DataScience

Machine Learning with Python

Introduction

Introduction

How People Spend Money?

What treatments for which illness?
How patient is responding?

Introduction

Car engine monitoring system

Banking


Learn spending
Pattern


BANK


CREDIT CARD
Fraud Alert


How People
Spend Money?

Hospital

Can we identify treatments that don't work as expected , quickly ?

What treatments for which illness?
How patient is responding?

Car Monitoring

Can a Intelligent Car give early warnings of failure?

Car engine monitoring system

Some Questions That Machine Learning Can Be Used To Answer

Machine Learning

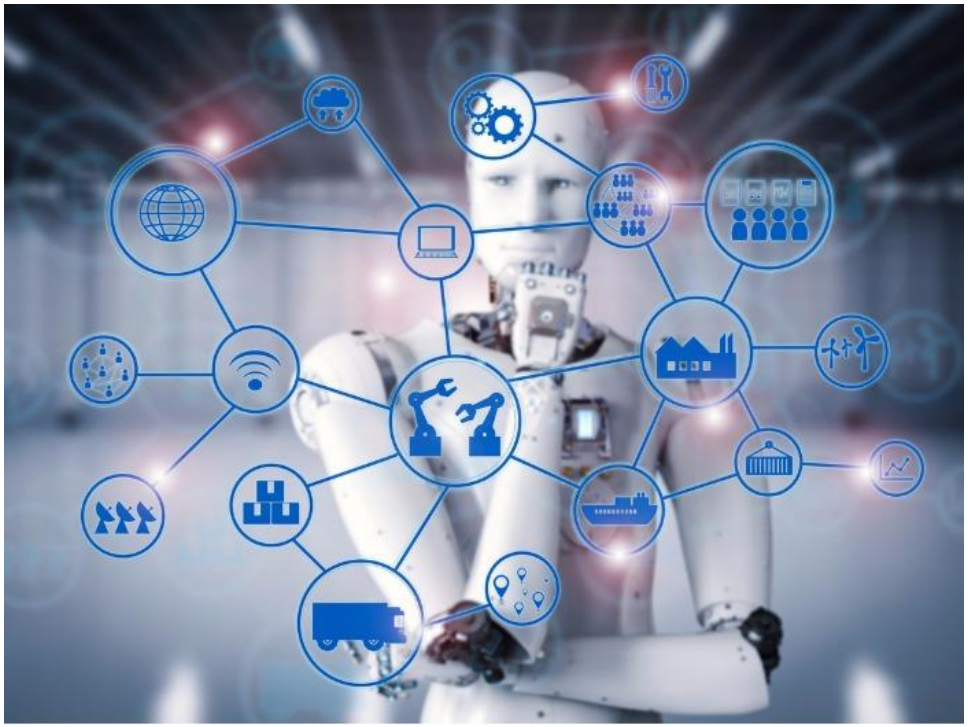
The size and complexity of these datasets mean that humans are unable to extract useful information from them.

Even the way that the data is stored works against us.

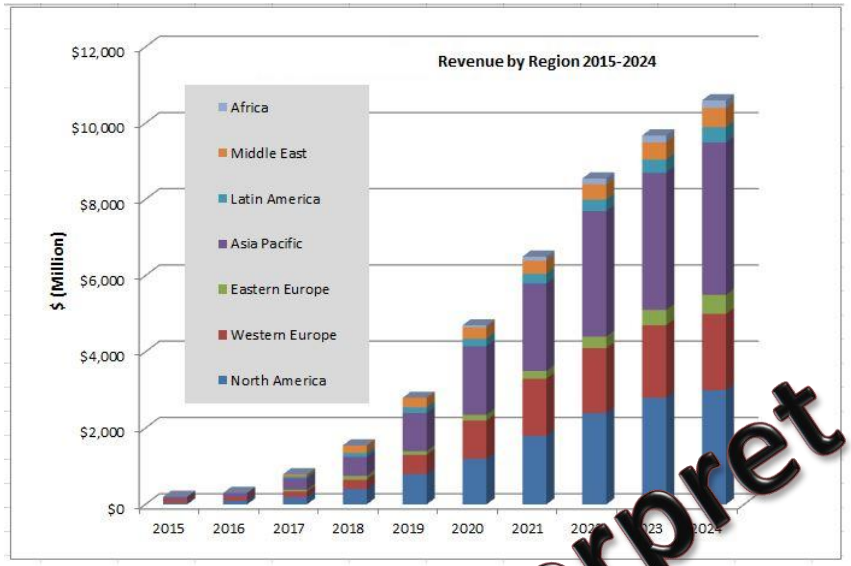
Given a file full of numbers, our minds generally turn away from looking at them for long.

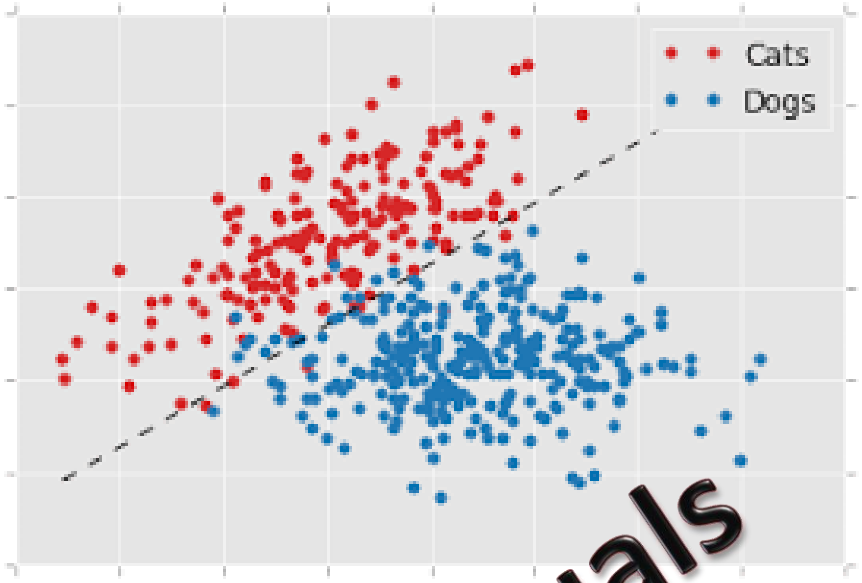
Machine Learning

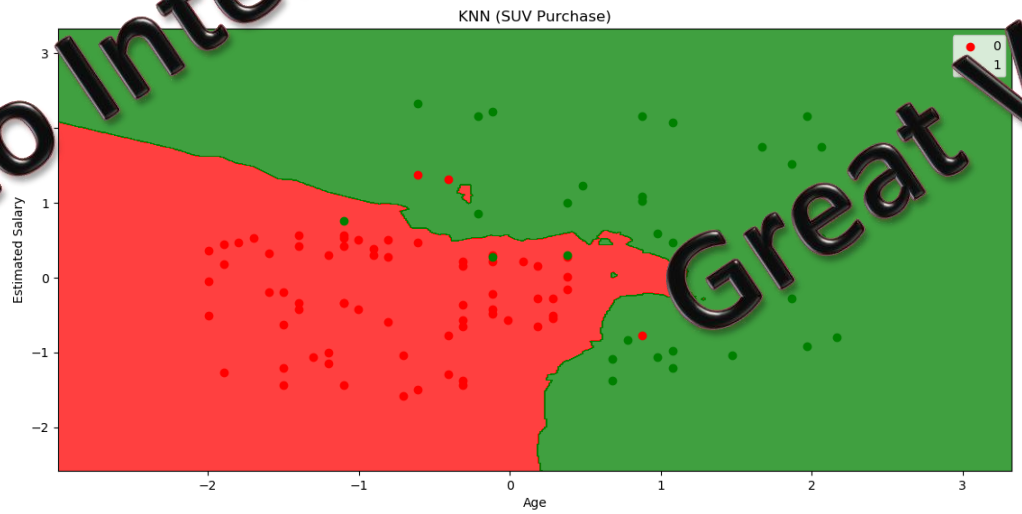
This is the reason why machine learning is becoming so popular — the problems of our human limitations go away if we can make computers do the dirty work for us.



Machine Learning





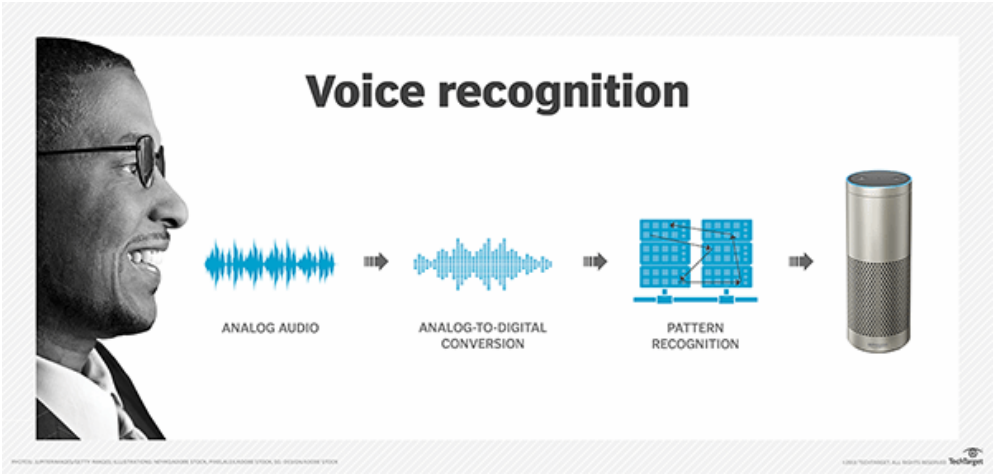


Easy to Interpret

Great Visuals

ML Algorithms

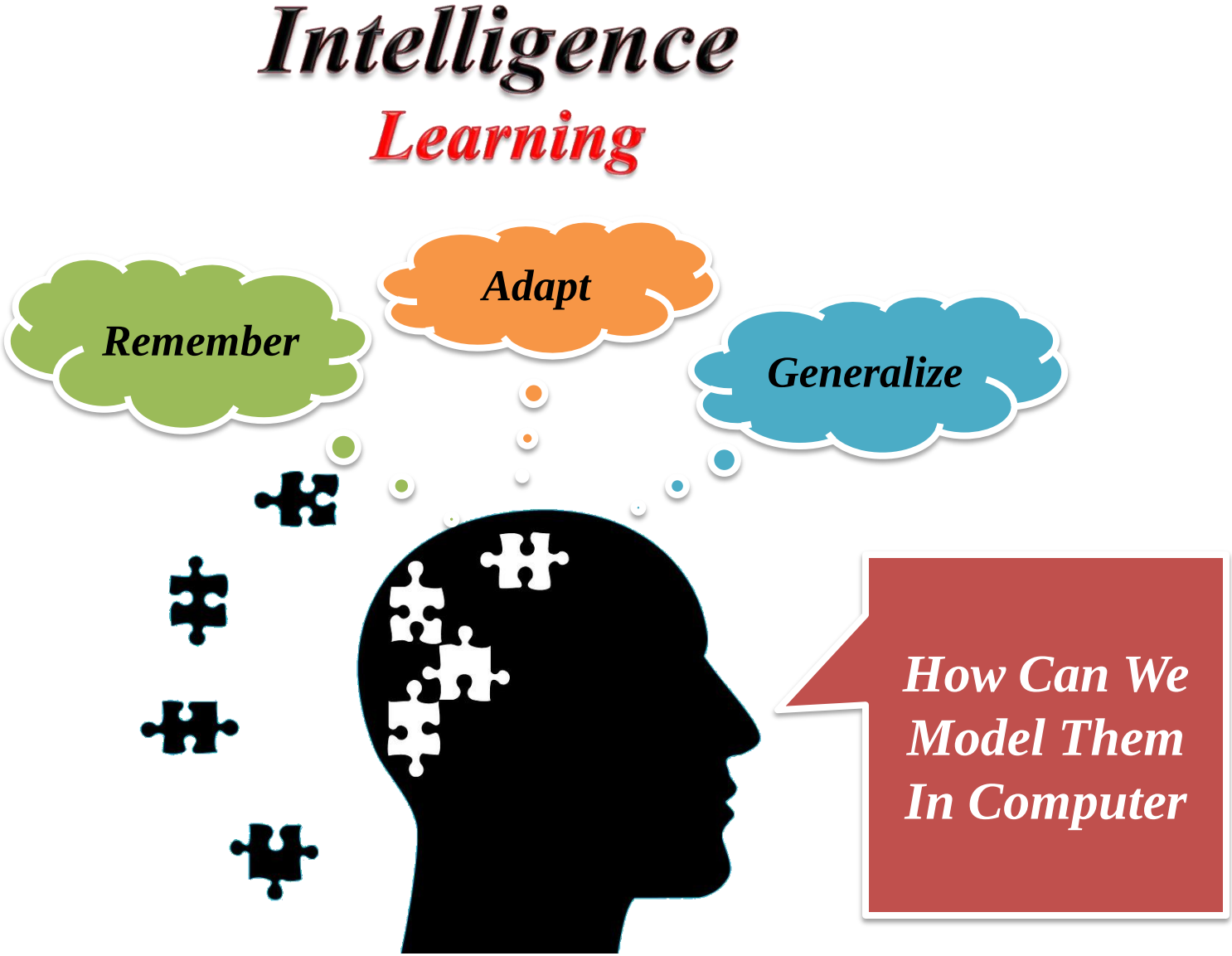
Spam Filters



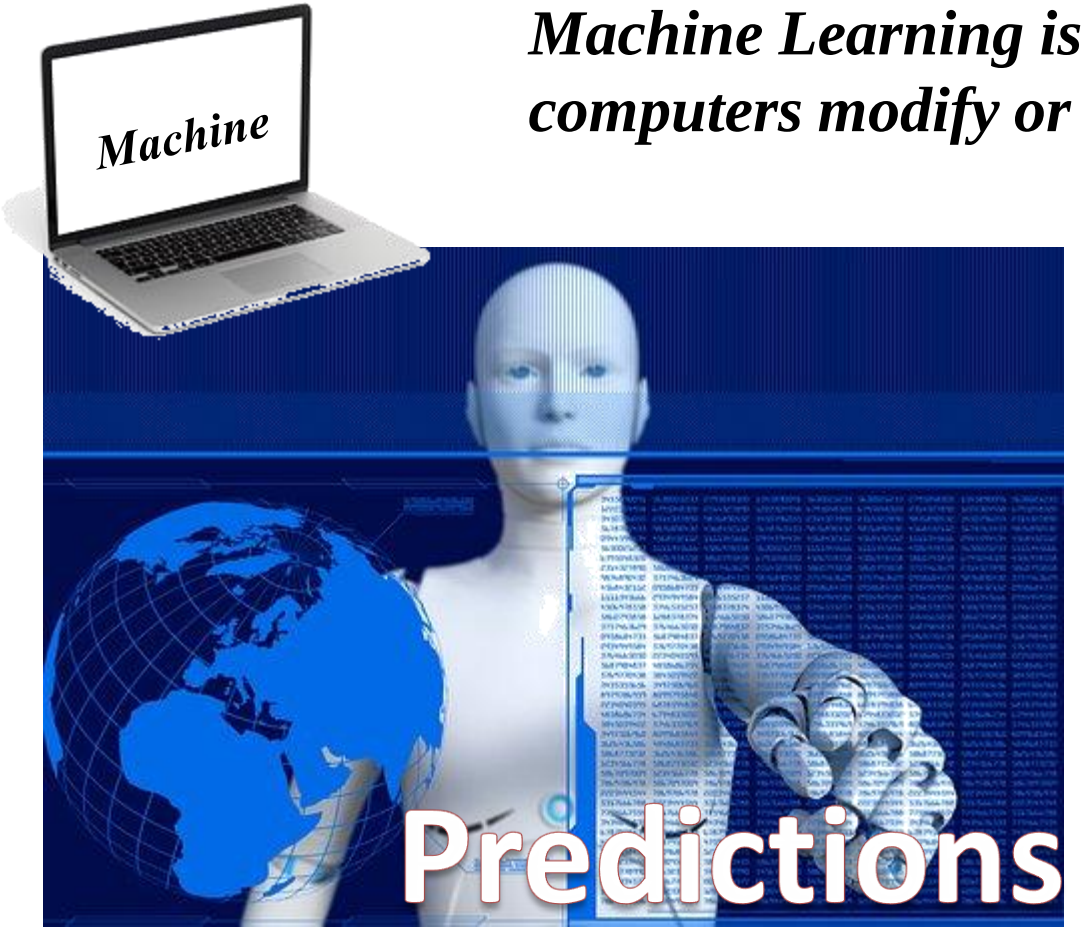
Computer Games

Algorithm to decide whether bank will give loan or not

Machine Learning



Machine Learning



Machine Learning is about making computers modify or adapt their actions

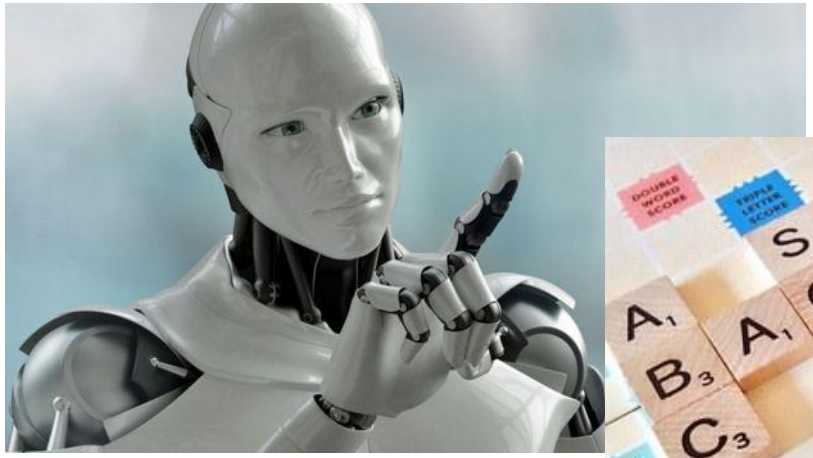
Accurate Actions

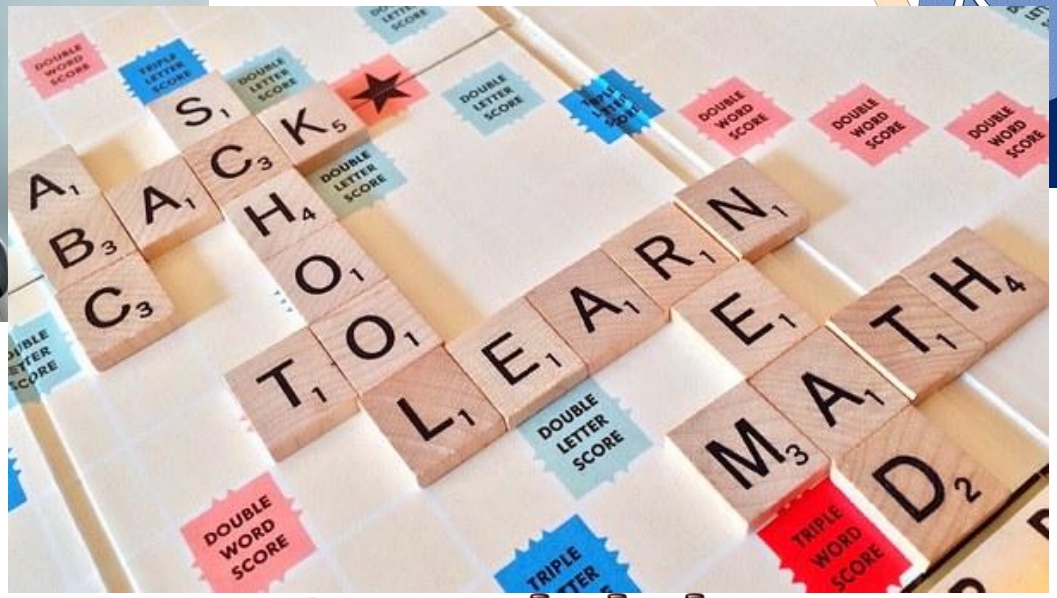
Accuracy is measured by how well the chosen action reflects the correct ones

Machine Learning

learn Win
learn learn Win
learn Win

Win
Win
loose
loose Win
loose





Scrabble

Machine Learning

- Machine Learning is a field that uses **algorithms** to **learn from data** and make **predictions**.
- We can **feed data** into an **algorithm** and use it to **make predictions** about what might happen in the future.
- This has vast range of applications e.g:
    - ***Self-driving car***
    - ***Stock Price Prediction***

Machine Learning

- **Supervised Learning**
- **Unsupervised Learning**

Supervised Learning

*A **training set** of examples with the correct responses (targets) is provided and based on this training set the algorithm **generalises** to respond correctly to all possible inputs.*

e.g:

- ***Linear Regression***
- ***K-Nearest Neighbour***
- ***Decision Trees***

Unsupervised Learning

Correct responses are not provided, but instead the algorithm tries to ***identify similarities*** *between the inputs that have something in common , are* ***categorised together***

e.g:

- ***K-Means Cluster algorithm***
- ***Anomaly detection***

More Data, More Questions, Better Answers

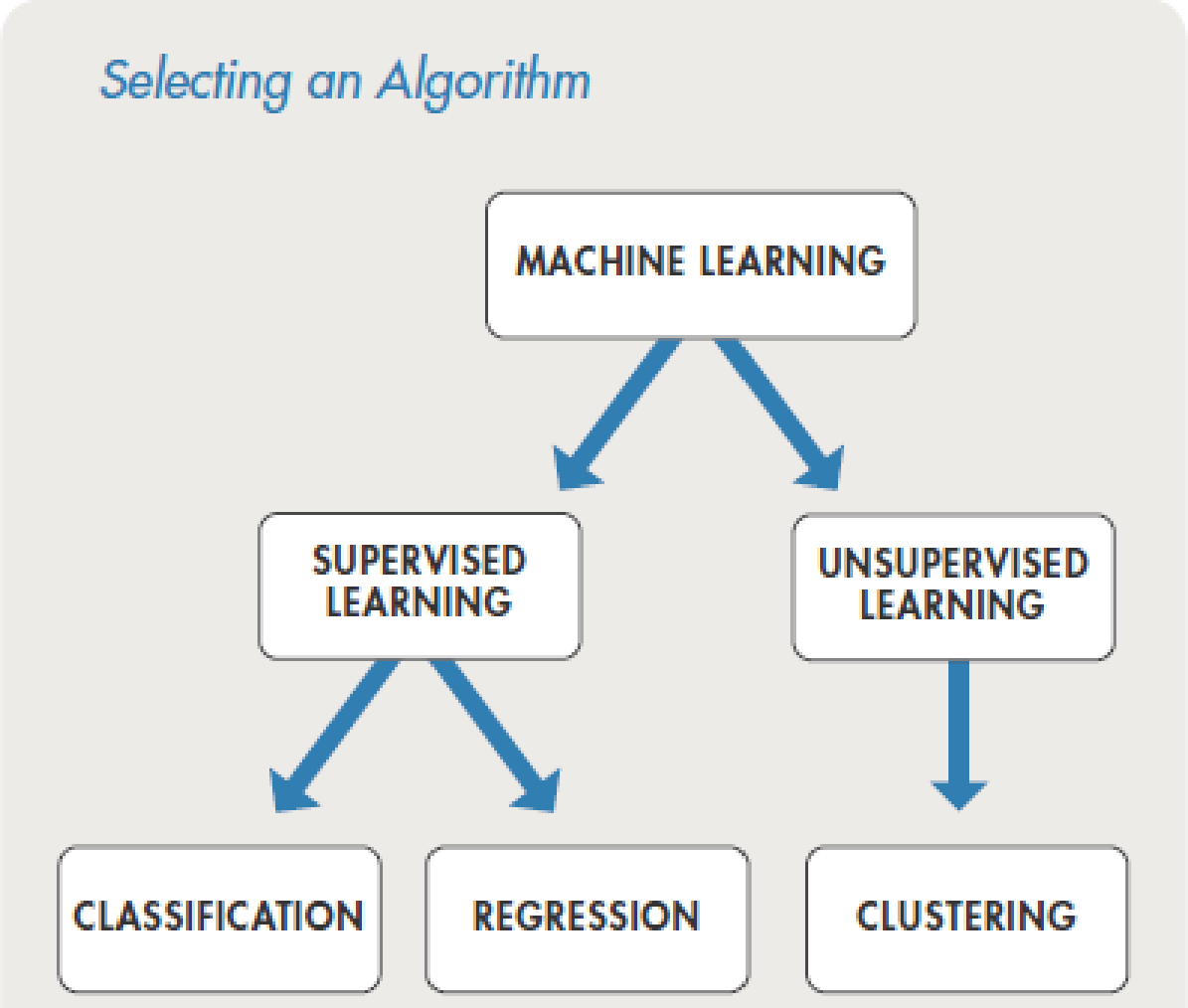
- Machine learning algorithms find natural patterns in data that generate insight and help you make better decisions and predictions.

- They are used every day to make critical decisions in medical diagnosis, stock trading, energy load forecasting, and more.

Better Data, Better Decisions

Today's Decisions Generate Data for future

No best method or one size fits all



Machine Learning Challenges

Data comes in all shapes and sizes

Real-world datasets can be messy, incomplete, and in a variety of formats. You might just have simple numeric data. But sometimes you're combining several different data types, such as sensor signals, text, and streaming images from a camera.

Machine Learning Challenges

Preprocessing your data might require specialized knowledge and tools.

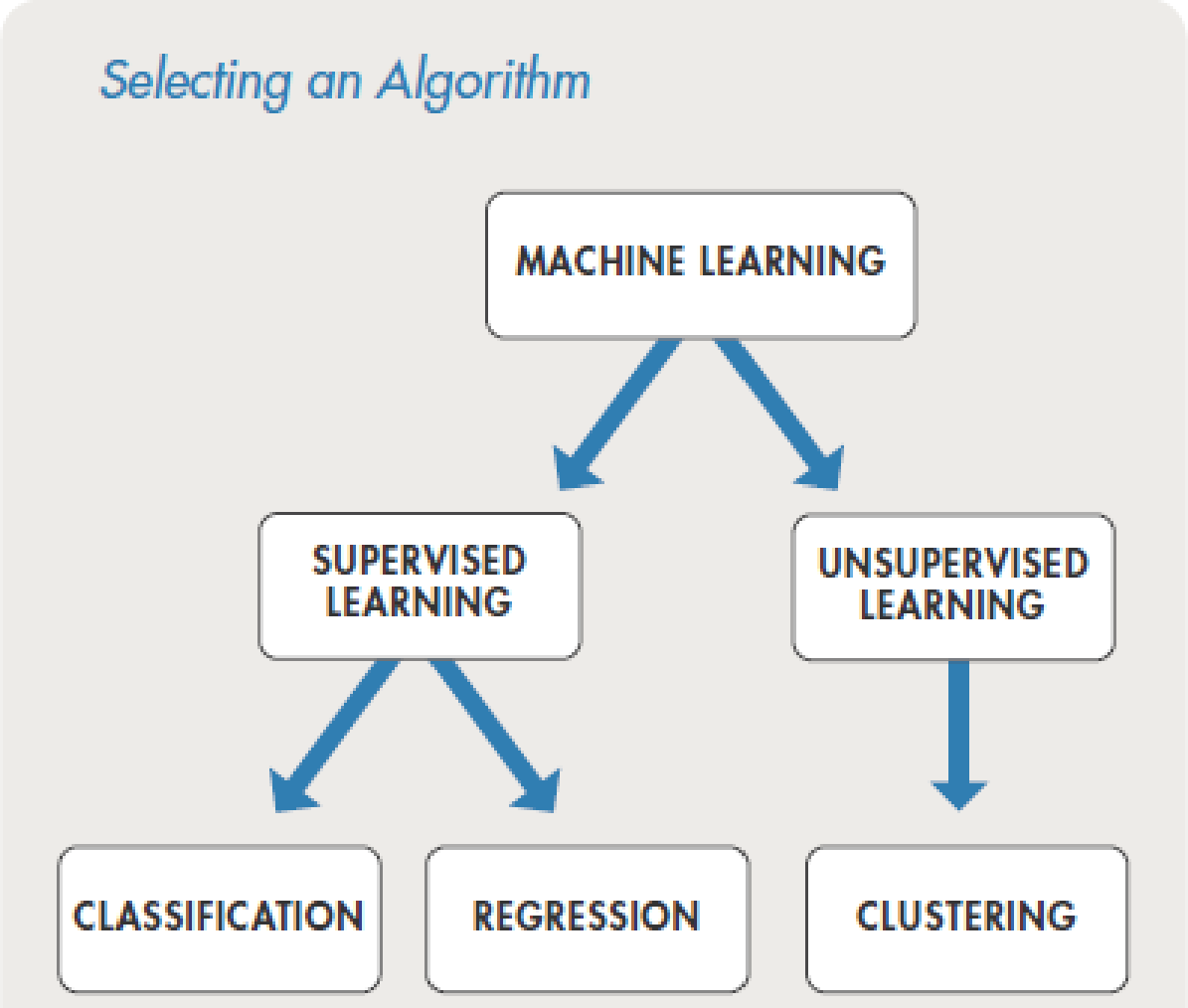
For example, to select features to train an object detection algorithm requires specialized knowledge of image processing. Different types of data require different approaches to preprocessing.

Machine Learning Challenges

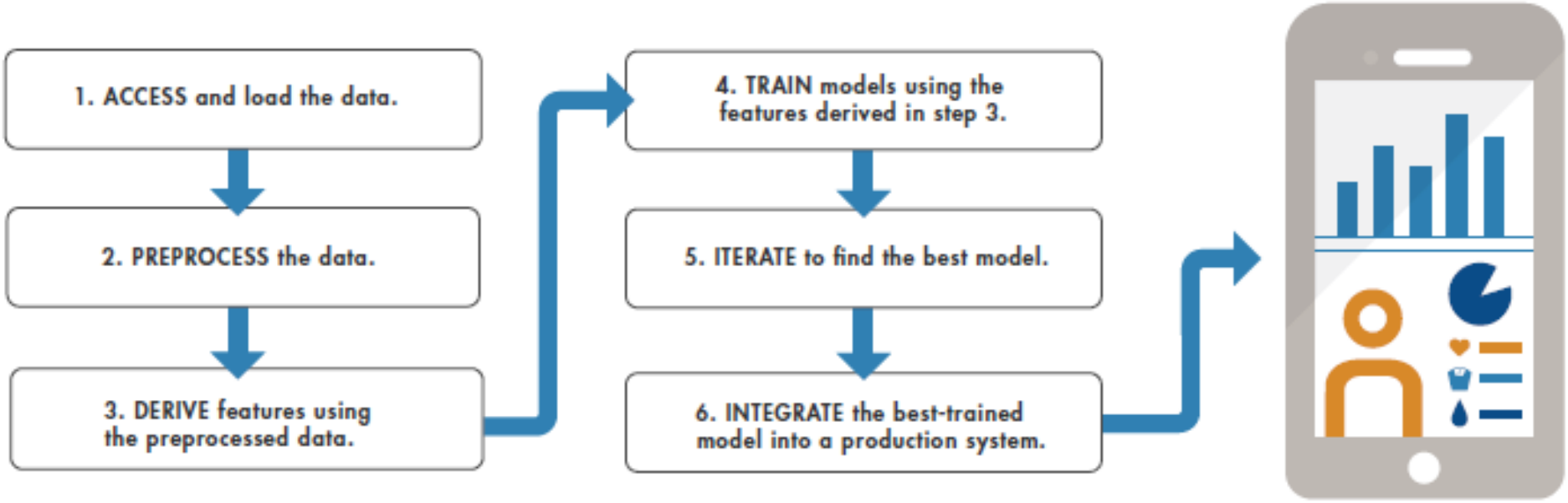
It takes time to find the best model to fit the data.

Choosing the right model is a balancing act. Highly flexible models tend to overfit data by modeling minor variations that could be noise. On the other hand, simple models may assume too much. There are always tradeoffs between model speed, accuracy, and complexity.

No best method or one size fits all



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

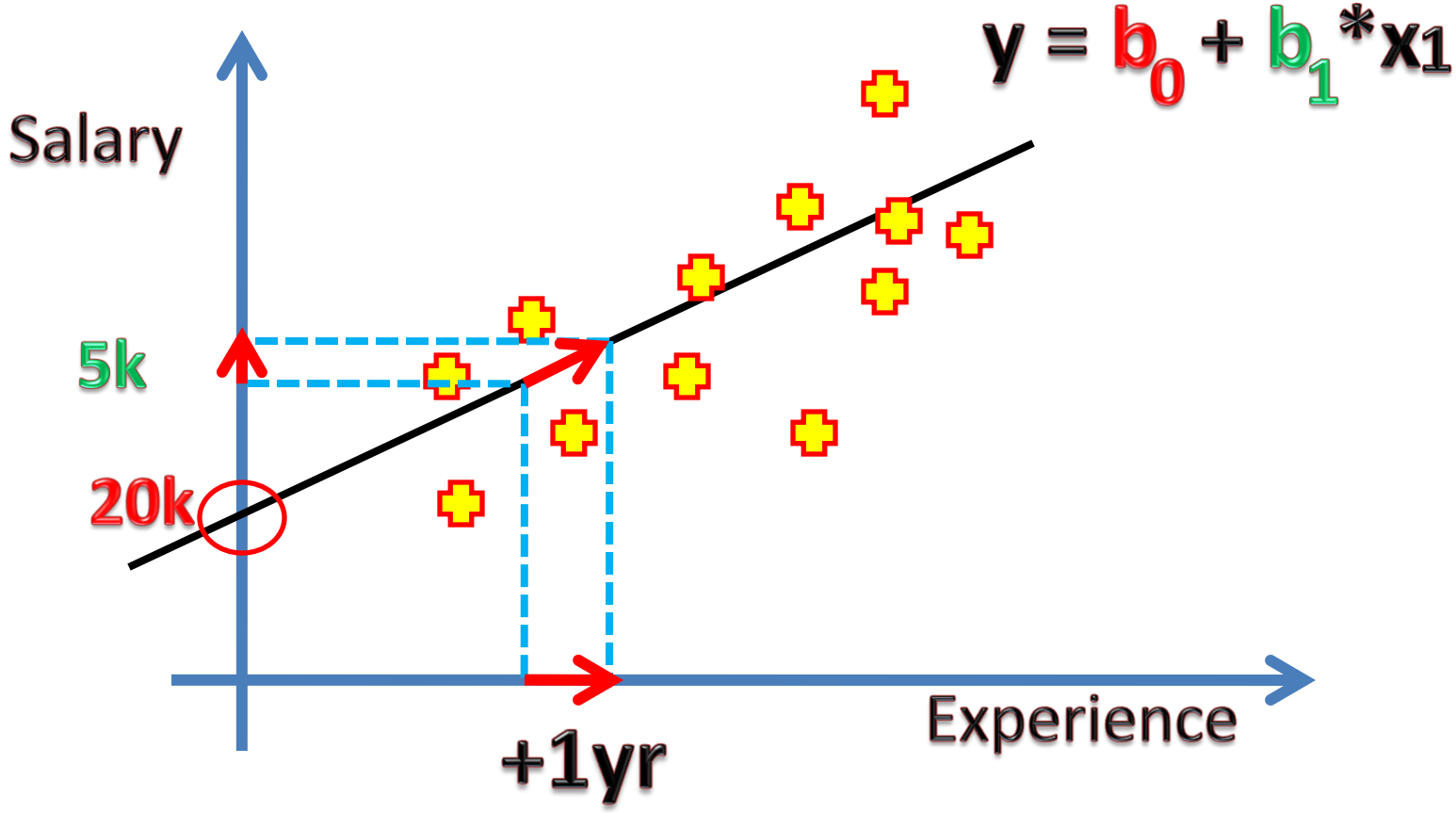
- StandardScaler will **transform** the data such that its distribution will have a **mean value 0** and **standard deviation of 1**
- This is useful while **comparing data** that corresponds to **different units**. In that case, we want to **remove the units**.
- This is done in a consistent way for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

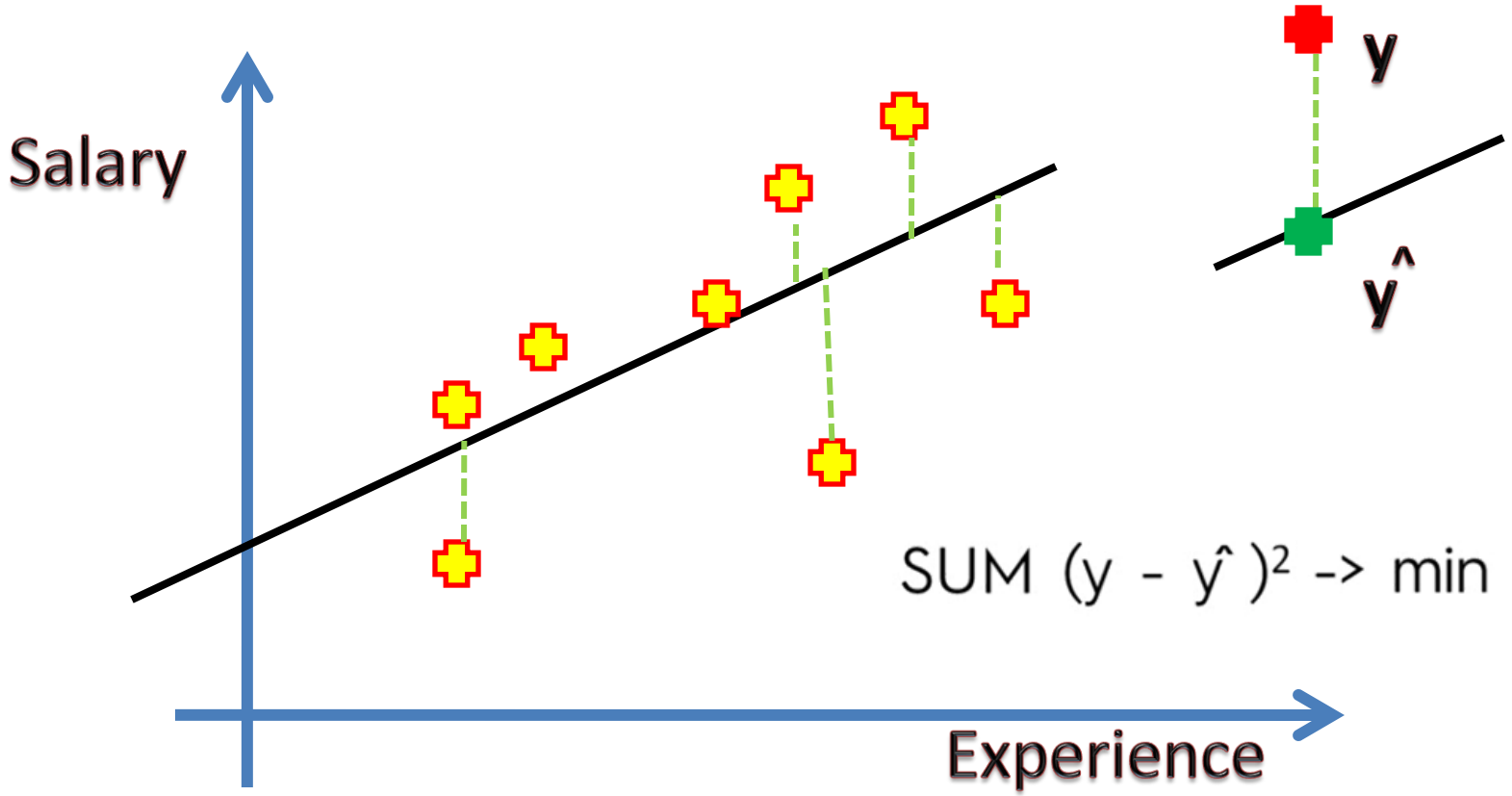
$$y = b_0 + b_1 * x_1$$

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

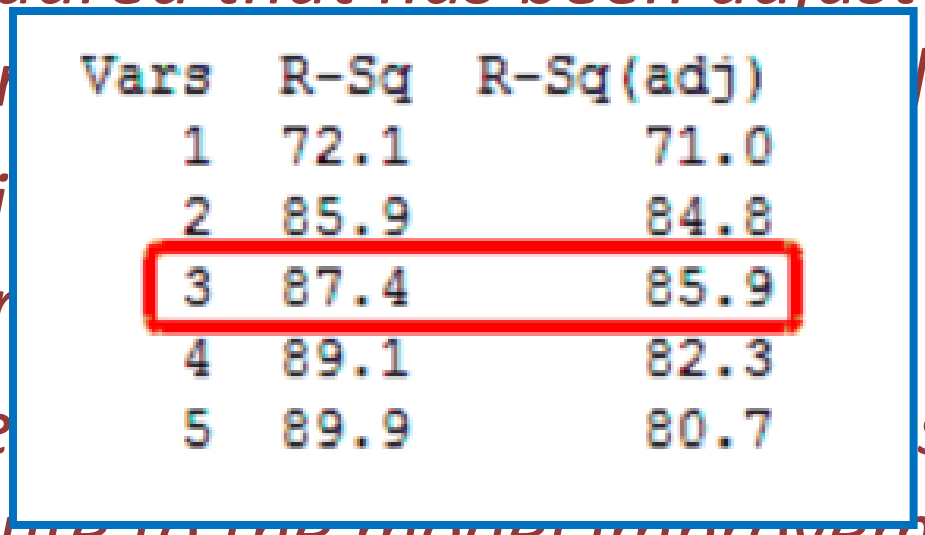
R-squared

- *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

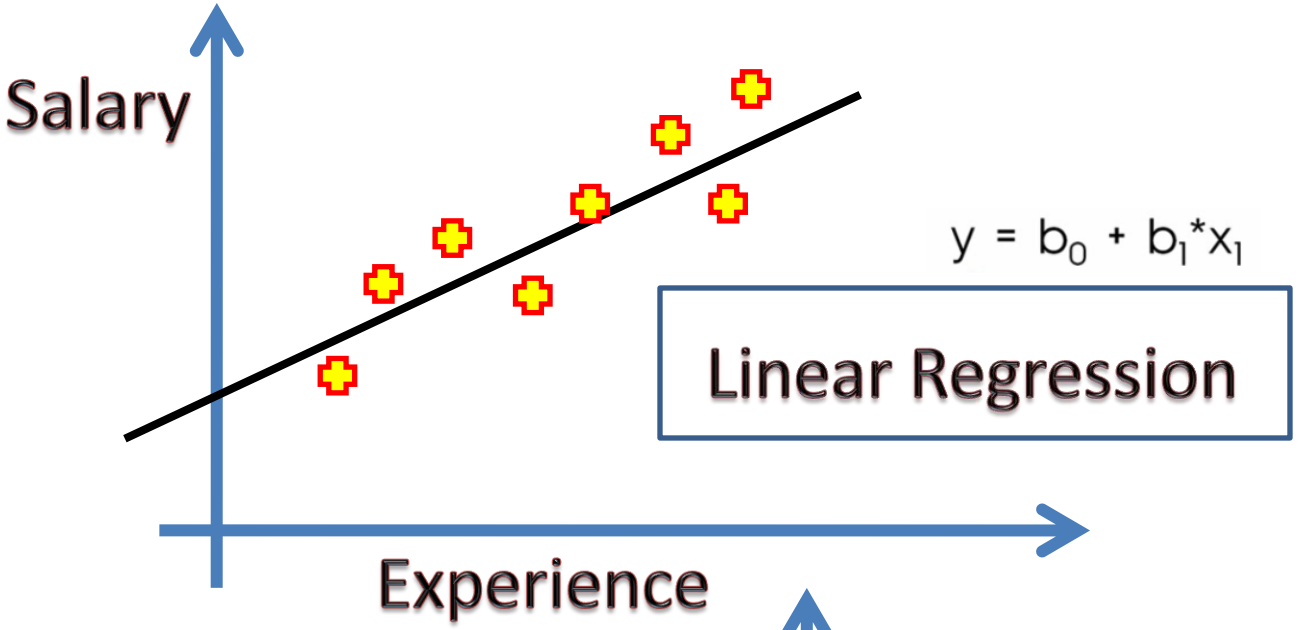
- *The adjusted R-squared is a modified version of R-squared that has been adjusted for the numbe*
- *The adj* *only if the new ter*
- *It decre* *sn't contribute to the model improvement.*

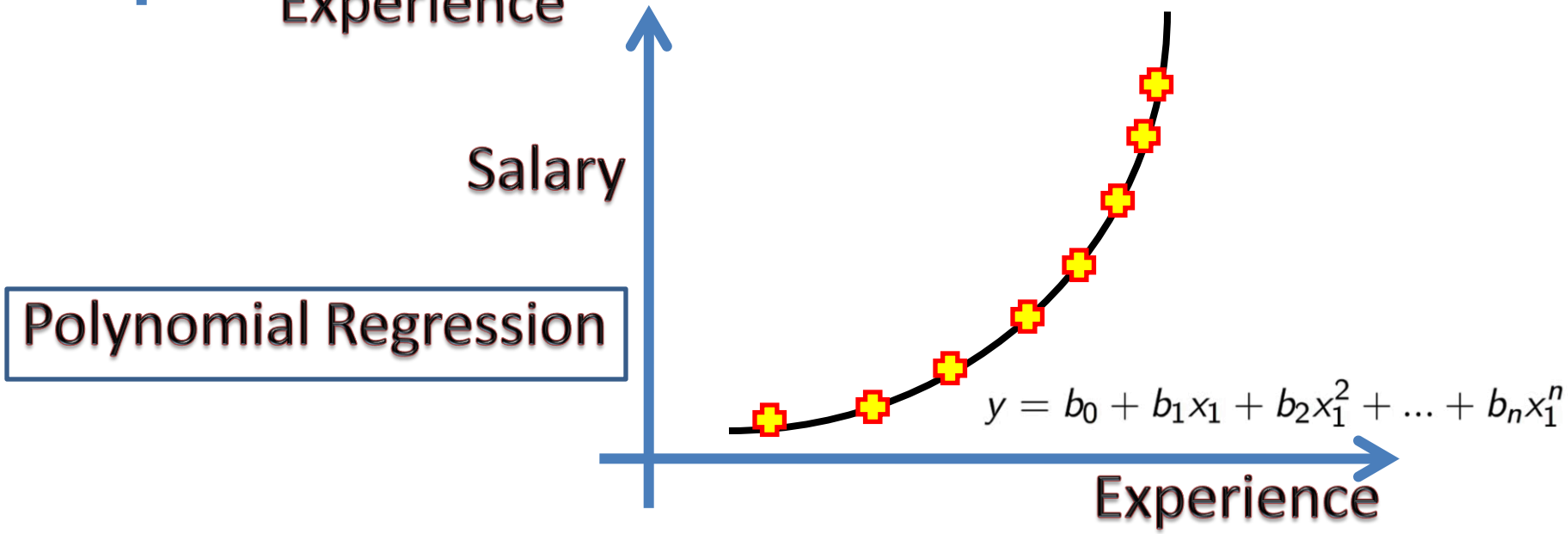


Vars	R-Sq	R-Sq(adj)
1	72.1	71.0
2	85.9	84.8
3	87.4	85.9
4	89.1	82.3
5	89.9	80.7

Polynomial Regression

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

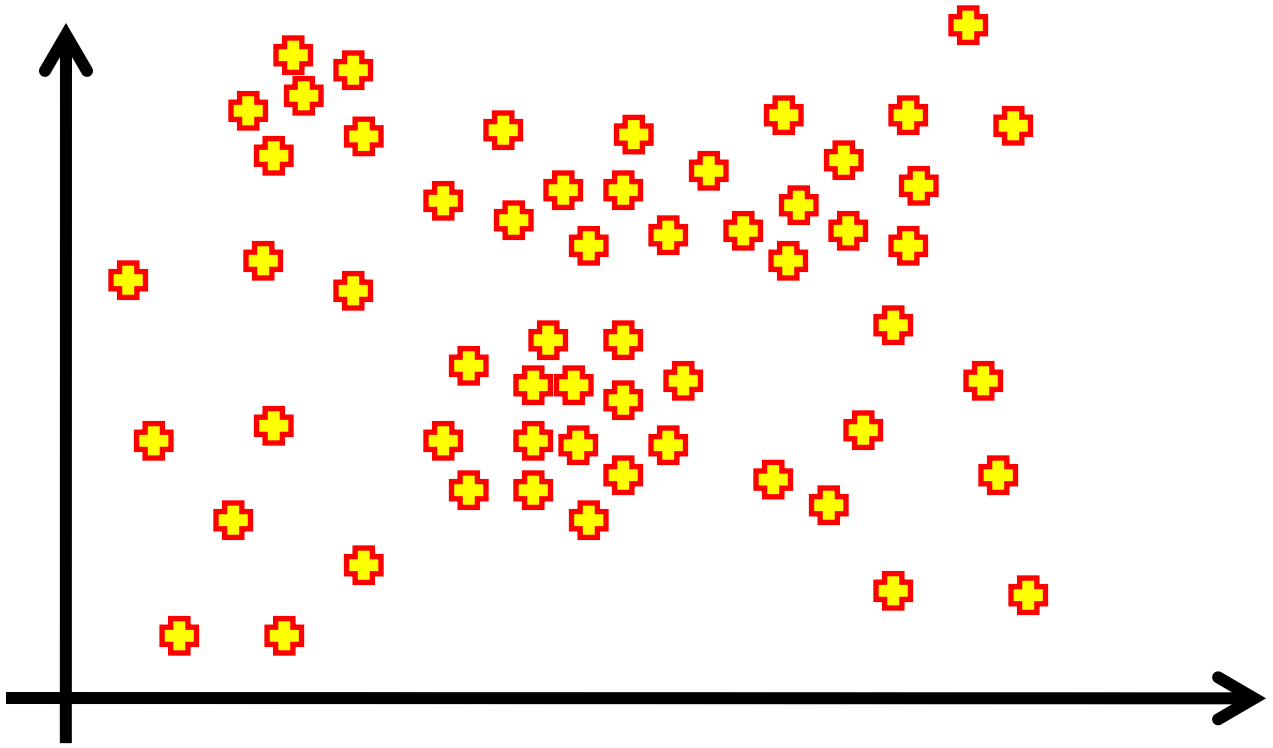

Salary
Experience
$y = b_0 + b_1 * x_1$
Linear Regression


Salary
Polynomial Regression
$y = b_0 + b_1 x_1 + b_2 x_1^2 + \ldots + b_n x_1^n$
Experience

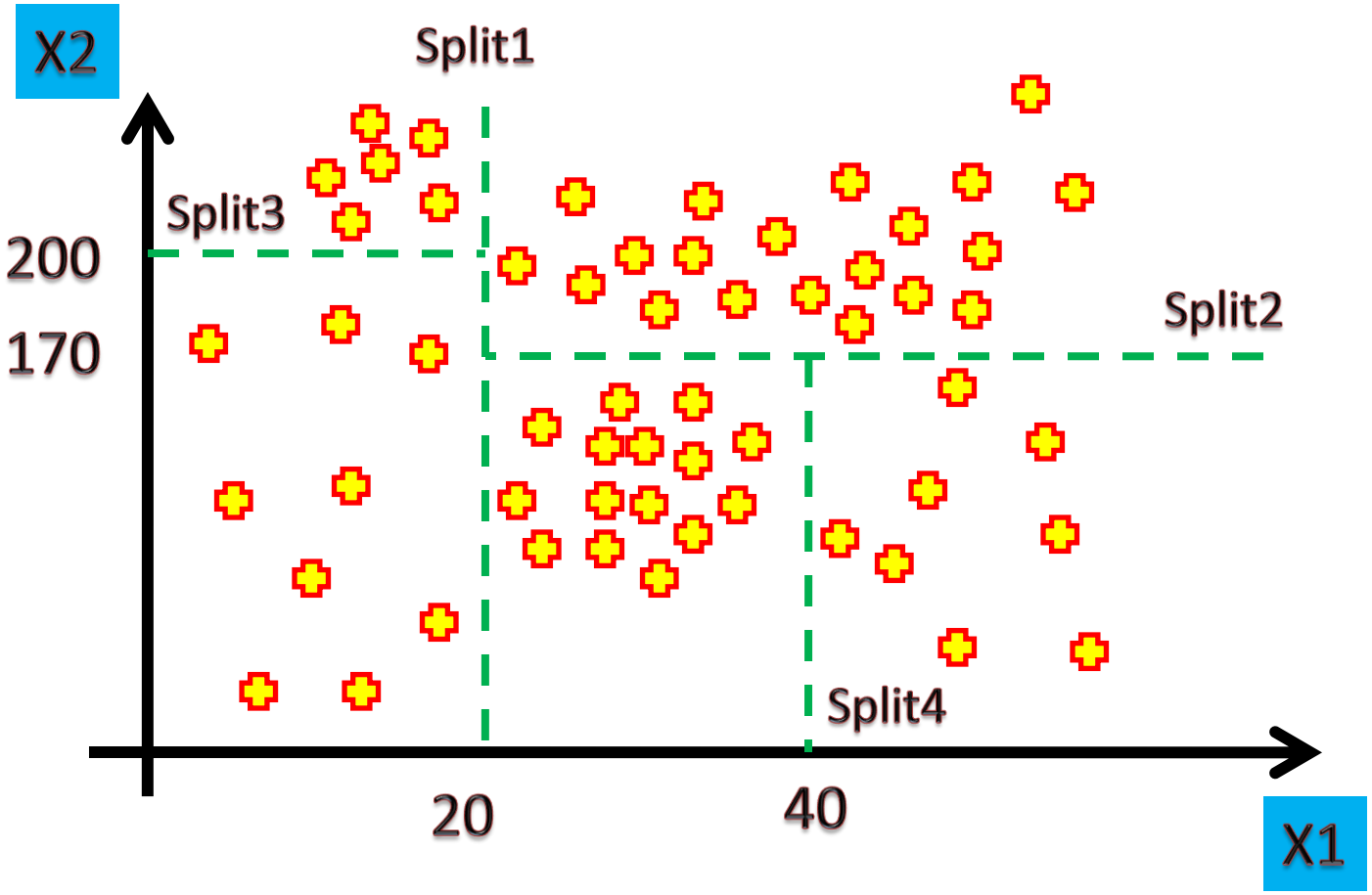
Decision Tree Regression

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.

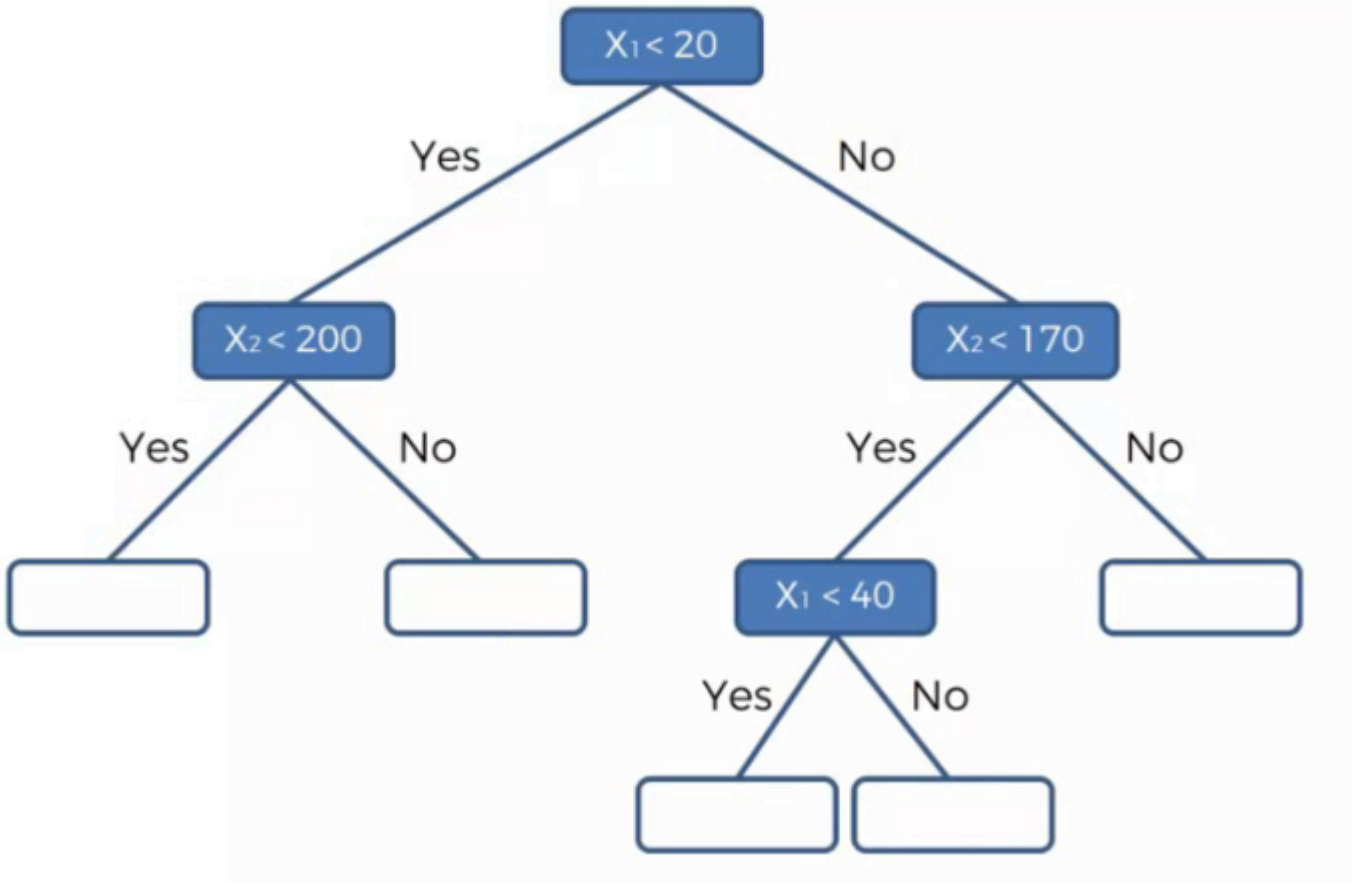
Decision Tree Regression



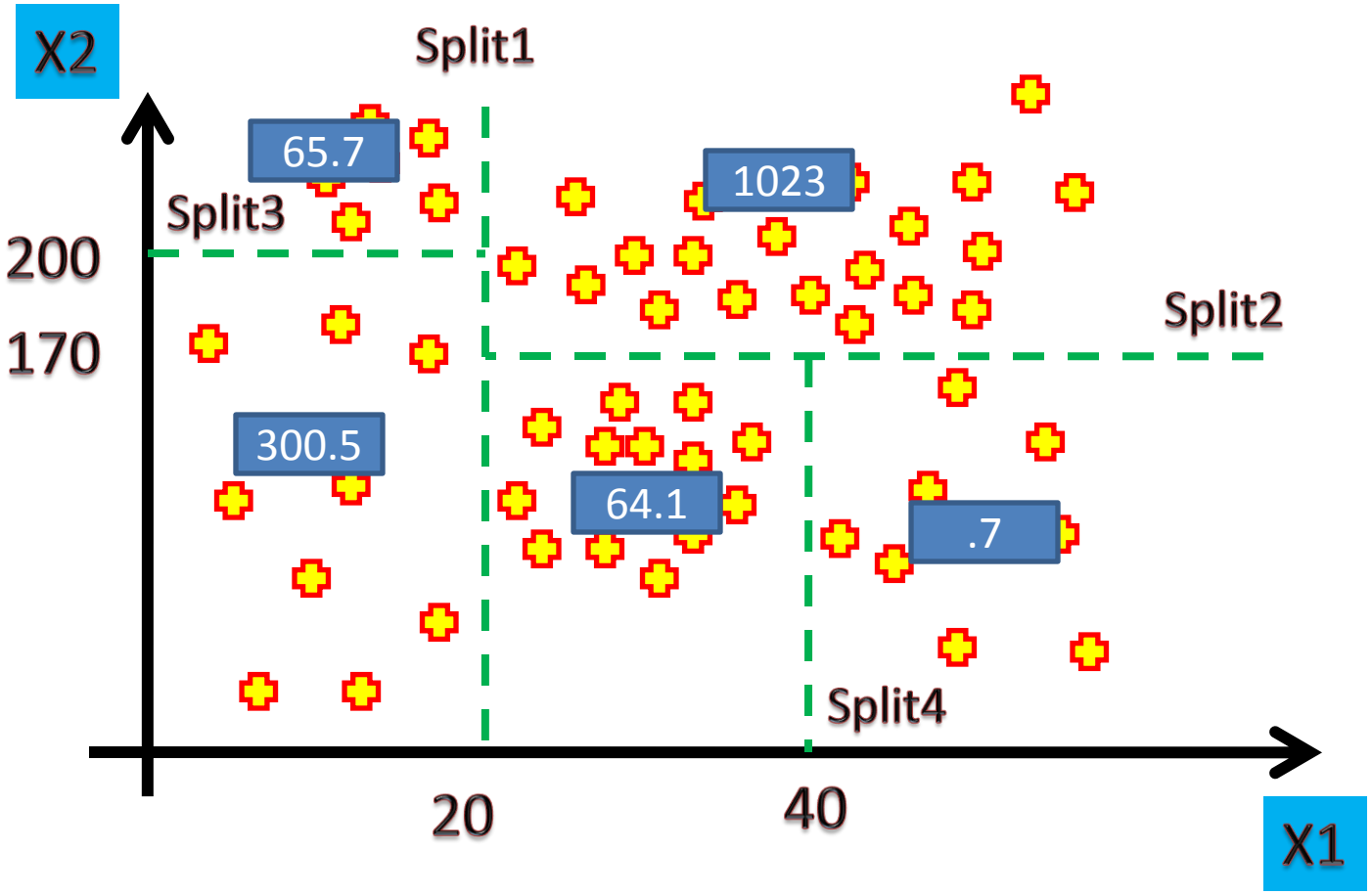
Decision Tree Regression



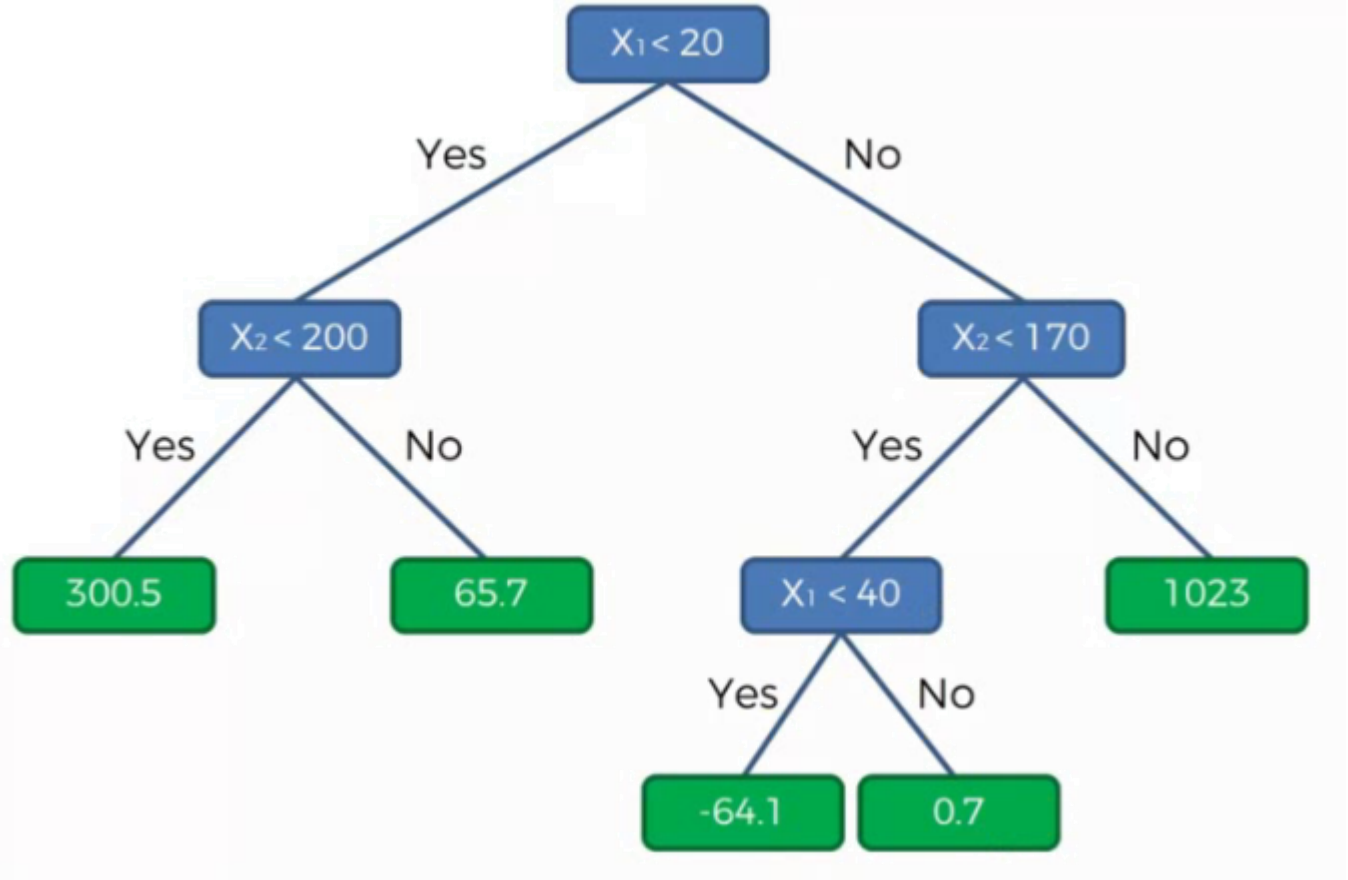
Decision Tree Regression



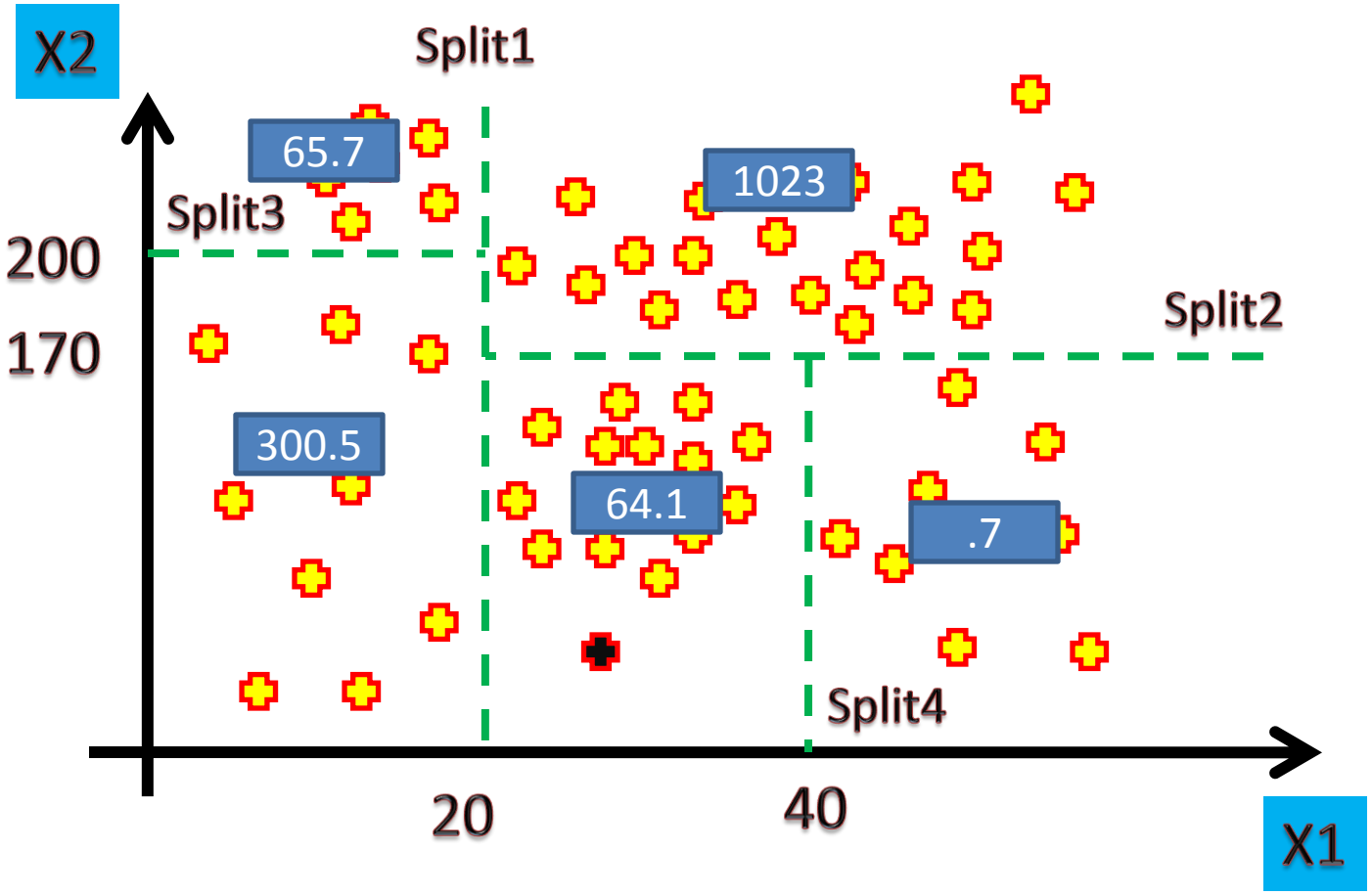
Decision Tree Regression



Decision Tree Regression

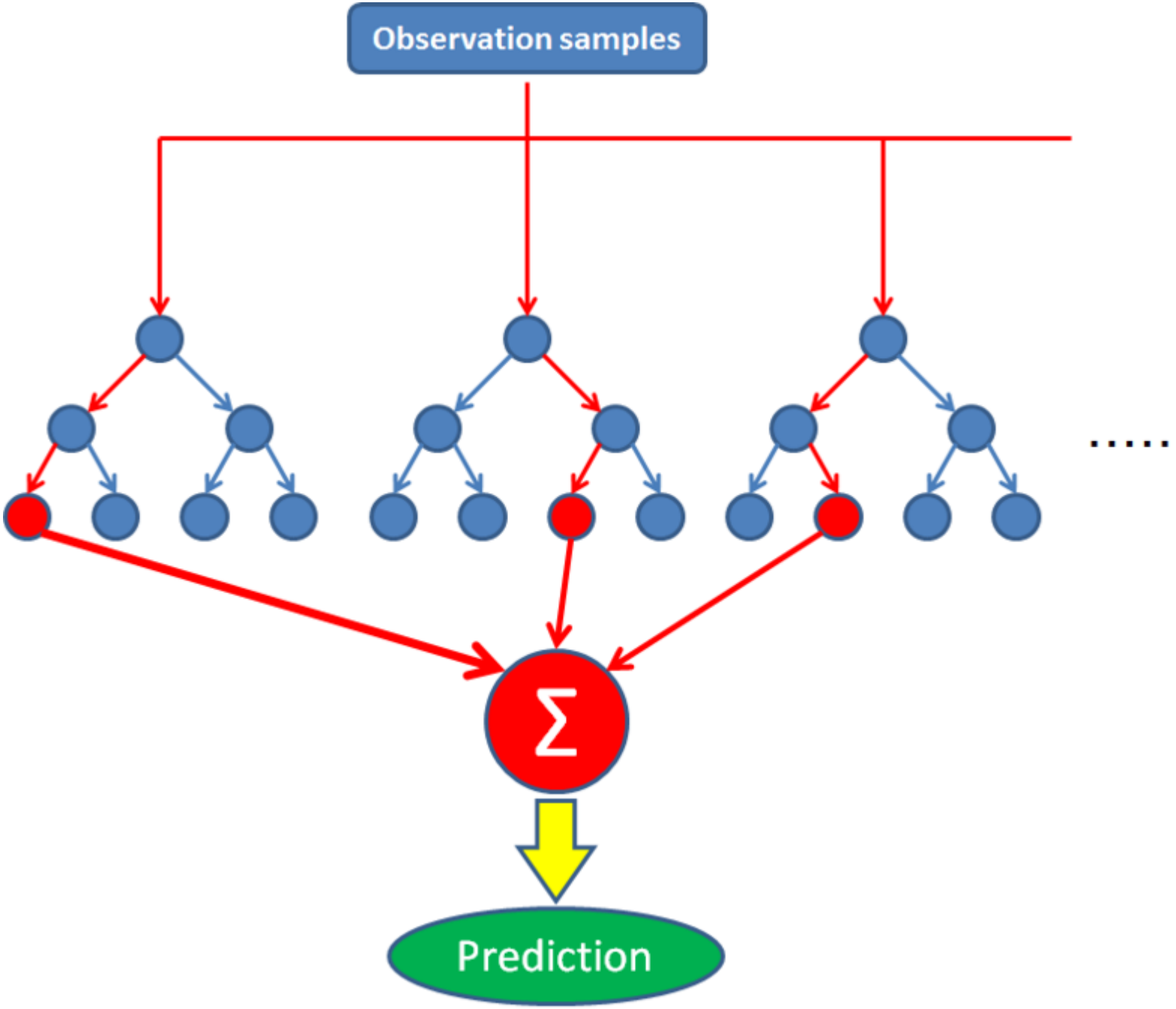


Decision Tree Regression



Random Forest Regression

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

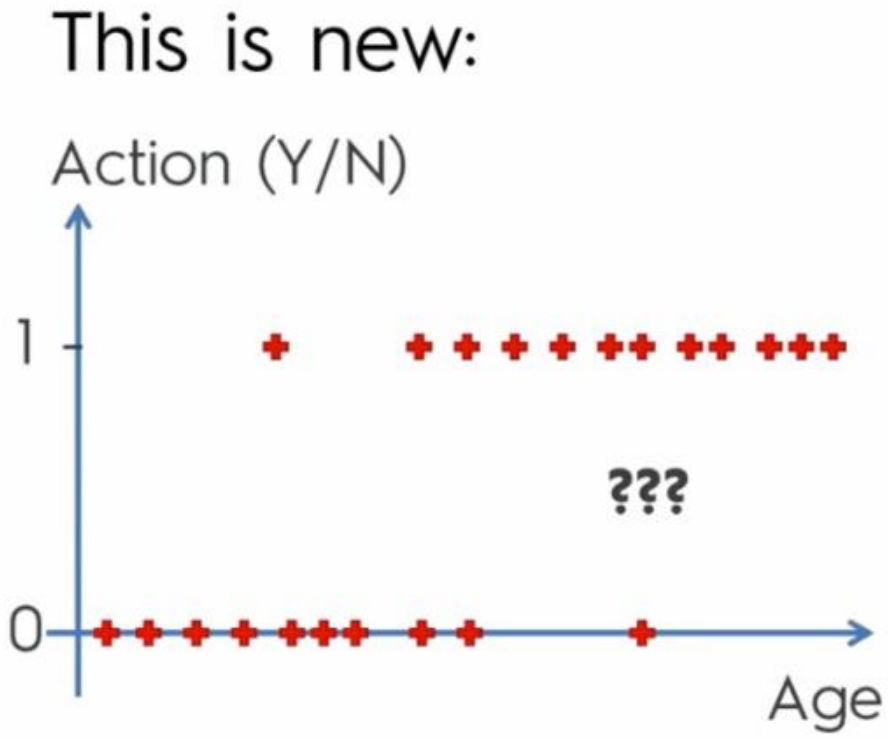

Observation samples
.....
Σ
Prediction

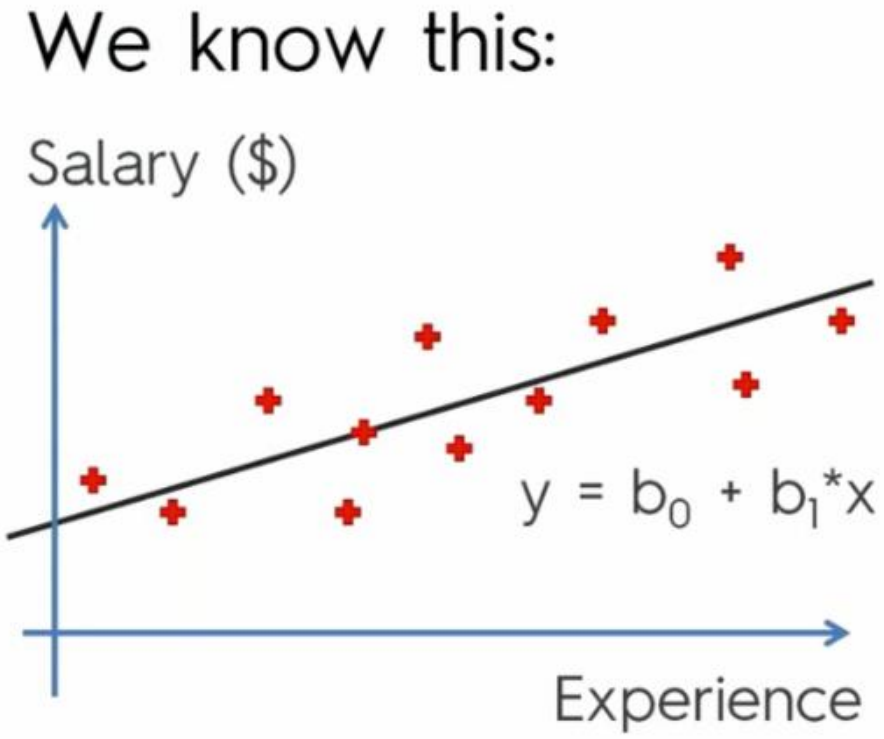
Classification

Confusion Matrix

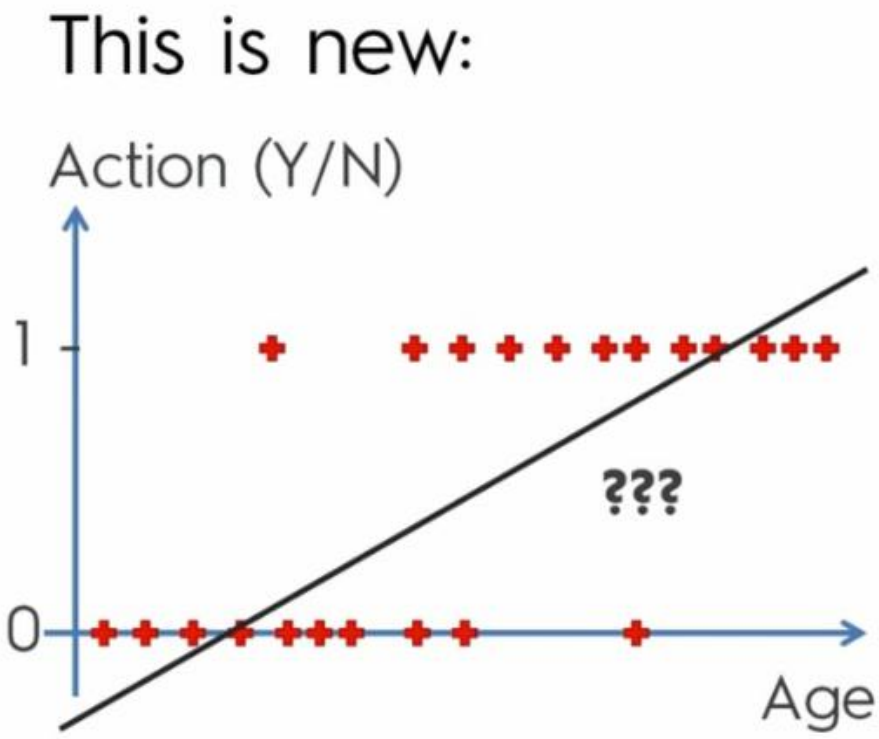
n=165	**Predicted: NO**	**Predicted: YES**	
Actual: NO	TN = 50	FP = 10	60
Actual: YES	FN = 5	TP = 100	105
	55	110	

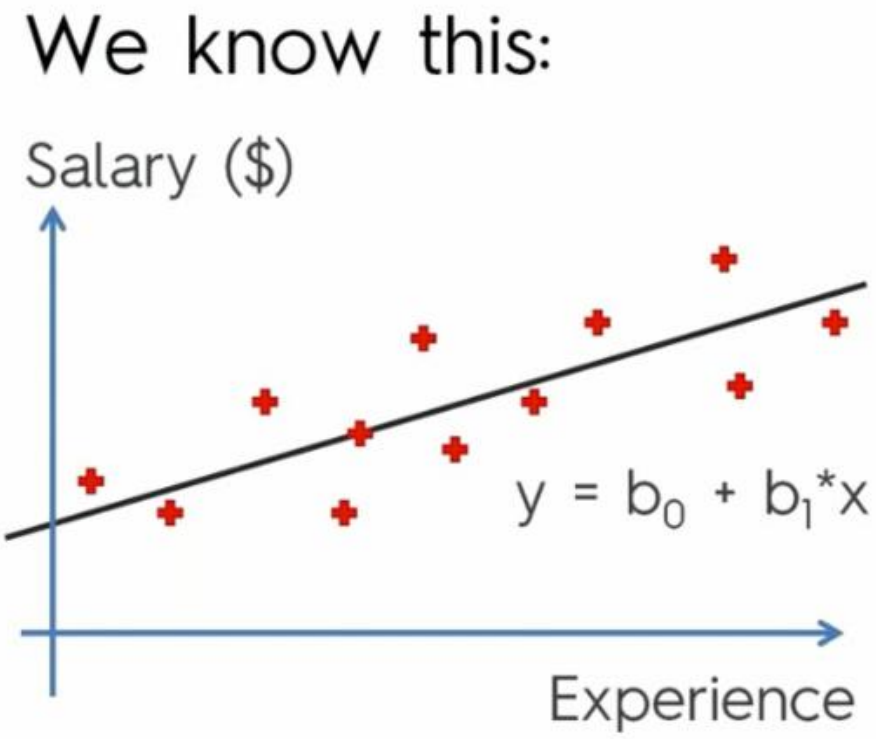
Logistic Regression



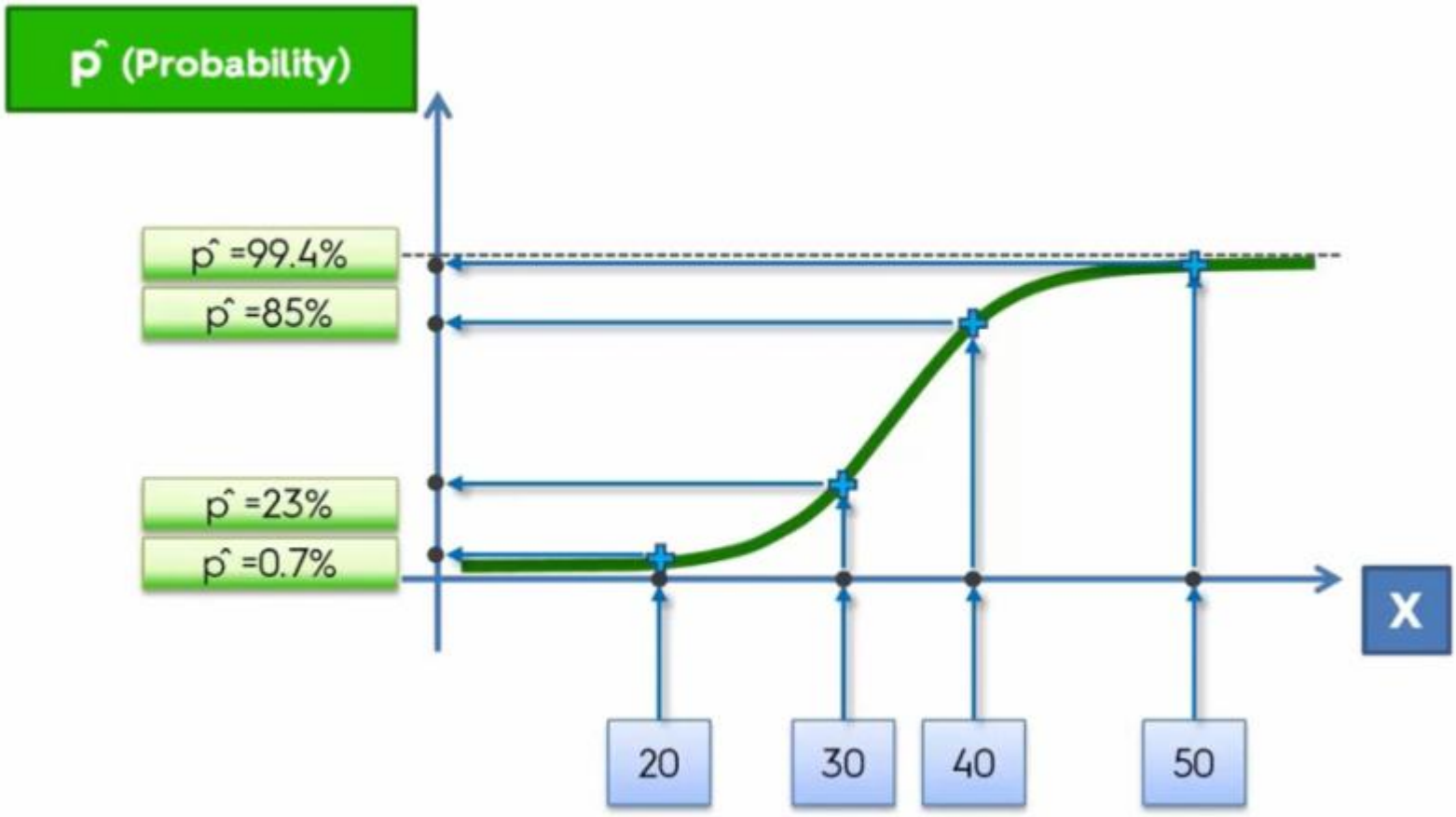


Logistic Regression





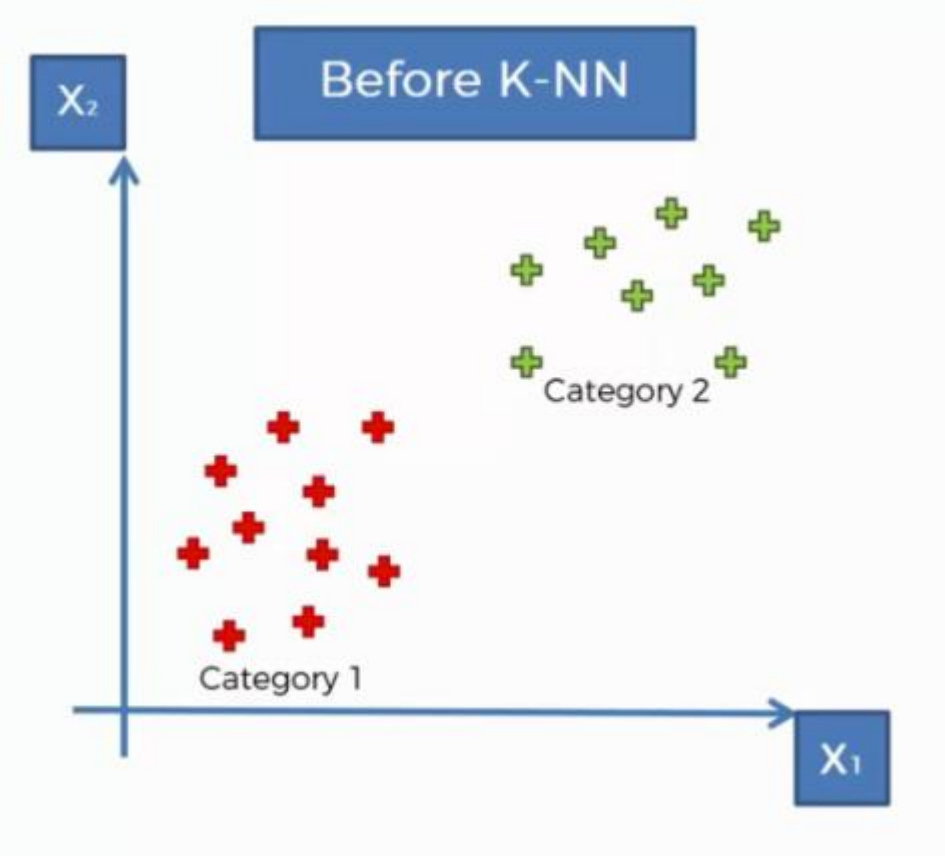
Logistic Regression

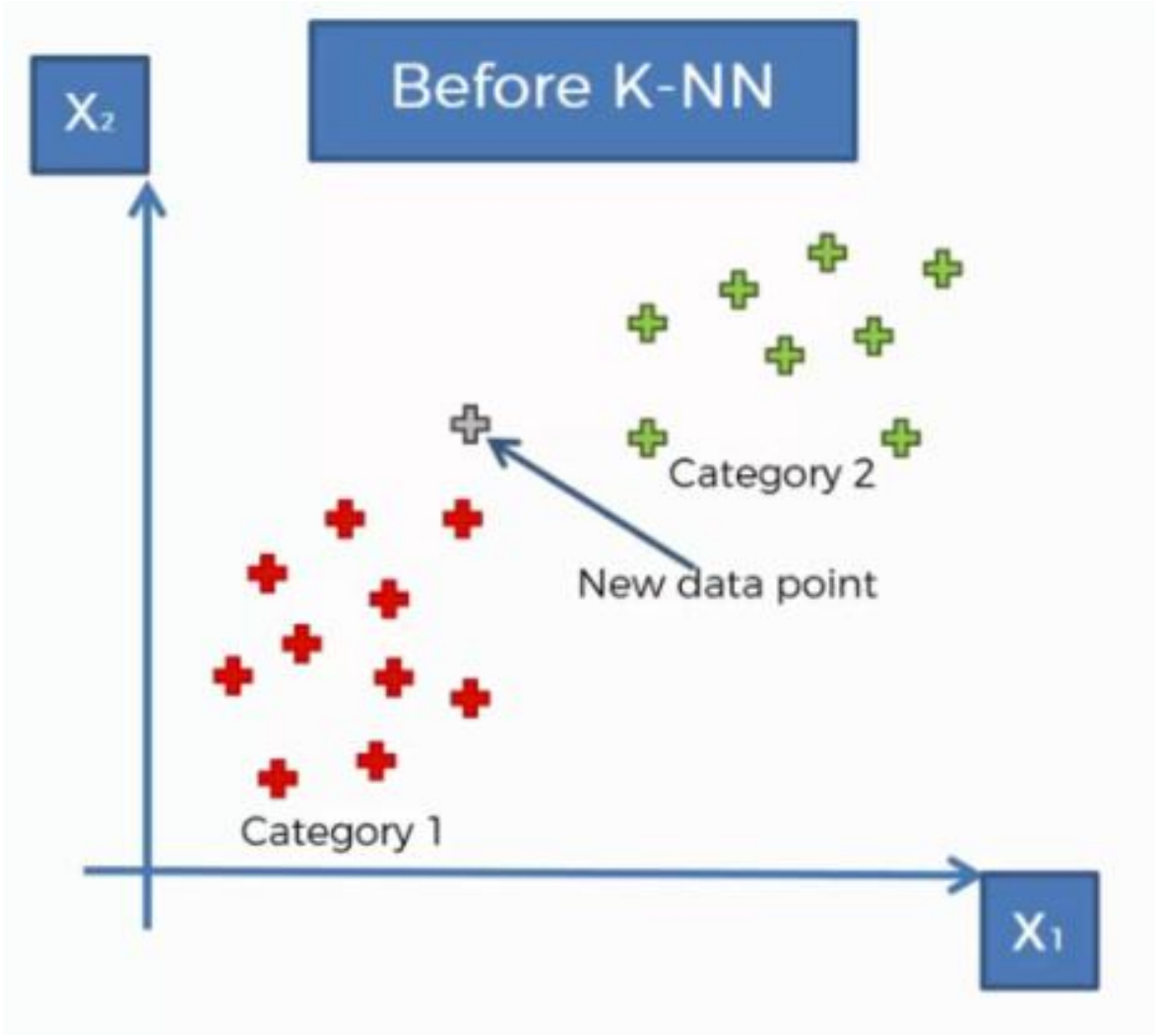


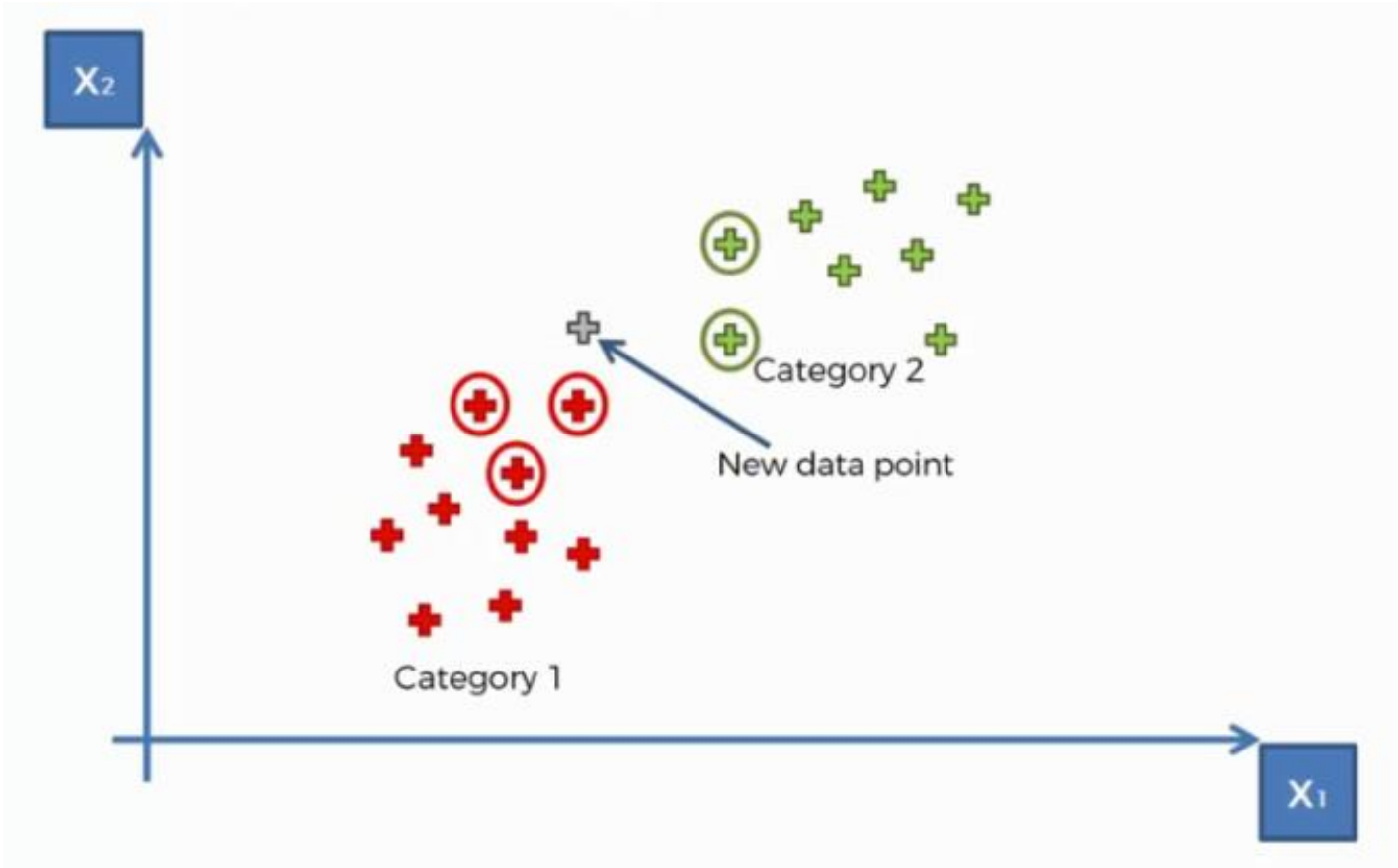
Feature Scaling

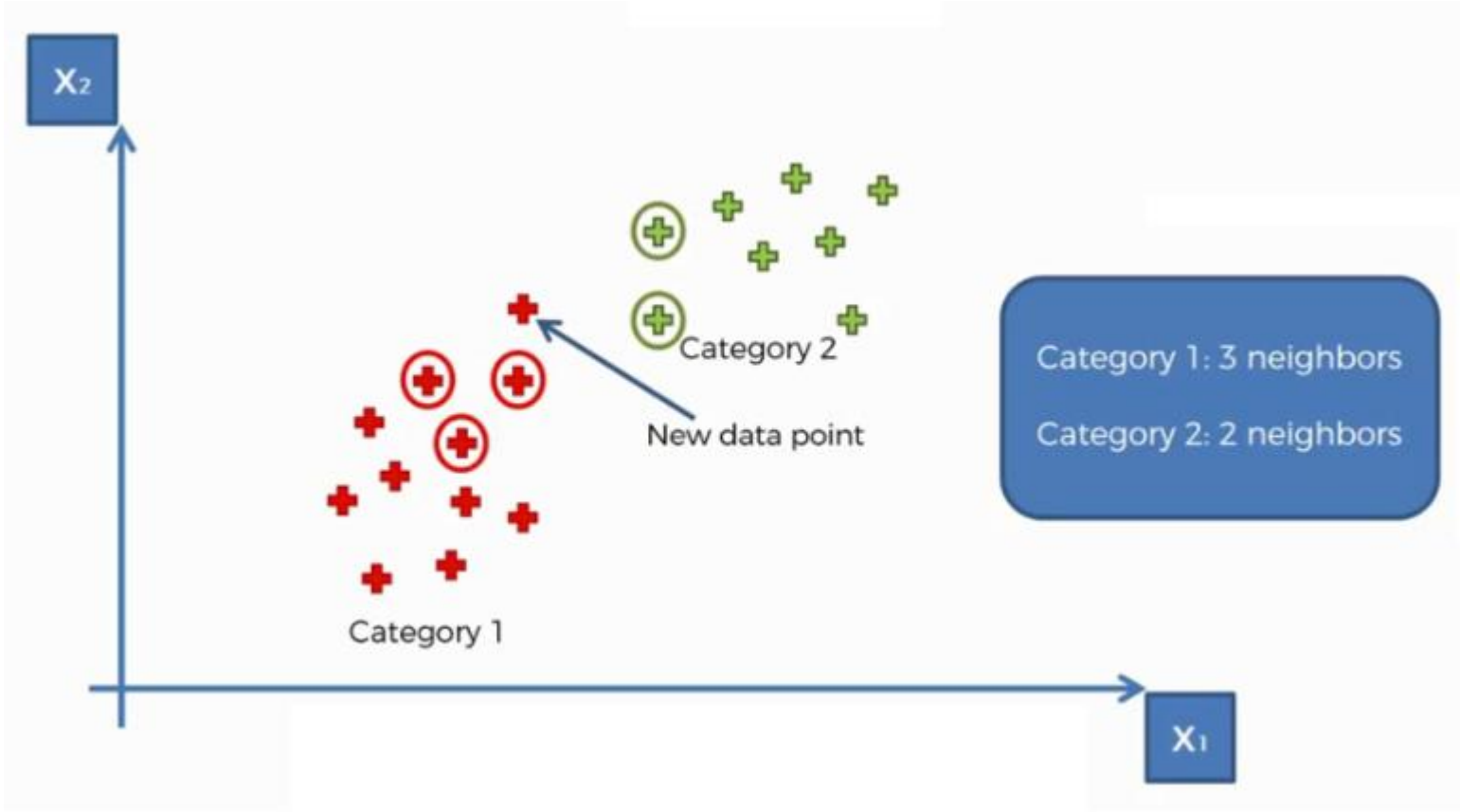
- StandardScaler will **transform** the data such that its distribution will have a **mean value 0** and **standard deviation of 1**
- This is useful while **comparing data** that corresponds to **different units**. In that case, we want to **remove the units**.
- This is done in a consistent way for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

KNN

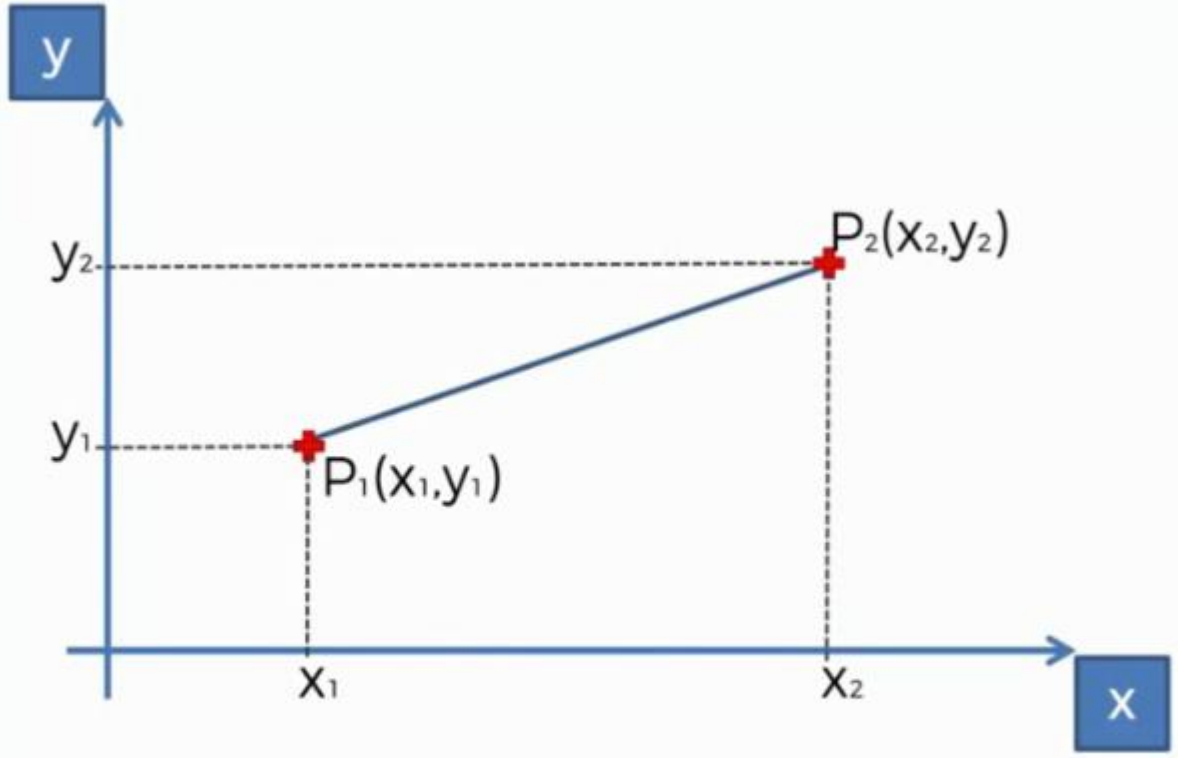



Before K-NN
X2
Category 2
New data point
Category 1
X1


X2
Category 2
New data point
Category 1
X1

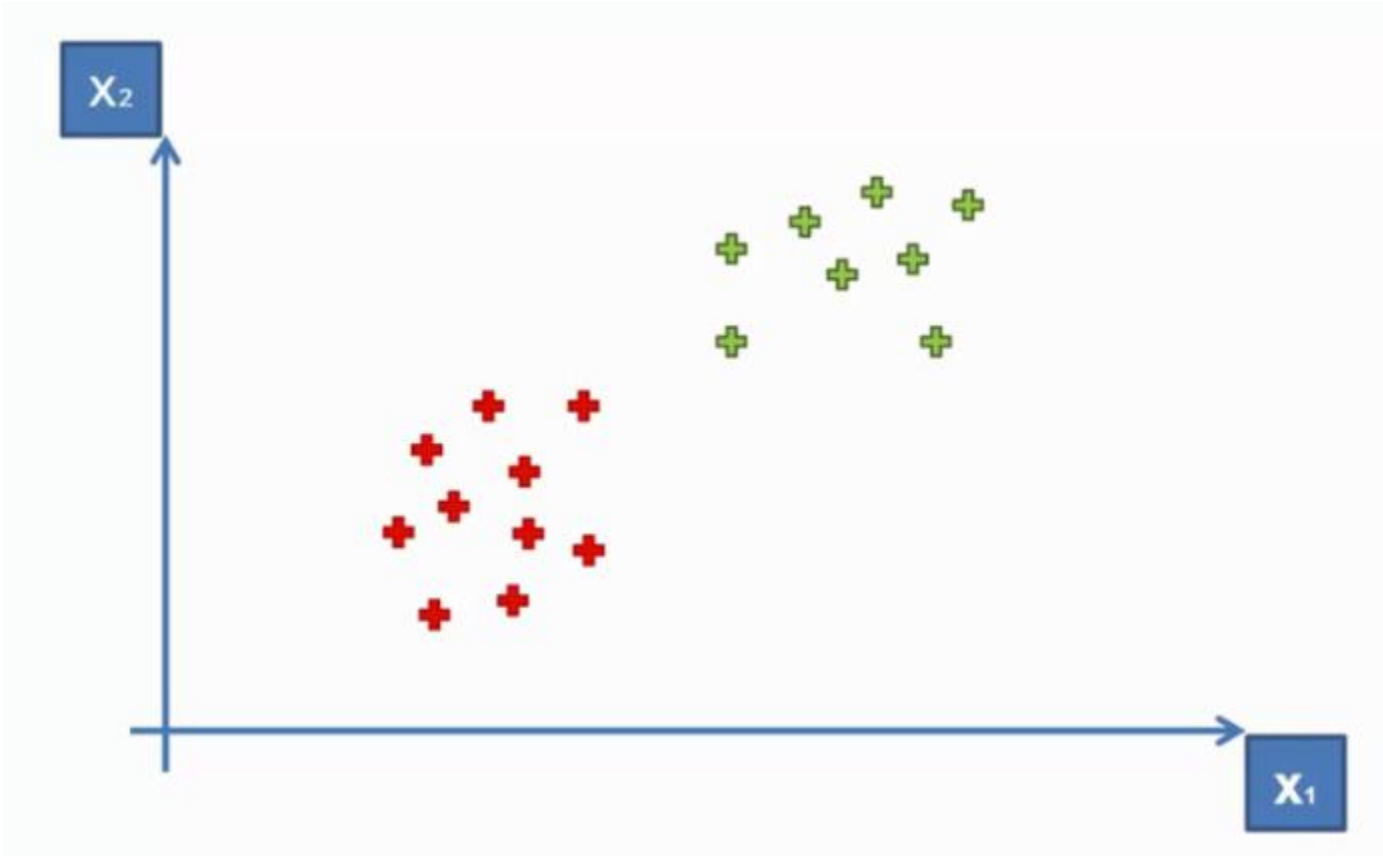

X2
Category 2
New data point
Category 1: 3 neighbors
Category 2: 2 neighbors
Category 1
X1

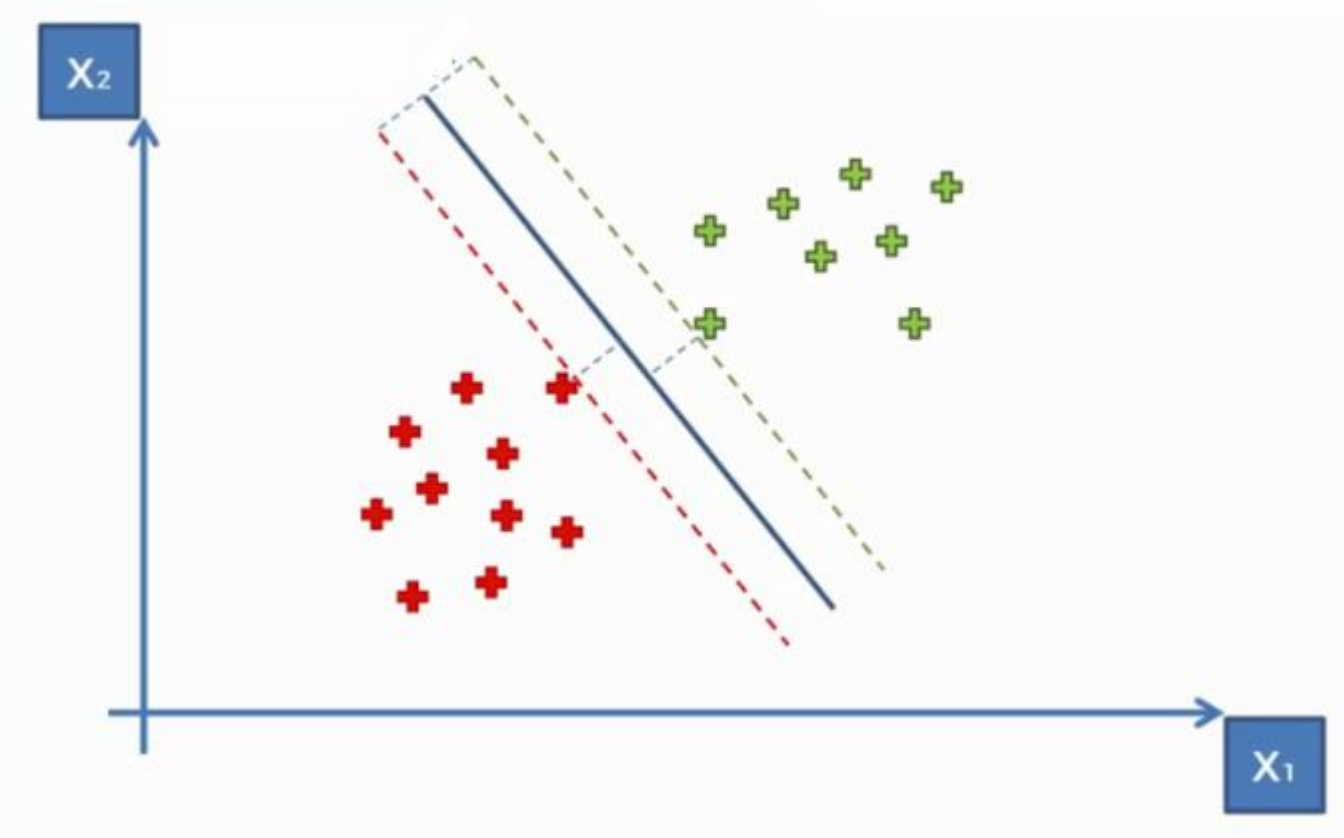
Euclidean Distance

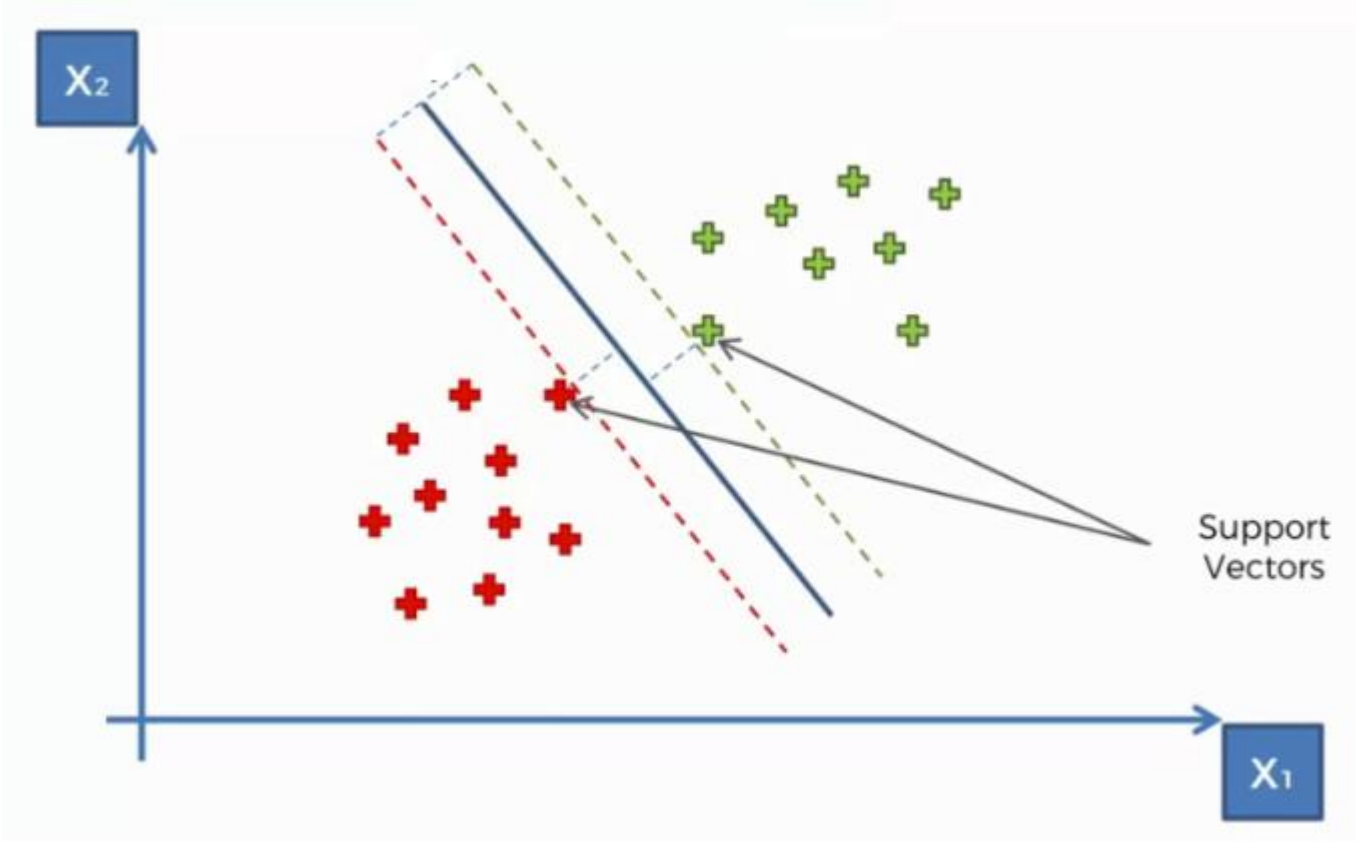


Euclidean Distance between P_1 and $P_2 = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$

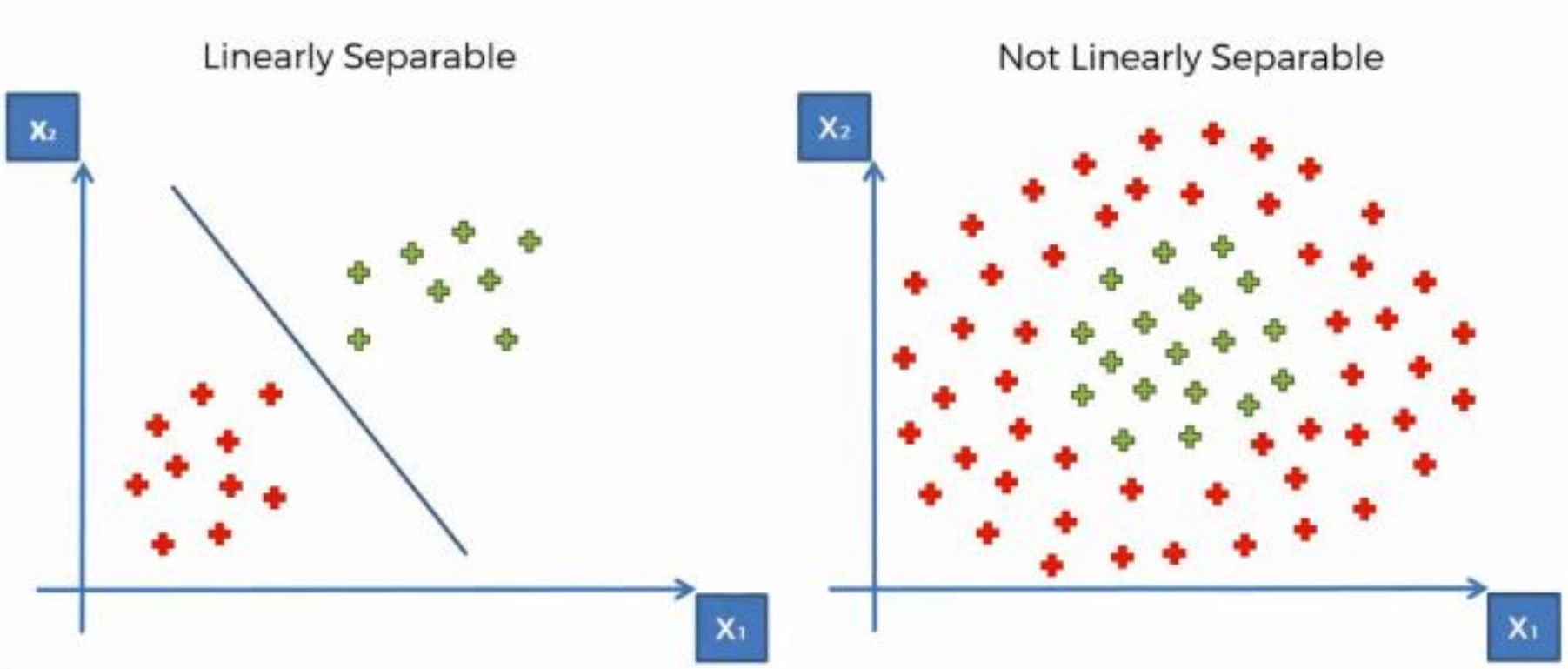
Support Vector Machine

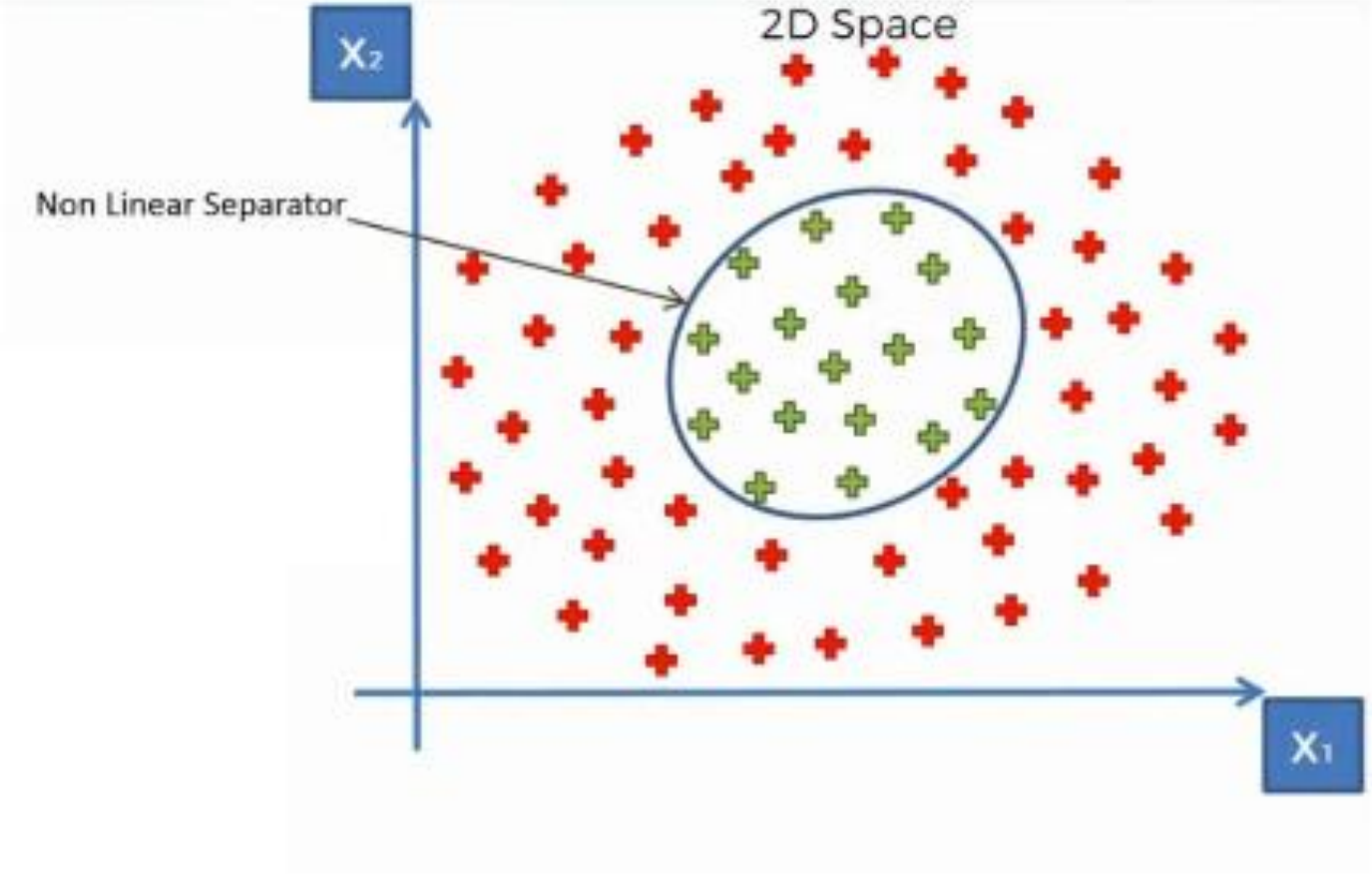



X2
X1


X2
Support
Vectors
X1

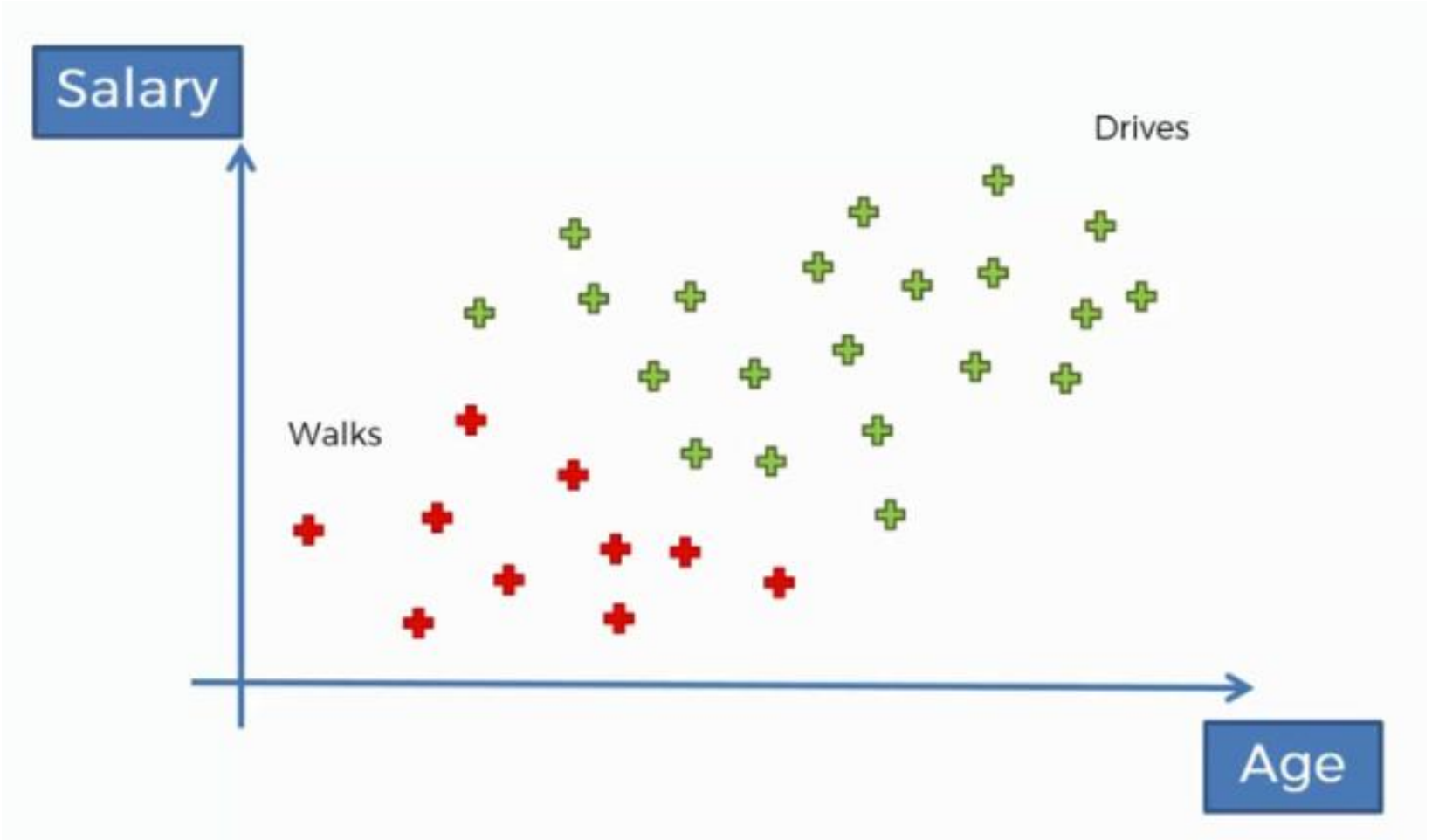
Kernel SVM

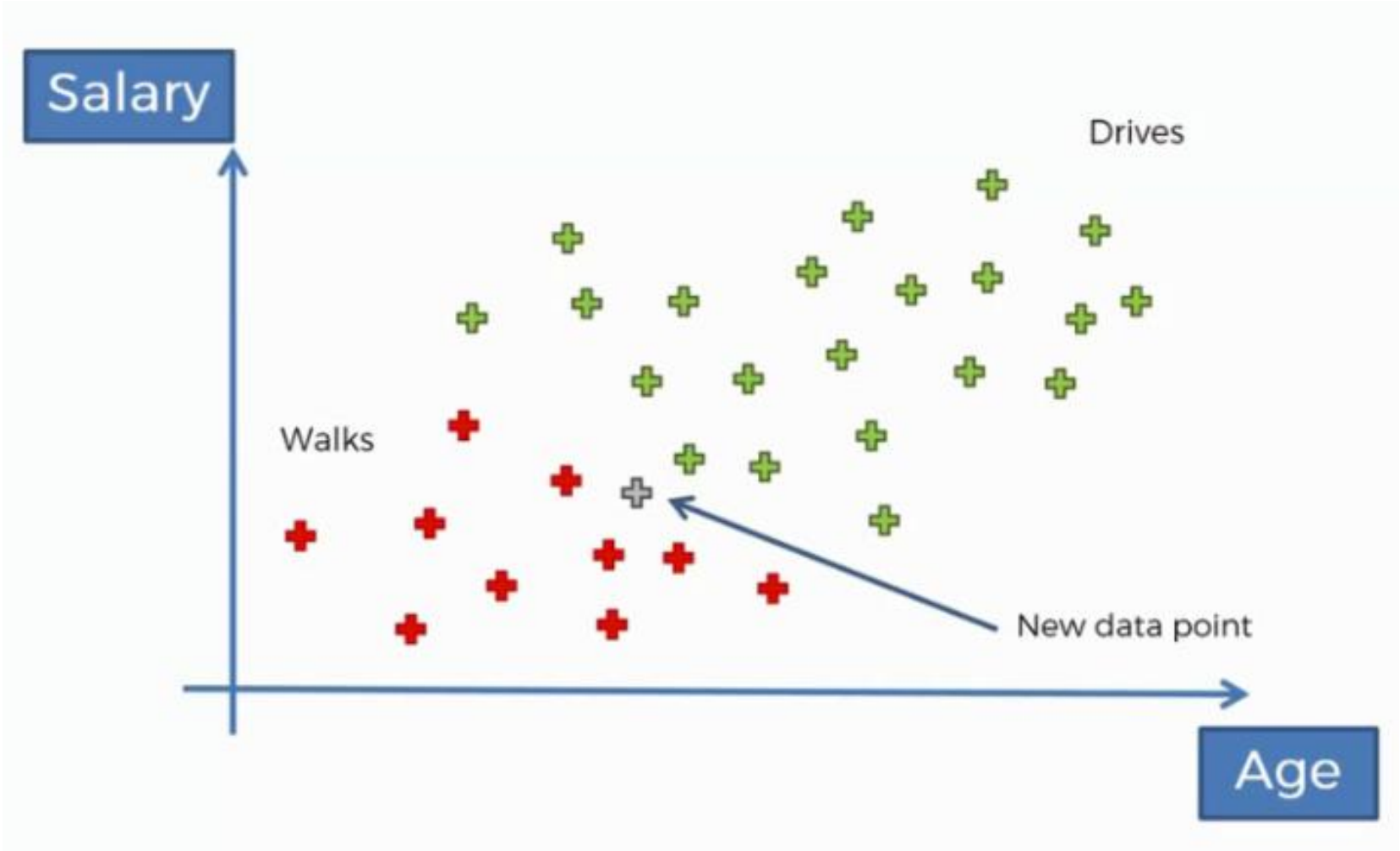



2D Space
X2
Non Linear Separator
X1

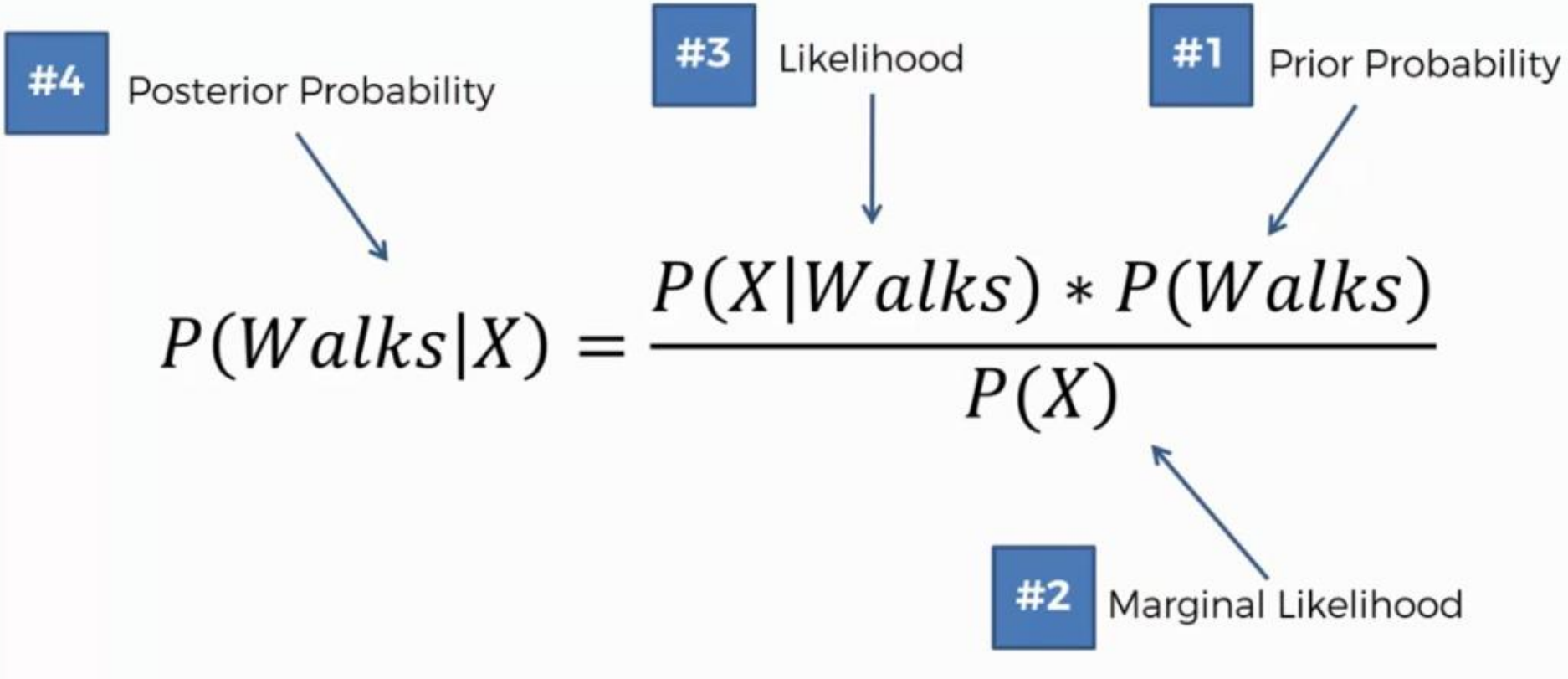
Naive Bayes

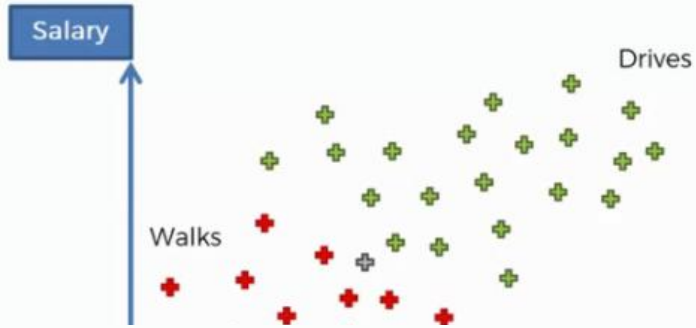
$$P(A|B) = \frac{P(B|A) * P(A)}{P(B)}$$


Salary
Drives
Walks
Age


Salary
Drives
Walks
New data point
Age

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

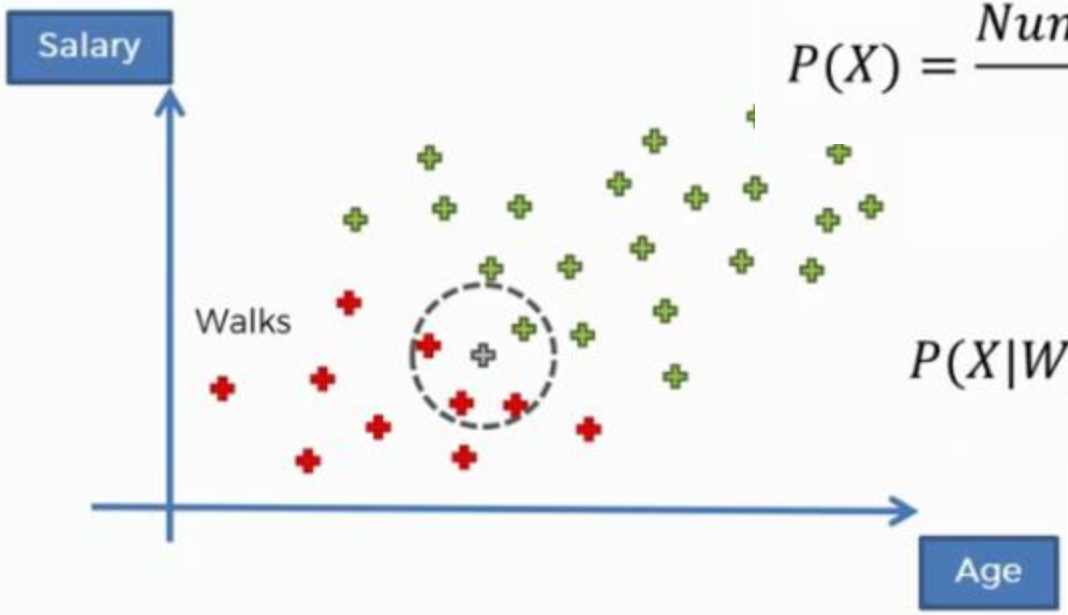

#4
Posterior Probability
#3
Likelihood
#1
Prior Probability
$P(Walks|X) = \frac{P(X|Walks) * P(Walks)}{P(X)}$
#2
Marginal Likelihood



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

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

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



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

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

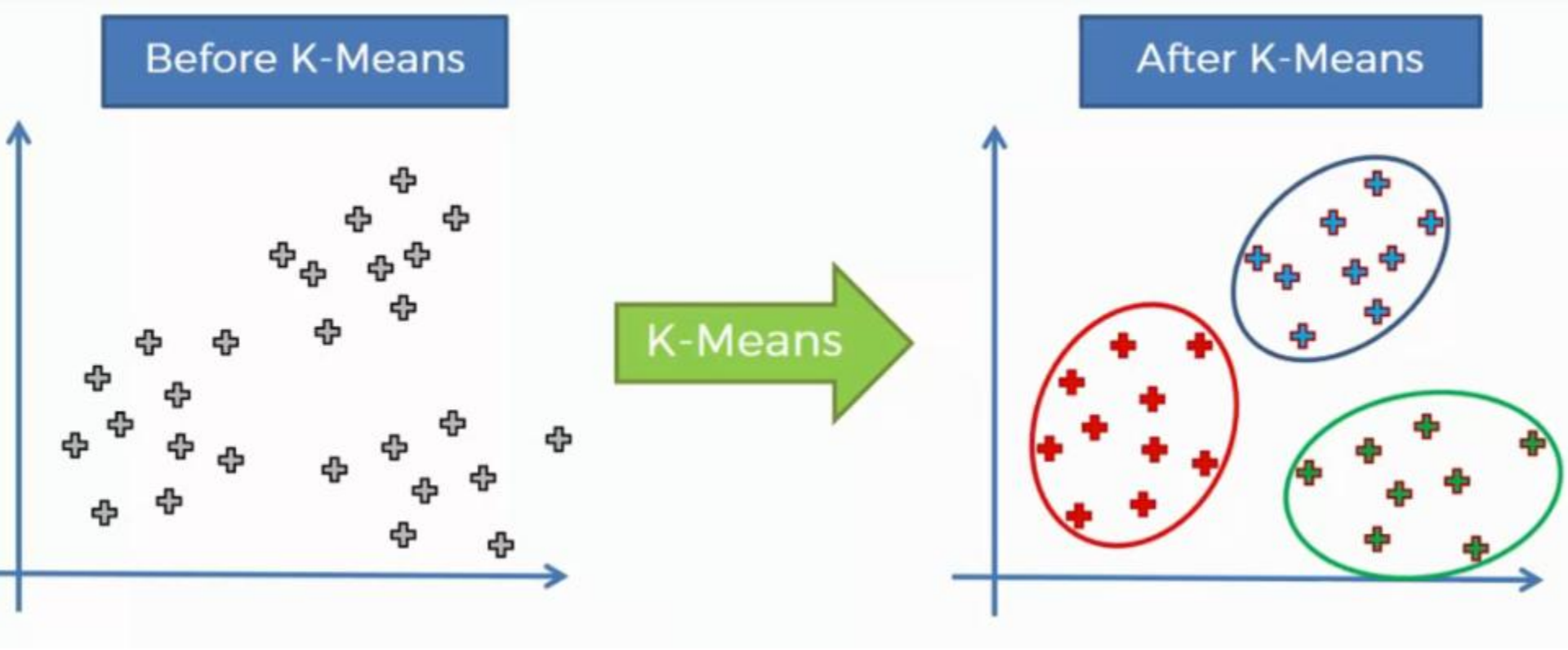
$$P(Walks|X) = \frac{\frac{3}{10} * \frac{10}{30}}{\frac{4}{30}} = 0.75$$

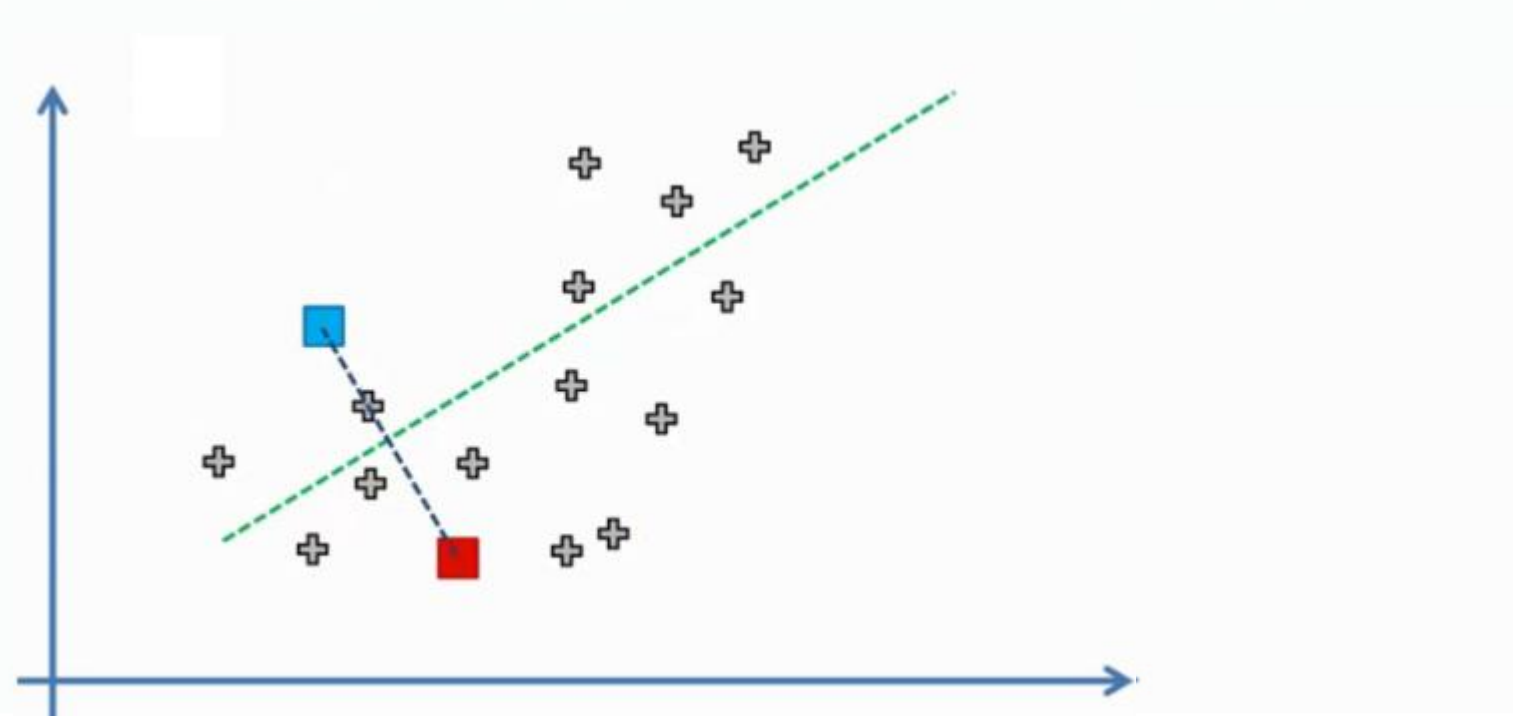
$$P(Walks|X) \; v.s. \; P(Drives|X)$$

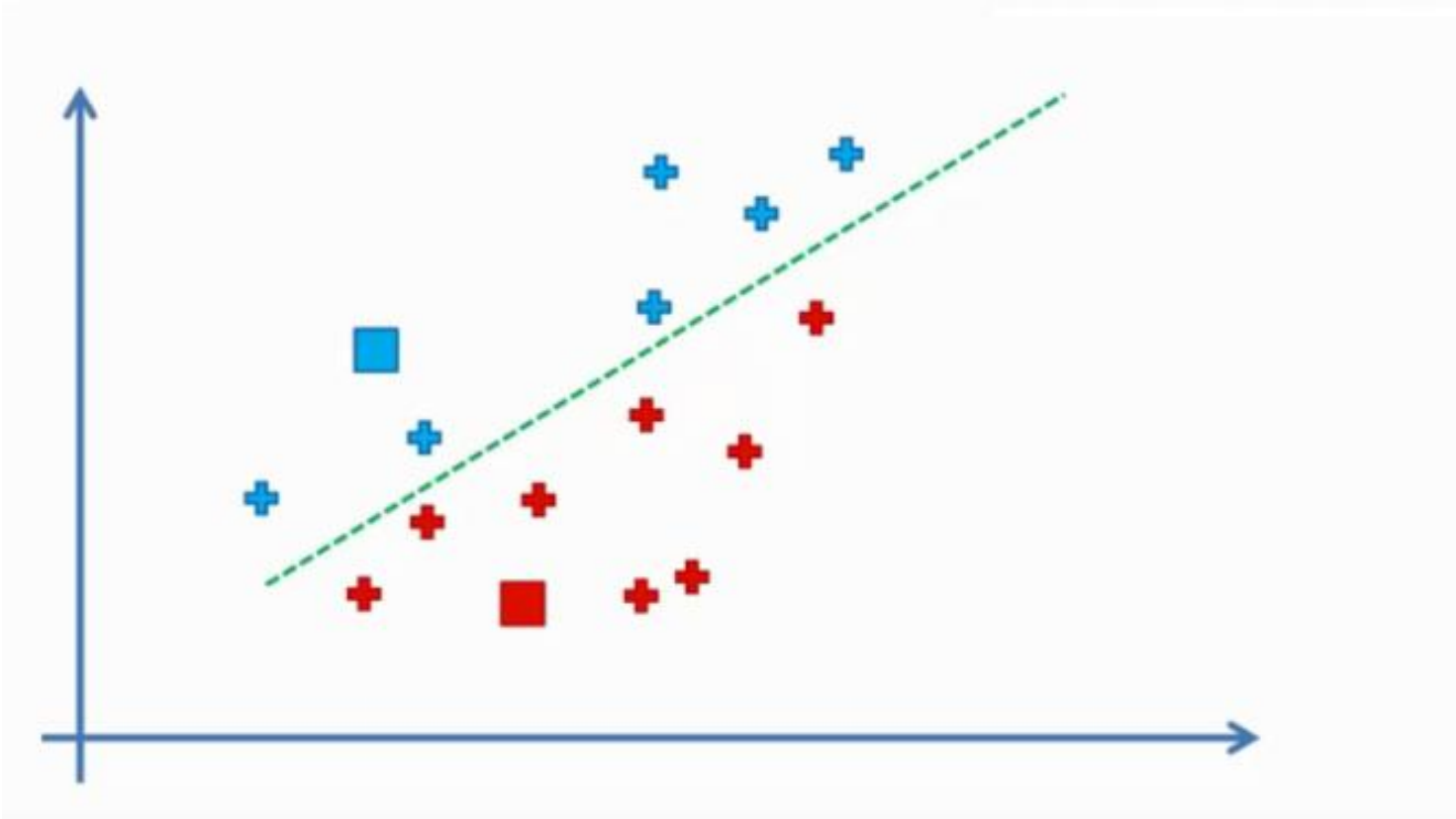
$$P(Drives|X) = \frac{\frac{1}{20} * \frac{20}{30}}{\frac{4}{30}} = 0.25$$

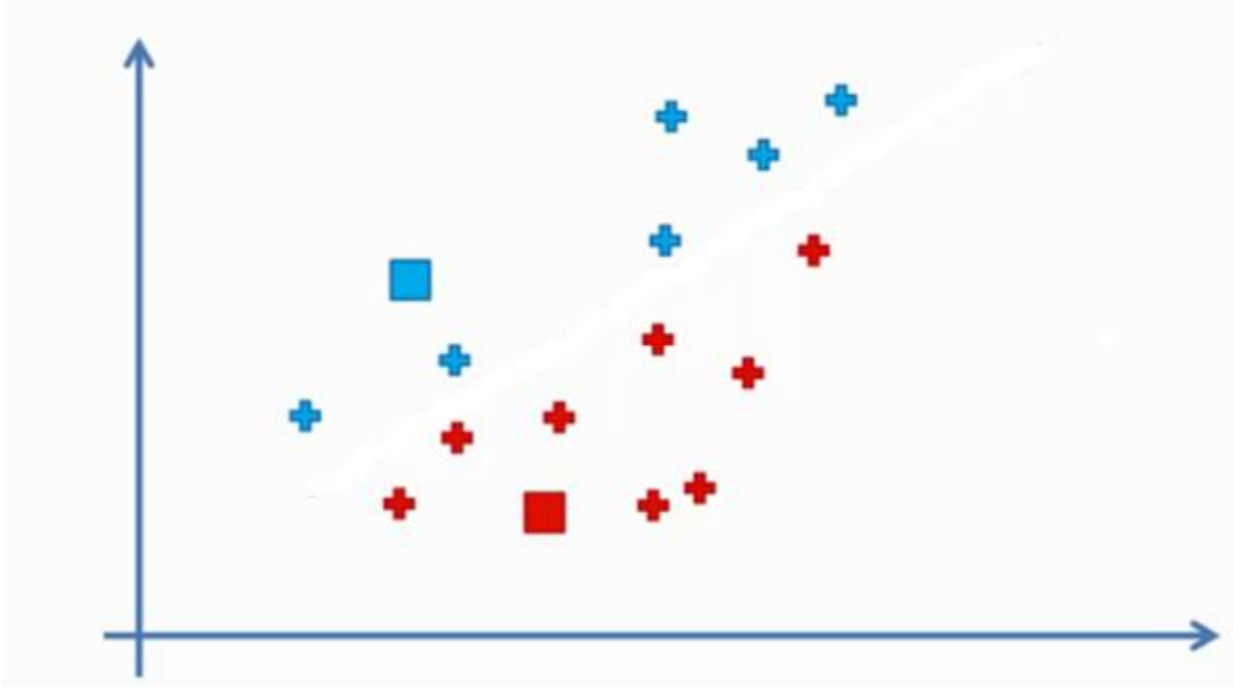
Clustering
Unsupervised

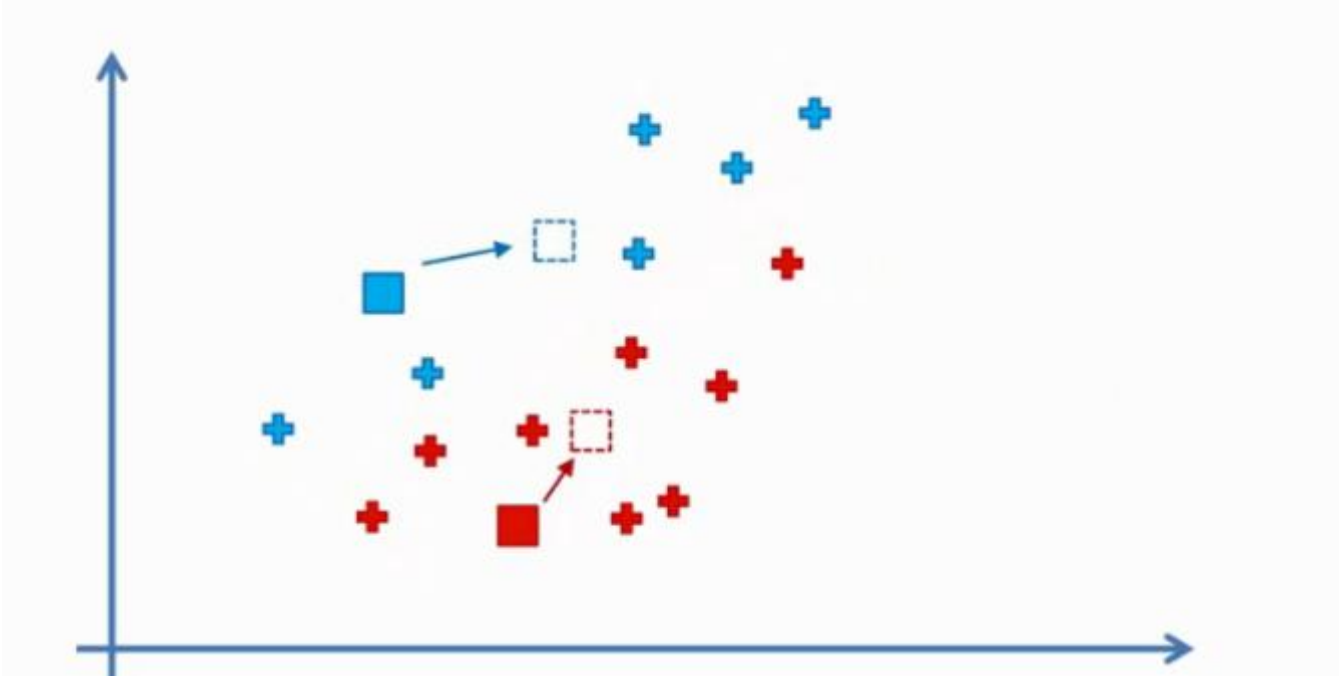
K-means

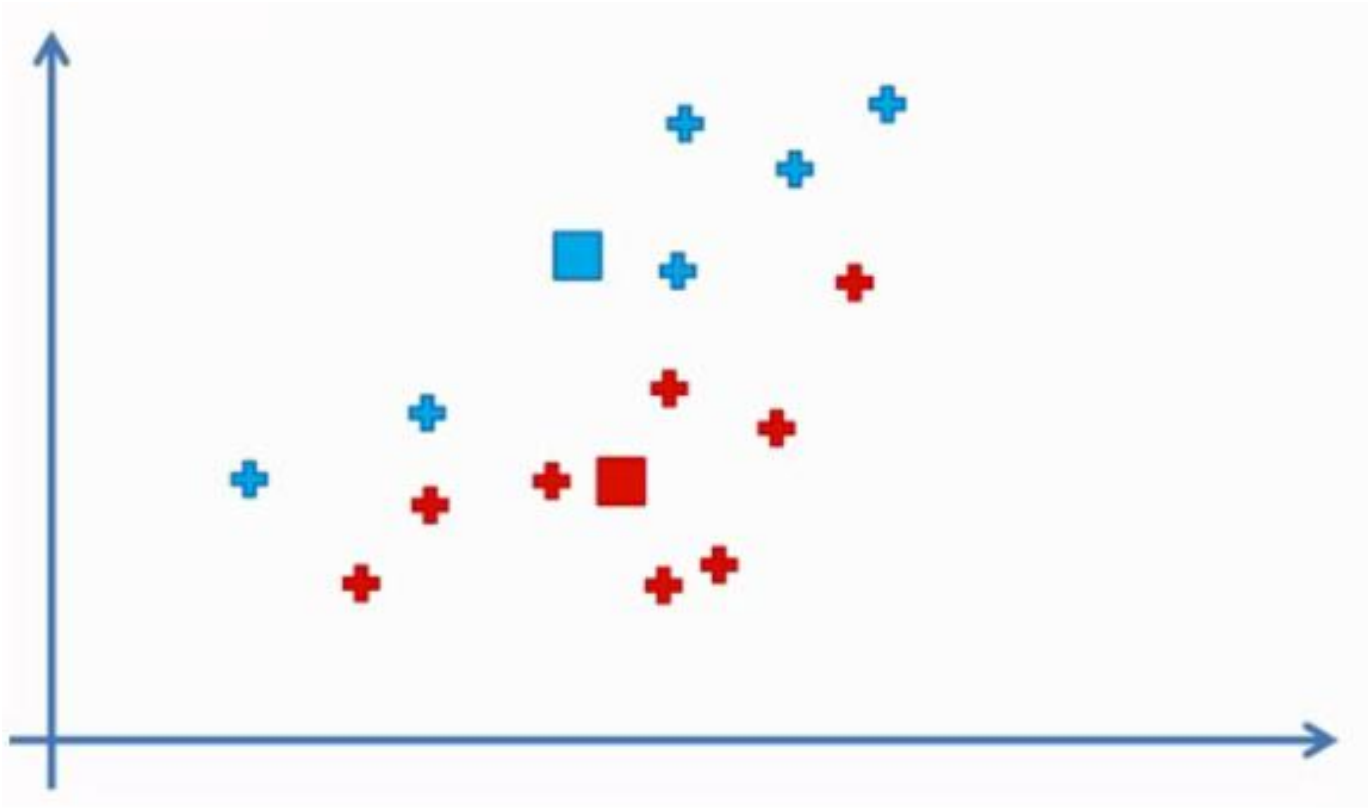


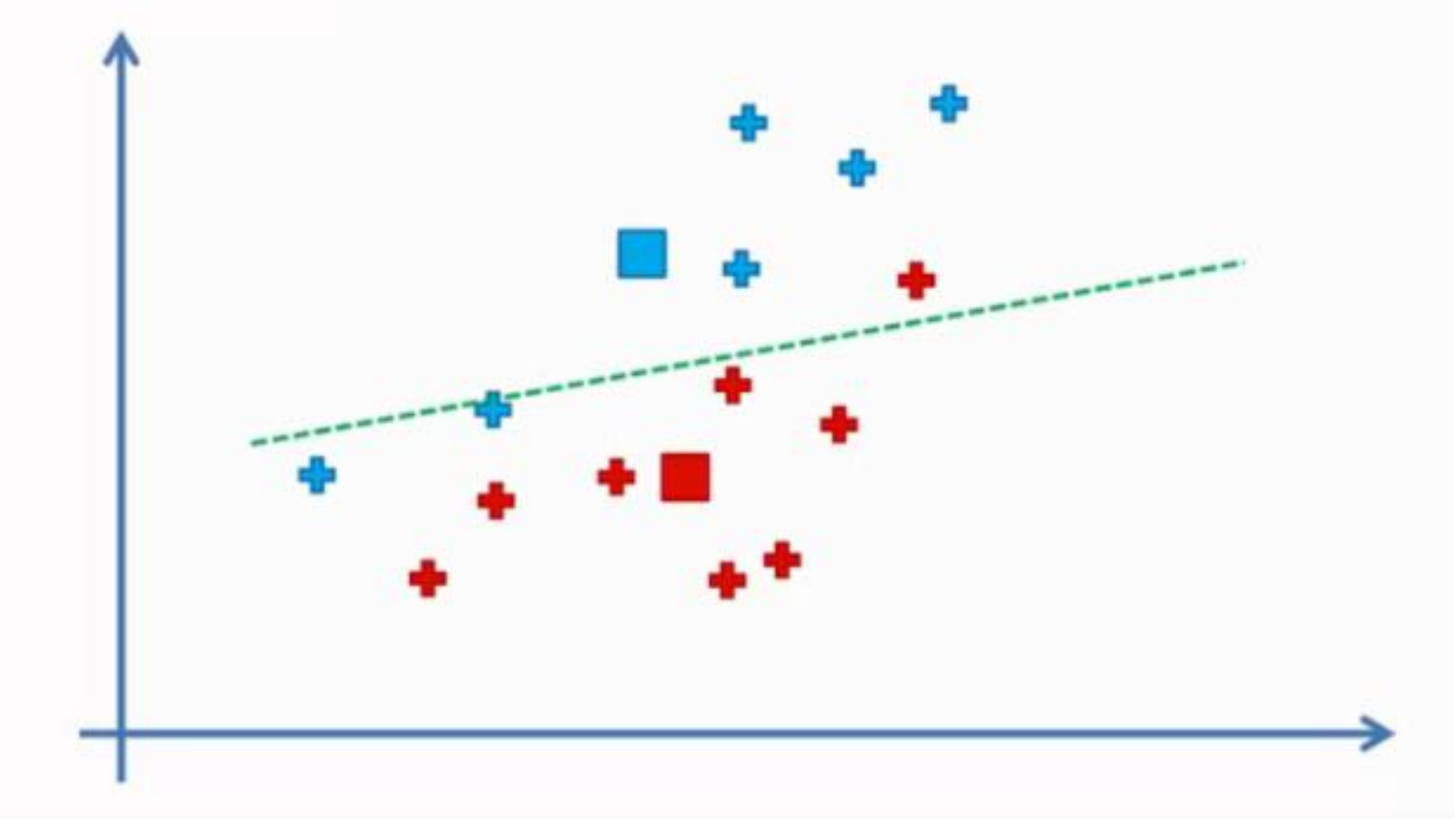


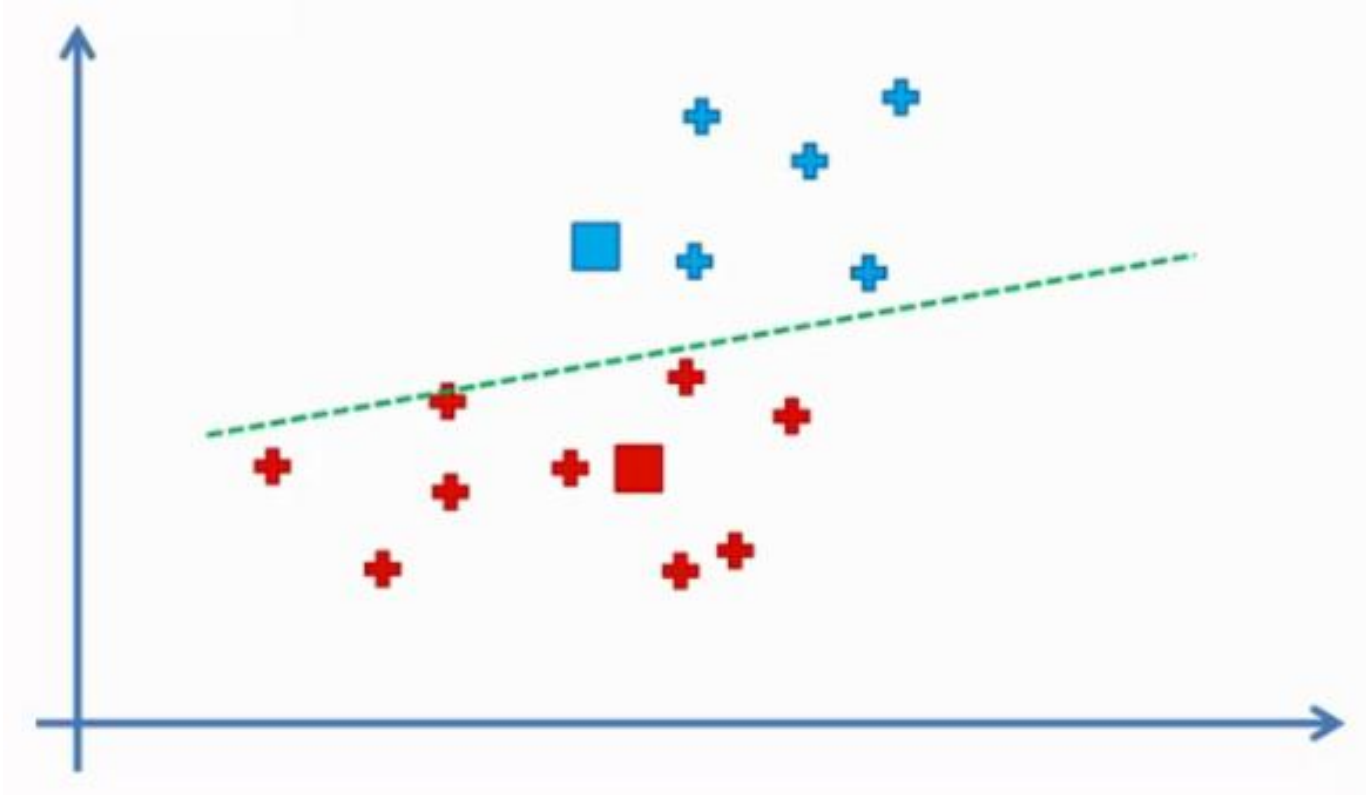


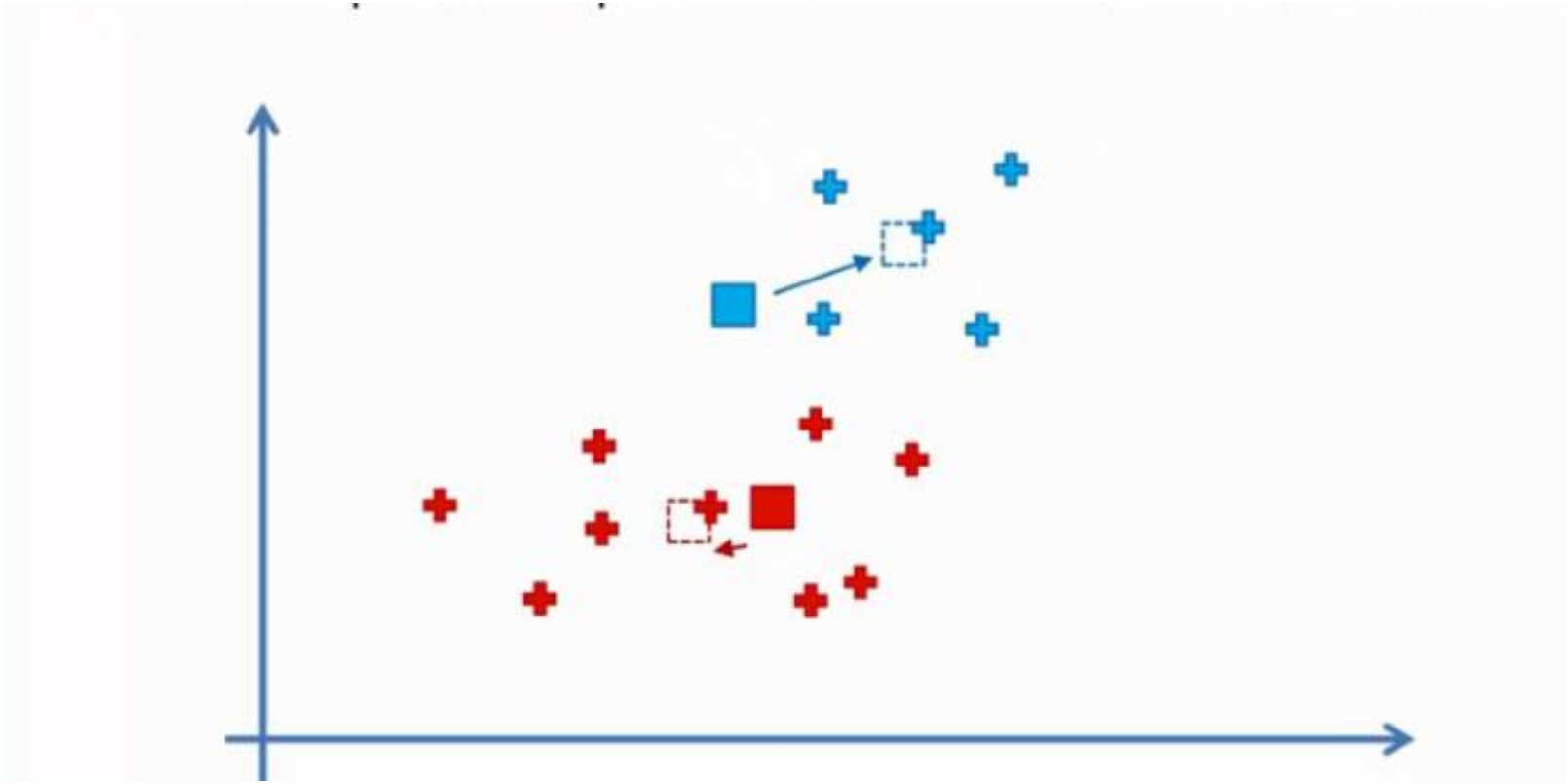


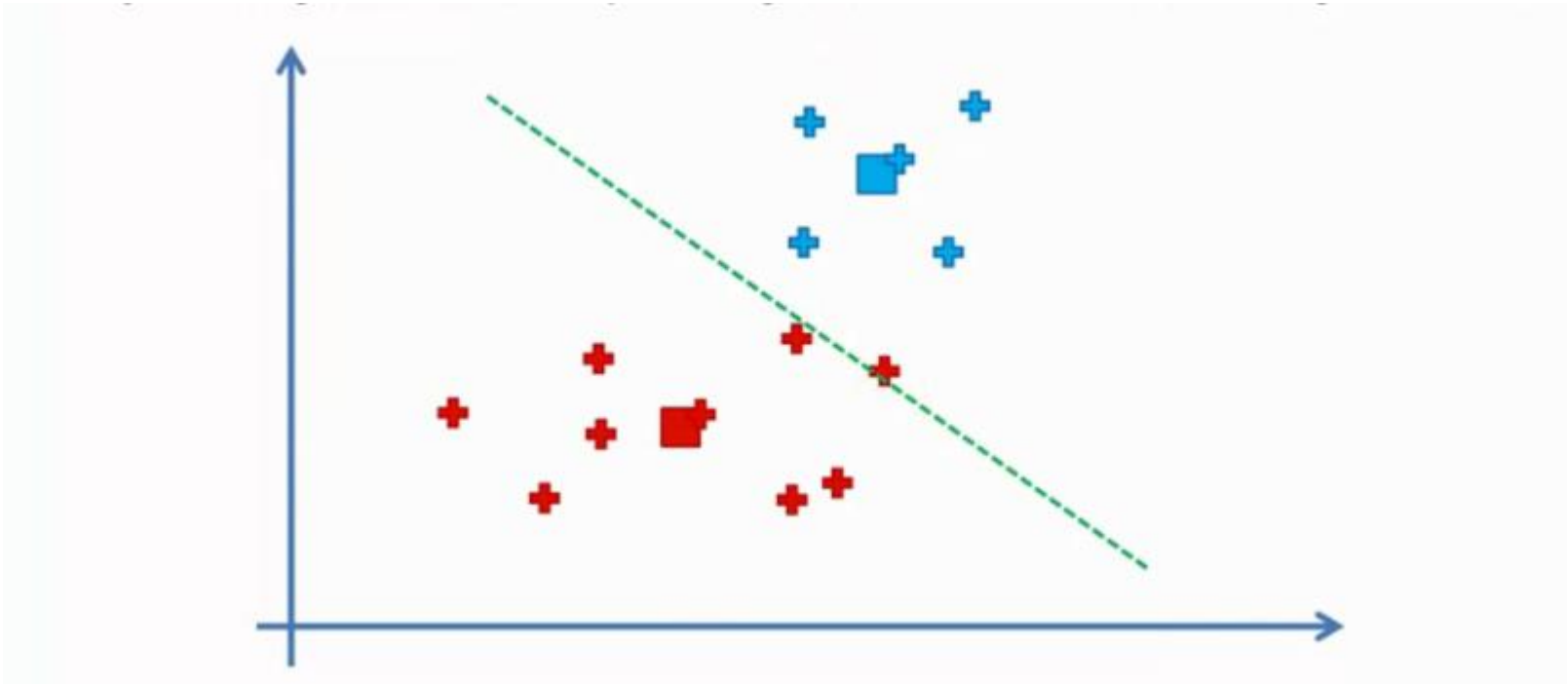


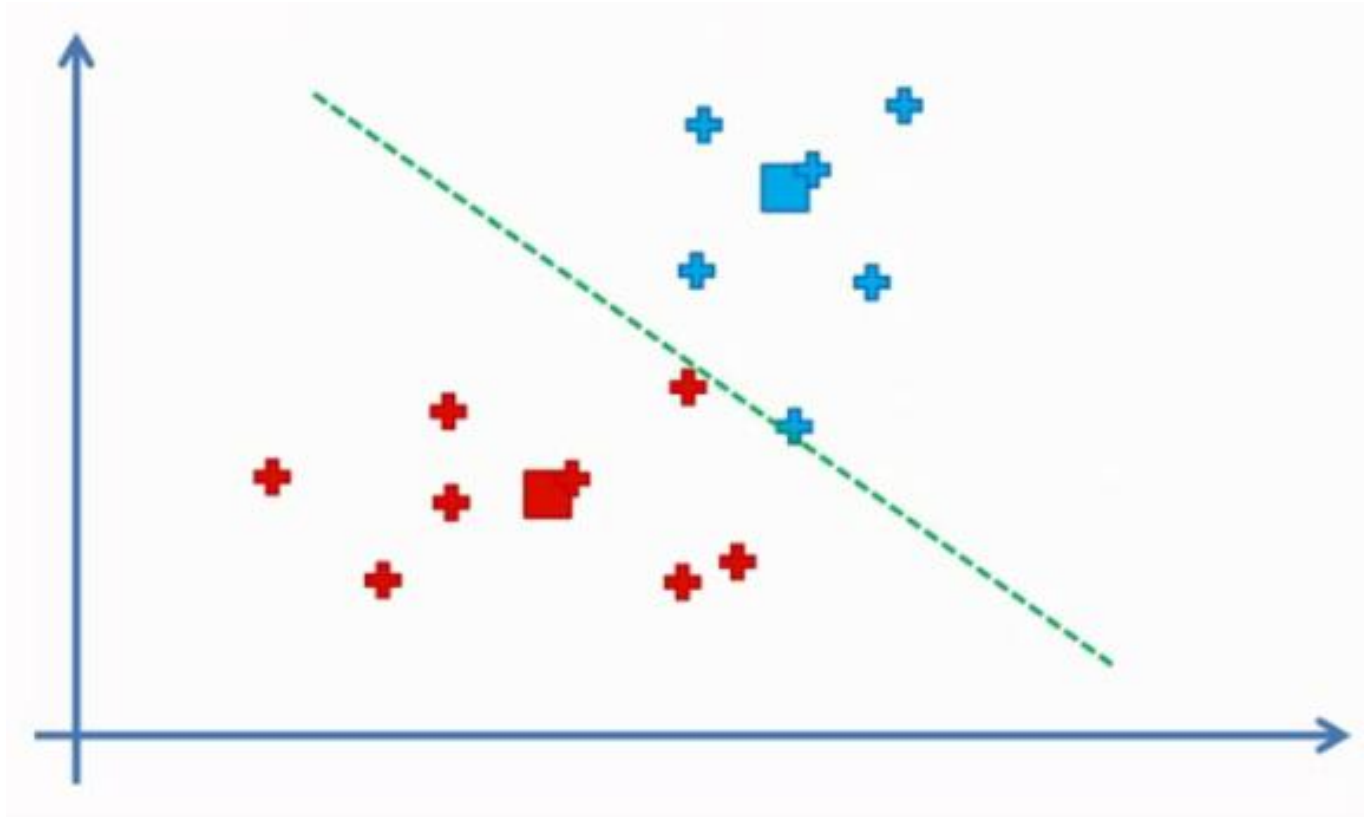


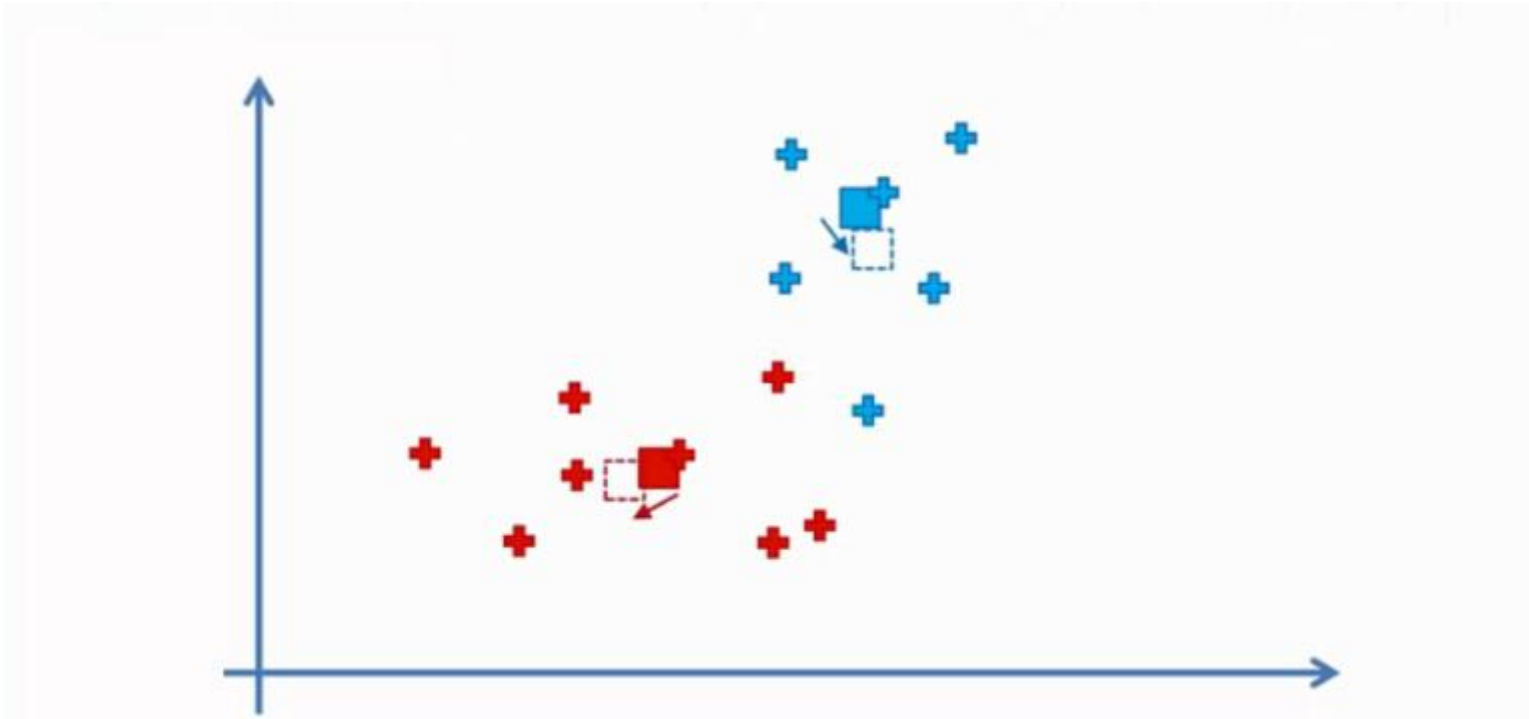


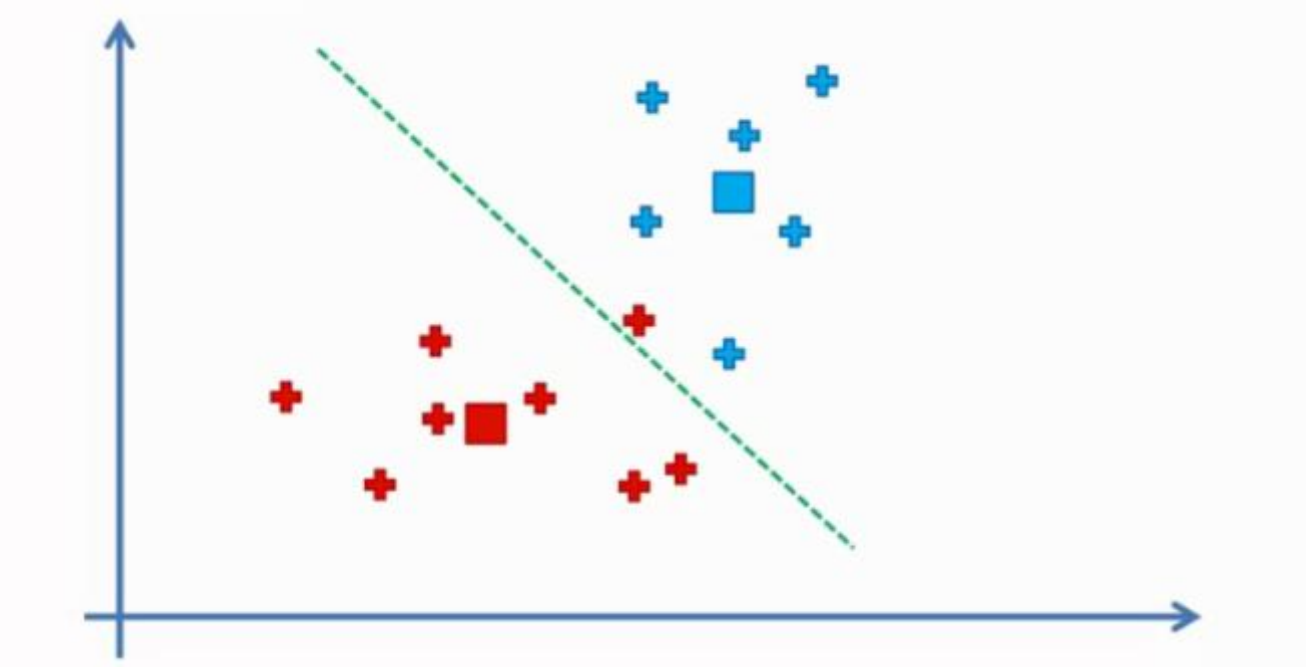


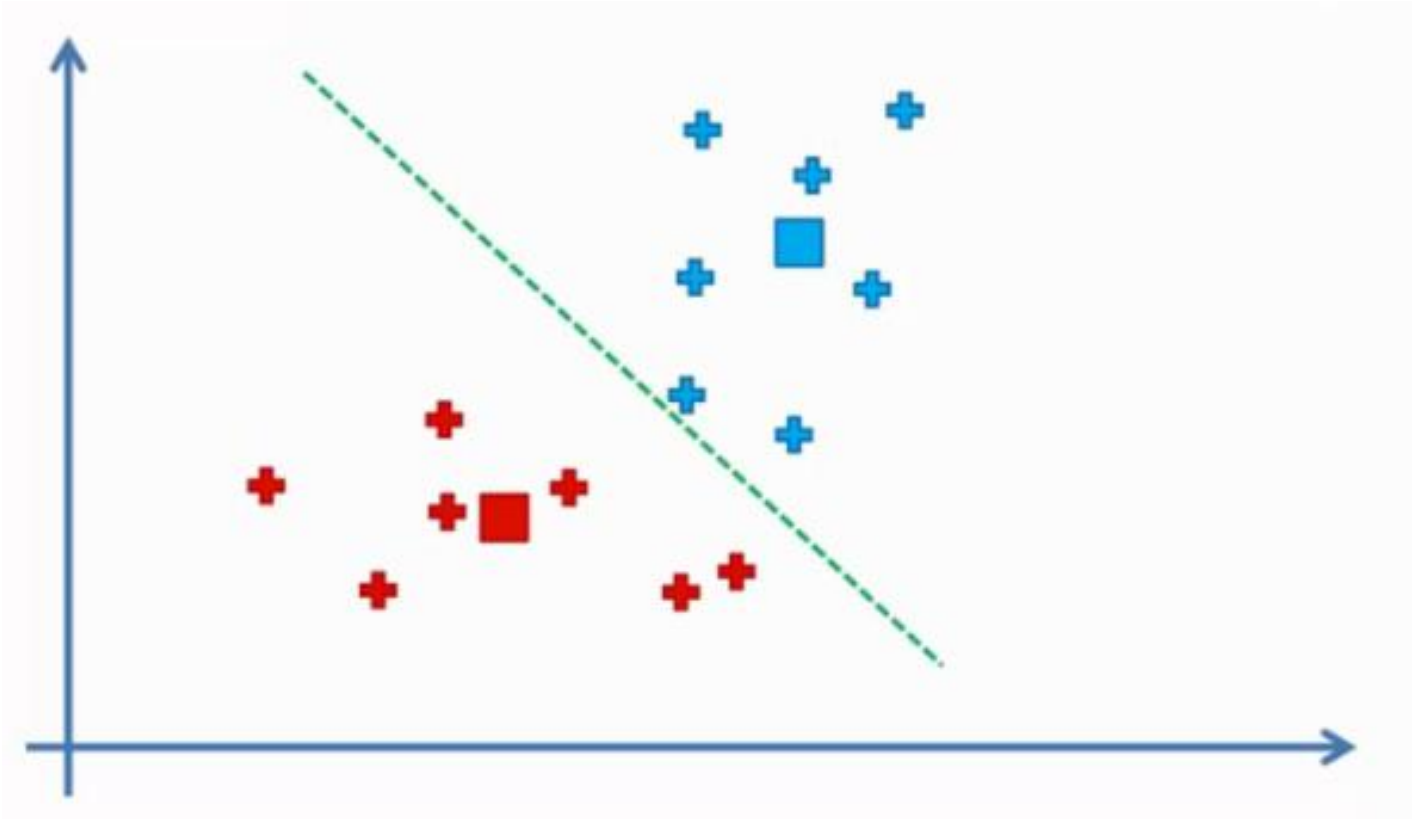


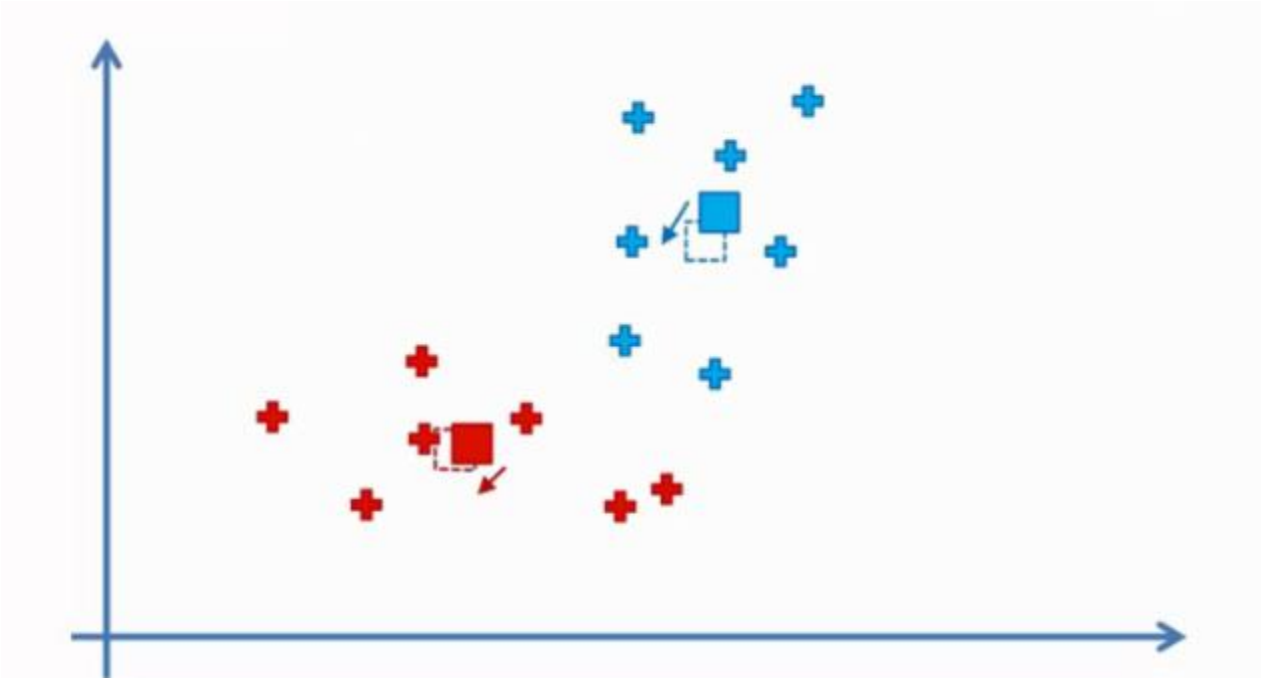


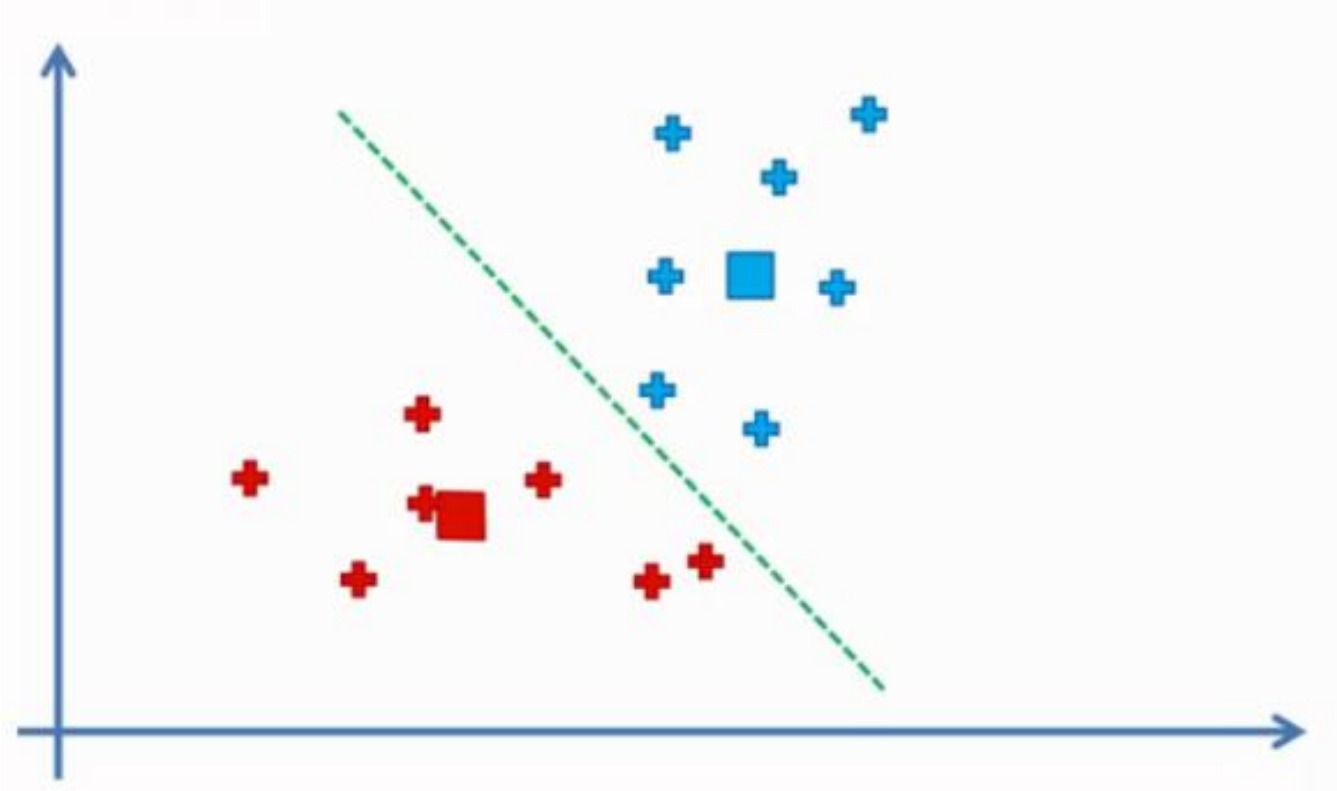


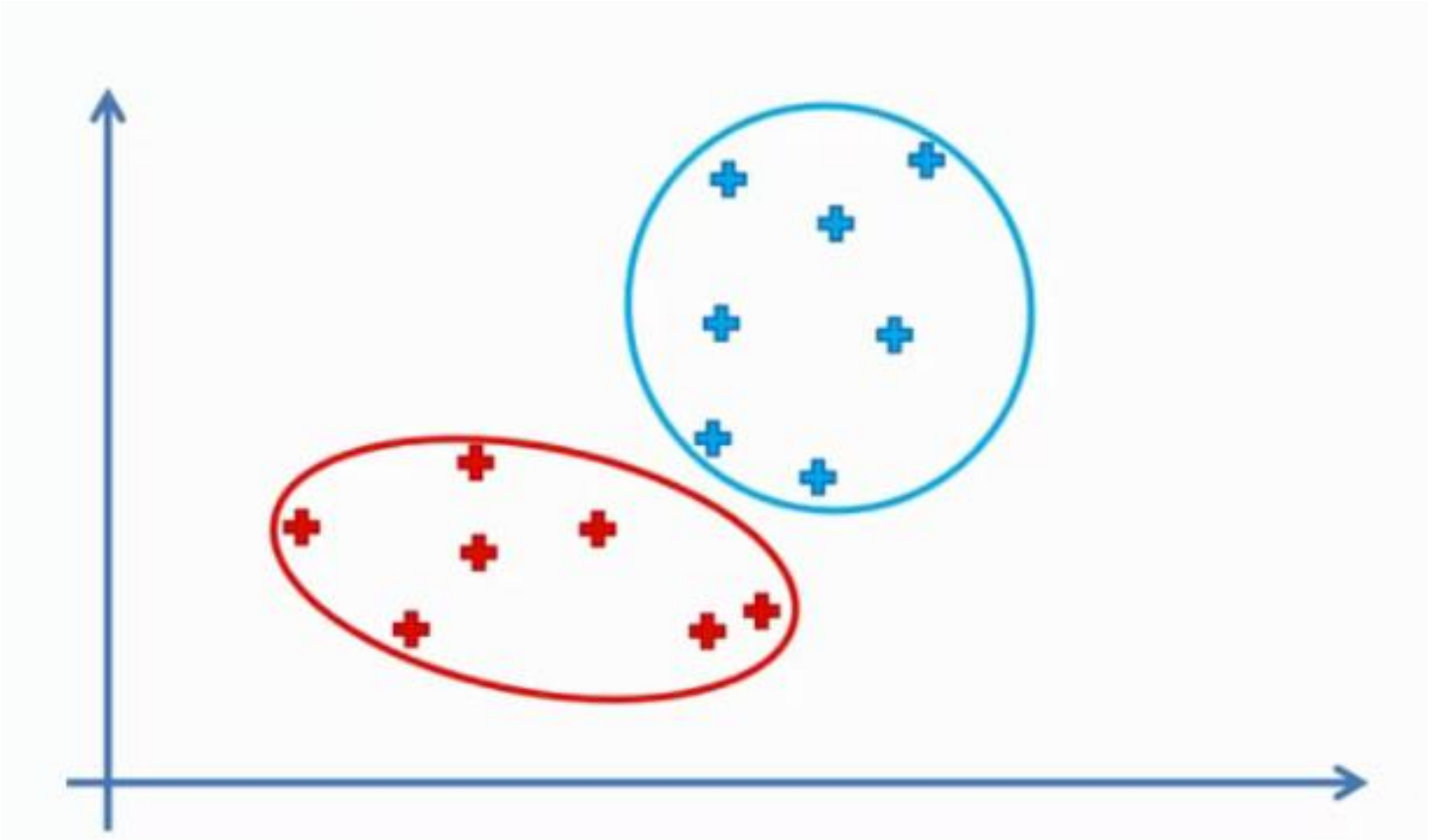


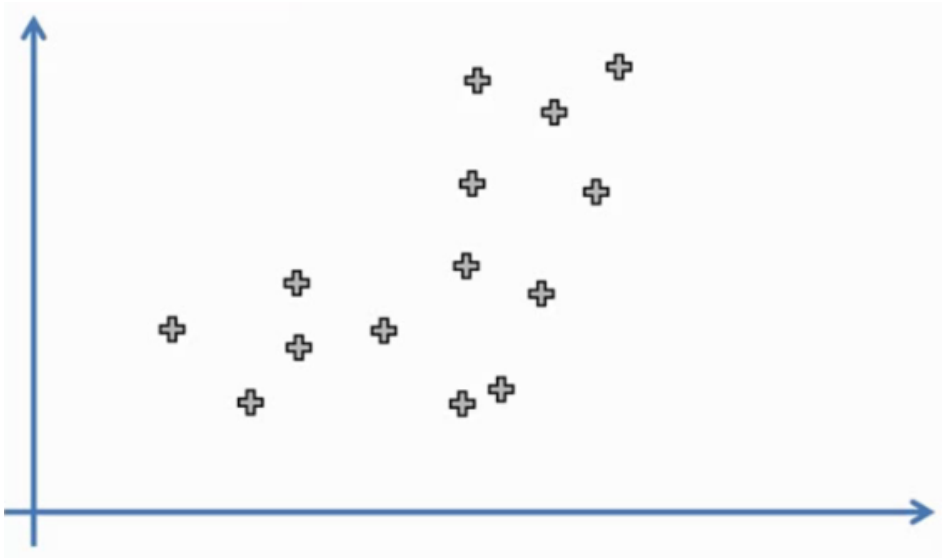


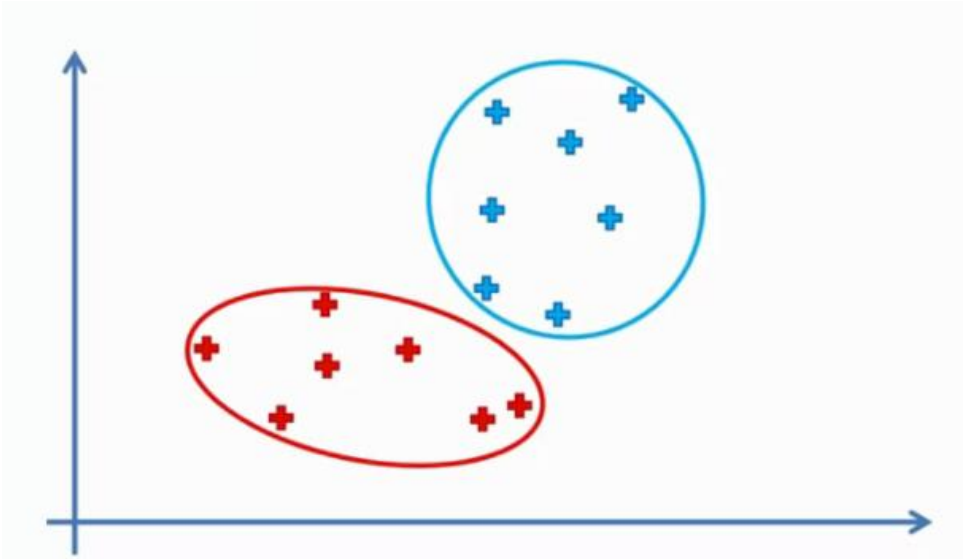












Upper Confidence Bound

- 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

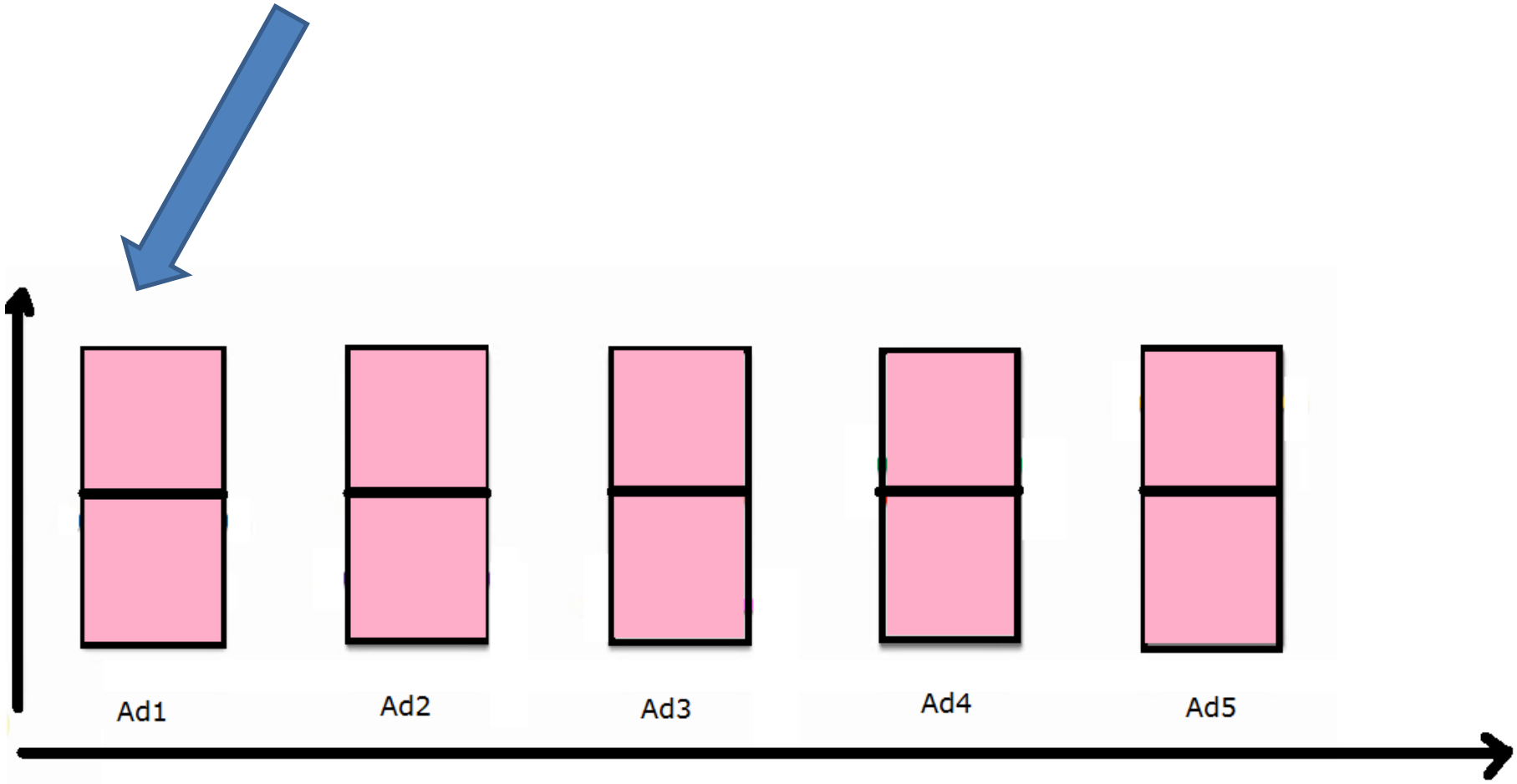
$$\bar{r}_i(n) = \frac{R_i(n)}{N_i(n)}$$

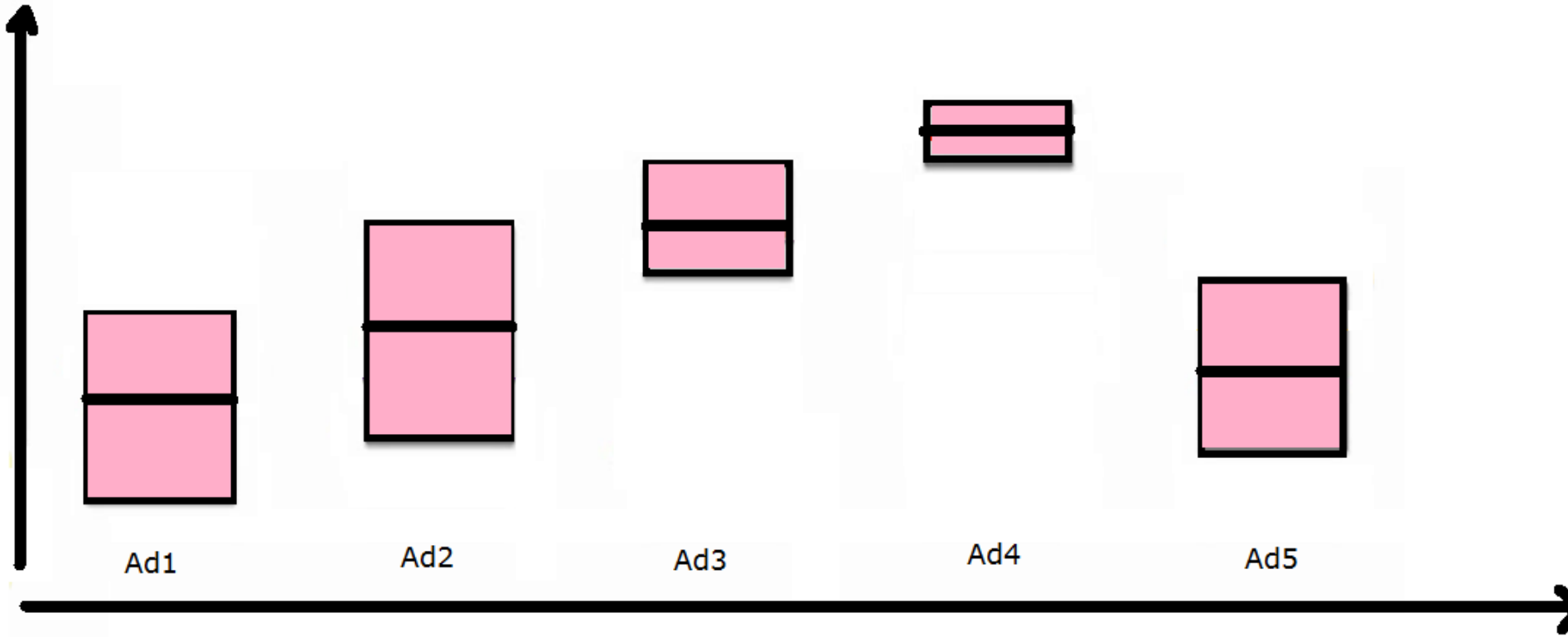
- UCB $\bar{r}_i(n) + \Delta_i(n)$

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

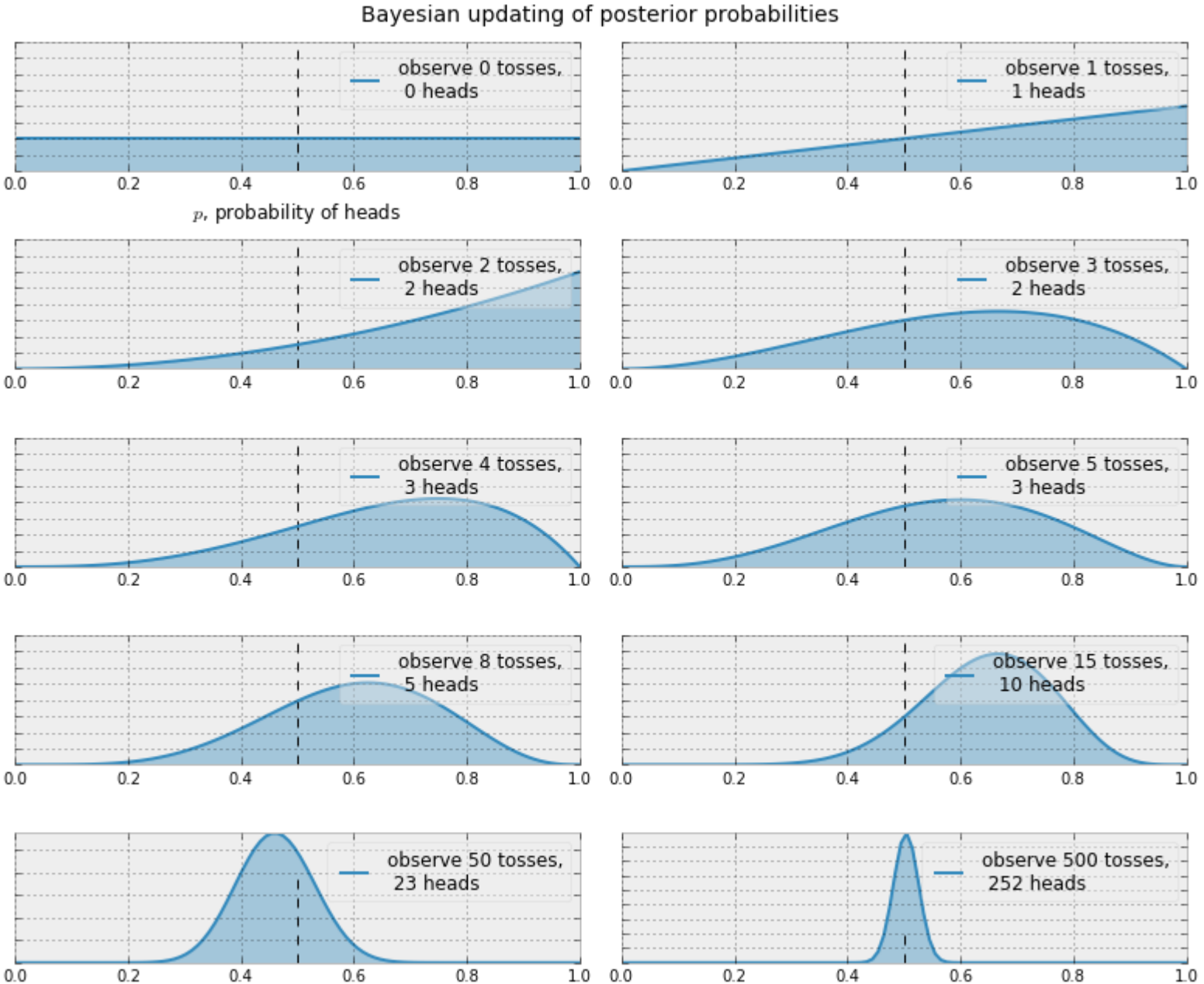
Step 3. We select the ad i that has the maximum UCB $\bar{r}_i(n) + \Delta_i(n)$.

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 $\theta_i(n)$.

Aprior

User ID	Movies liked
46578	Movie1, Movie2, Movie3, Movie4
98989	Movie1, Movie2
71527	Movie1, Movie2, Movie4
78981	Movie1, Movie2
89192	Movie2, Movie4
61557	Movie1, Movie3

Potential Rules:

Movie1 ⇨ Movie2

Movie2 ⇨ Movie4

Movie1 ⇨ Movie3

Transaction ID	Products purchased
46578	Burgers, French Fries, Vegetables
98989	Burgers, French Fries, Ketchup
71527	Vegetables, Fruits
78981	Pasta, Fruits, Butter, Vegetables
89192	Burgers, Pasta, French Fries
61557	Fruits, Orange Juice, Vegetables
87923	Burgers, French Fries, Ketchup, Mayo

Potential Rules:

Burgers ⇨ French Fries

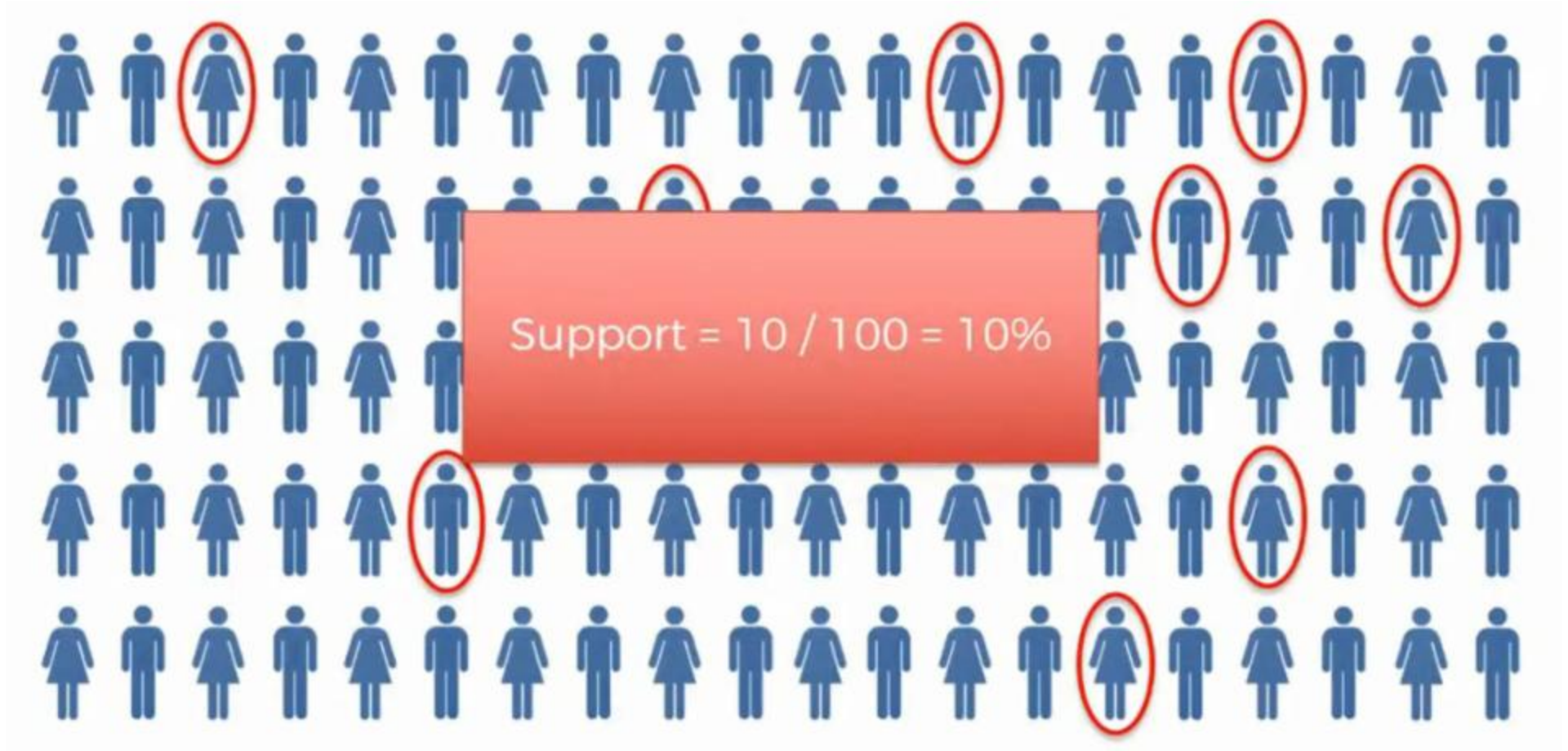
Vegetables ⇨ Fruits

Burgers, French Fries ⇨ Ketchup

Support

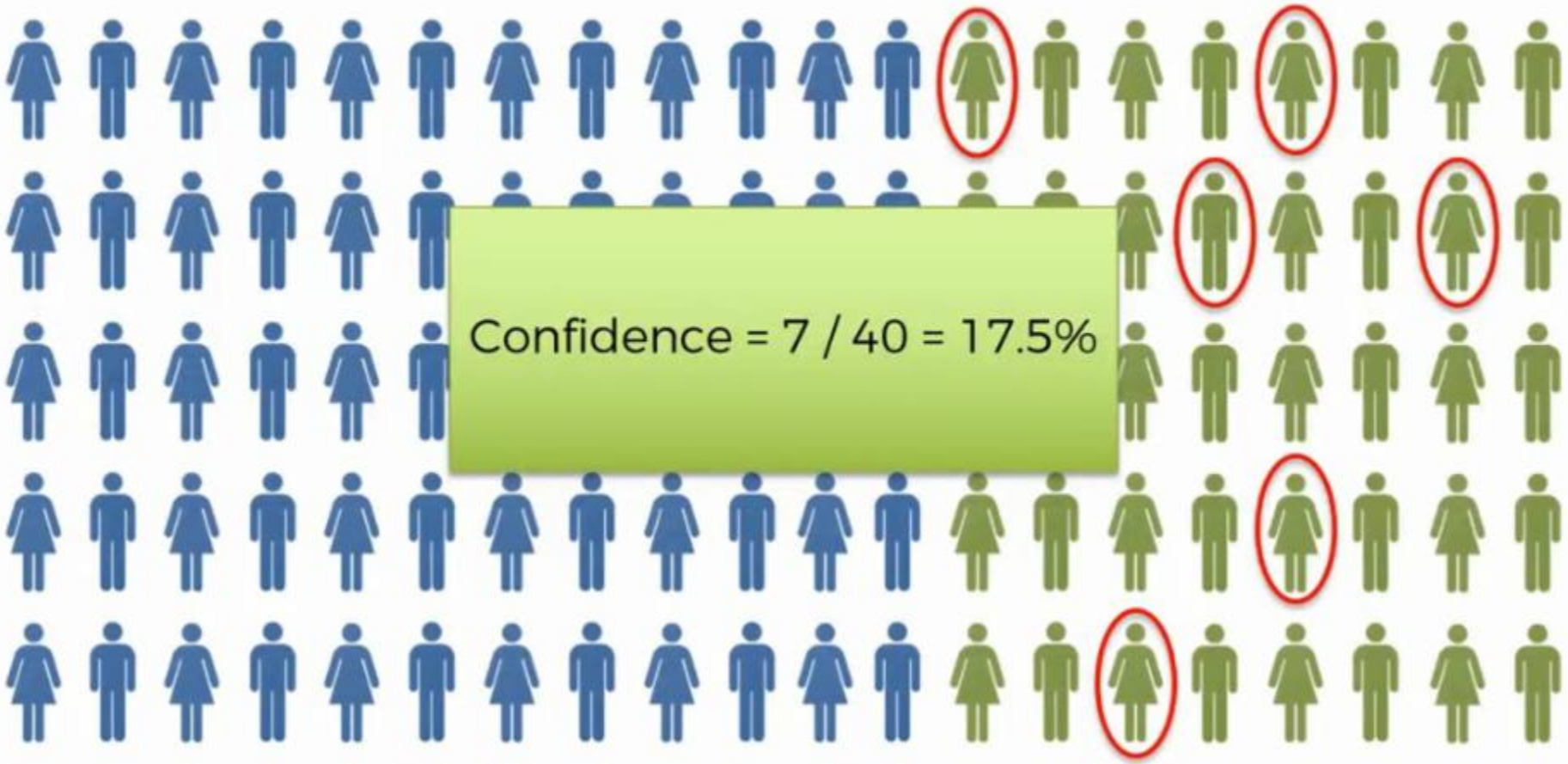
Movie Recommendation: $$\text{support}(\boldsymbol{M}) = \frac{\#\text{ user watchlists containing } \boldsymbol{M}}{\#\text{ user watchlists}}$$

Market Basket Optimisation: $$\text{support}(\boldsymbol{I}) = \frac{\#\text{ transactions containing } \boldsymbol{I}}{\#\text{ transactions}}$$


Support = 10 / 100 = 10%

Confidence

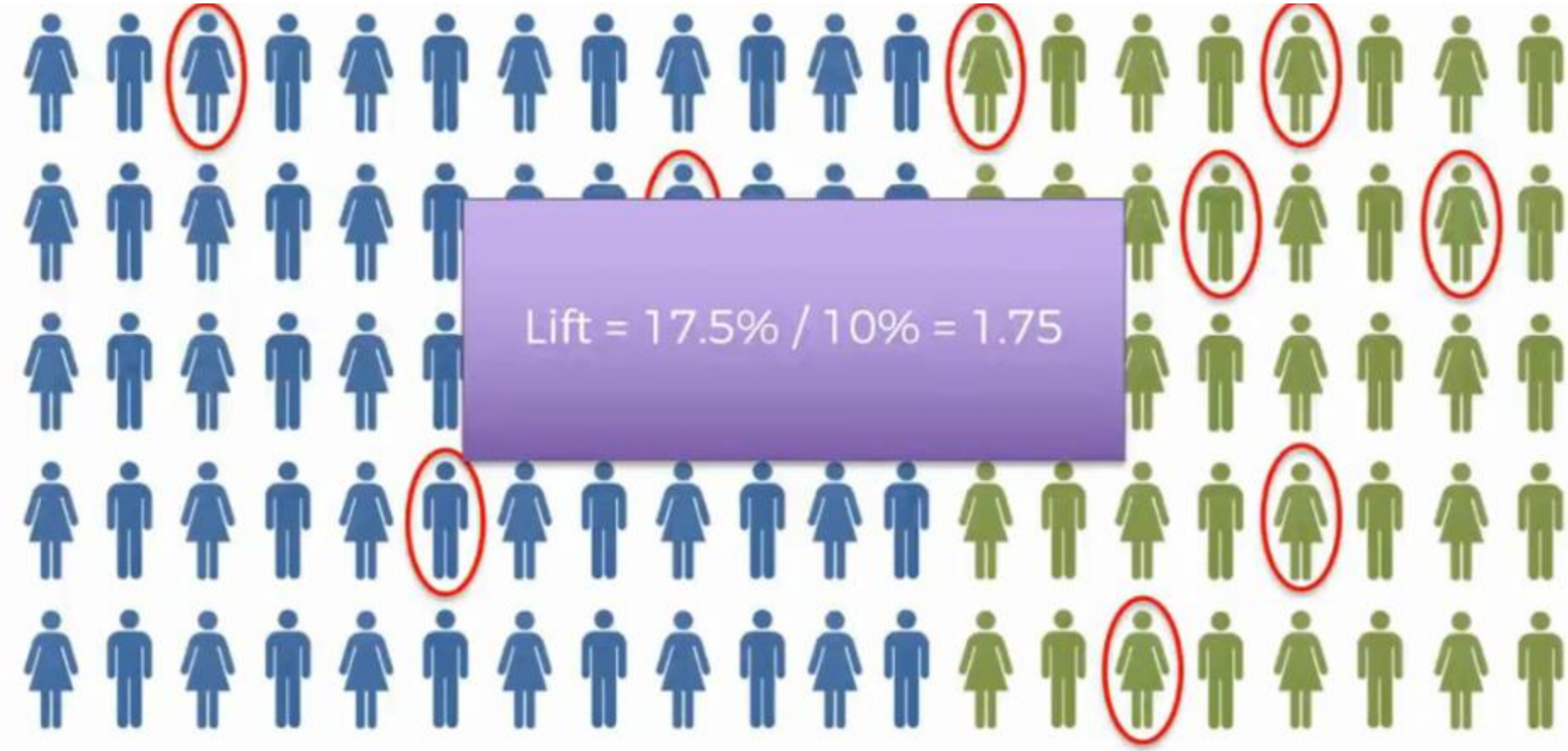
Movie Recommendation: $\text{confidence}(\boldsymbol{M_1} \rightarrow \boldsymbol{M_2}) = \frac{\#\text{ user watchlists containing } \boldsymbol{M_1} \text{ and } \boldsymbol{M_2}}{\#\text{ user watchlists containing } \boldsymbol{M_1}}$


Confidence = 7 / 40 = 17.5%

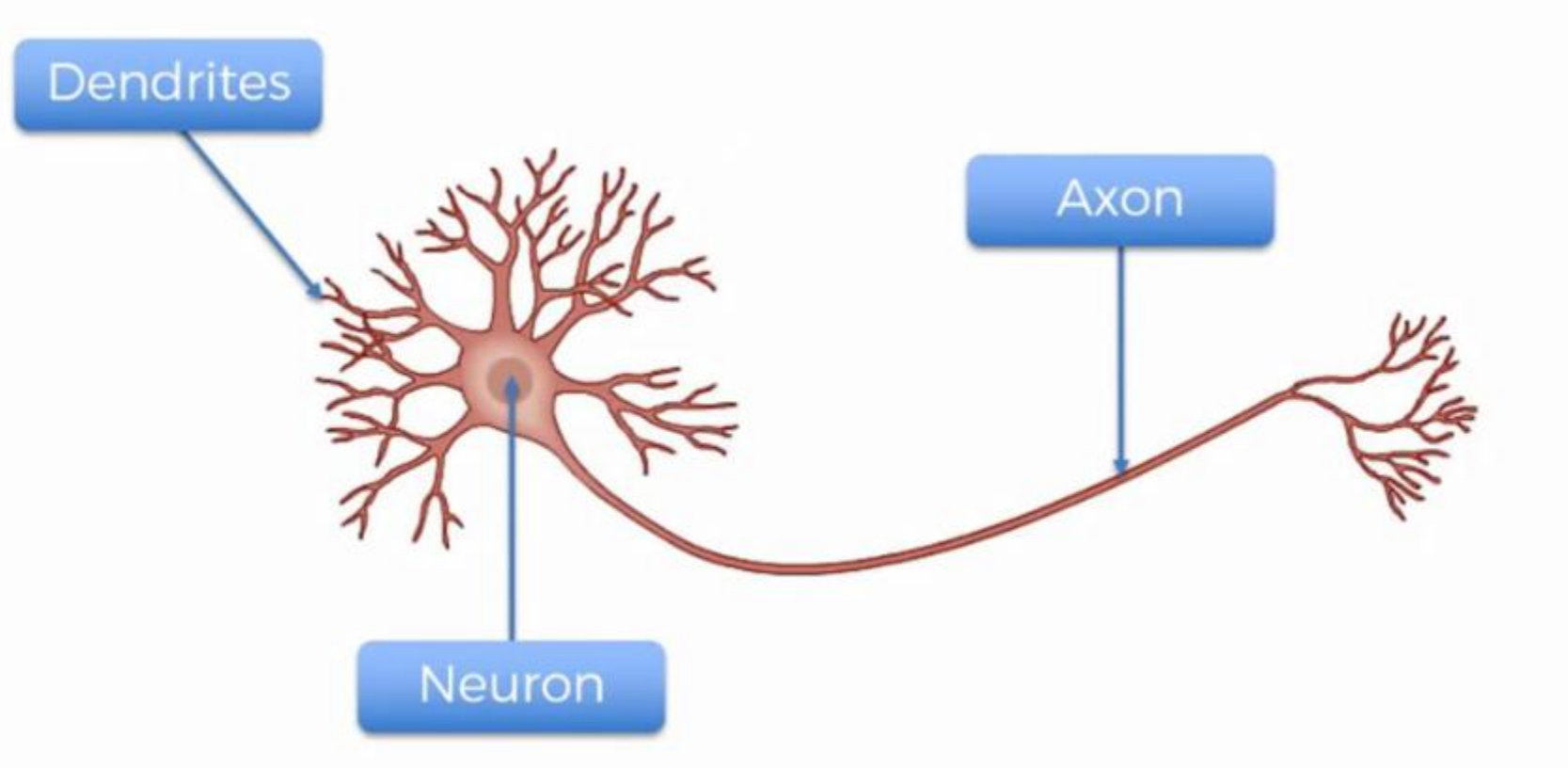
Lift

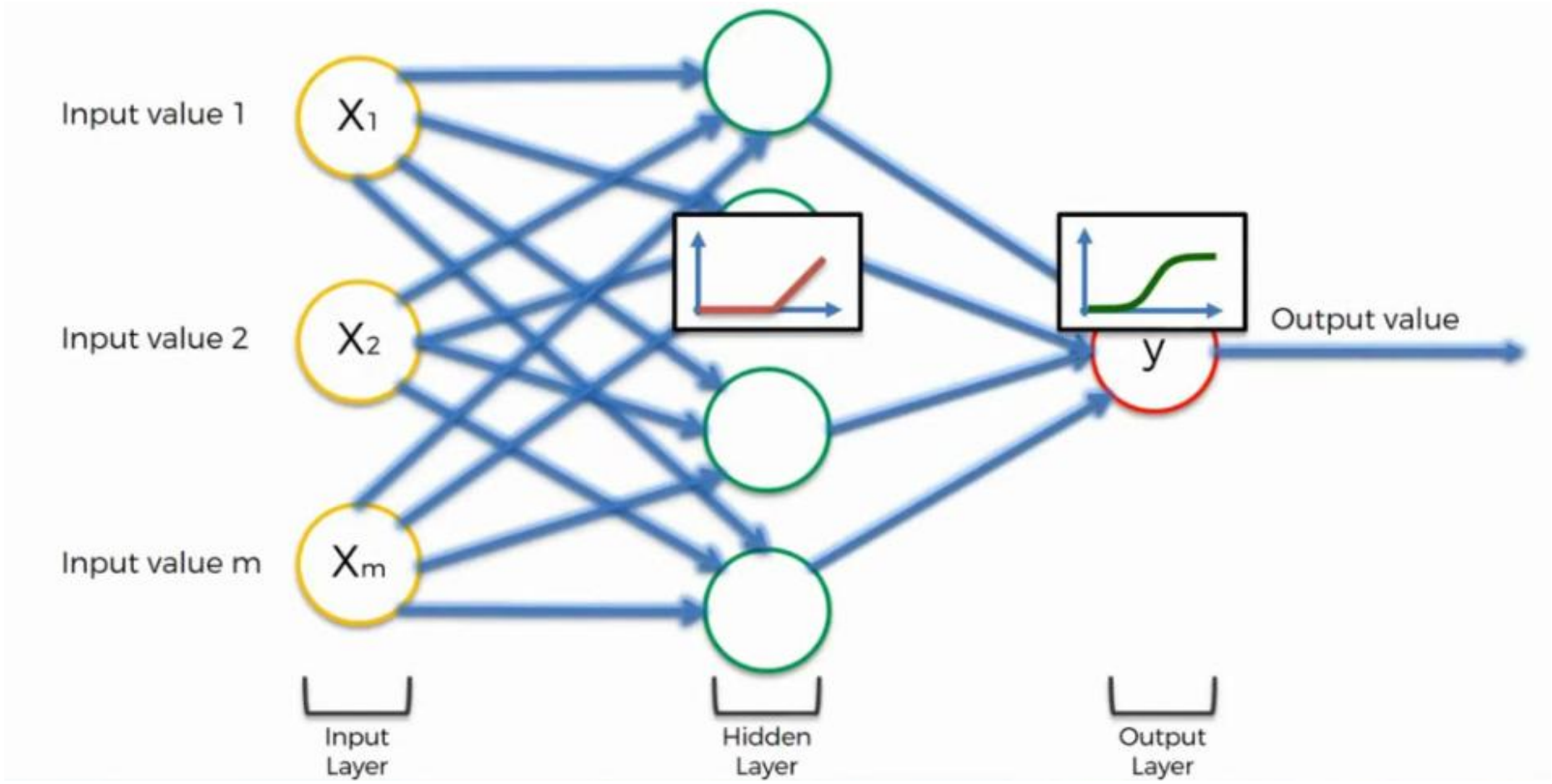
Movie Recommendation:

$$\text{lift}(M_1 \rightarrow M_2) = \frac{\text{confidence}(M_1 \rightarrow M_2)}{\text{support}(M_2)}$$


Lift = 17.5% / 10% = 1.75

ANN




Input value 1
Input value 2
Input value m
X1
X2
Xm
y
Output value
Input Layer
Hidden Layer
Output Layer

Input Layer

- This layer accepts input features.
- It provides information from the outside world to the network, no computation is performed at this layer, nodes here just pass on the information(features) to the hidden layer.

Hidden Layer

- Nodes of this layer are not exposed to the outer world, they are the part of the abstraction provided by any neural network.
- Hidden layer performs all sort of computation on the features entered through the input layer and transfer the result to the output layer.

Output Layer

- This layer returns the final output computed by the network to the application.

Activation Function

- Activation function decides, whether a neuron should be activated or not by calculating weighted sum and further adding bias with it.

- The purpose of the activation function is to *introduce non-linearity* into the output of a neuron.

Activation Function

- Neural network has neurons that work in correspondence of *weight, bias* and their respective activation function.
- In a neural network, we would update the weights and biases of the neurons on the basis of the error at the output.
- This process is known as *back-propagation*.
- Activation functions make the back-propagation possible

Activation Function

- **Sigmoid Function :**
- Usually used in output layer of a binary classification, where result is either 0 or 1
- As value for sigmoid function lies between 0 and 1 only
- Result can be predicted easily to be 1 if value is greater than 0.5 and 0 otherwise.

Activation Function

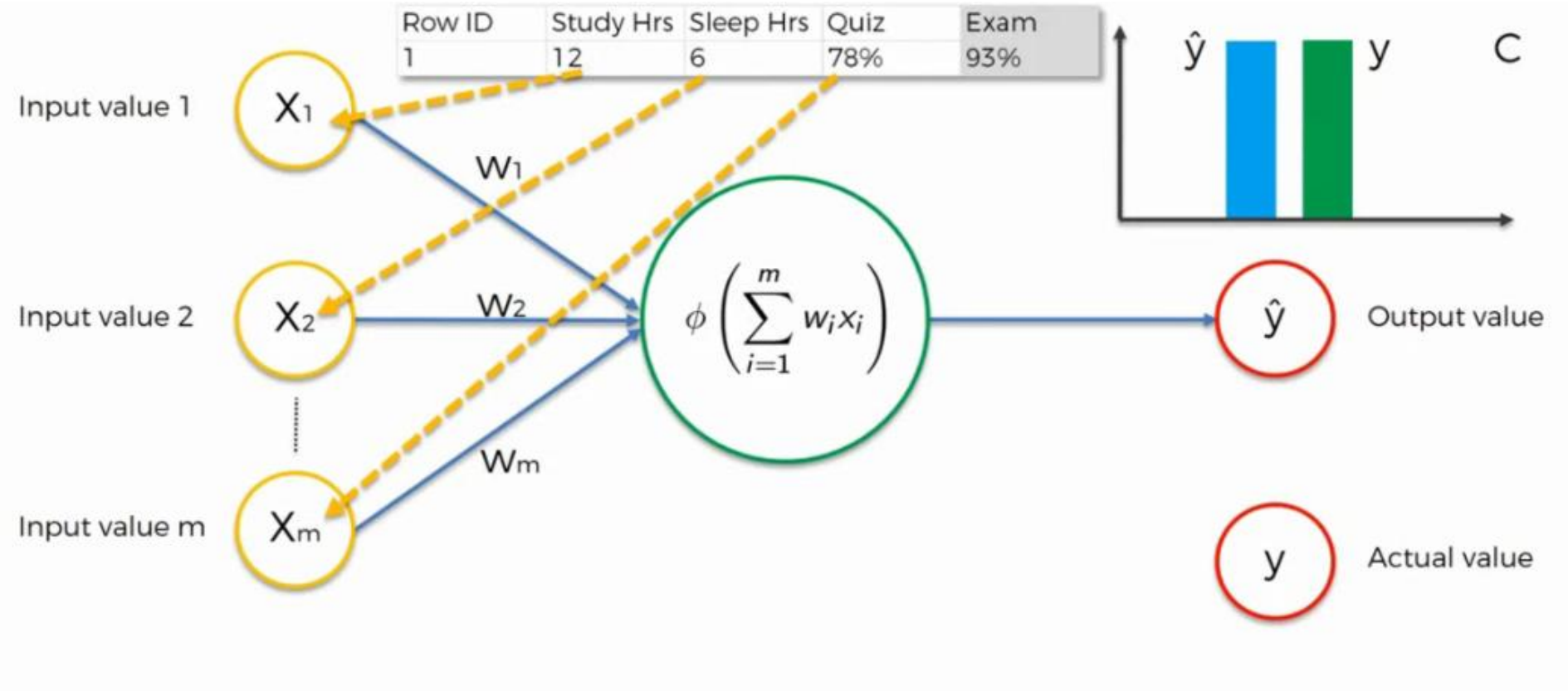
- **RELU (***Rectified linear unit***)**

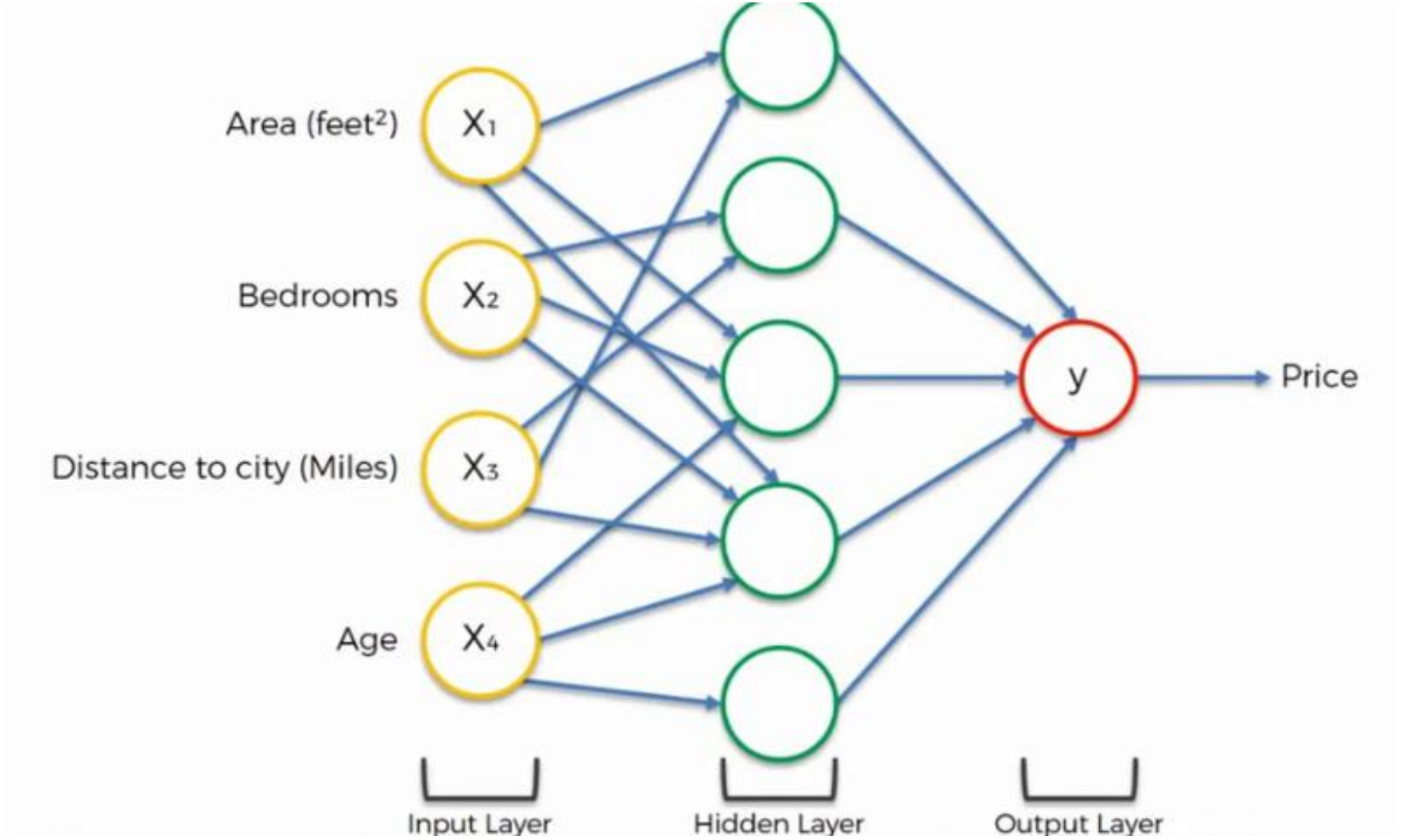
- It is the most widely used activation function.

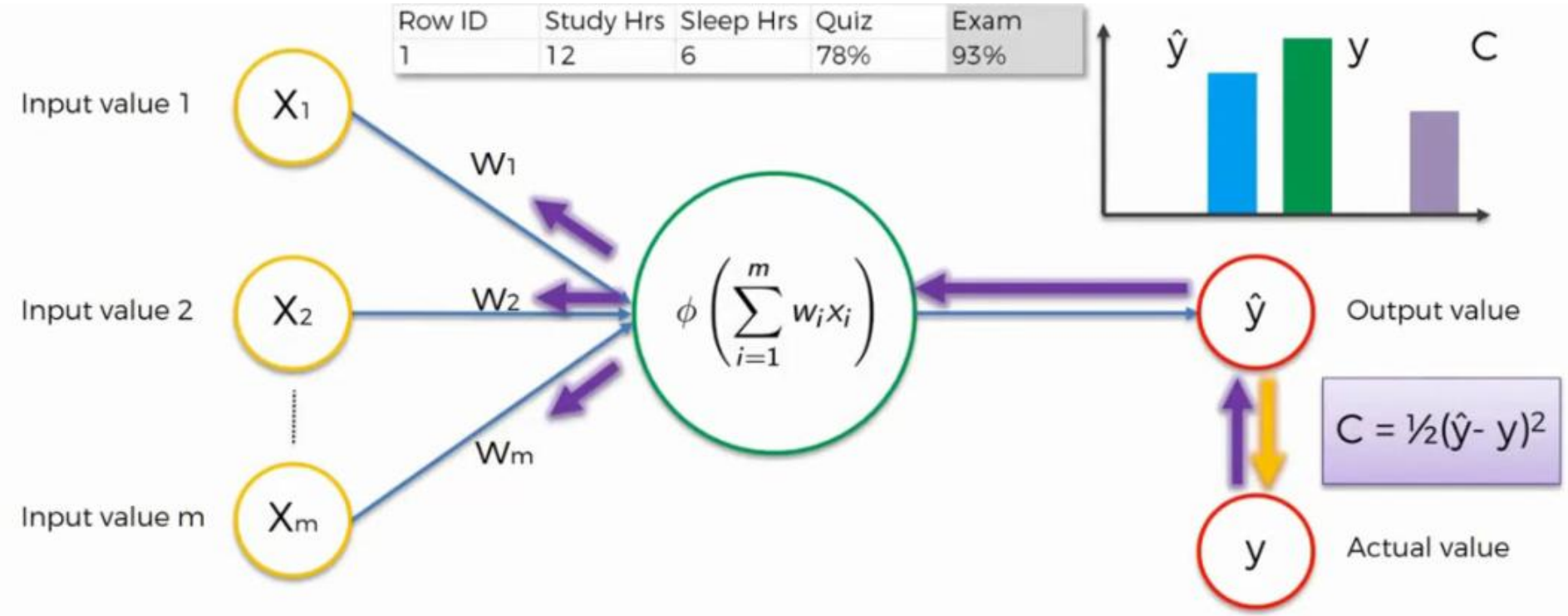
- ReLu is less computationally expensive because it involves simpler mathematical operations.

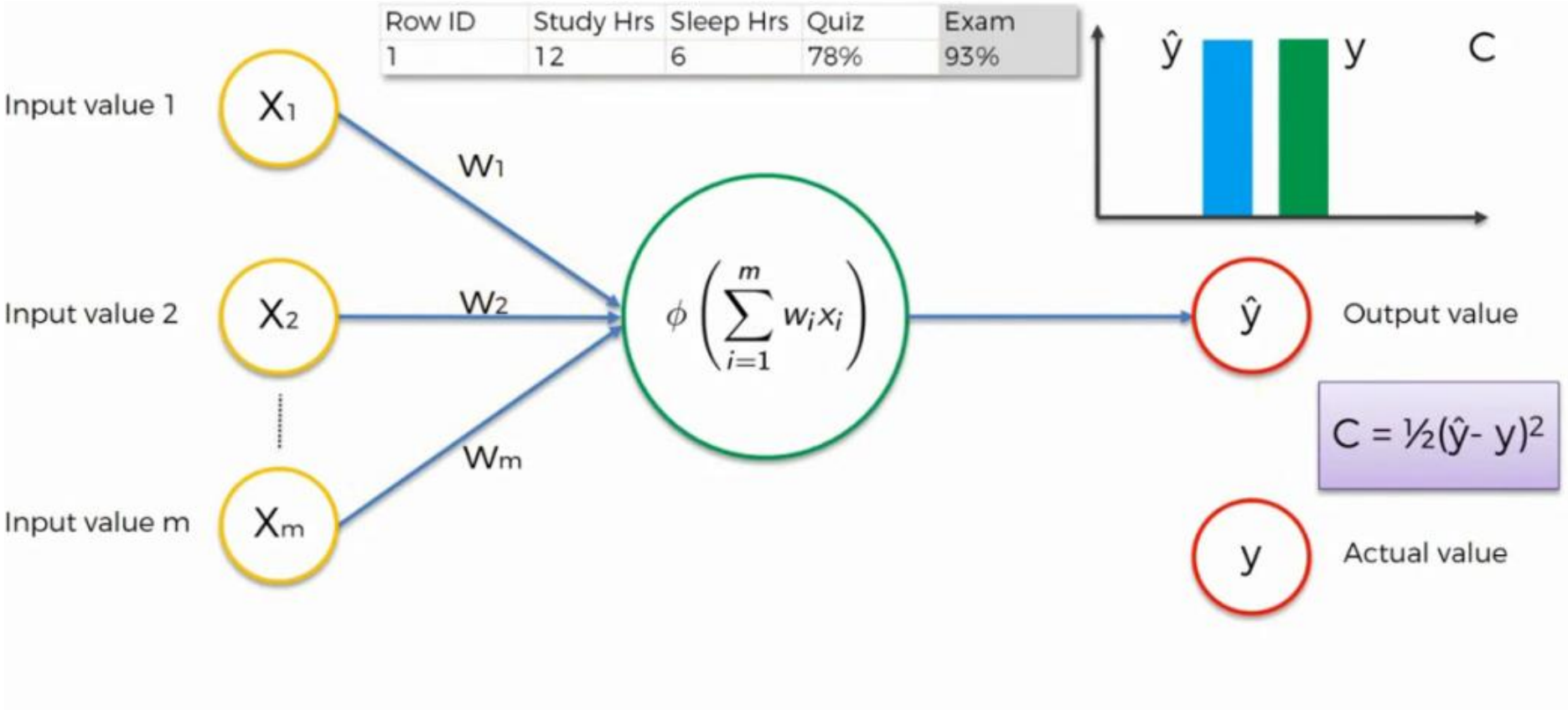
Gradient Descent

- Gradient Descent is used while training a machine learning model.
- A gradient measures how much the output of a function changes if you change the inputs a little bit.
- It simply measures the change in all weights with regard to the change in error.


Row ID | Study Hrs | Sleep Hrs | Quiz | Exam
1 | 12 | 6 | 78% | 93%
ŷ
y
C
Input value 1
Input value 2
Input value m
X1
X2
Xm
W1
W2
Wm
$\phi\left(\sum_{i=1}^{m} w_i x_i\right)$
ŷ
Output value
y
Actual value


Area (feet²)
X1
Bedrooms
X2
Distance to city (Miles)
X3
Age
X4
y
Price
Input Layer
Hidden Layer
Output Layer


Row ID | Study Hrs | Sleep Hrs | Quiz | Exam
1 | 12 | 6 | 78% | 93%
ŷ
y
C
Input value 1
X1
W1
Input value 2
X2
W2
Input value m
Xm
Wm
$\phi\left(\sum_{i=1}^{m} w_i x_i\right)$
ŷ
Output value
C = ½(ŷ- y)²
y
Actual value


Row ID | Study Hrs | Sleep Hrs | Quiz | Exam
1 | 12 | 6 | 78% | 93%
ŷ
y
C
Input value 1
X1
W1
Input value 2
X2
W2
Input value m
Xm
Wm
$\phi\left(\sum_{i=1}^{m} w_i x_i\right)$
ŷ
Output value
C = ½(ŷ- y)²
y
Actual value