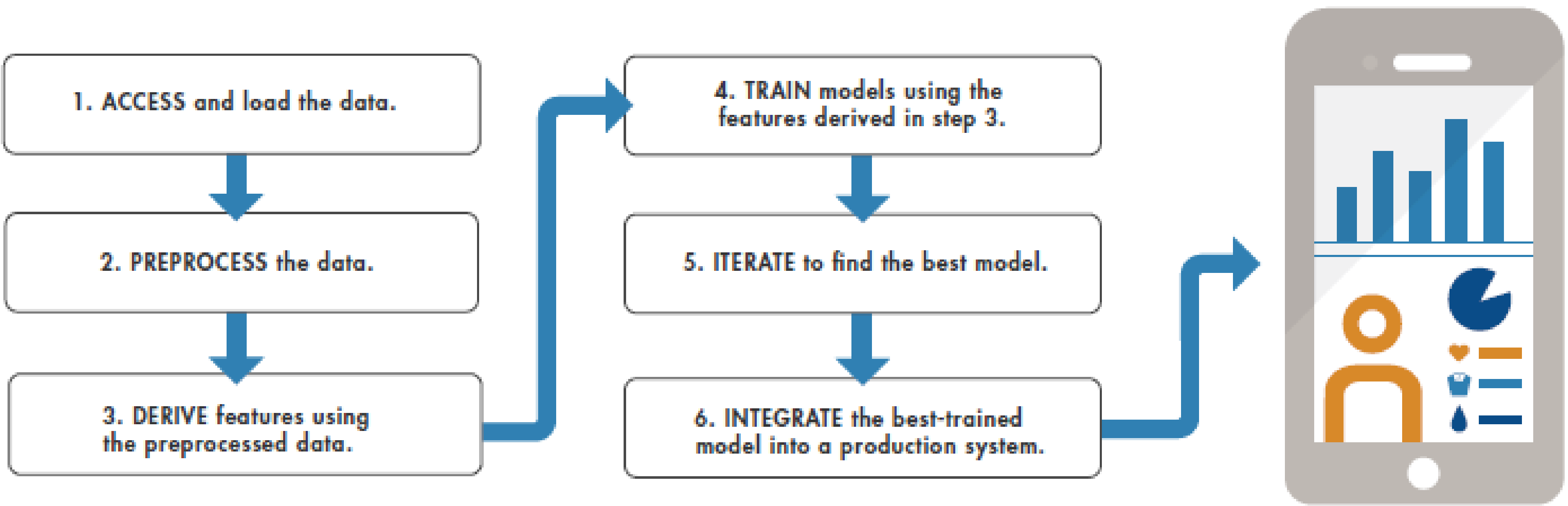
ML Workflow


1. ACCESS and load the data.
2. PREPROCESS the data.
3. DERIVE features using the preprocessed data.
4. TRAIN models using the features derived in step 3.
5. ITERATE to find the best model.
6. INTEGRATE the best-trained model into a production system.

Data PreProcessing

- ***Take care of Missing Data***

- ***Encoding categorical dataset***

- ***Splitting the dataset into training set and test set***

- ***Feature Scaling***

Dummy Variables

- A Dummy variable or Indicator Variable is an artificial variable created to represent an attribute with two or more distinct categories/levels.

- Regression analysis treats all independent (X) variables in the analysis as numerical.

- To include columns like Gender, Product Brand, then Dummy variables are created in this situation to trick the regression algorithm into correctly analyzing column variables.

Encoding Categorical Variable

Feature Scaling

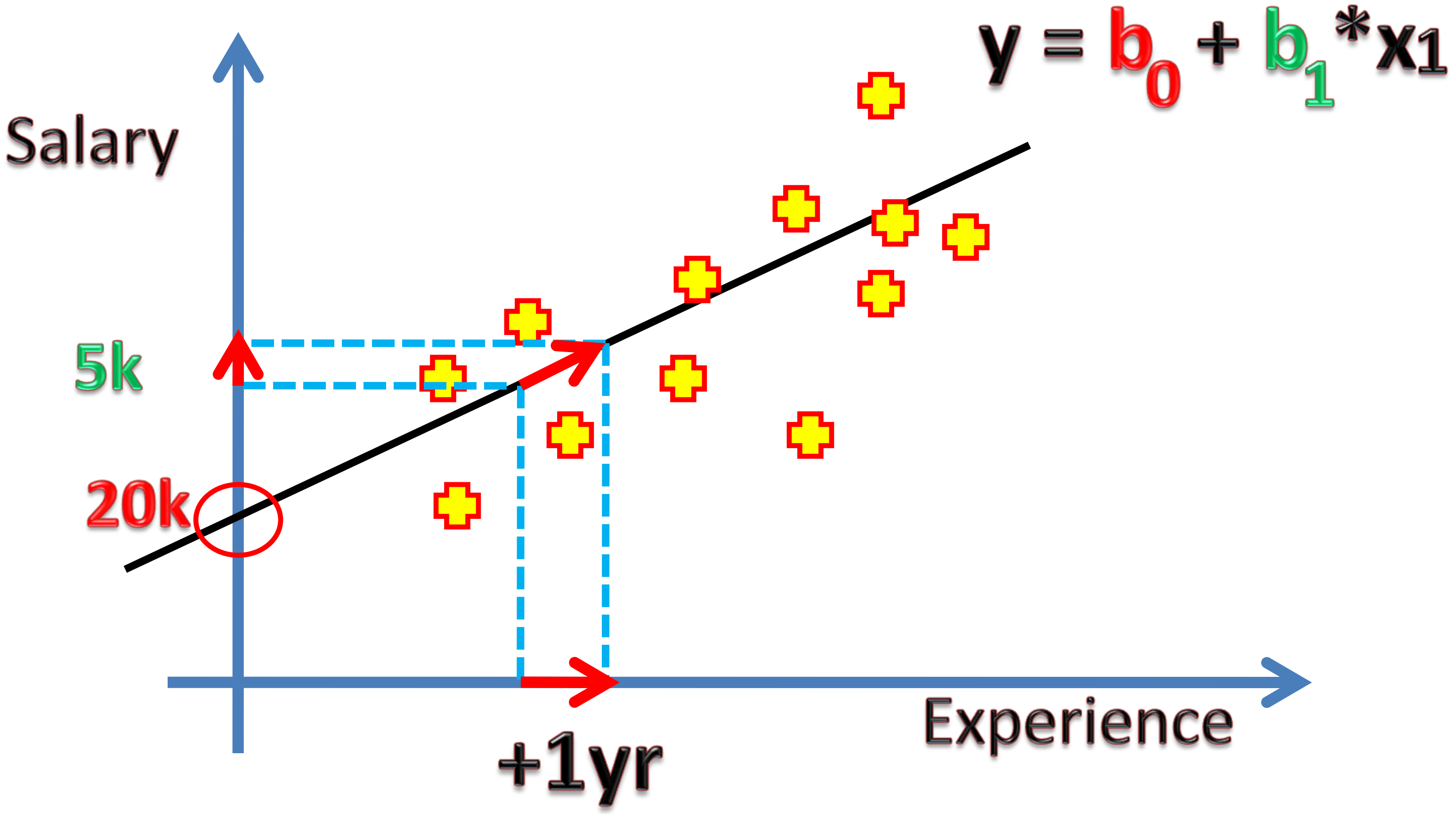
- StandardScaler will **transform** the data such that its distribution will have a **mean value 0** and **standard deviation of 1**
- This is useful while **comparing data** that corresponds to **different units**. In that case, we want to **remove the units**.
- This is done in a consistent way for all the data, we **transform the data** in a way that the **variance is unitary** and the **mean of the series is 0**.

Linear Regression

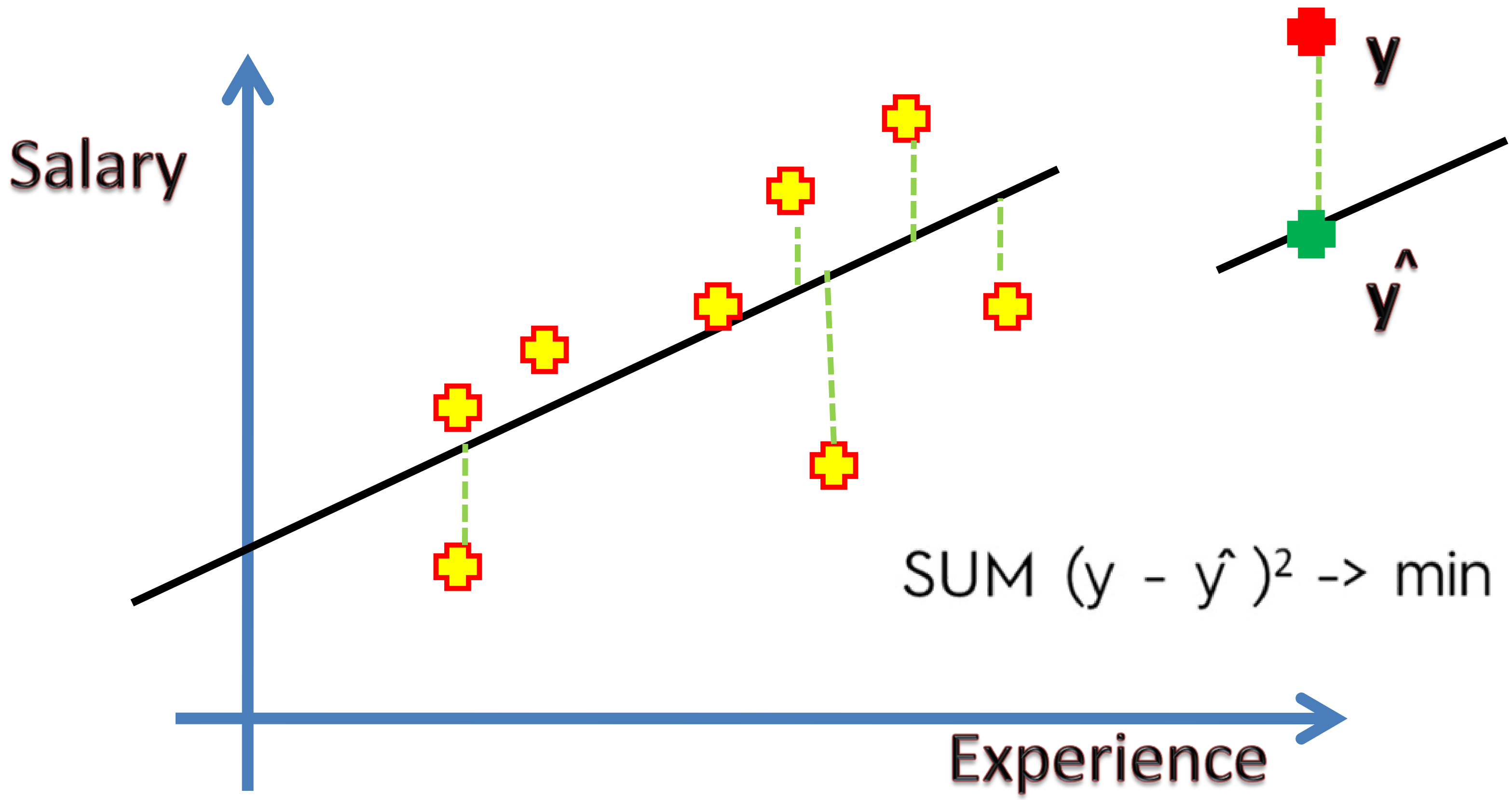
$$y = b_0 + b_1 * x_1$$

y is the dependent variable
x1 is independent variable
b1 is coefficient of X or the slope of the line
b0 is the constant called intercept

Linear Regression



Linear Regression



Multiple Linear Regression

$$y = b_0 + b_1 * x_1 + b_2 * x_2 + \ldots + b_n * x_n$$

Dummy Variables

- A Dummy variable or Indicator Variable is an artificial variable created to represent an attribute with two or more distinct categories/levels.
- Regression analysis treats all independent (X) variables in the analysis as numerical.
- To include columns like Gender, Product Brand, then Dummy variables are created in this situation to trick the regression algorithm into correctly analyzing column variables.

Encoding Categorical Variable

R-squared

- *R-Squared is the proportion of variation in the dependent (response) variable that has been explained by the model.*

Also known as coefficient of determination, it tells us how much is the variation in the dependent variable (salary) can be explained by the independent variable (Experience)

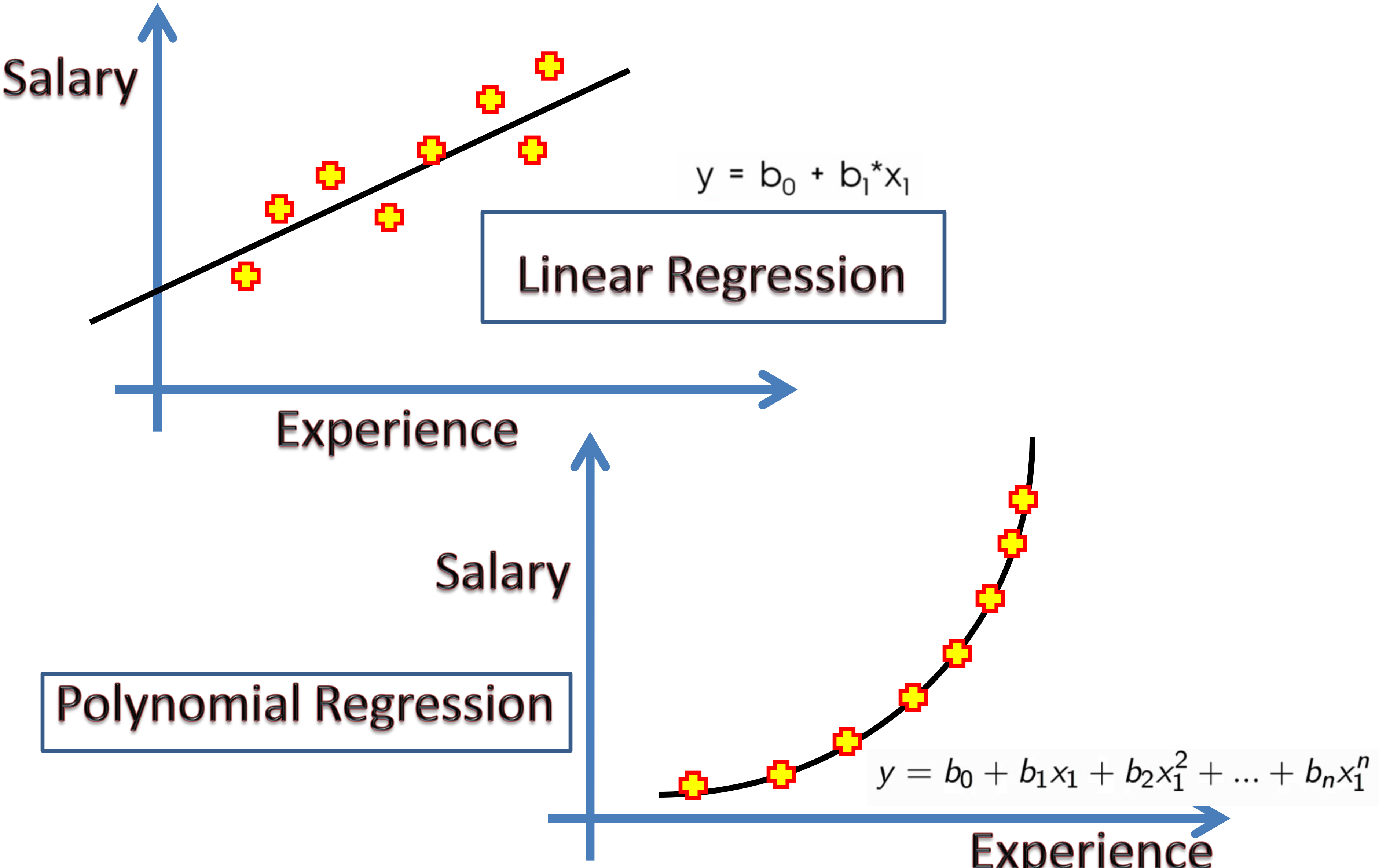
Adjusted R-squared

- *The adjusted R-squared is a modified version of R-squared that has been adjusted for the number ... l.*
- *The adj ... only if the new ter ...*
- *It decre ... sn't contribute to the model improvement.*

Vars	R-Sq	R-Sq(adj)
1	72.1	71.0
2	85.9	84.8
3	87.4	85.9
4	89.1	82.3
5	89.9	80.7

Polynomial Regression

$$y = b_0 + b_1 x_1 + b_2 x_1^2 + \dots + b_n x_1^n$$

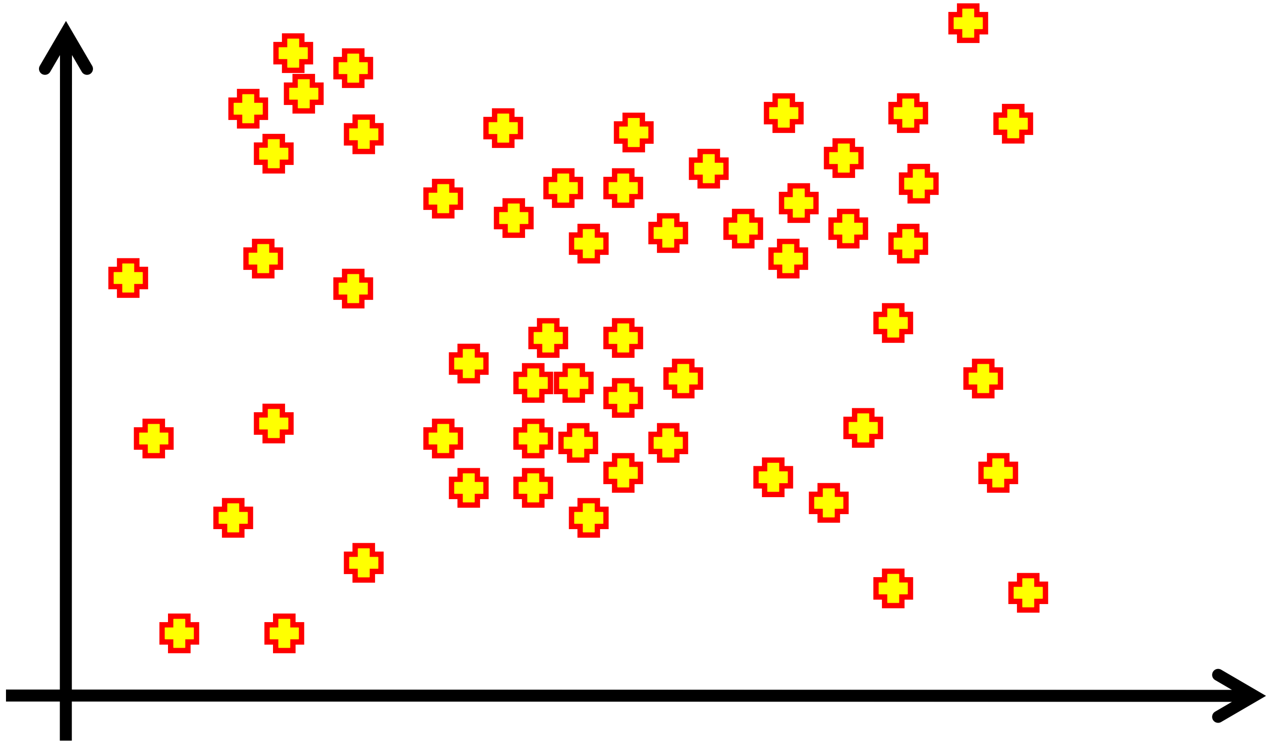


Salary
Experience
$y = b_0 + b_1{*}x_1$
Linear Regression
Salary
Experience
Polynomial Regression
$y = b_0 + b_1x_1 + b_2x_1^2 + ... + b_nx_1^n$

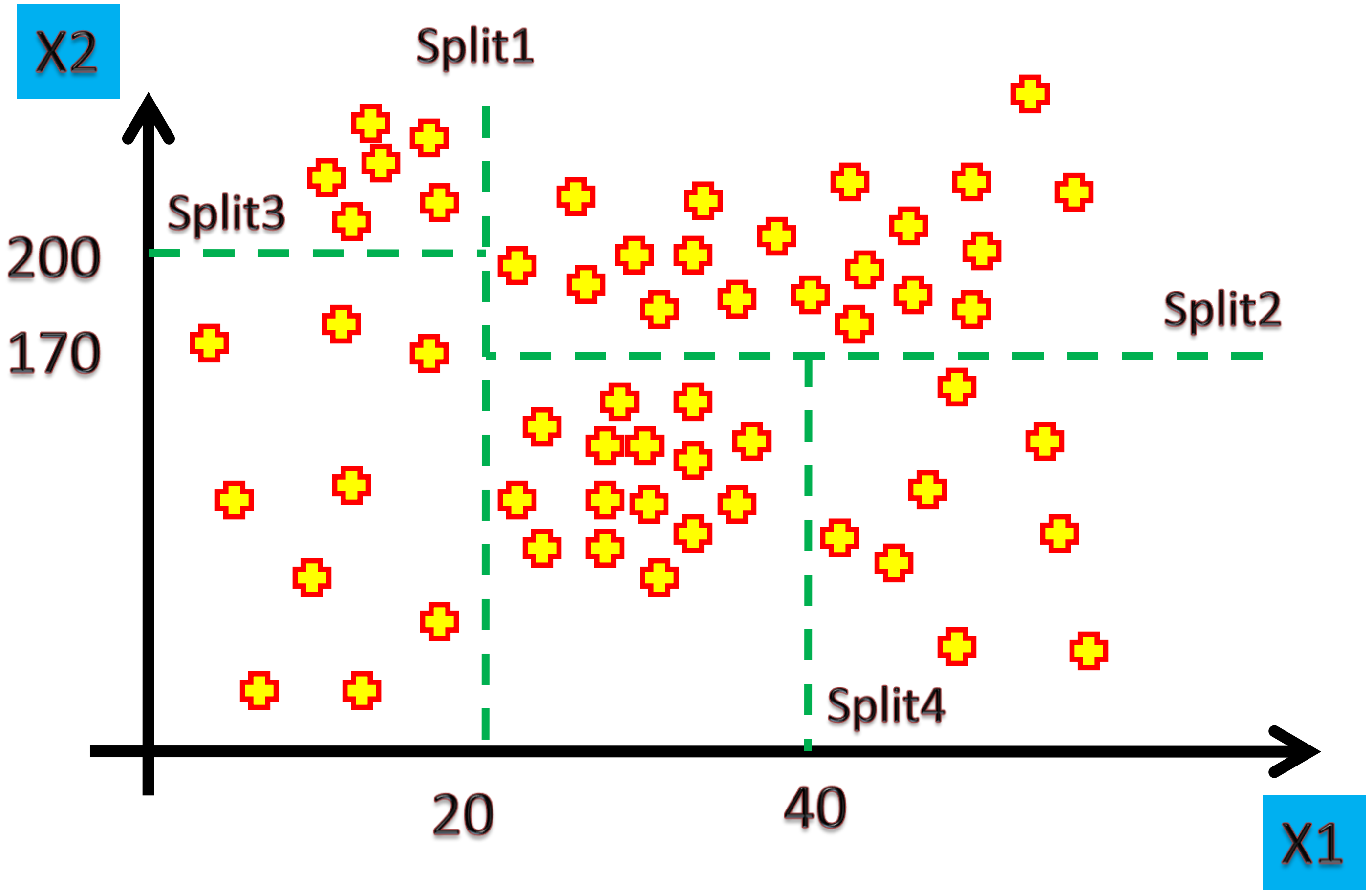
Decision Tree Regression

Decision tree builds regression or classification models in the form of a tree structure. It brakes down a dataset into smaller and smaller subsets while at the same time an associated decision tree is incrementally developed.

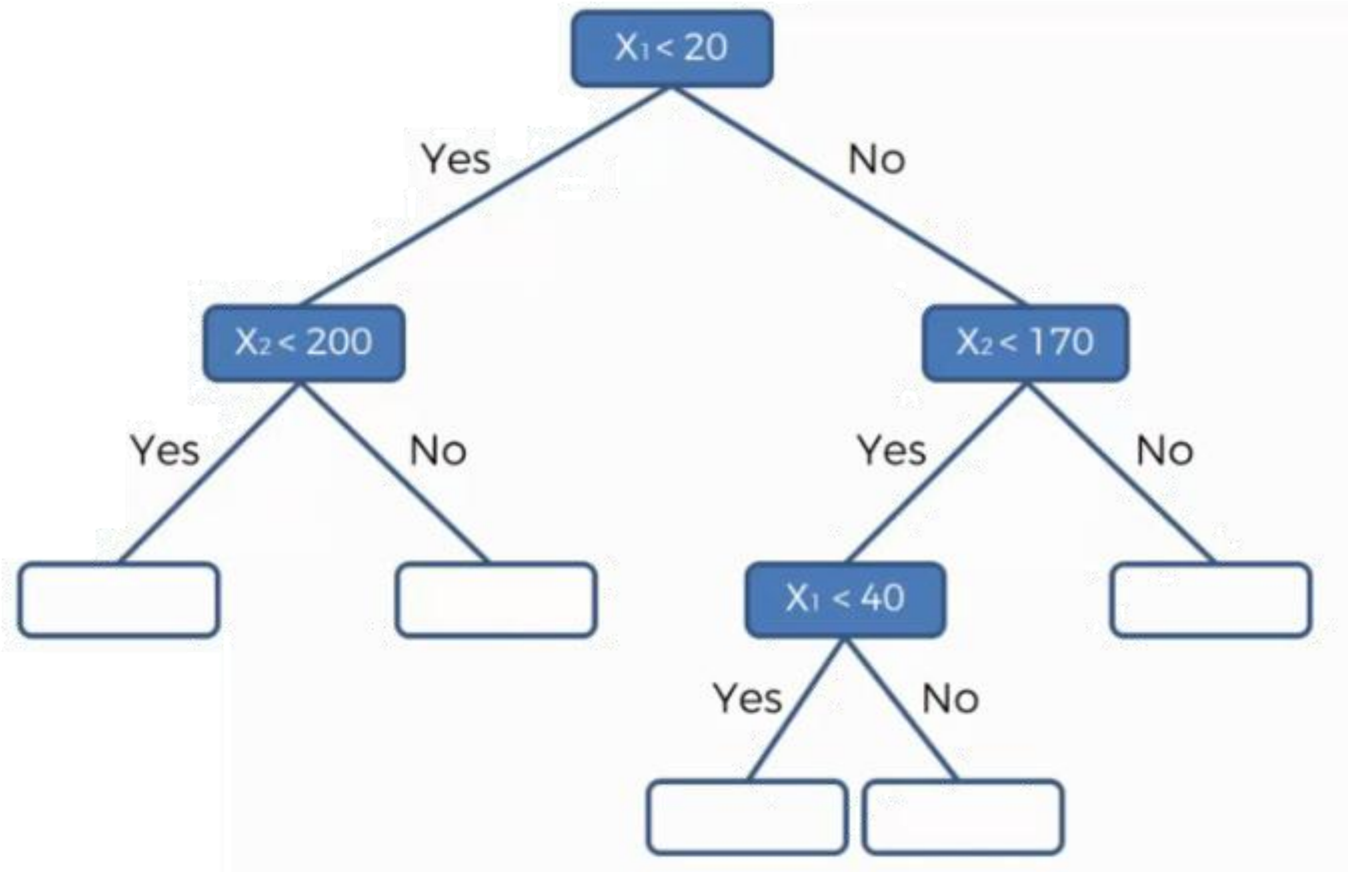
Decision Tree Regression



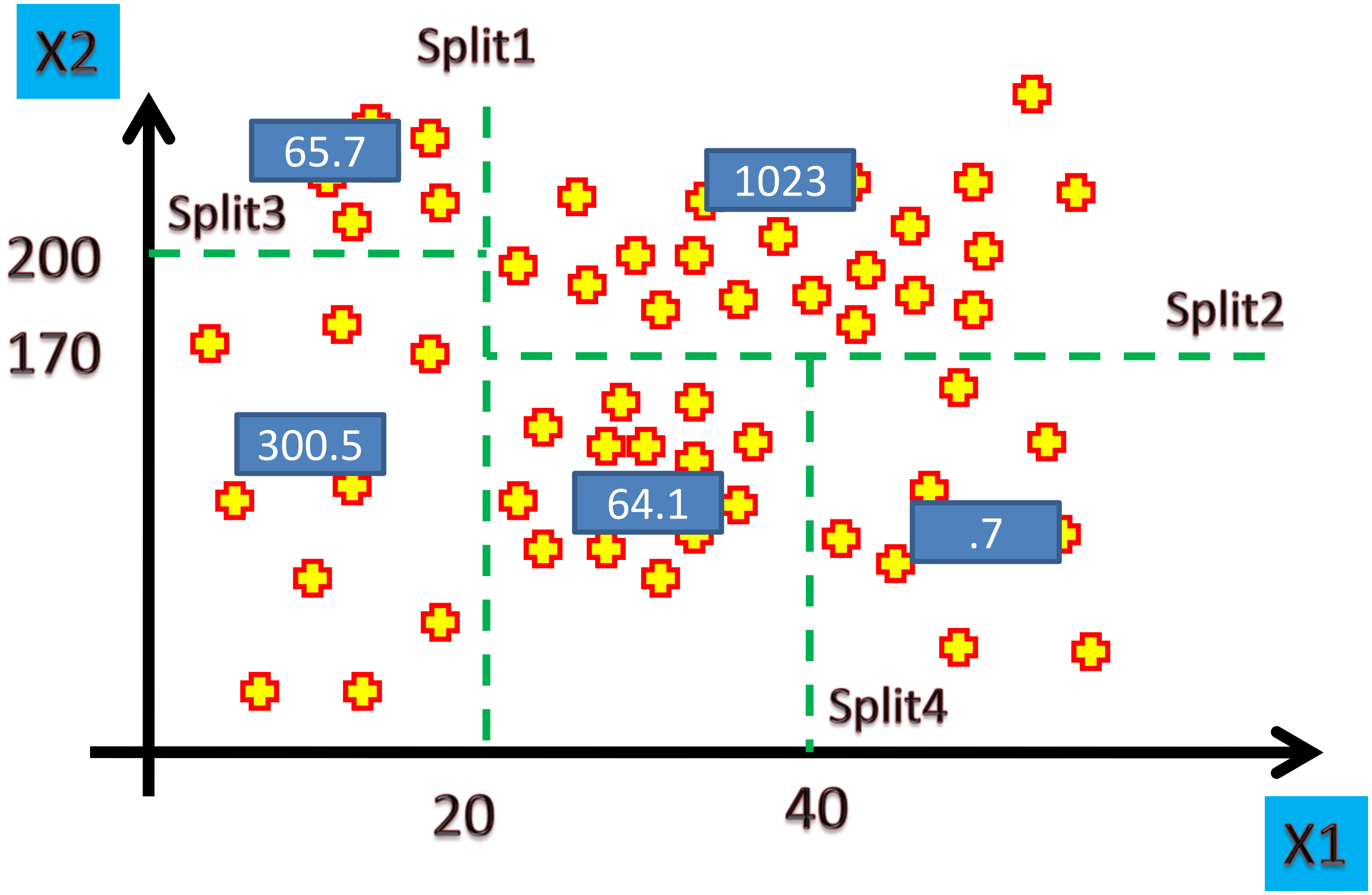
Decision Tree Regression



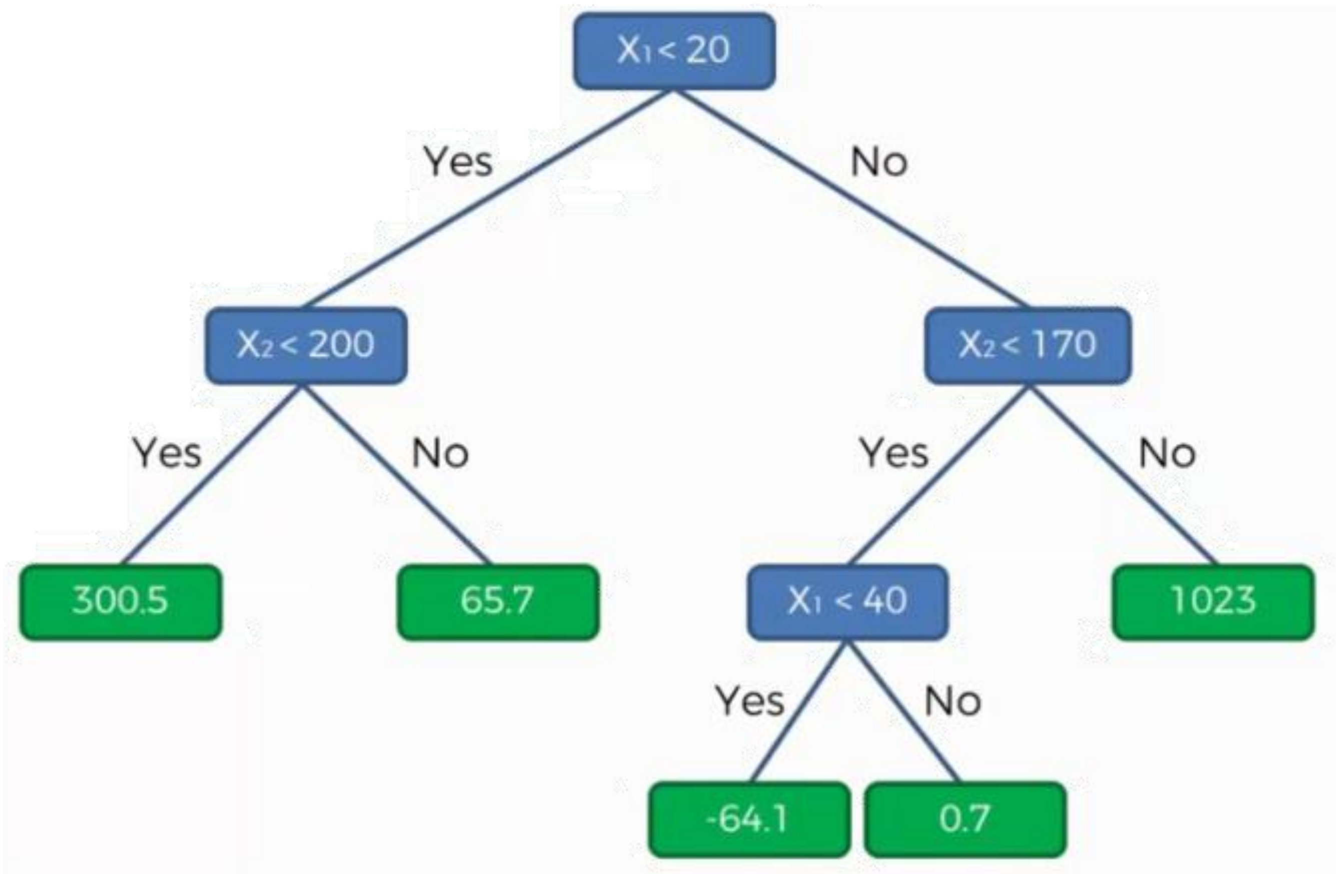
Decision Tree Regression



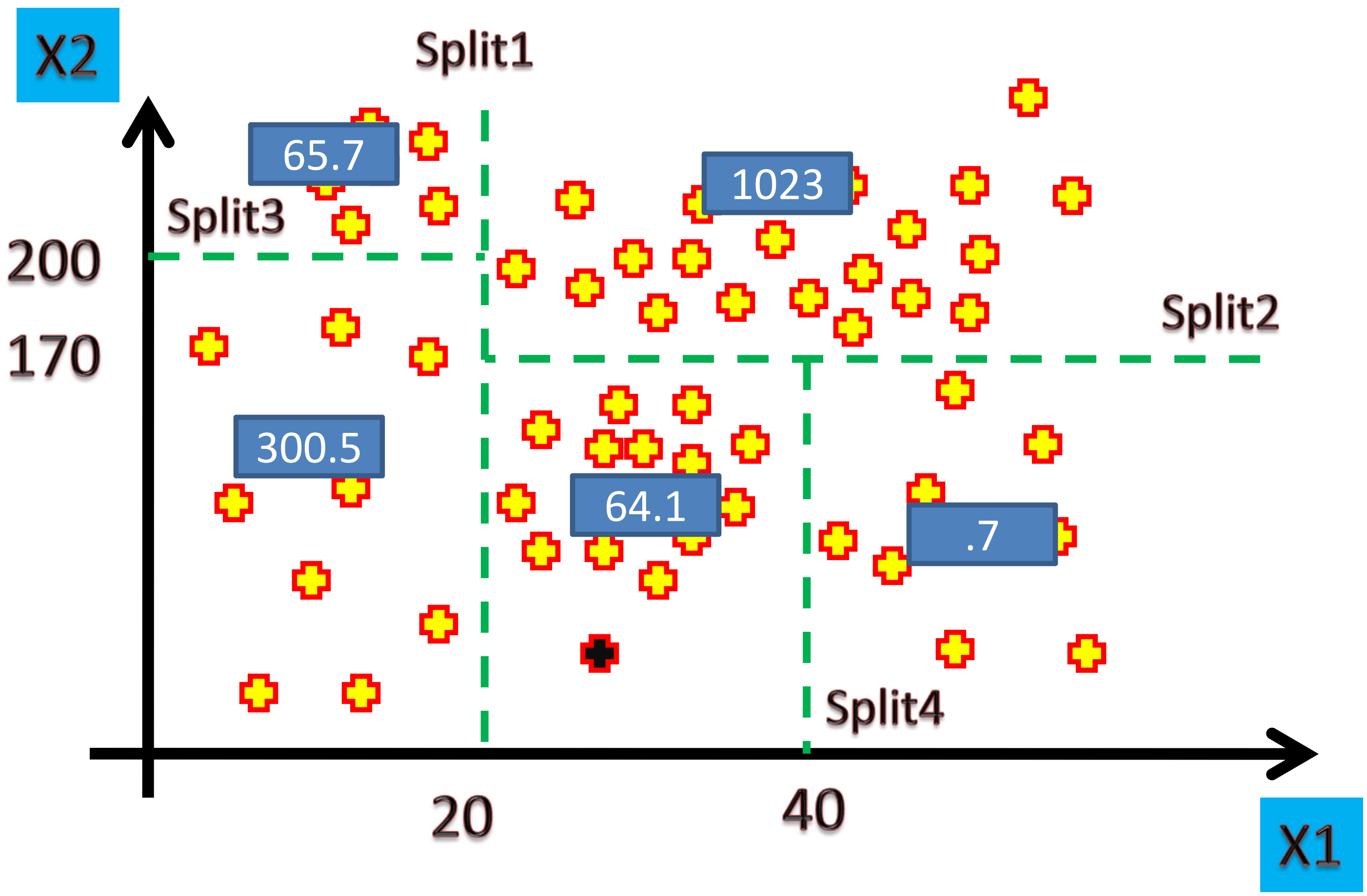
Decision Tree Regression



Decision Tree Regression

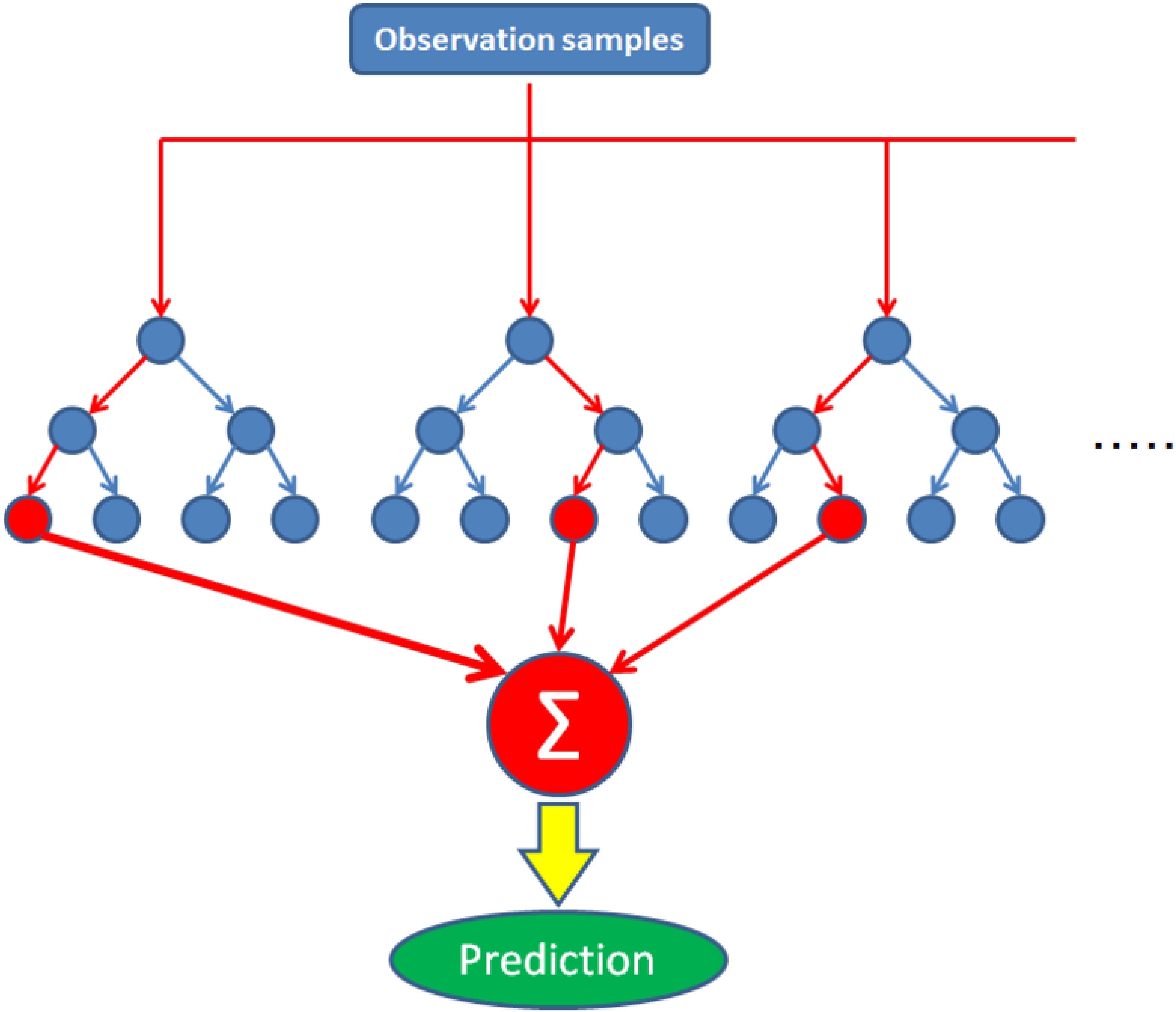


Decision Tree Regression



Random Forest Regression

Random forest builds multiple decision trees and merges them together to get a more accurate and stable prediction.

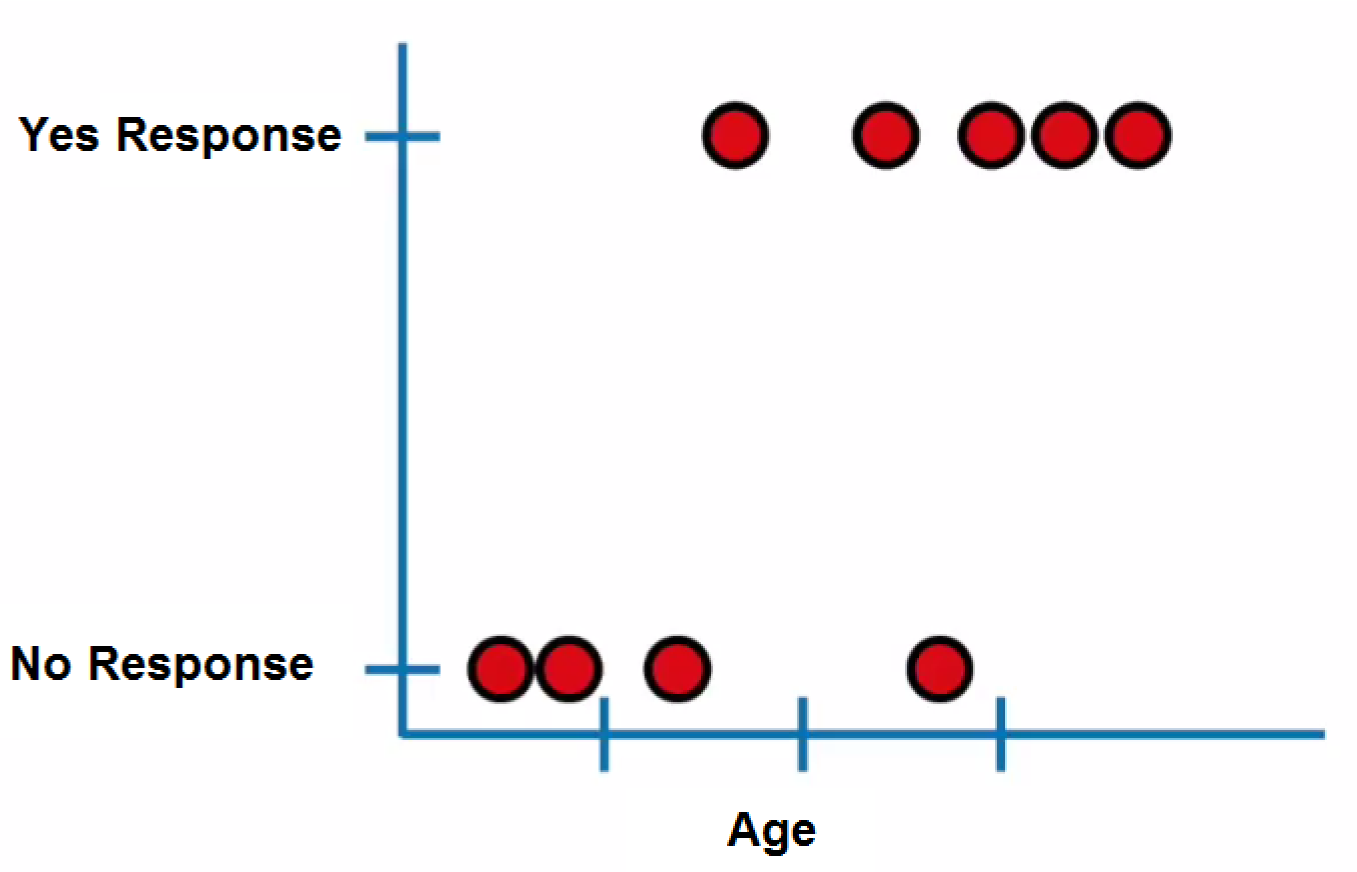

Observation samples
.....
Σ
Prediction

Classification

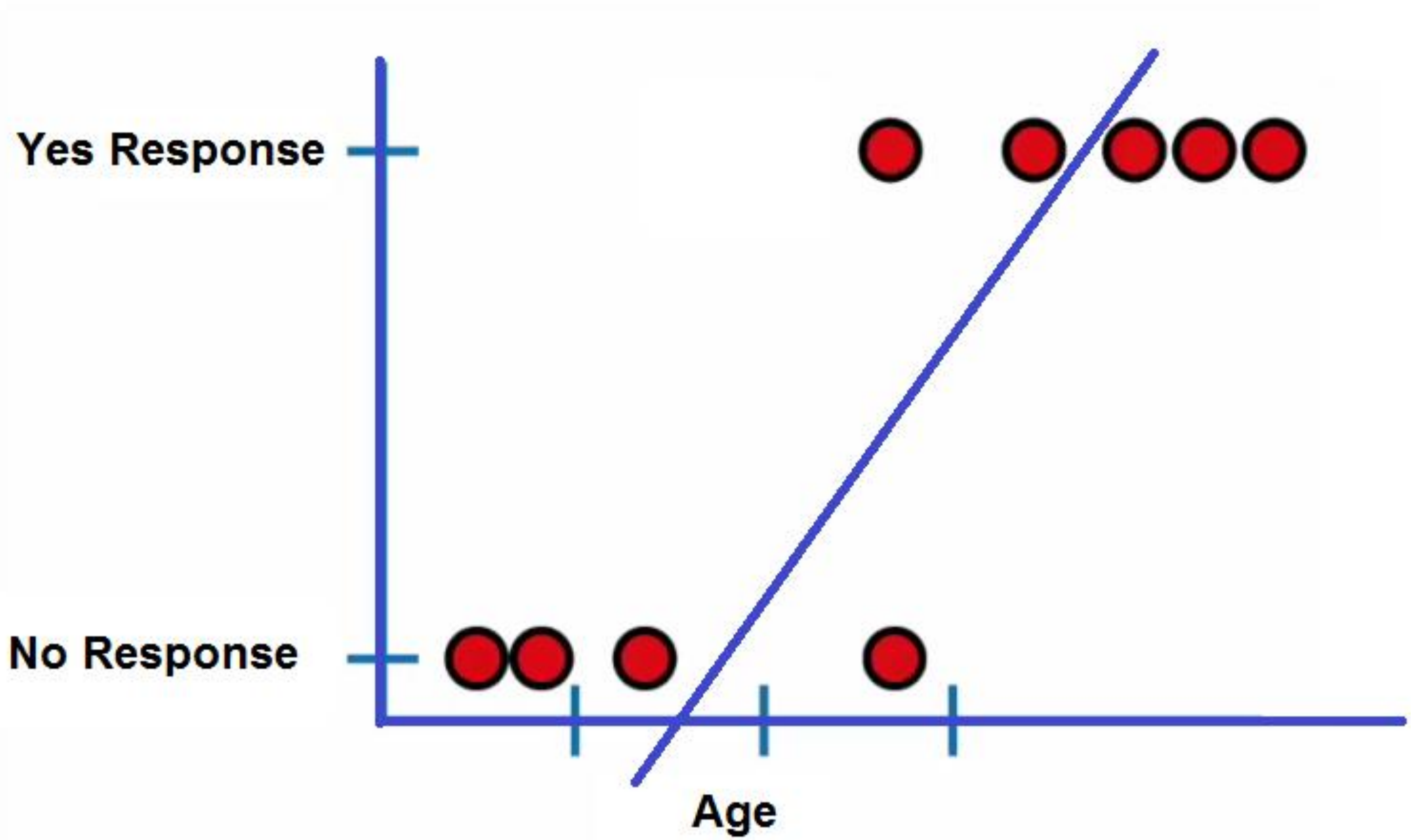
Confusion Matrix

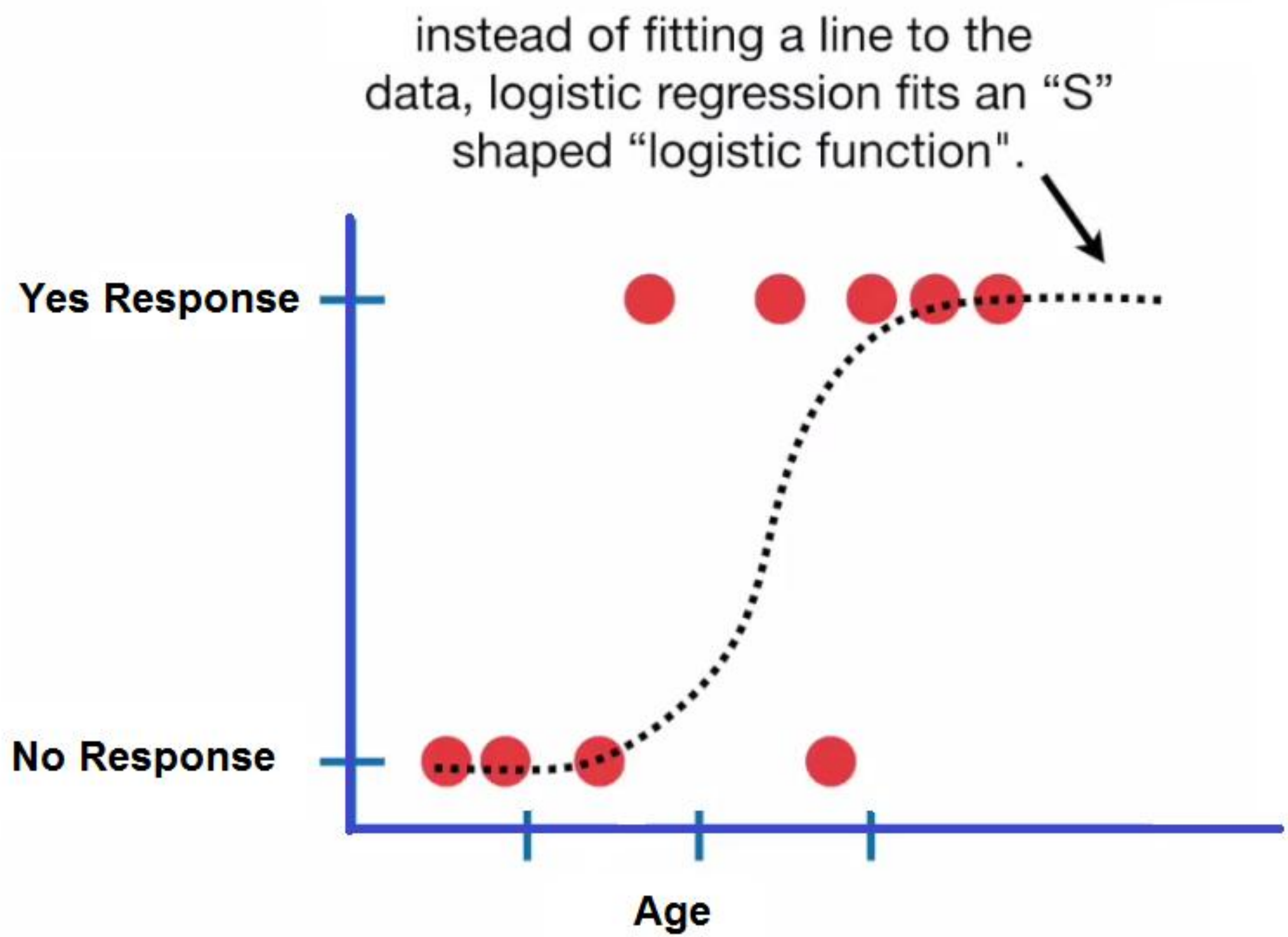
n=165	Predicted: NO	Predicted: YES	
Actual: NO	TN = 50	FP = 10	60
Actual: YES	FN = 5	TP = 100	105
	55	110	

Logistic Regression

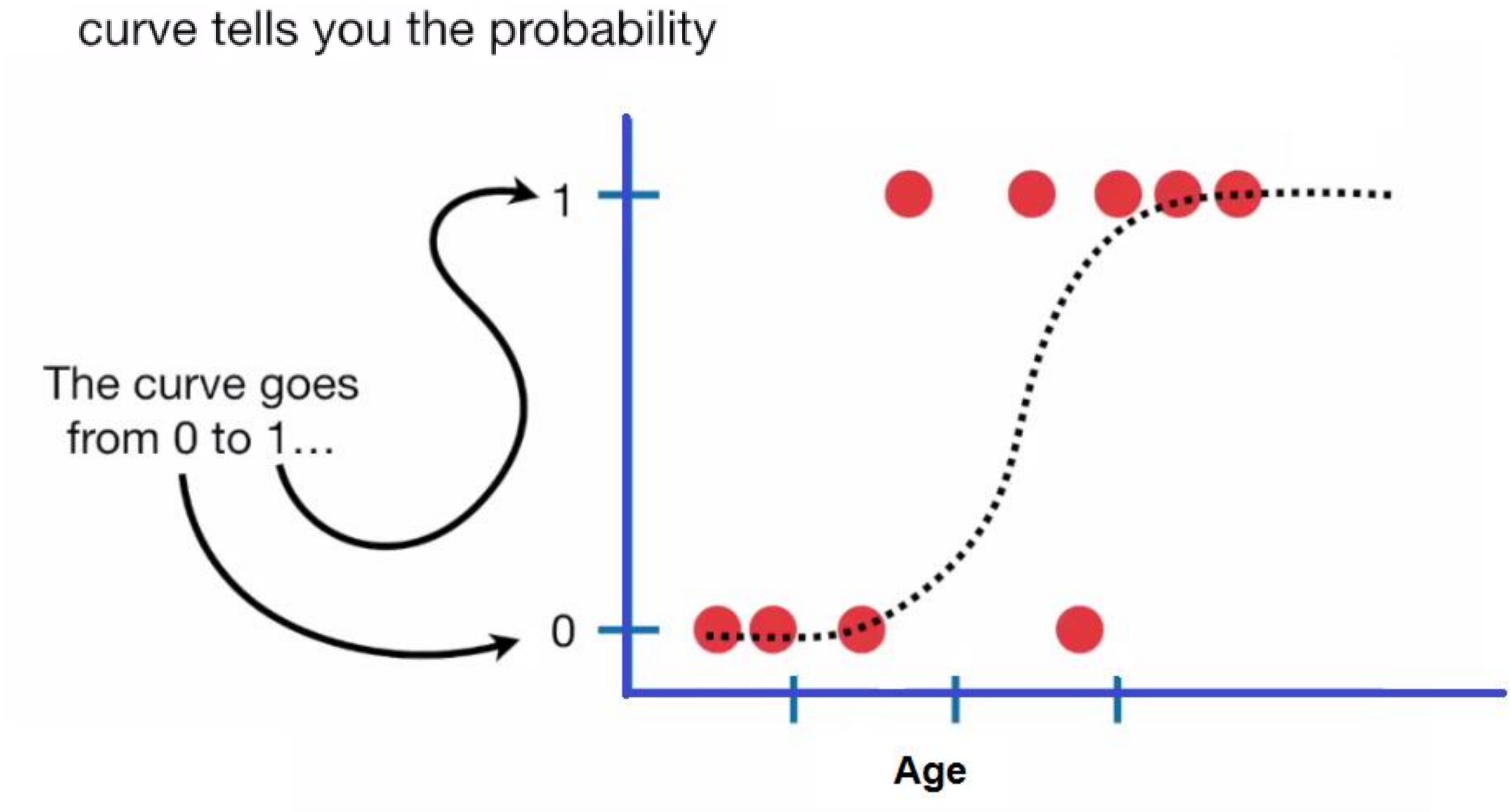


Logistic Regression




instead of fitting a line to the data, logistic regression fits an “S” shaped “logistic function".
Yes Response
No Response
Age

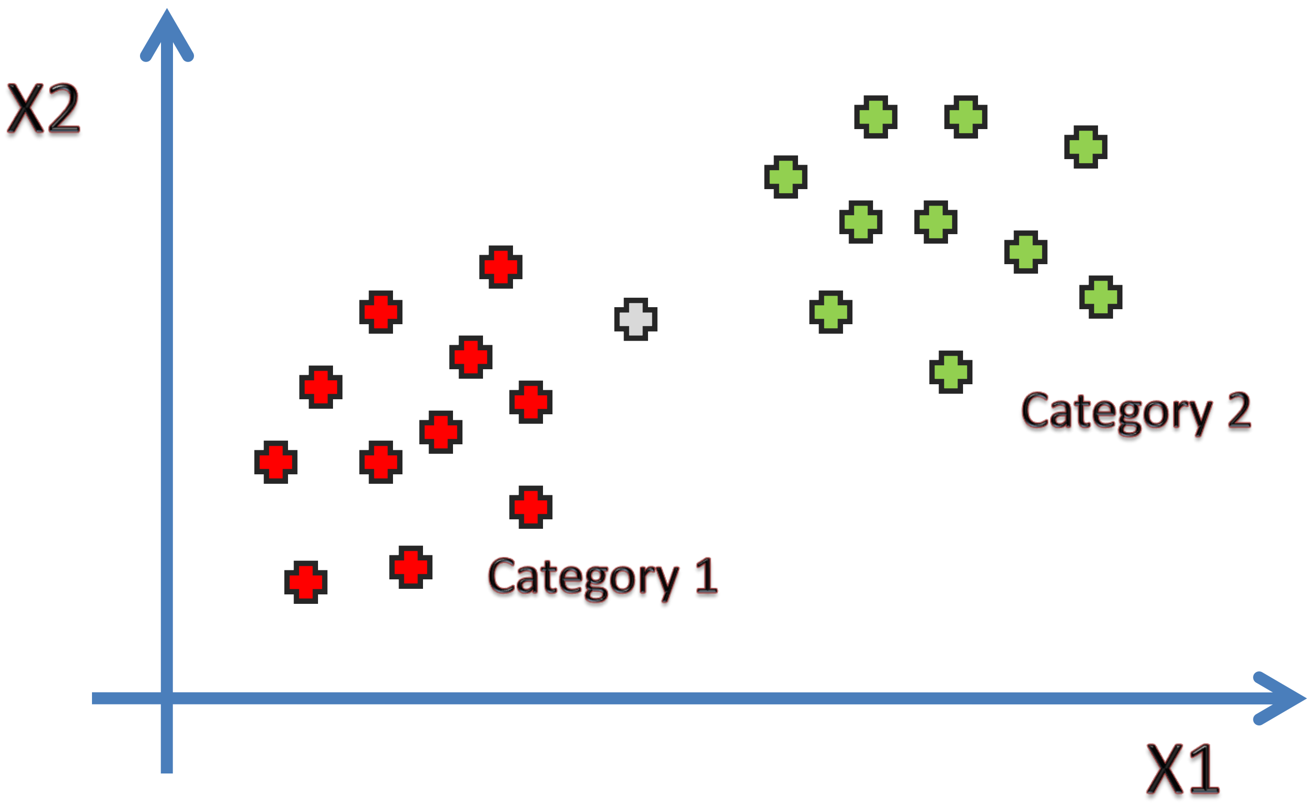
Logistic Regression



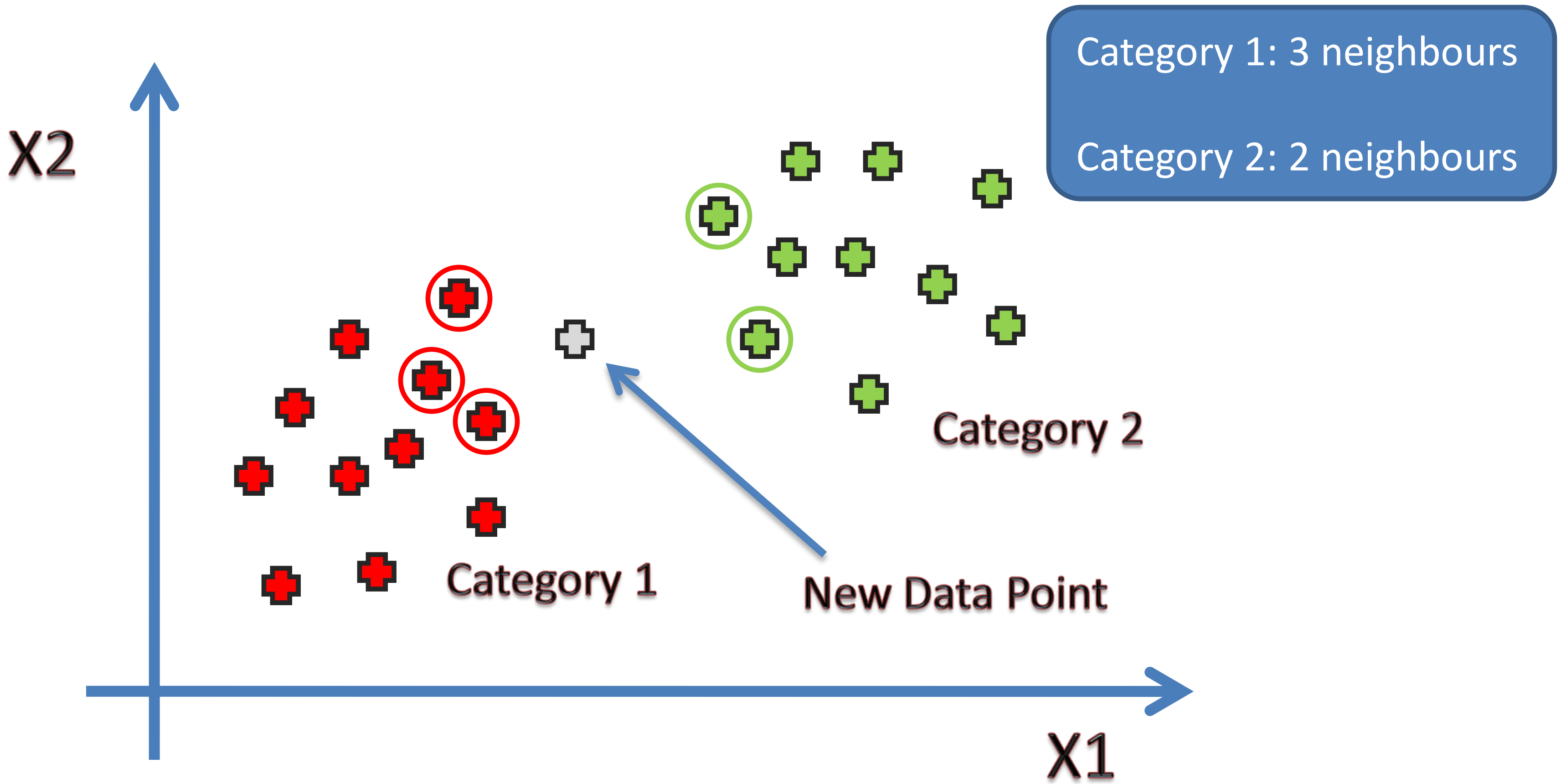
Feature Scaling

- StandardScaler will **transform** the data such that its distribution will have a **mean value 0** and **standard deviation of 1**
- This is useful while **comparing data** that corresponds to **different units**. In that case, we want to **remove the units**.
- This is done in a consistent way for all the data, we **transform the data** in a way that the **variance is unitary** and the **mean of the series is 0**.

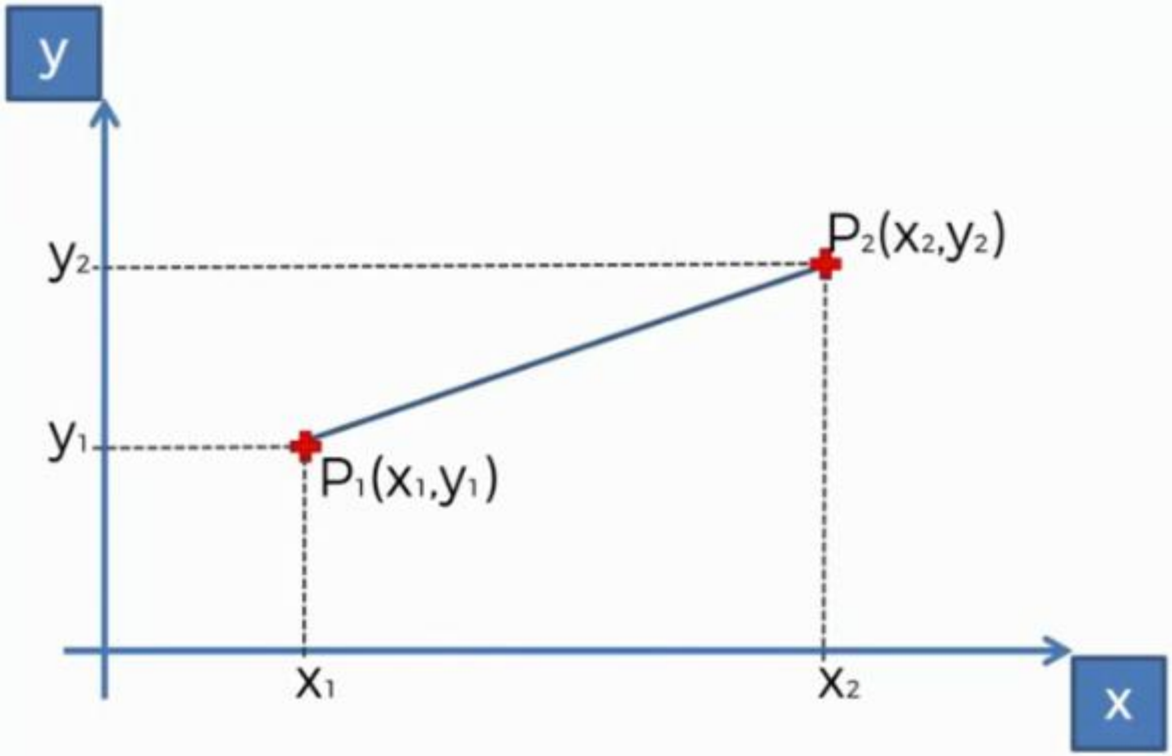
KNN



KNN

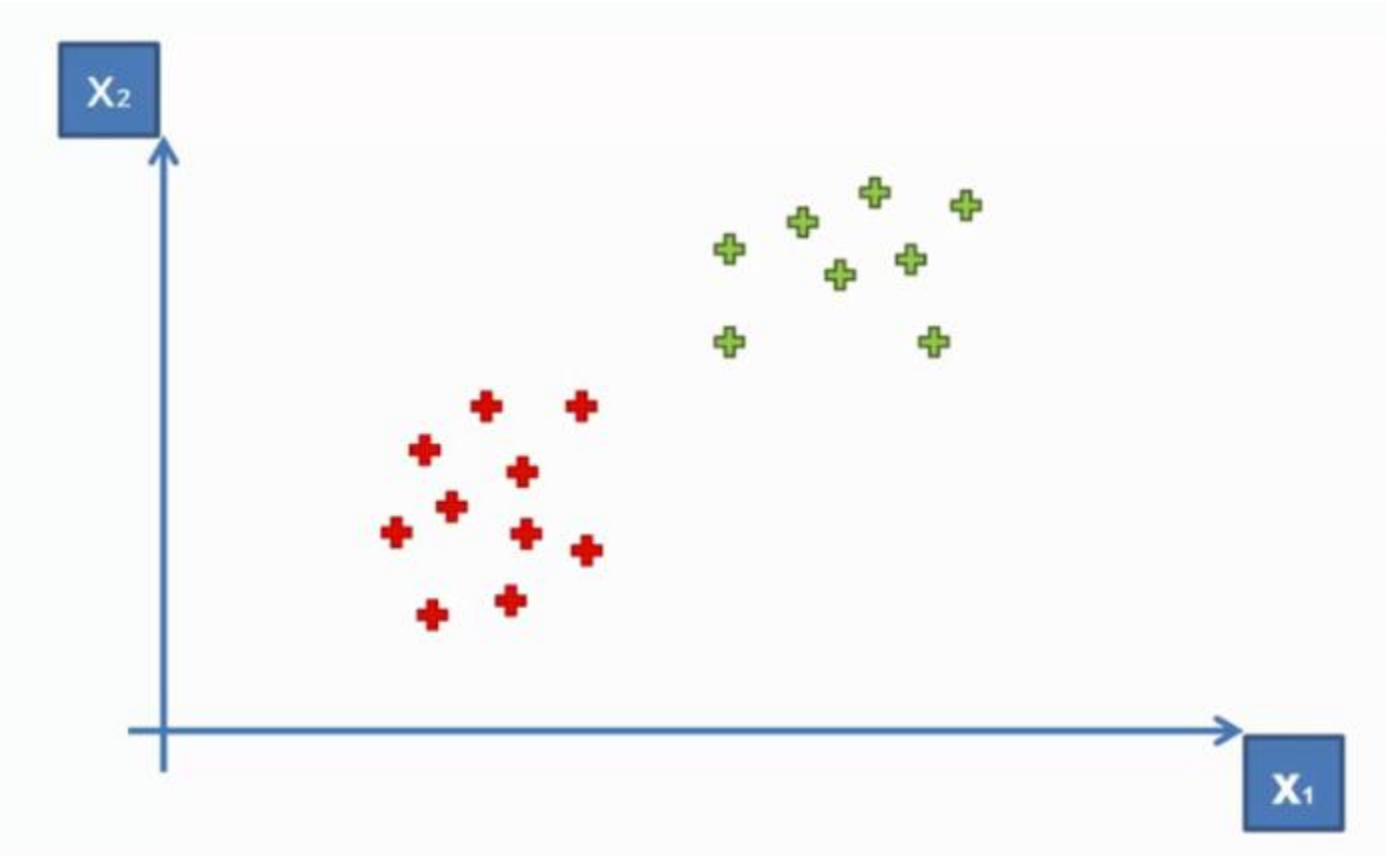

Category 1: 3 neighbours
Category 2: 2 neighbours
X2
Category 2
Category 1
New Data Point
X1

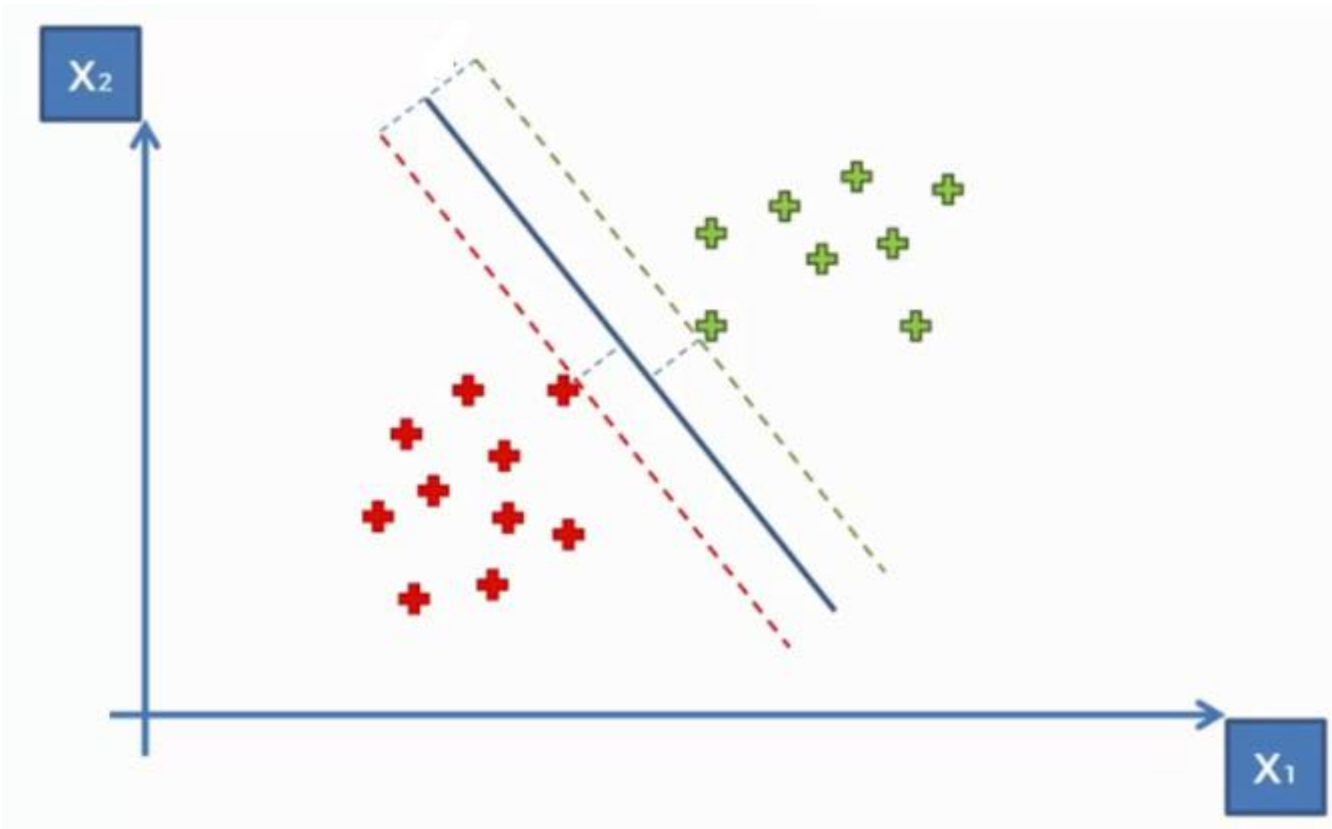
Euclidean Distance

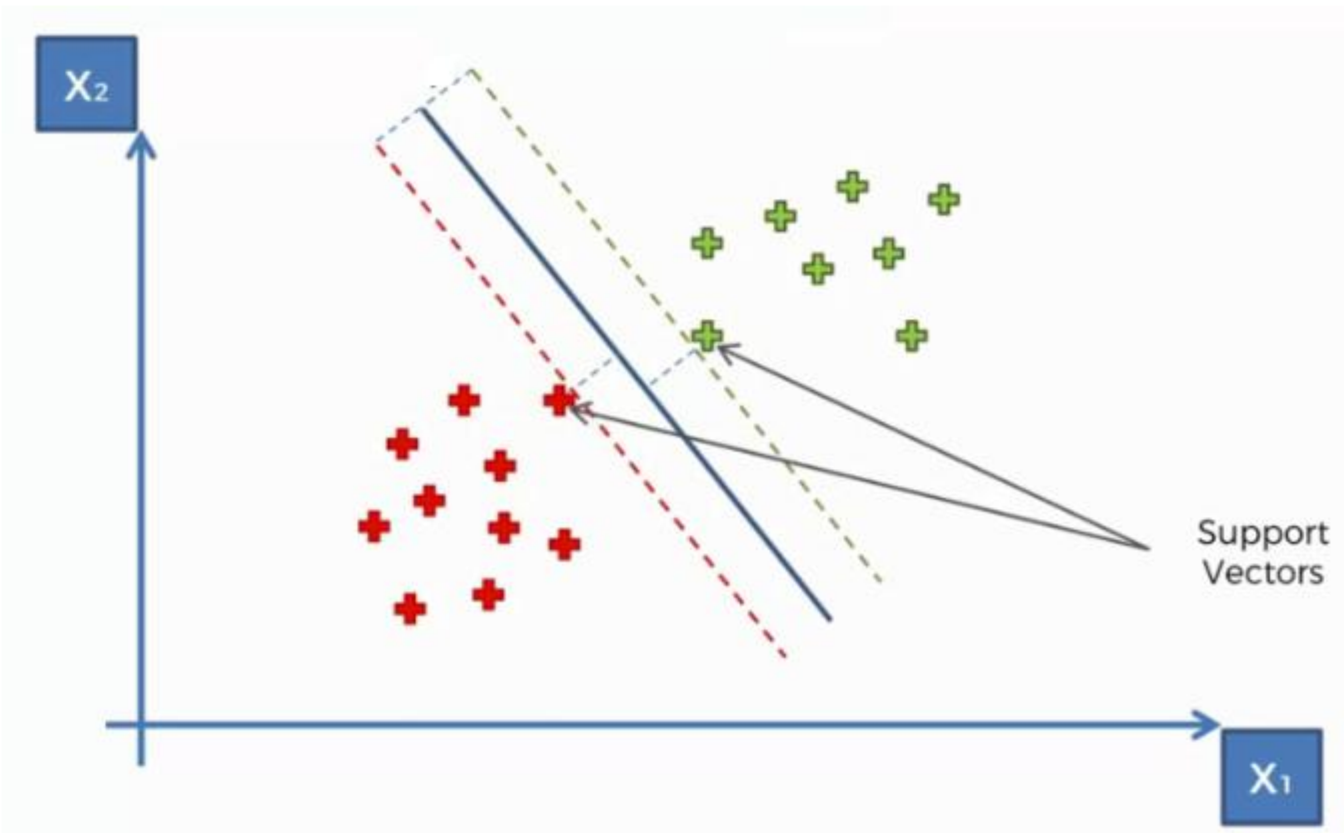


$$\text{Euclidean Distance between } P_1 \text{ and } P_2 = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$$

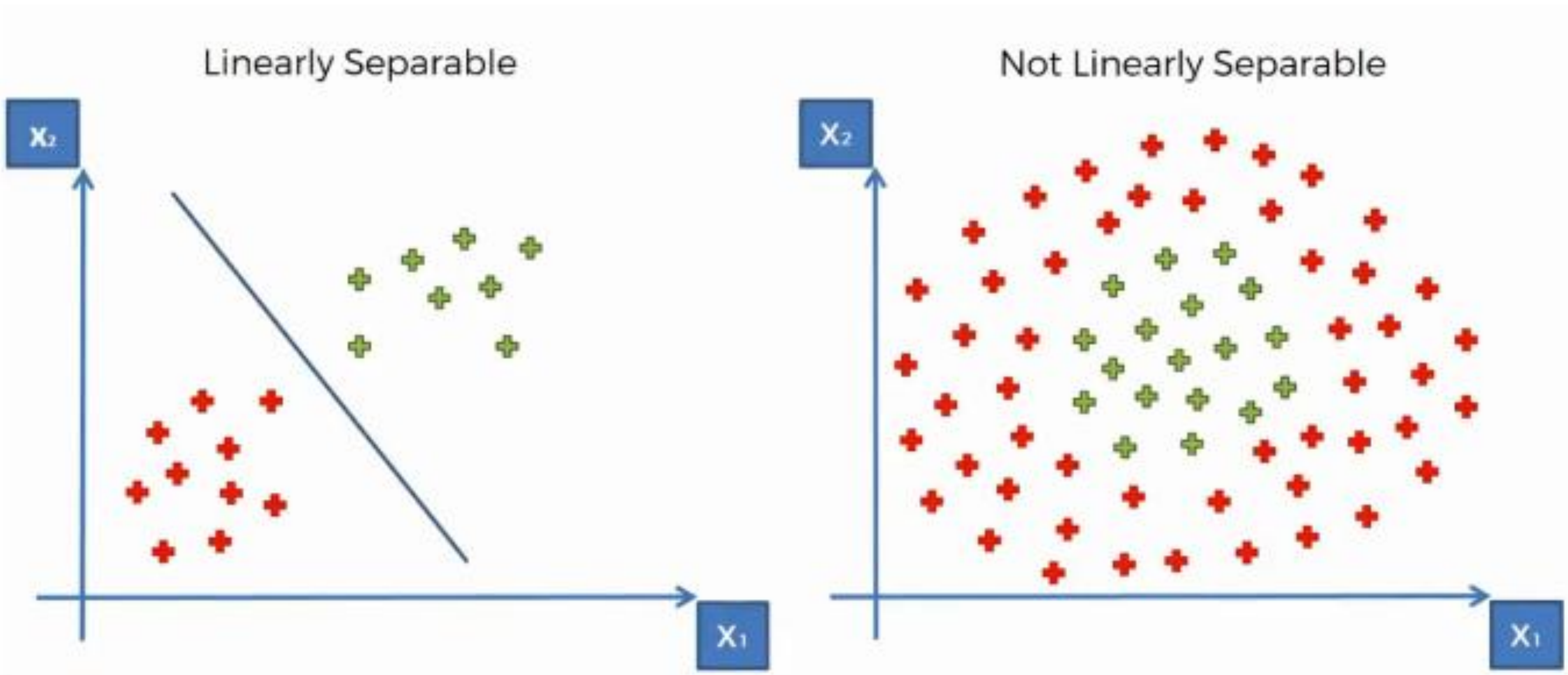
Support Vector Machine

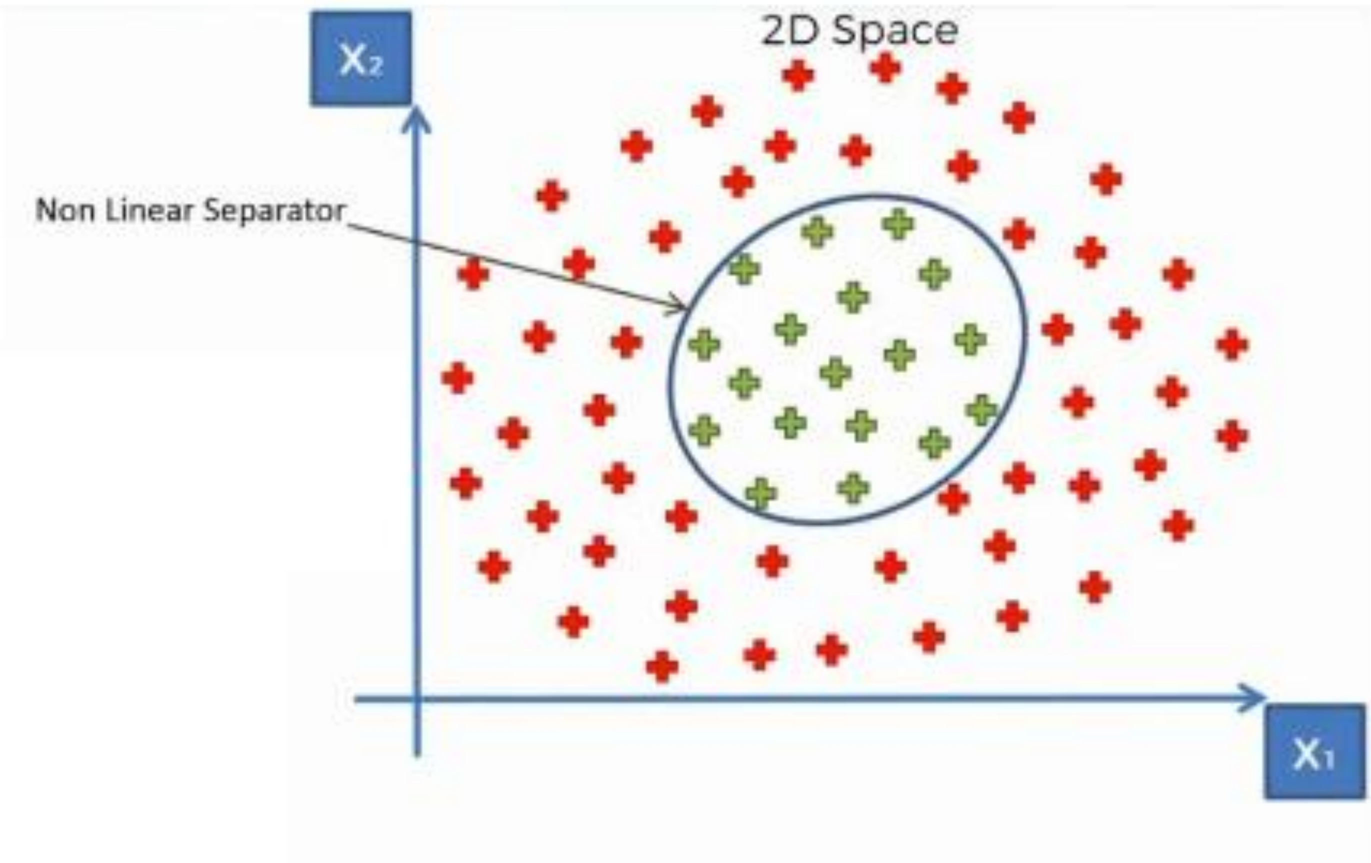



X2
X1


X2
Support
Vectors
X1

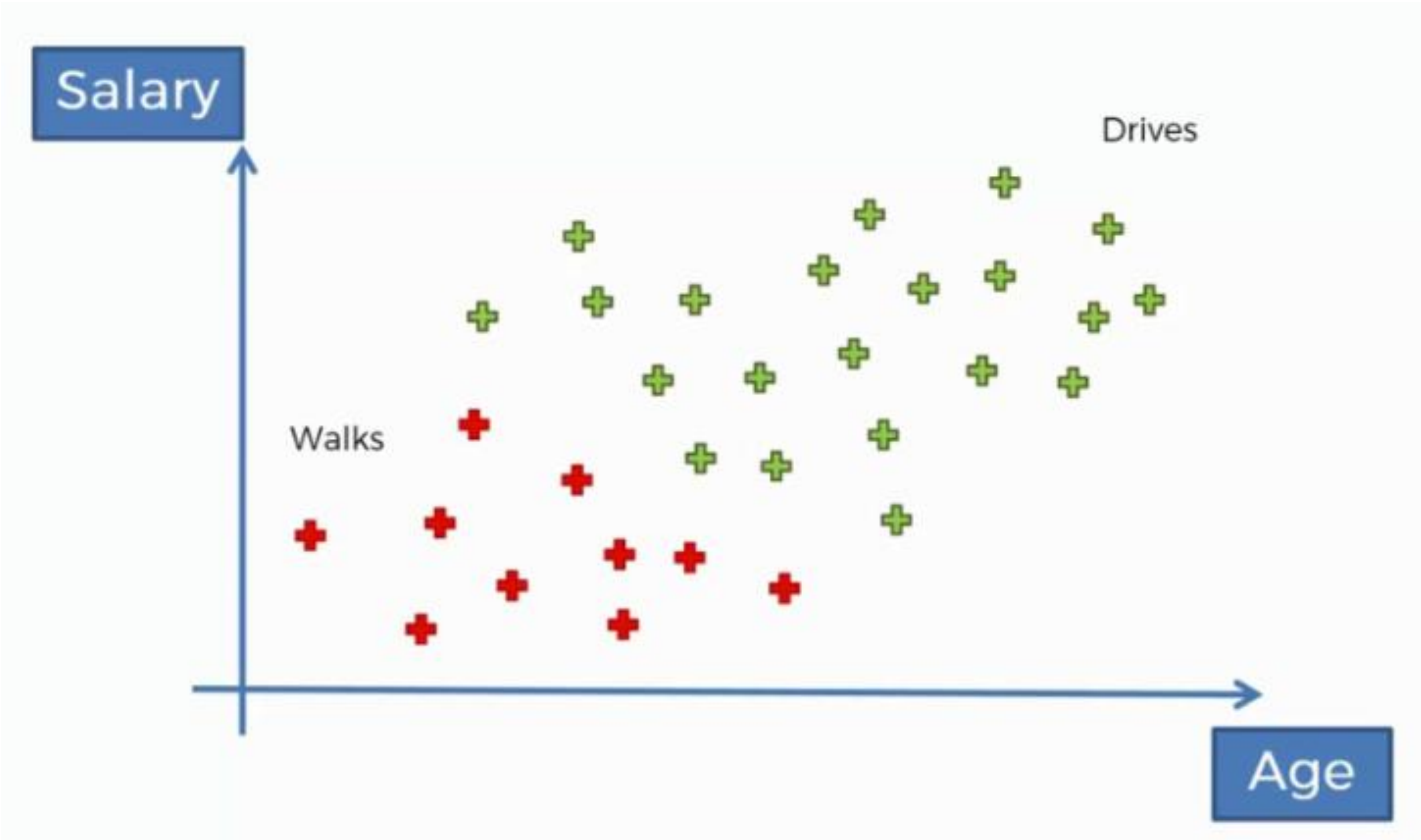
Kernel SVM




2D Space
X2
Non Linear Separator
X1

Naive Bayes

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


Salary
Drives
Walks
Age


Salary
Drives
Walks
New data point
Age

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

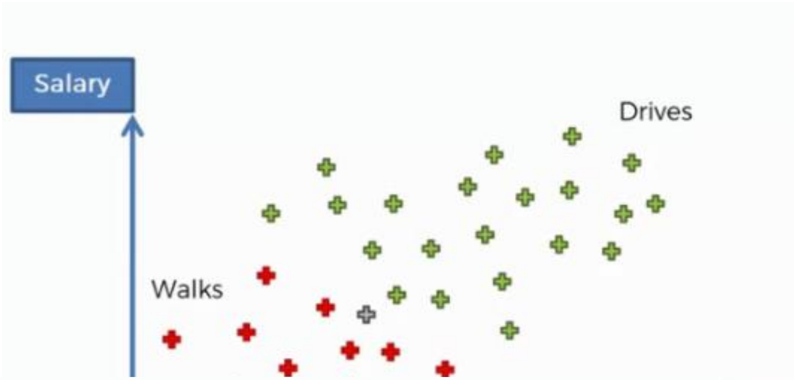
#4 Posterior Probability

#3 Likelihood

#1 Prior Probability

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

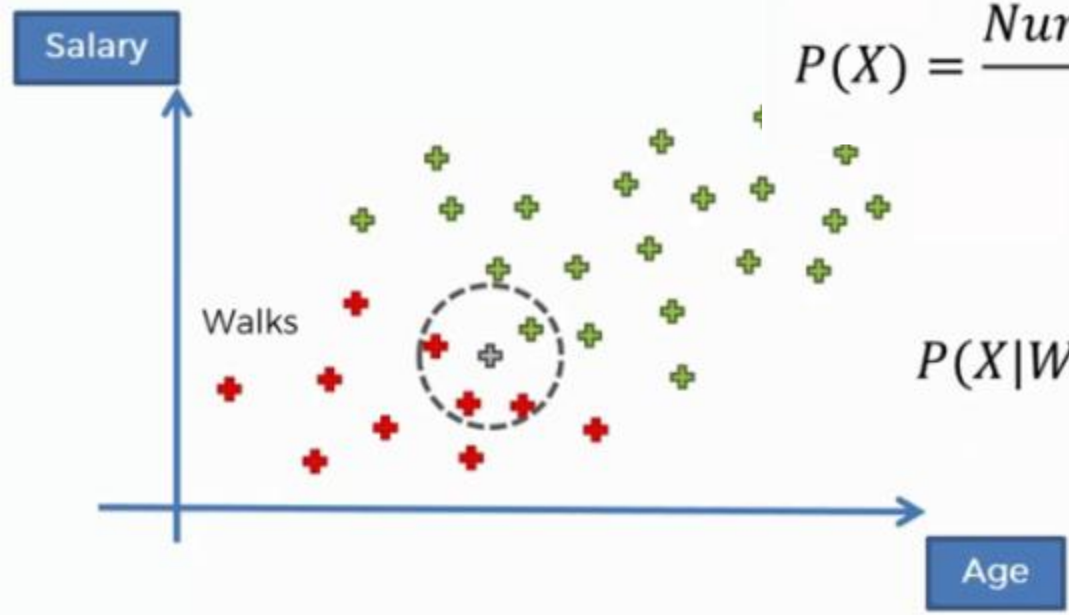
#2 Marginal Likelihood



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

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

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



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

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

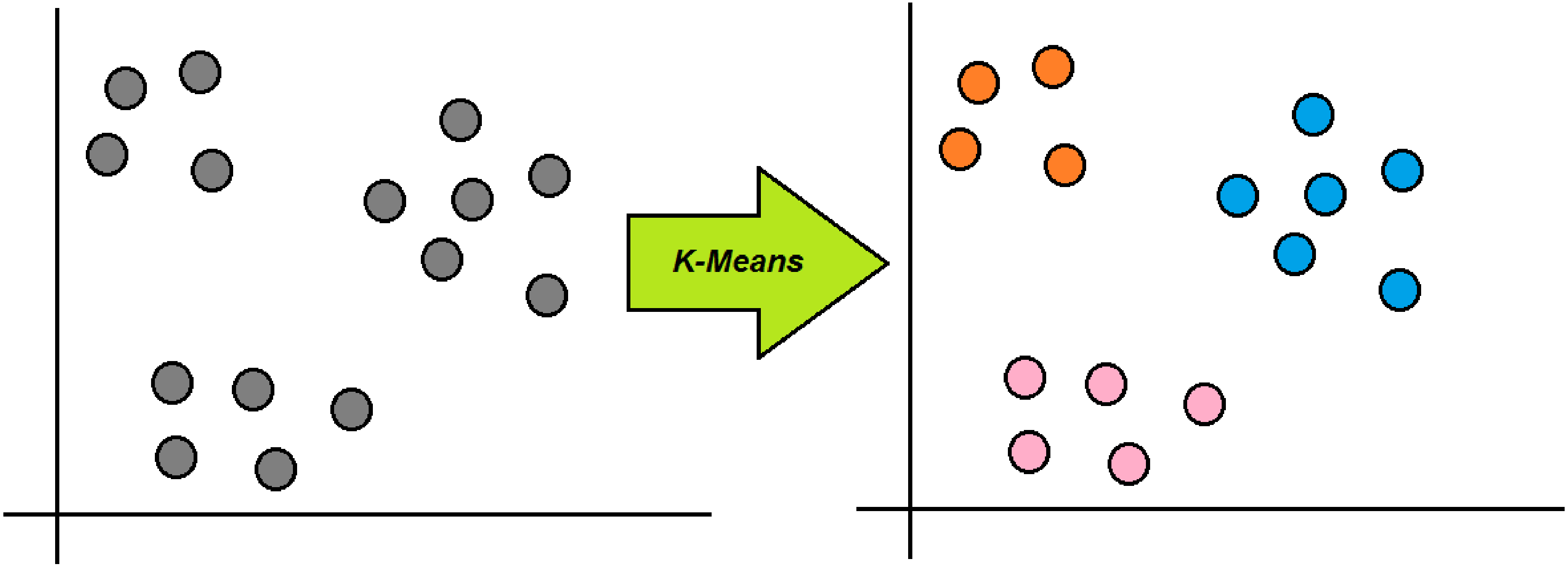
$$P(Walks|X) = \frac{\frac{3}{10} * \frac{10}{30}}{\frac{4}{30}} = 0.75$$

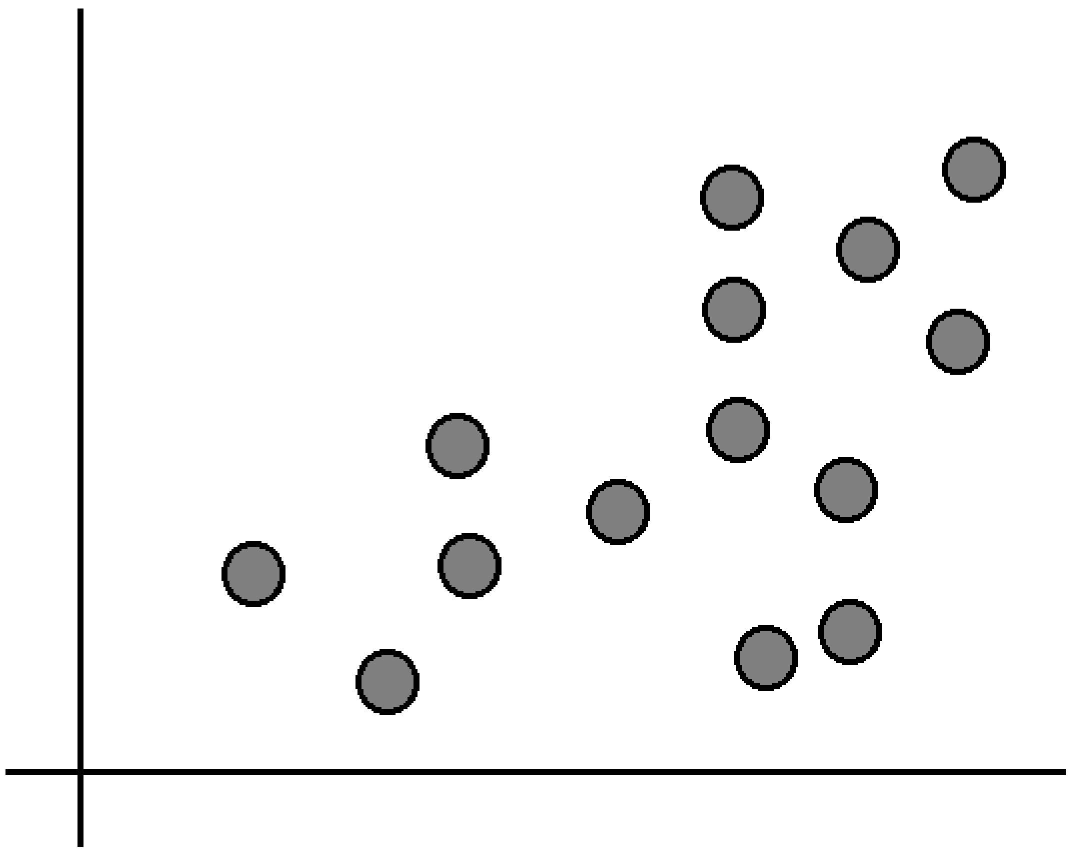
$$P(Walks|X) \ v.s. \ P(Drives|X)$$

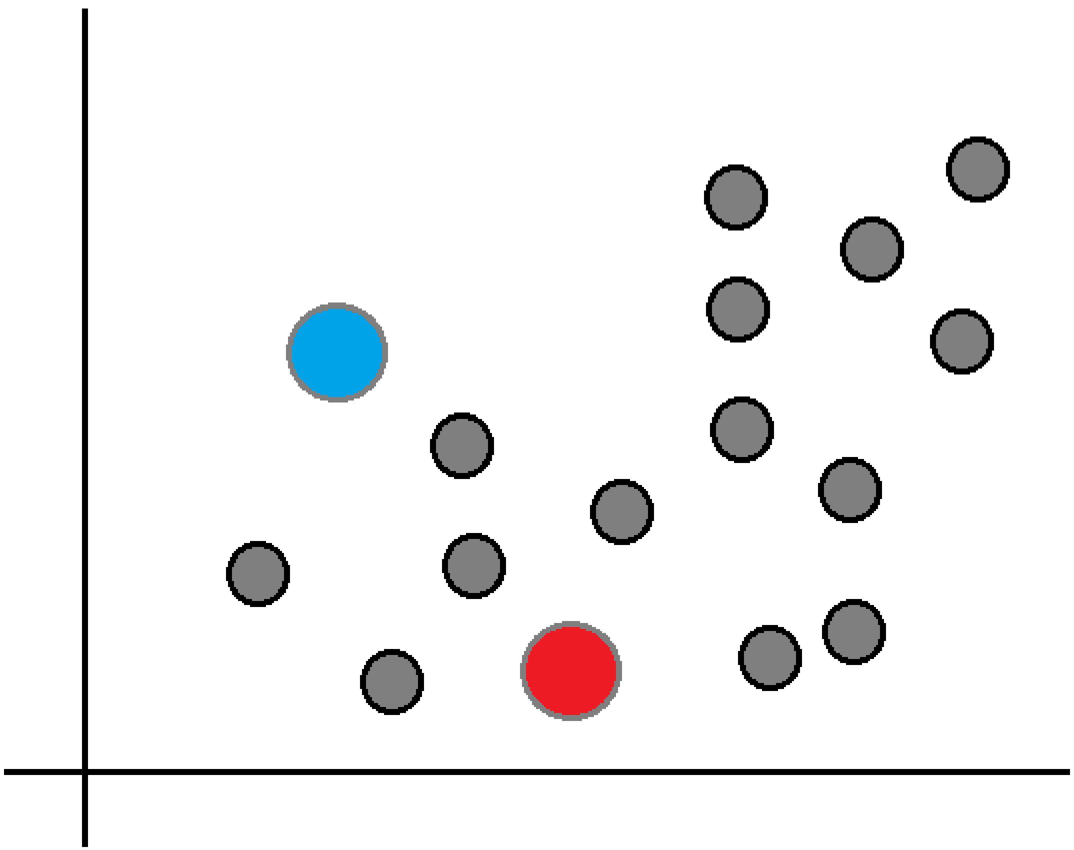
$$P(Drives|X) = \frac{\frac{1}{20} * \frac{20}{30}}{\frac{4}{30}} = 0.25$$

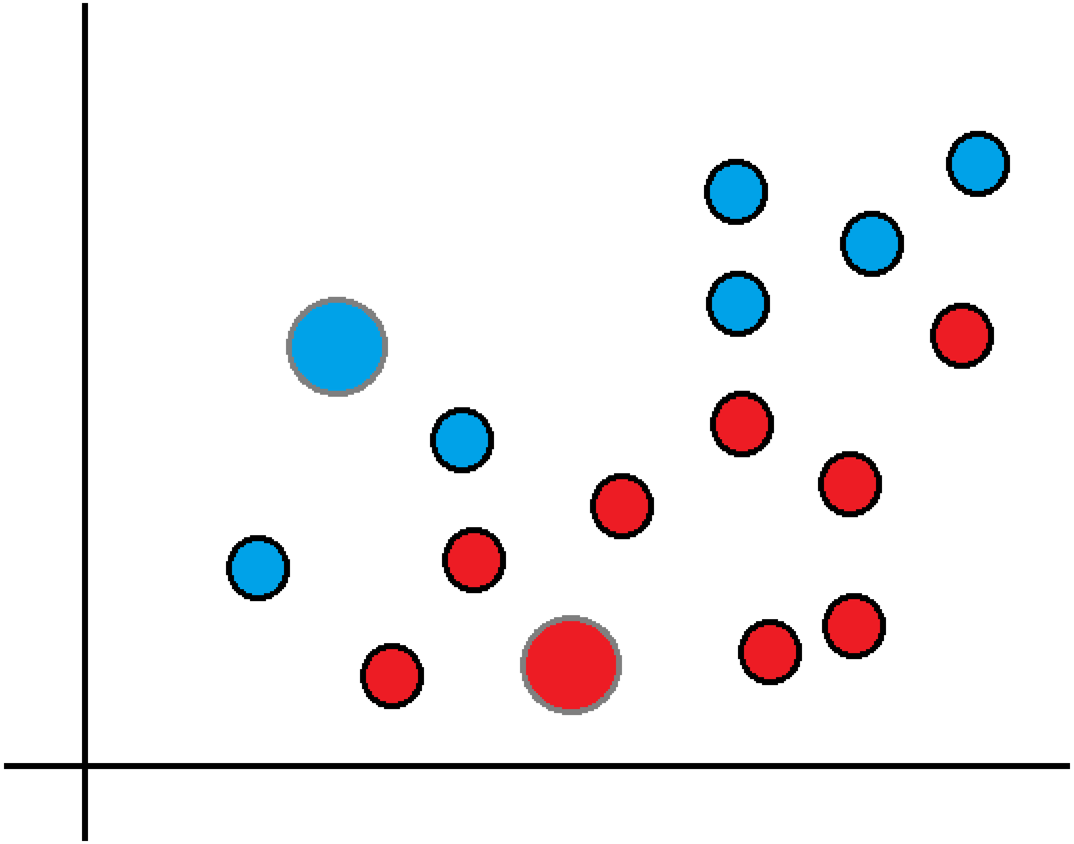
Clustering
Unsupervised

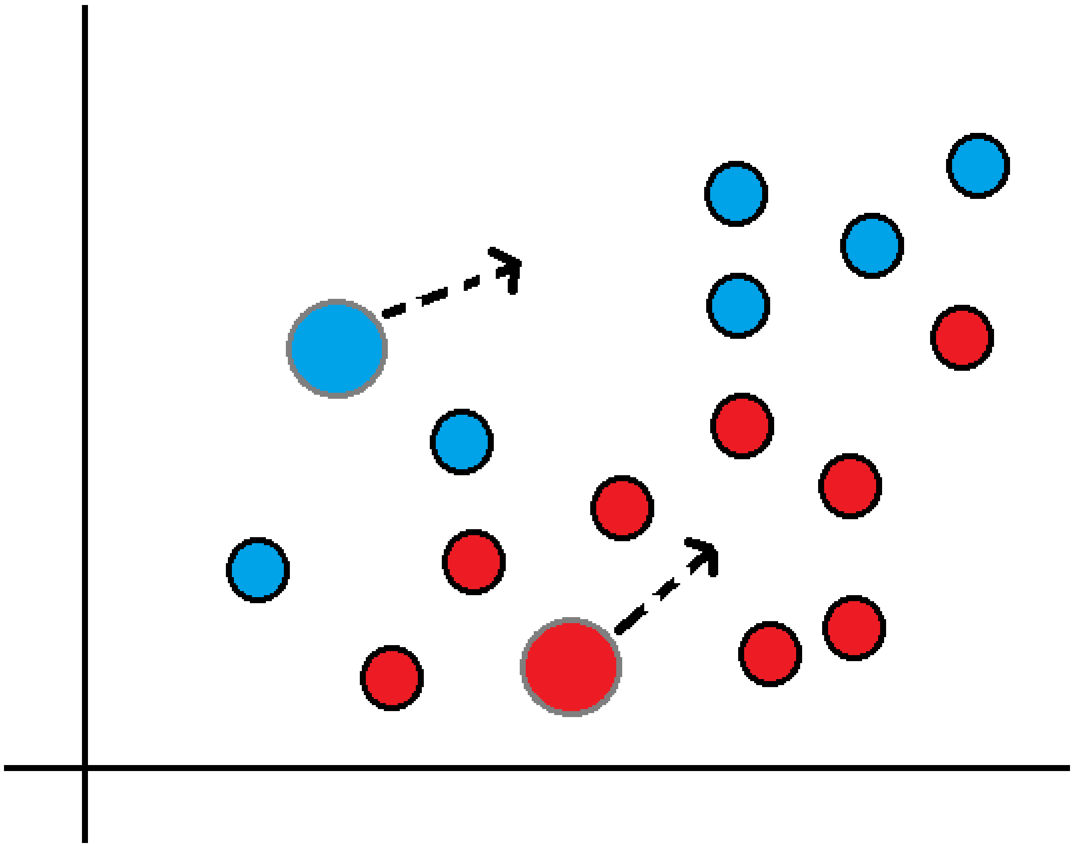
K-means

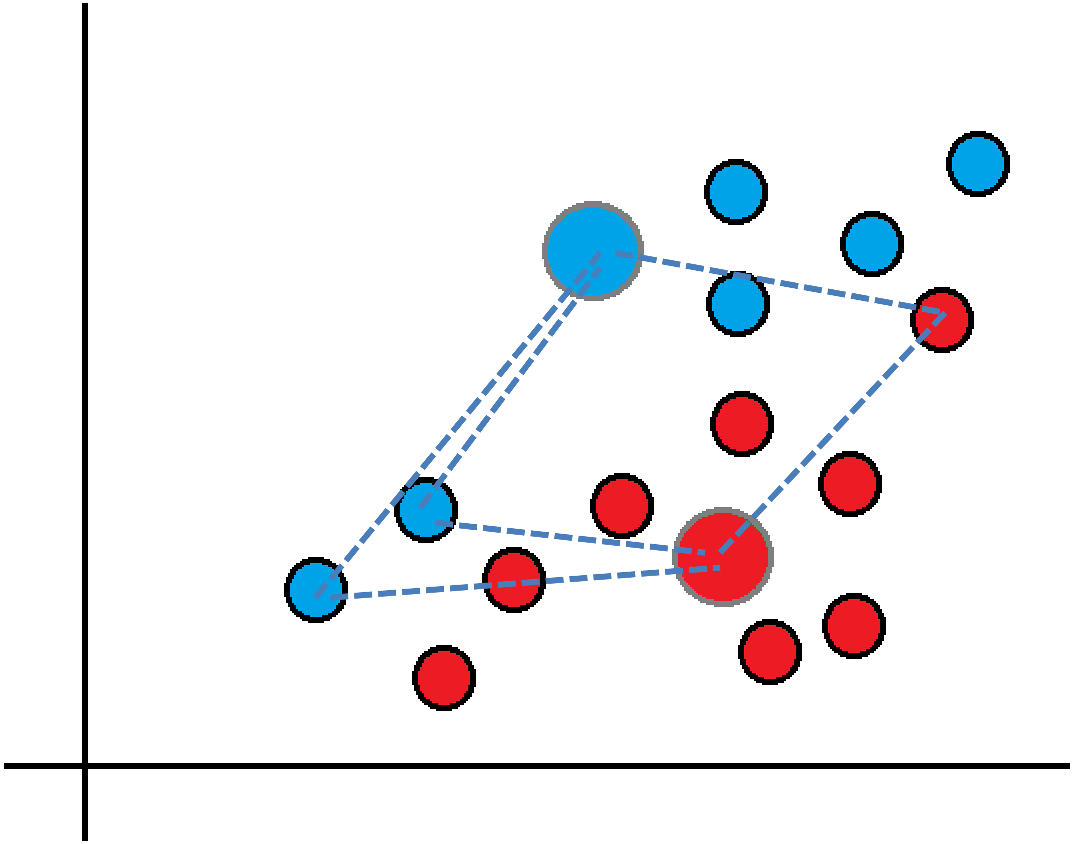


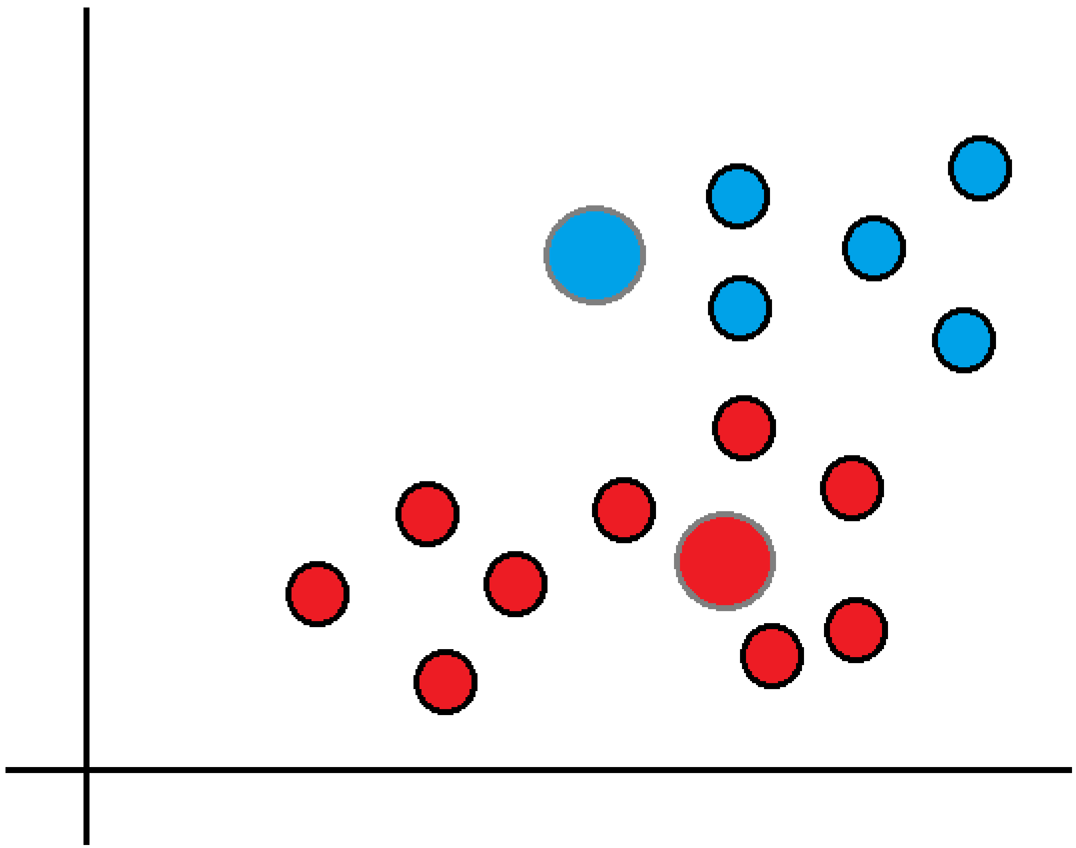


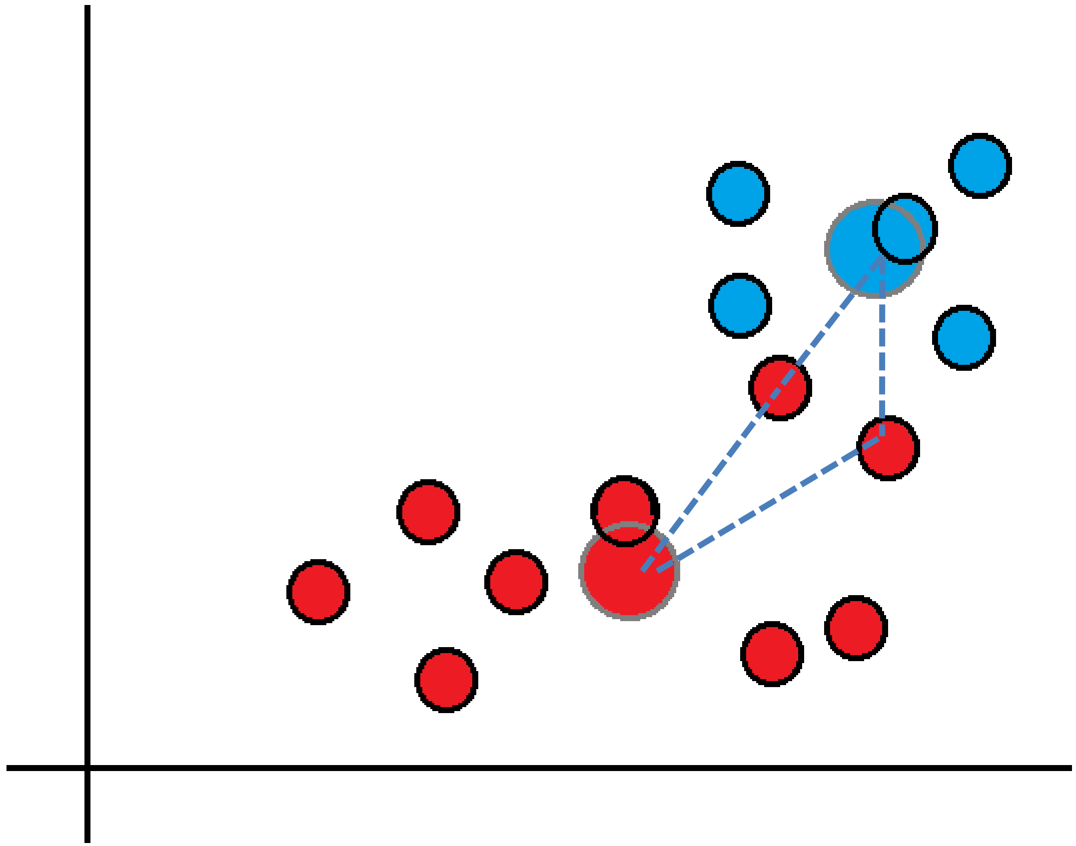


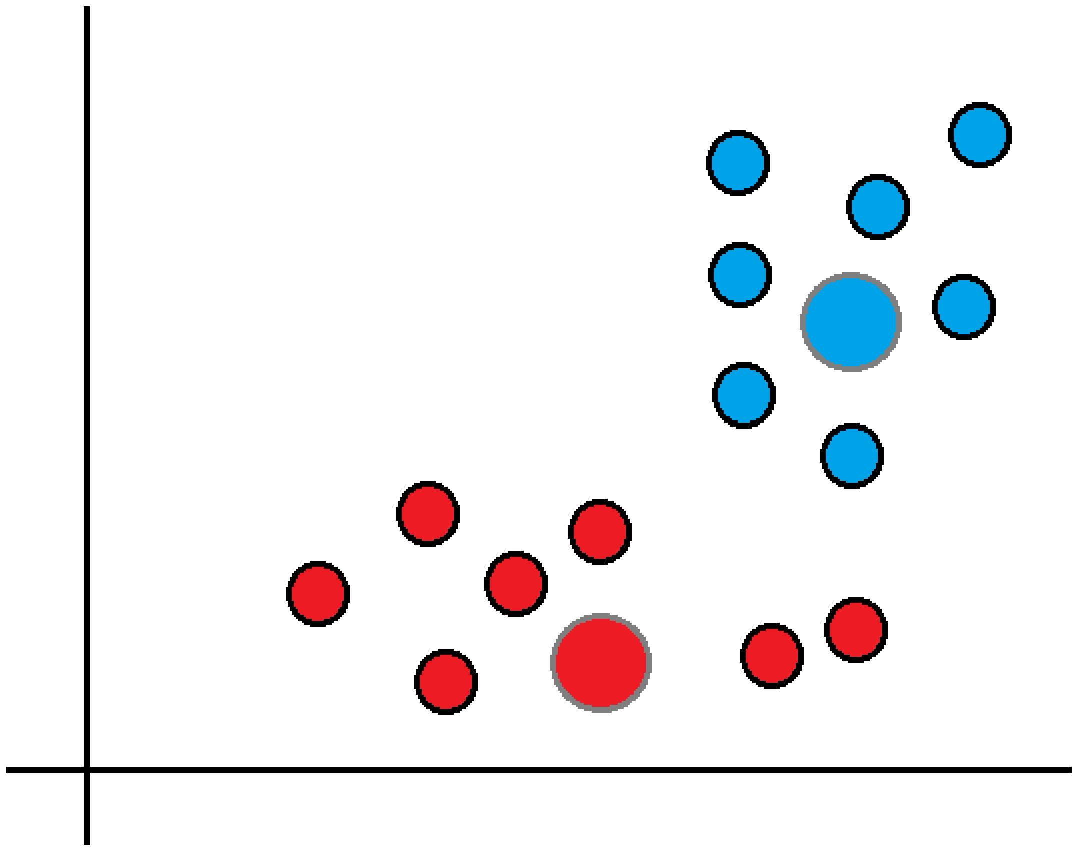


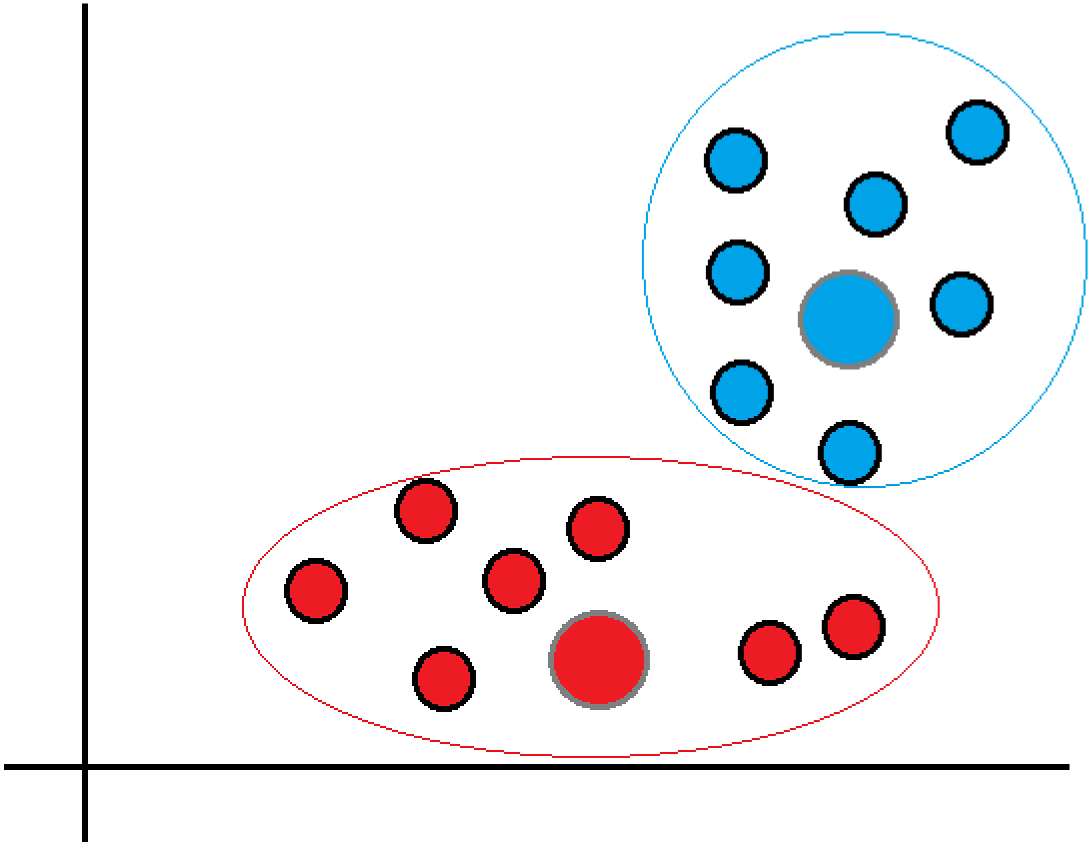












Upper Confidence Bound

- There are many algorithms to optimize the decision making behaviour of the agent, some perform better than others.
- A very popular method is the UCB exploration strategy
- This algorithm chooses the arm based on the average reward mean plus an exploration bonus.
- The exploration bonus is dependent on the number of times the action has been tried out before and the total number of action selections.

- We have **d** Ads that we display to users each time they connect to web page.
- Each time a user connects to this web page, that makes a **round**
- At **each round n**, we choose one Ad to display to the user.
- At each **round n**, **Ad i** gets reward
- if the user clicked on Ad $r_i(n) \in \{0, 1\}: \ r_i(n) = 1$
- if the user didnt then 0
- The goal is to **maximize the total reward** we get over many rounds

Step 1. At each round n, we consider two numbers for each ad i:

- $N_i(n)$ - the number of times the ad i was selected up to round n,
- $R_i(n)$ - the sum of rewards of the ad i up to round n.

Step 2. From these two numbers we compute:

- the average reward of ad i up to round n

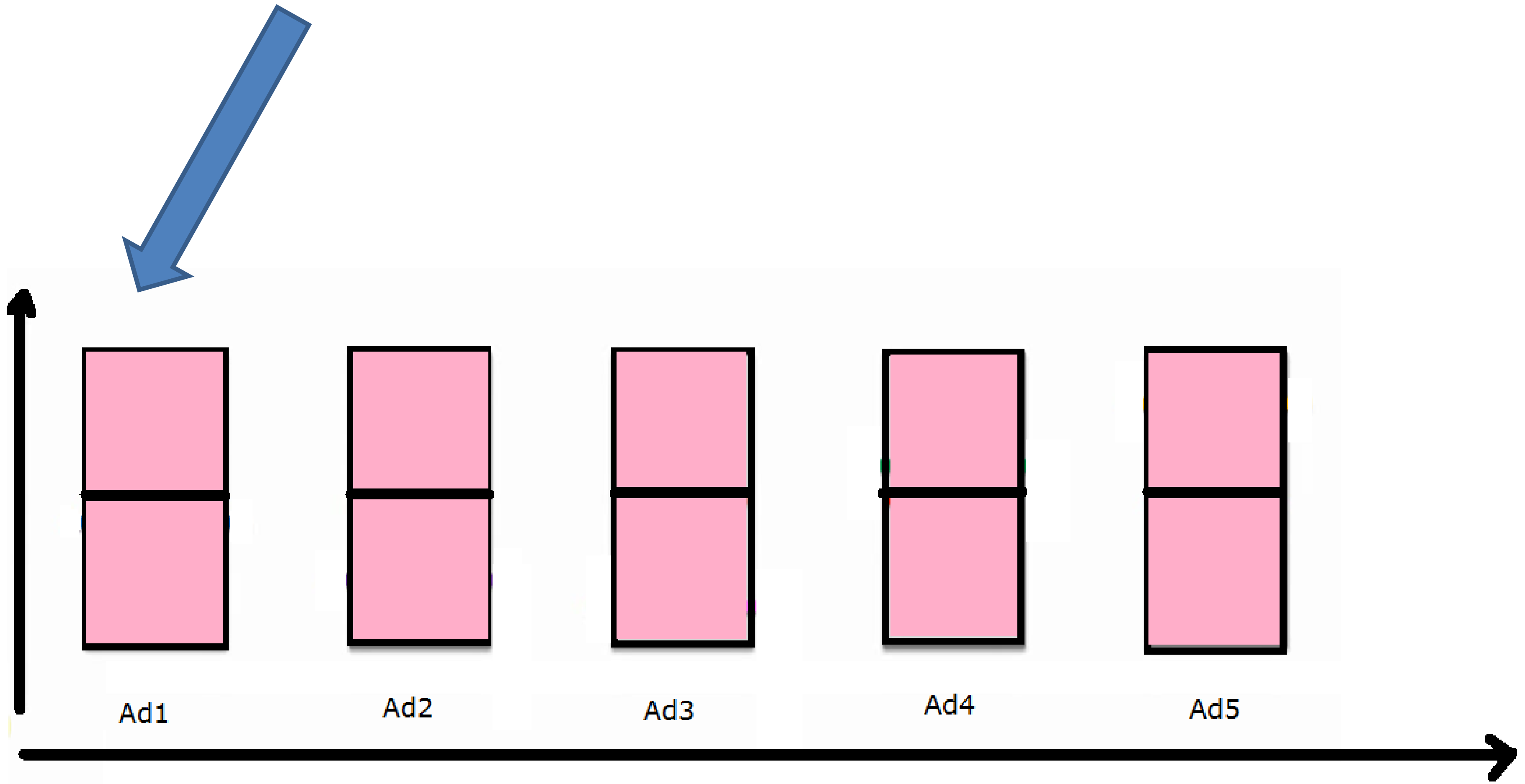
$$\bar{r}_i(n) = \frac{R_i(n)}{N_i(n)}$$

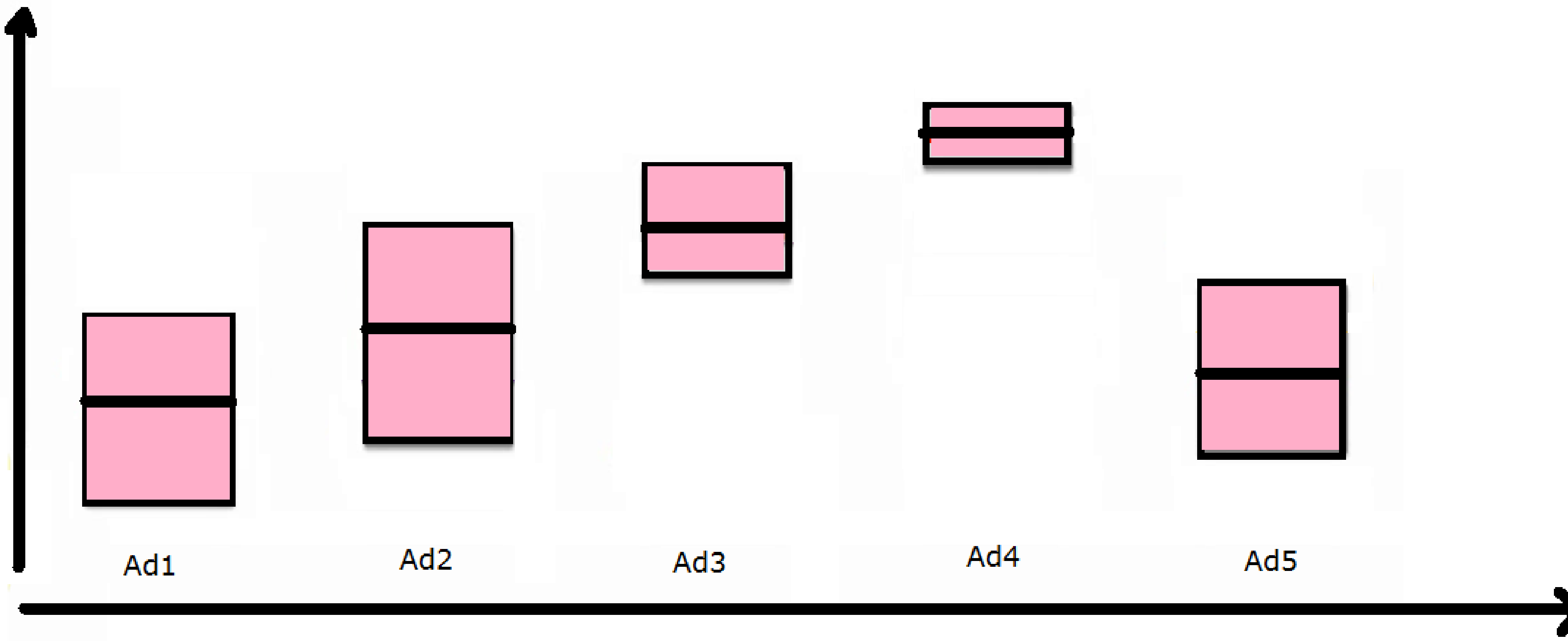
- UCB $\bar{r}_i(n) + \Delta_i(n)$

$$\Delta_i(n) = \sqrt{\frac{3}{2}\frac{\log(n)}{N_i(n)}}$$

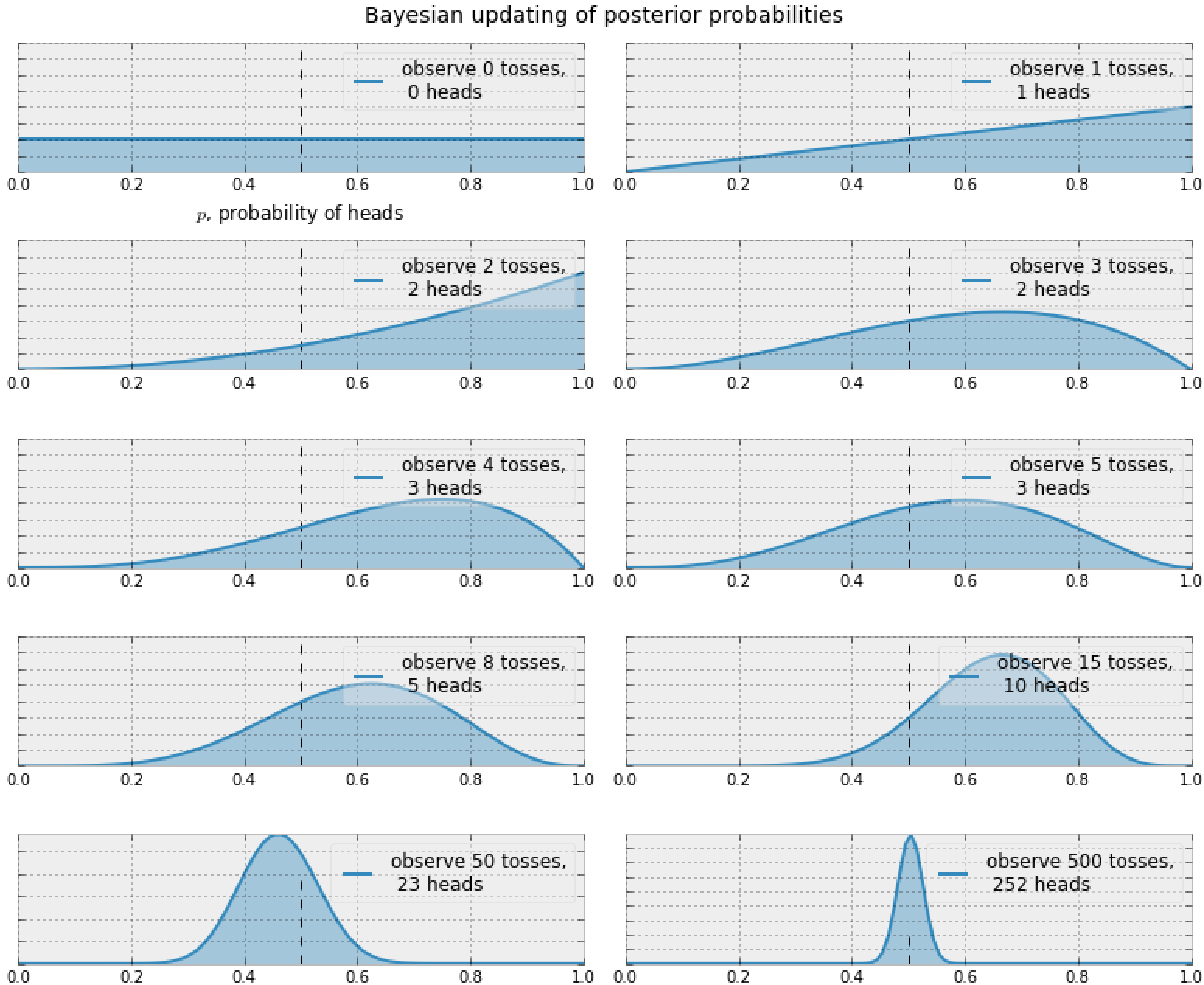
Step 3. We select the ad i that has the maximum UCB $\bar{r}_i(n) + \Delta_i(n)$.

Confidence Band Generated for all Ads




Ad1
Ad2
Ad3
Ad4
Ad5

Thompson Sampling



Thompson Sampling

Step 1. At each round n, we consider two numbers for each ad i:

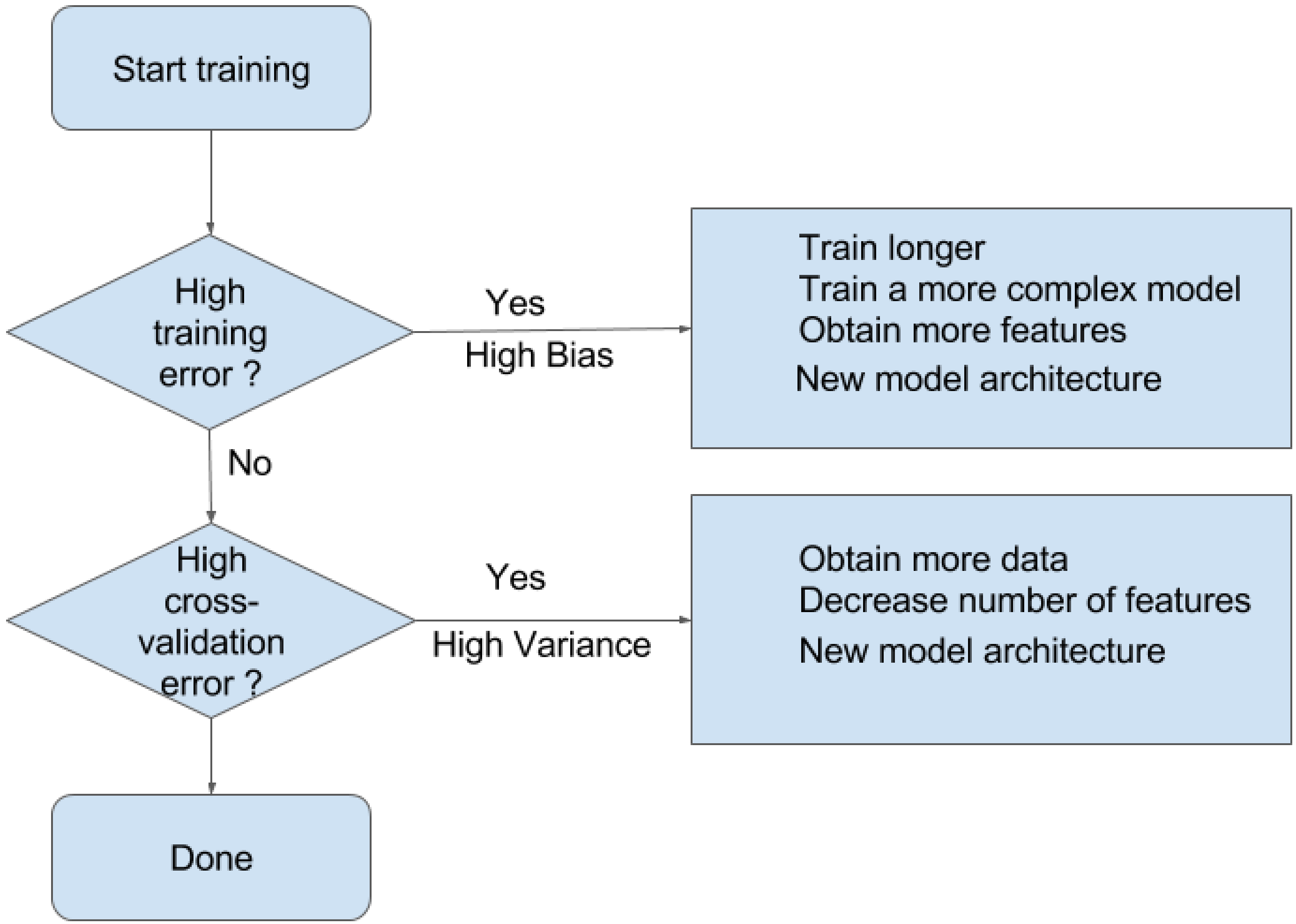
- $N_i^1(n)$ - the number of times the ad i got reward 1 up to round n,
- $N_i^0(n)$ - the number of times the ad i got reward 0 up to round n.

Step 2. For each ad i, we take a random draw from the distribution below:

$$\theta_i(n) = \beta(N_i^1(n) + 1, N_i^0(n) + 1)$$

Step 3. We select the ad that has the highest $\theta_i(n)$.

OverFitting & UnderFitting



NLP

- TF-IDF
- TF => Trem Frequency
- IDF => Inverse document frequency

TF	=	Term Frequency
IDF	=	Inverse Document Frequency
TF-TDF	=	TF * IDF

TF

$$\frac{(Number\ of\ occurrences\ of\ a\ word\ in\ a\ document)}{(Number\ of\ words\ in\ that\ document)}$$

"to be or not to be"

$$\text{to} = \frac{1+1}{6}$$

to = 0.33

be = 0.33

or = 0.16

"It is going to rain today"

"Today I am not going outside"

"I am going to watch the season premiere"

Sentence 1	Sentence 2	Sentence 3
it	today	i
is	i	am
going	am	going
to	not	to
rain	going	watch
today	outside	the
		season
		premiere

Words/ Documents	Document 1	Document 2	Document 3
going	0.16	0.16	0.12
to	0.16	0	0.12
today	0.16	0.16	0
i	0	0.16	0.12
am	0	0.16	0.12
it	0.16	0	0
is	0.16	0	0
rain	0.16	0	0

IDF

Formula

$$log\left(\frac{(Number\ of\ documents)}{(Number\ of\ documents\ containing\ word)}\right)$$

$$log\left(\frac{(Number\ of\ documents)}{(Number\ of\ documents\ containing\ word)}\right)$$

"to be or not to be"
"i have to be"
"you got to be"

$$\text{to} = \log\left(\frac{3}{3}\right)$$

$$\text{to} = 0$$

$$\text{be} = \log\left(\frac{3}{3}\right)$$

$$\text{be} = 0$$

$$\text{have} = \log\left(\frac{3}{1}\right)$$

Words	IDF Value
going	log(3/3)
to	log(3/2)
today	log(3/2)
i	log(3/2)
am	log(3/2)
It	log(3/1)
is	log(3/1)
rain	log(3/1)

"it is going to rain today"

"today i am not going outside"

"i am going to watch the season premiere"

Words	IDF Value
going	0
to	0.41
today	0.41
i	0.41
am	0.41
It	1.09
is	1.09
rain	1.09

Words/ Documents	Document 1	Document 2	Document 3
going	0.16	0.16	0.12
to	0.16	0	0.12
today	0.16	0.16	0
i	0	0.16	0.12
am	0	0.16	0.12
it	0.16	0	0
is	0.16	0	0
rain	0.16	0	0

Words/ Documents	going	to	today	i	am	it	is	rain
Document 1	0	0.07	0.07	0	0	0.17	0.17	0.17
Document 2	0	0	0.07	0.07	0.07	0	0	0
Document 3	0	0.05	0	0.05	0.05	0	0	0

$$TFIDF(Word) = TF(Document, Word) * IDF(Word)$$