

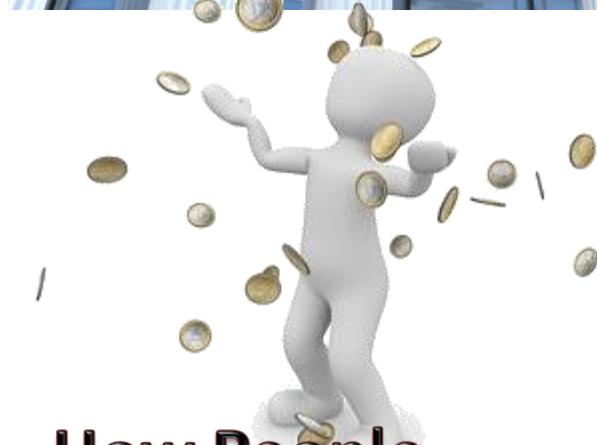
# DataScience

## Machine Learning with Python

# Introduction



# Introduction



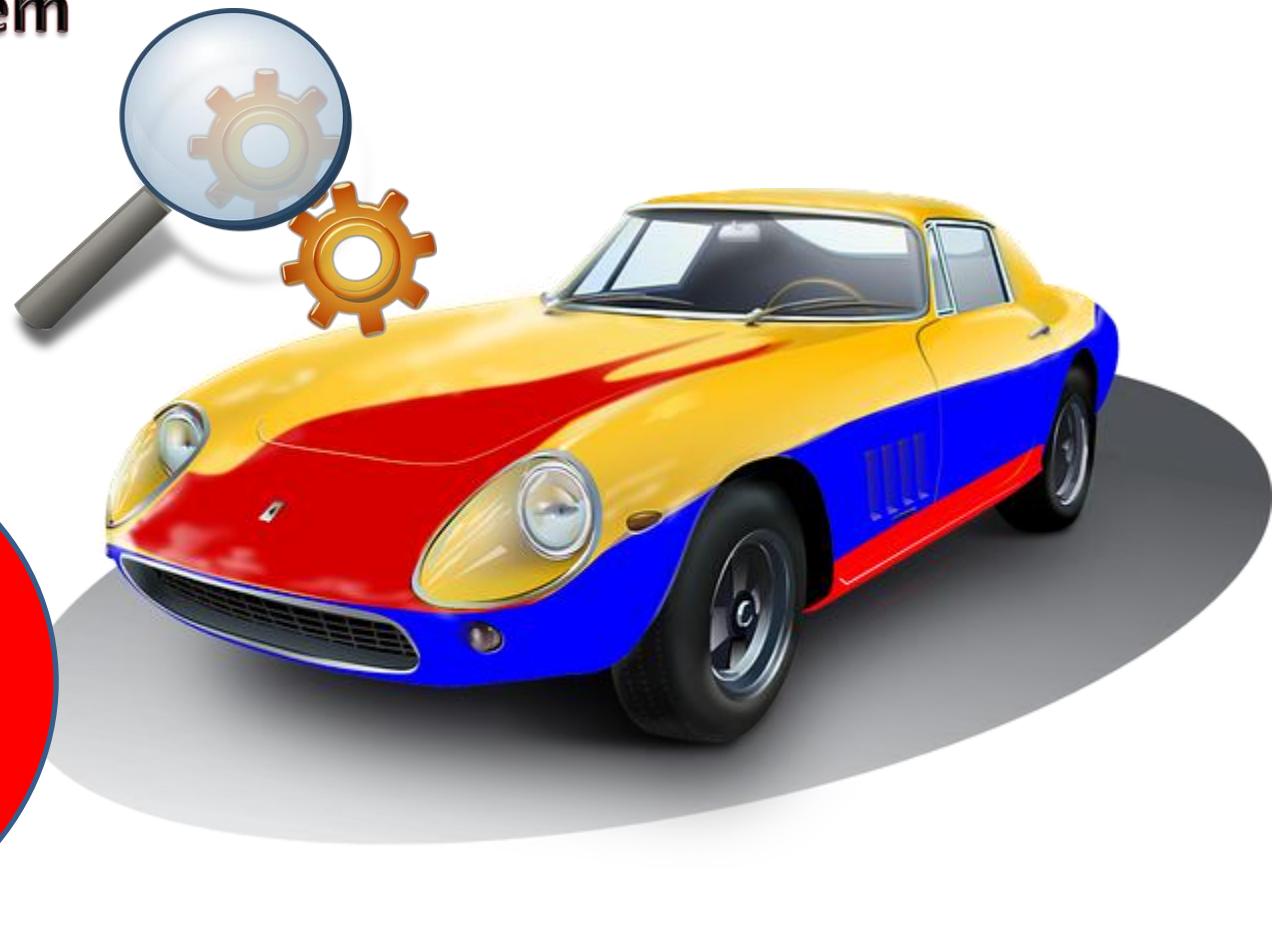
**How People  
Spend Money?**



**What treatments for which illness?  
How patient is responding?**

# Introduction

**Car engine  
monitoring system**



# Banking



Learn spending  
Pattern



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?**

Some Questions That Machine  
Learning Can Be Used To  
Answer

**Car engine  
monitoring system**



# Machine Learning



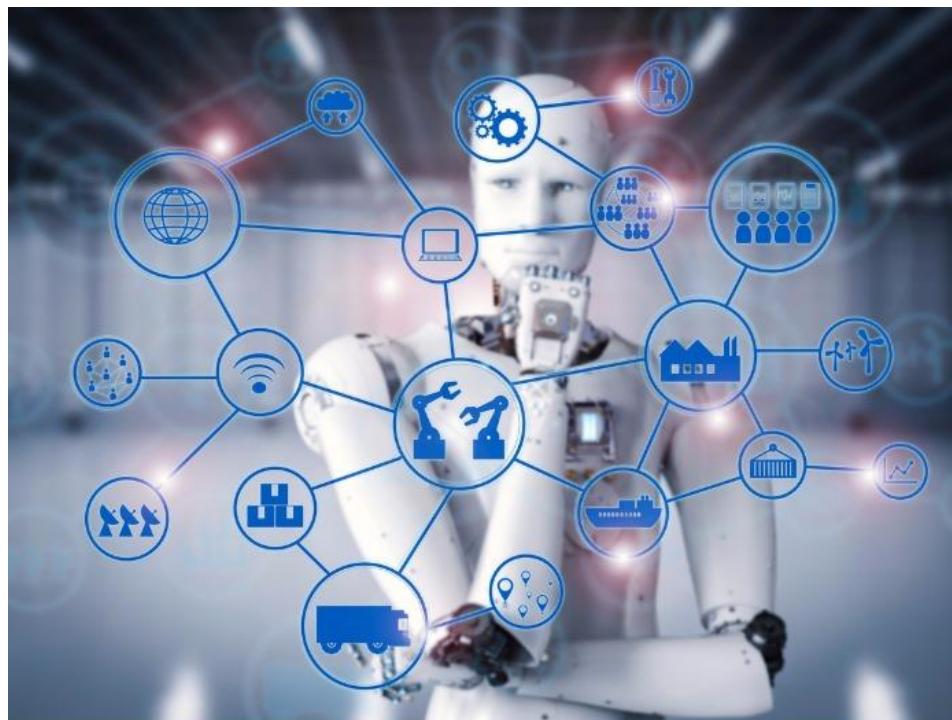
*The size and complexity of these datasets mean that humans are unable to extract useful information from them.*

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

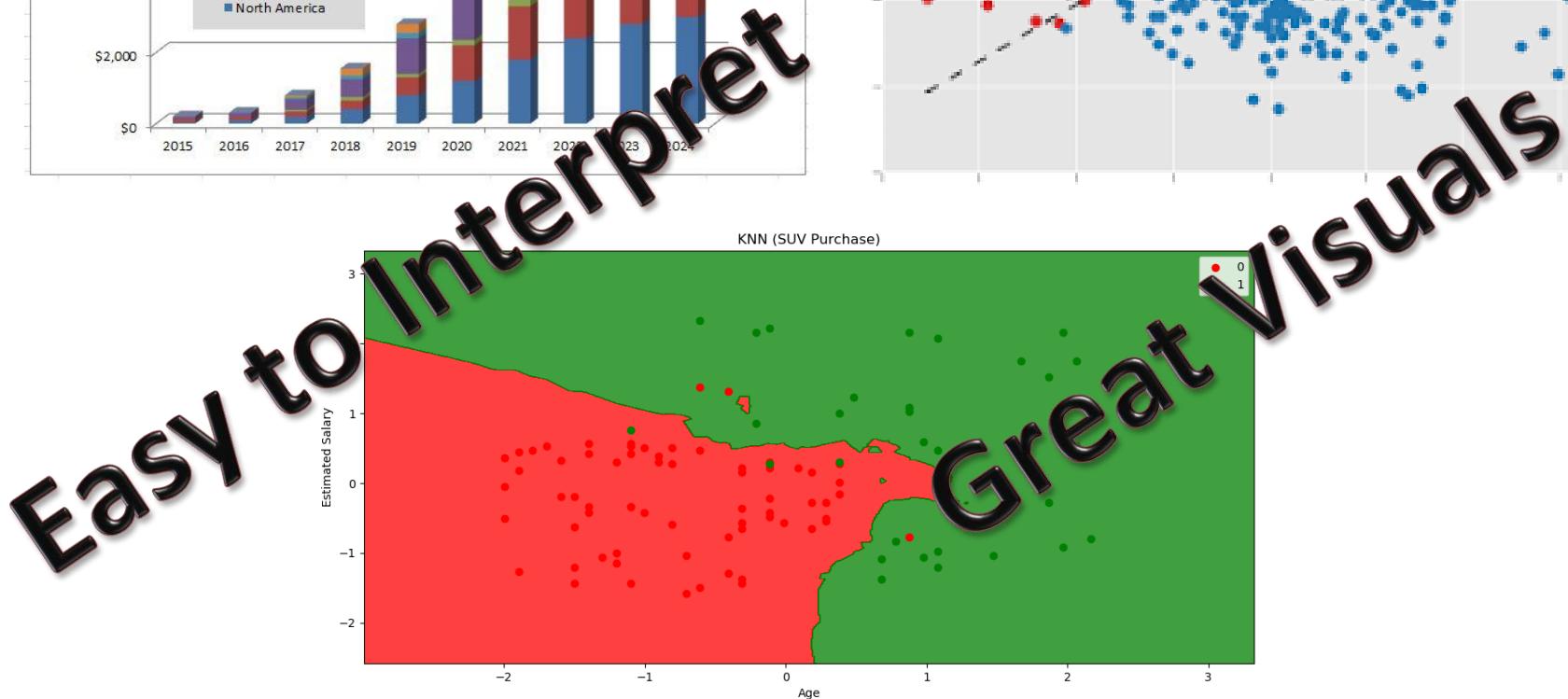
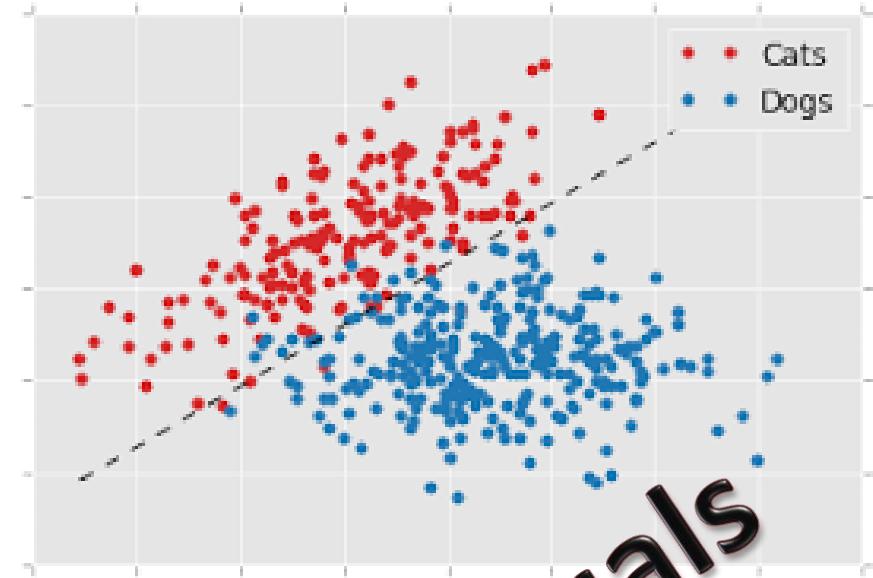
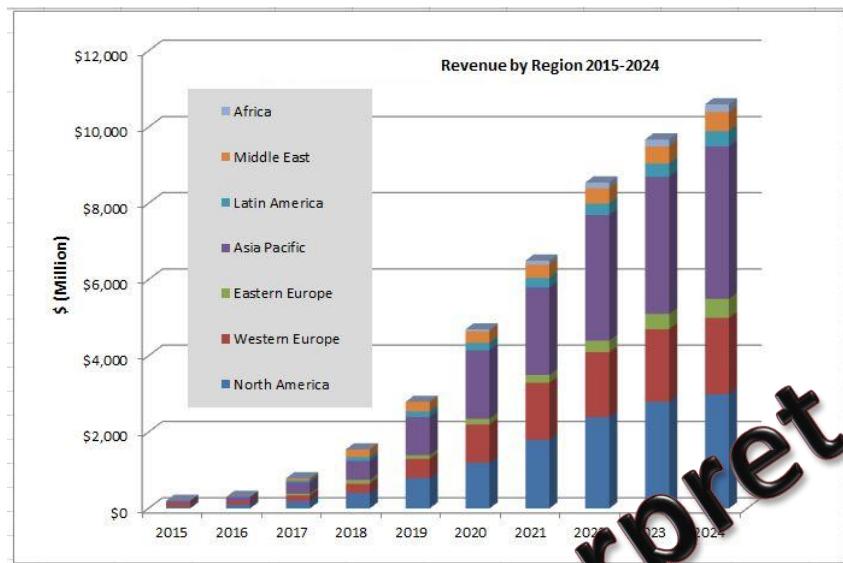
*Given a file full of numbers, our minds generally turn away from looking at them for long.*

# 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.

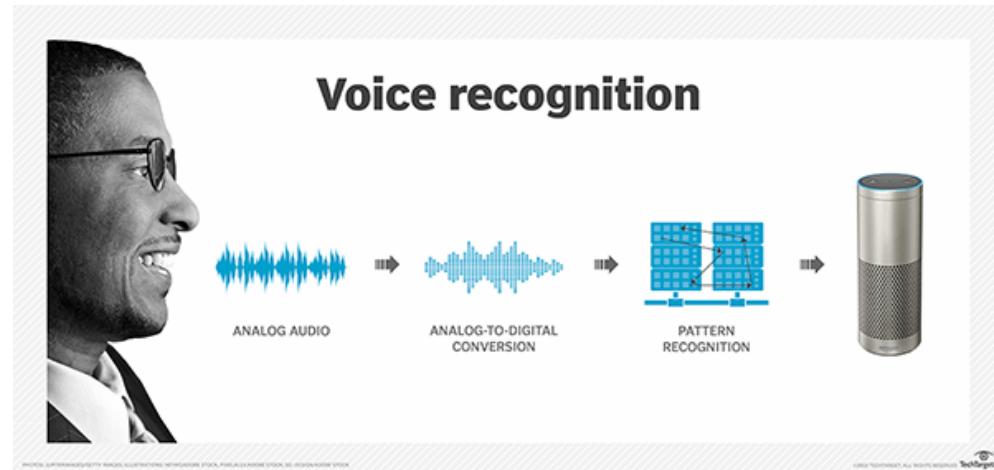


# Machine Learning



# ML Algorithms

## *Spam Filters*



## *Computer Games*

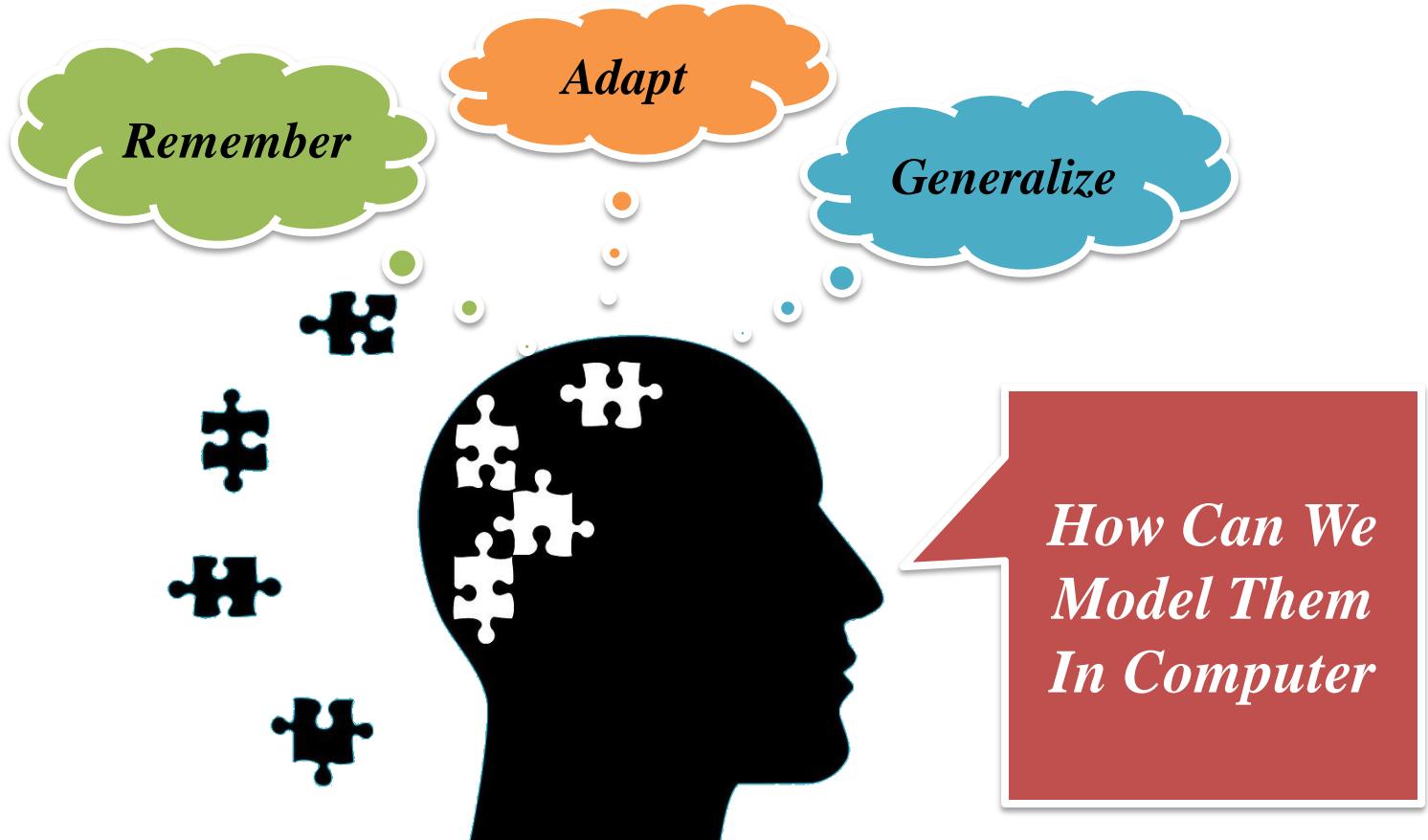


*Algorithm to decide  
whether bank will  
give loan or not*

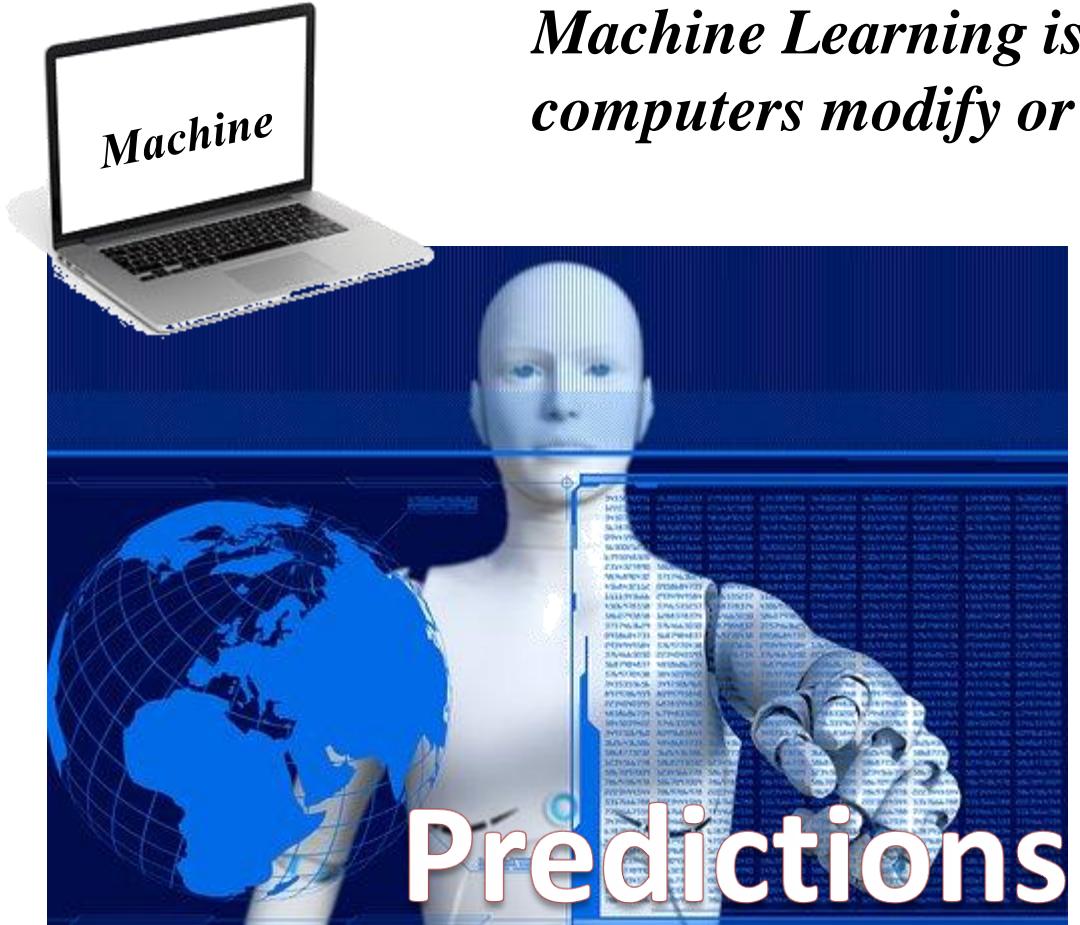


# Machine Learning

## *Intelligence Learning*



# Machine Learning



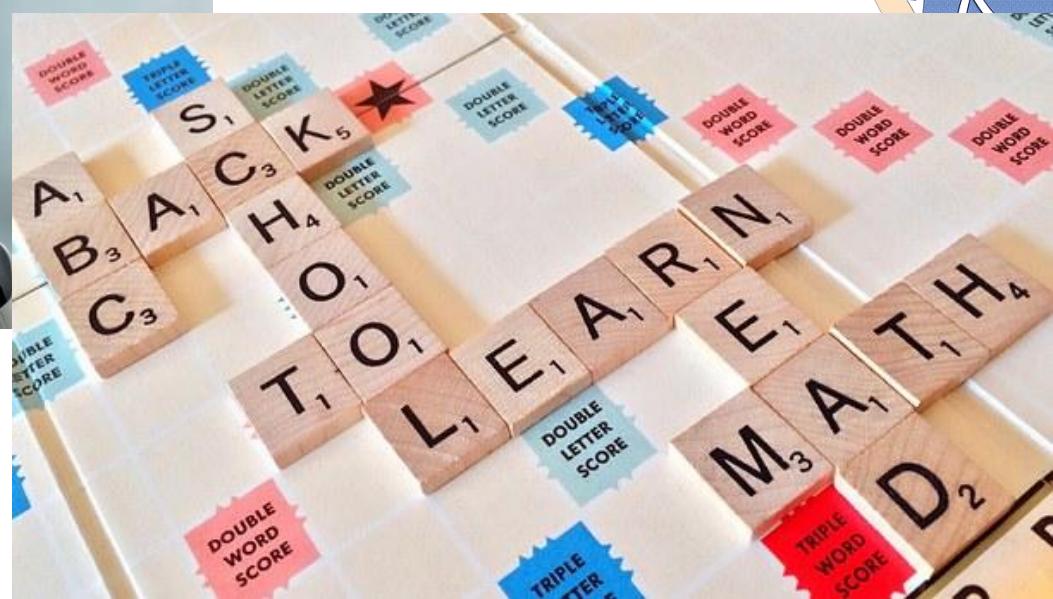
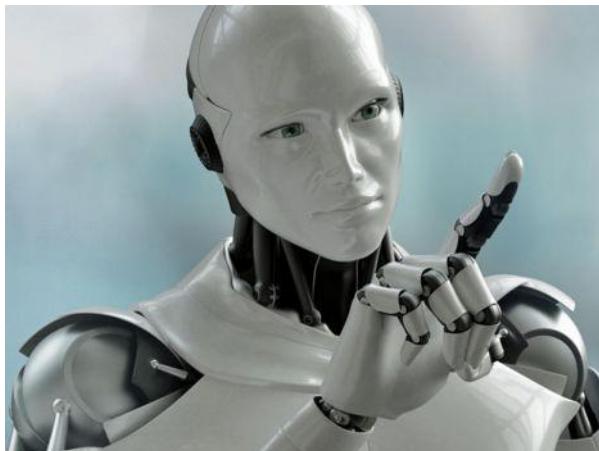
*Machine Learning is about making computers modify or adapt their actions*

*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:*

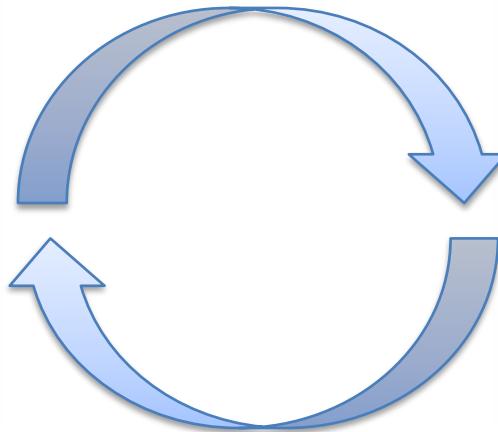
- *K-Means Cluster algorithm*
- *Anomaly detection*

# 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.



**DATA**



**DECISIONS**

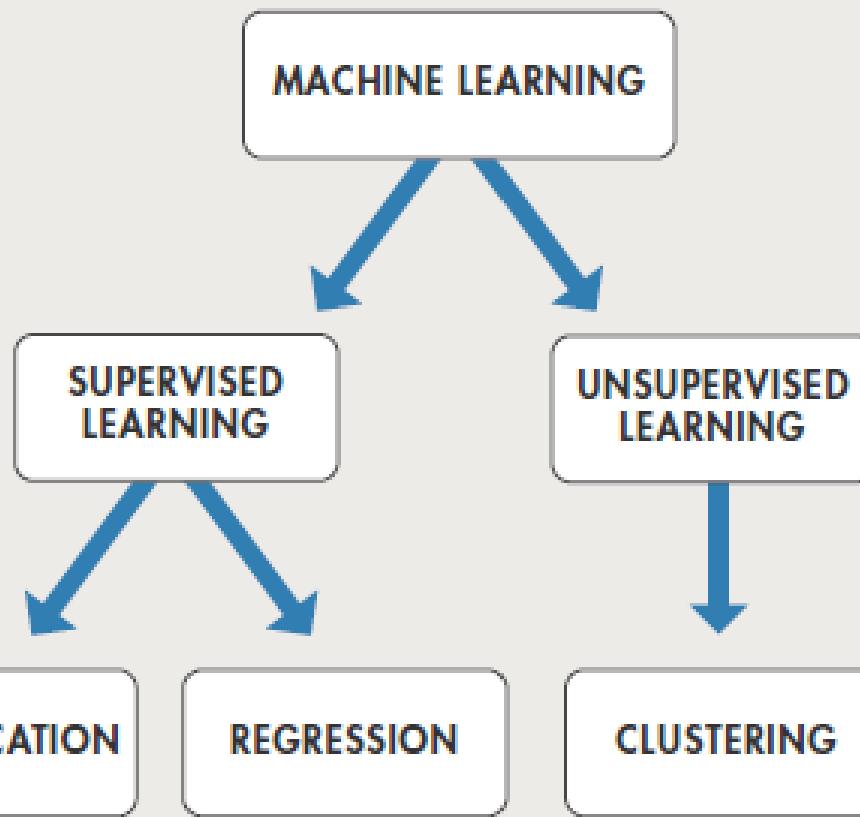


***Better Data, Better Decisions***

***Today's Decisions Generate Data for future***

# No best method or one size fits all

## Selecting an Algorithm



# 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.

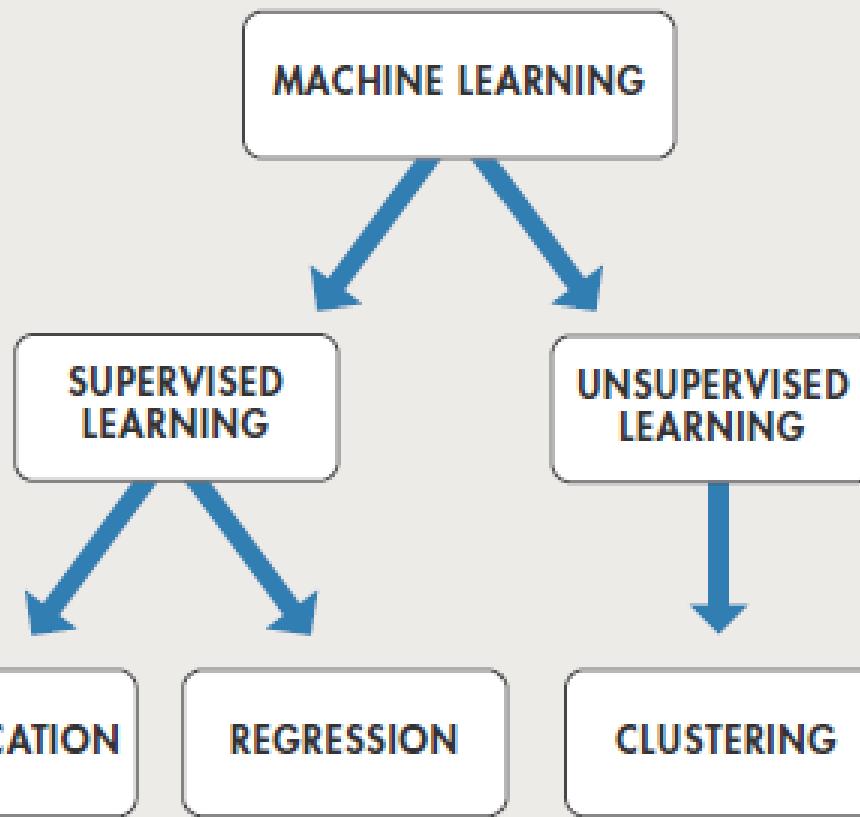
# Machine Learning Challenges

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

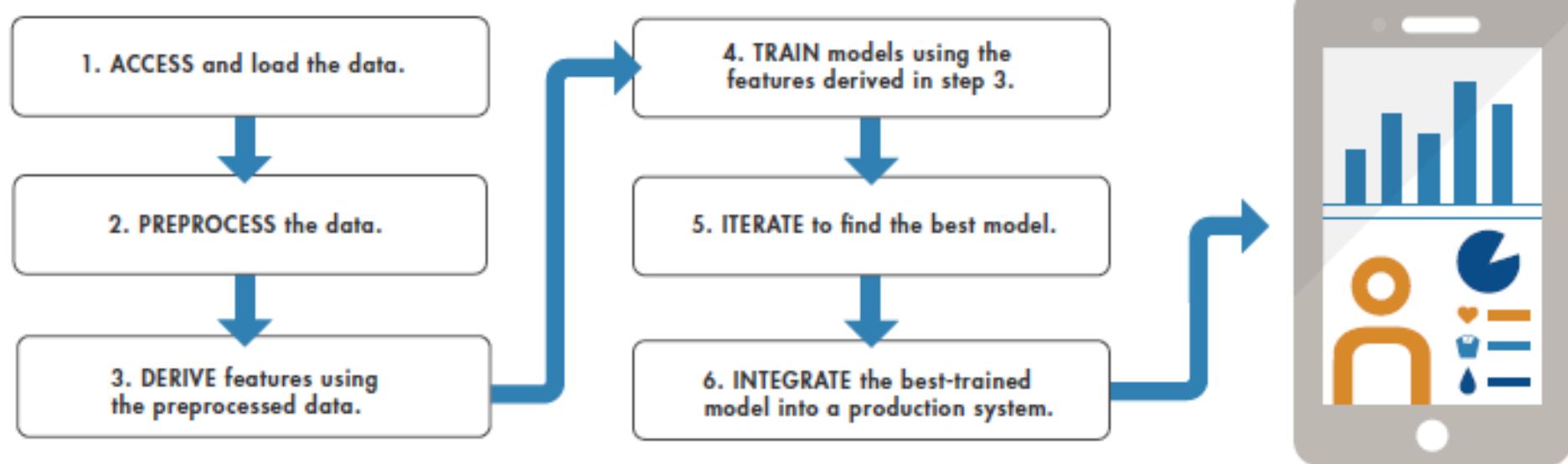
Choosing the right model is a balancing act. Highly flexible models tend to overfit data by modeling minor variations that could be noise. On the other hand, simple models may assume too much. There are always tradeoffs between model speed, accuracy, and complexity.

# No best method or one size fits all

## Selecting an Algorithm



# ML Workflow



# 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

$$y = b_0 + b_1 * x_1$$

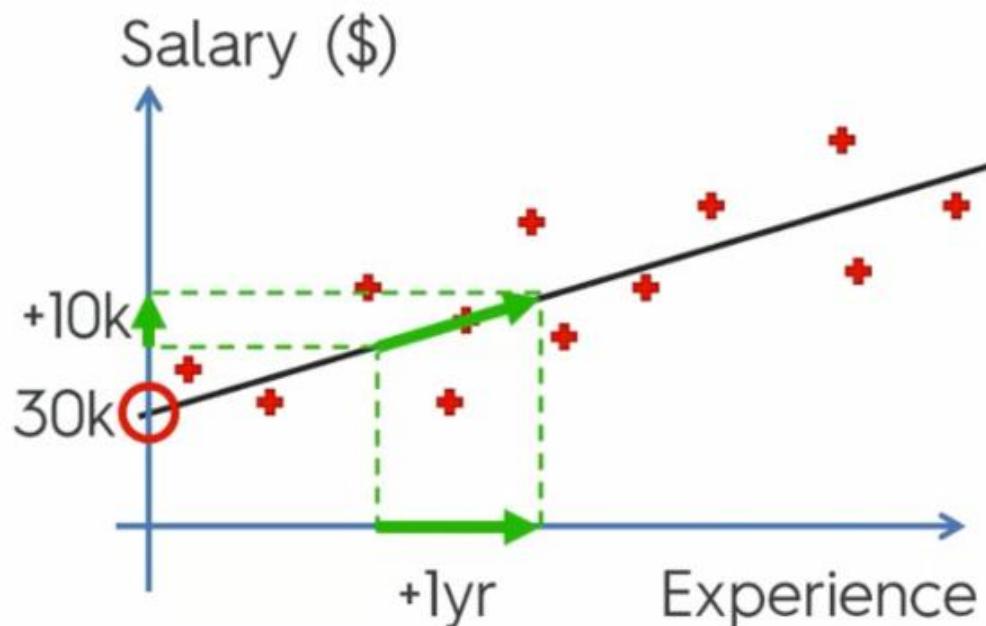
**y** is the dependent variable

**x1** is independent variable

**b1** is coefficient of X or the slope of  
the line

**b0** is the constant called intercept

## Simple Linear Regression:



$$y = b_0 + b_1 * x$$



$$\text{Salary} = \textcircled{b}_0 + \textcircled{b}_1 * \text{Experience}$$

## Simple Linear Regression:

Salary (\$)



Experience

$$\text{SUM } (y - \hat{y})^2 \rightarrow \min$$

# Multiple Linear Regression

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

# 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

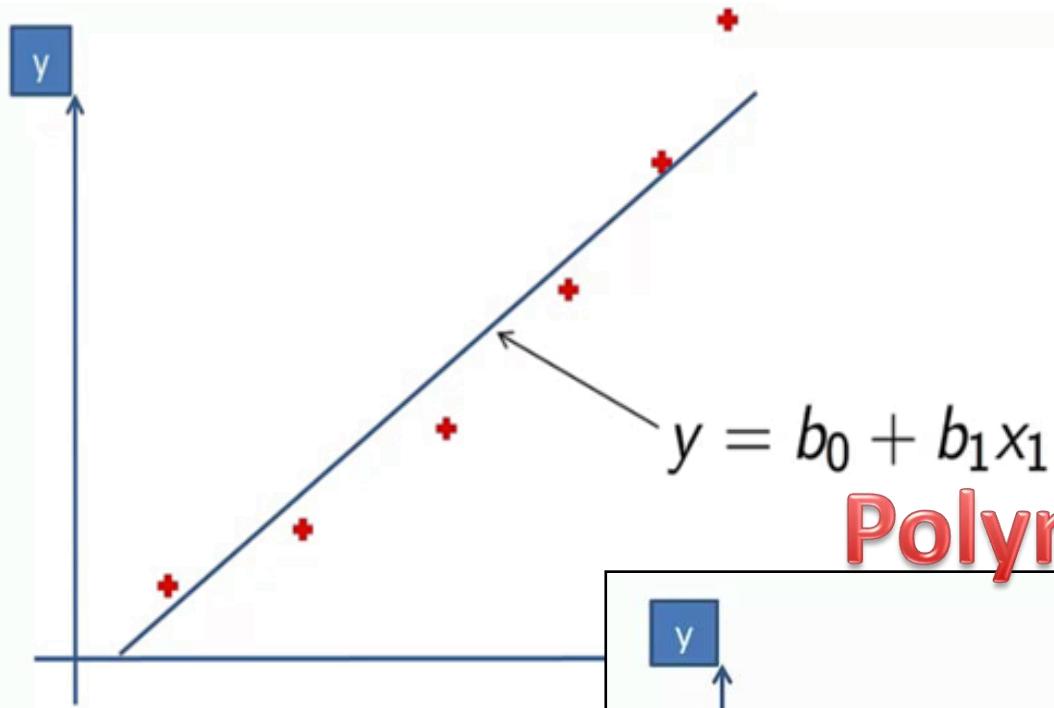
- The adjusted R-squared is a modified version of R-squared that has been adjusted for the number of predictors in the model.
- The adjusted R-squared increases only if the new term improves the model.
- It decreases if the new term doesn't contribute to the model improvement.

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

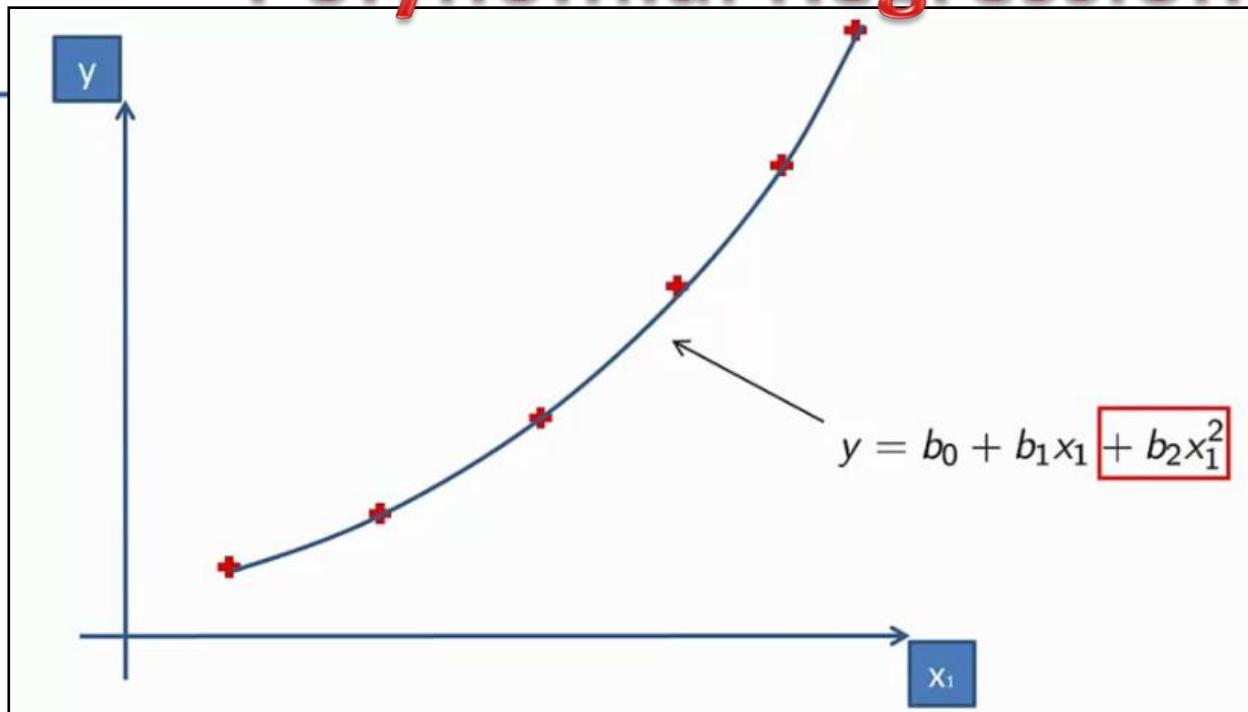
# Polynomial Regression

$$y = b_0 + b_1x_1 + b_2x_1^2 + \dots + b_nx_1^n$$

# Simple Linear Regression



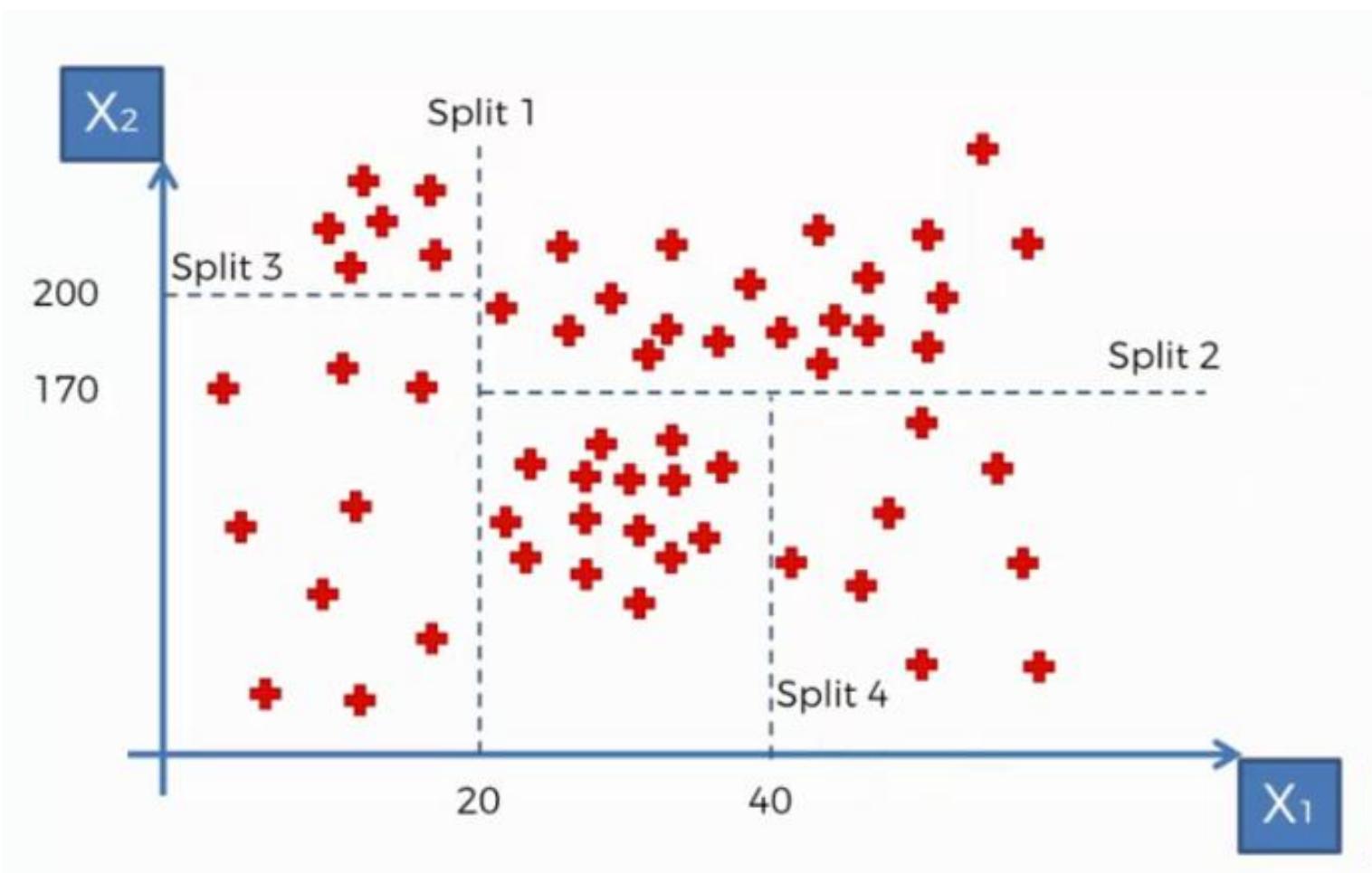
# Polynomial Regression



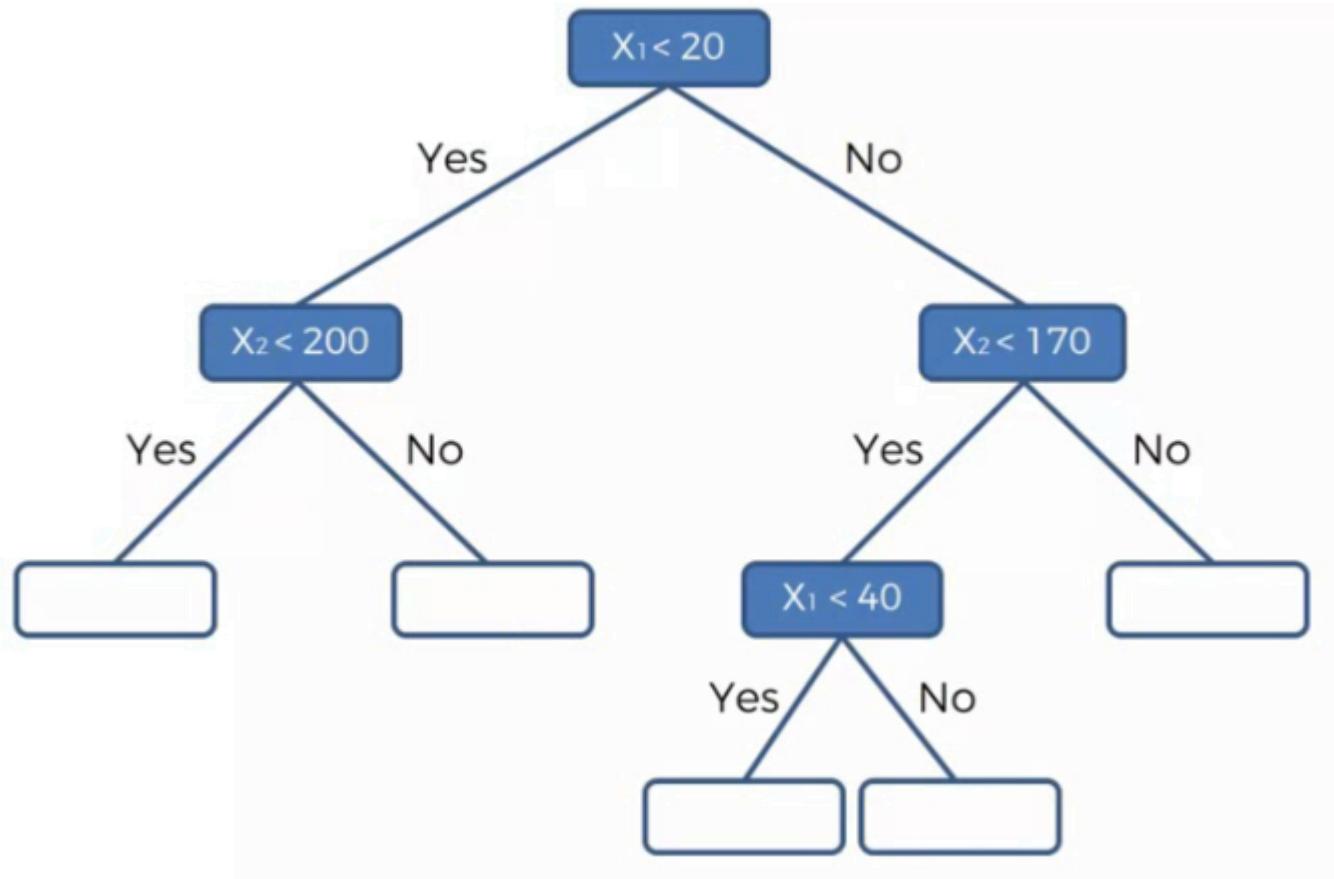
# Decision Tree Regression

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

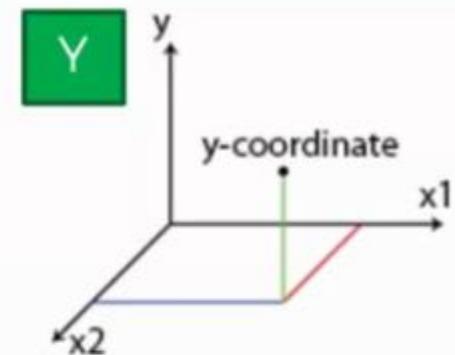
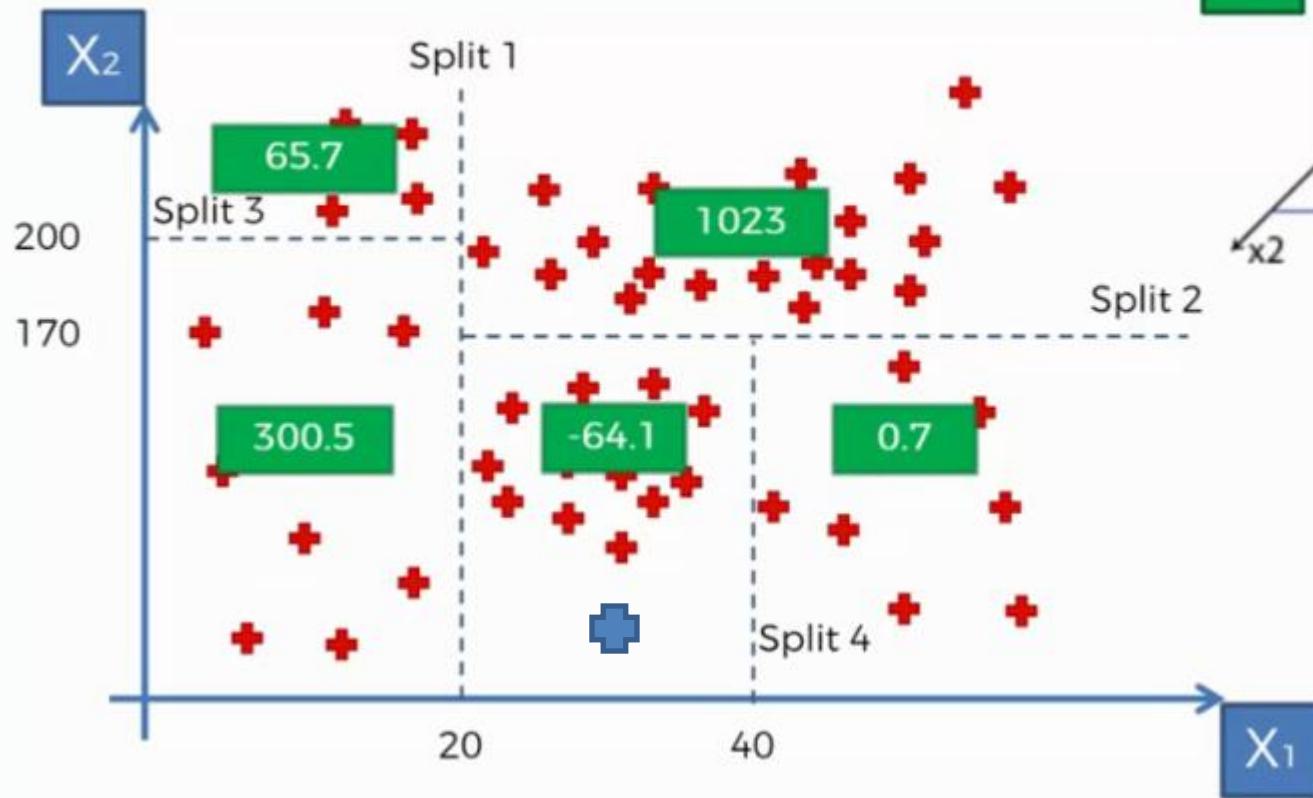
# Decision Tree Regression



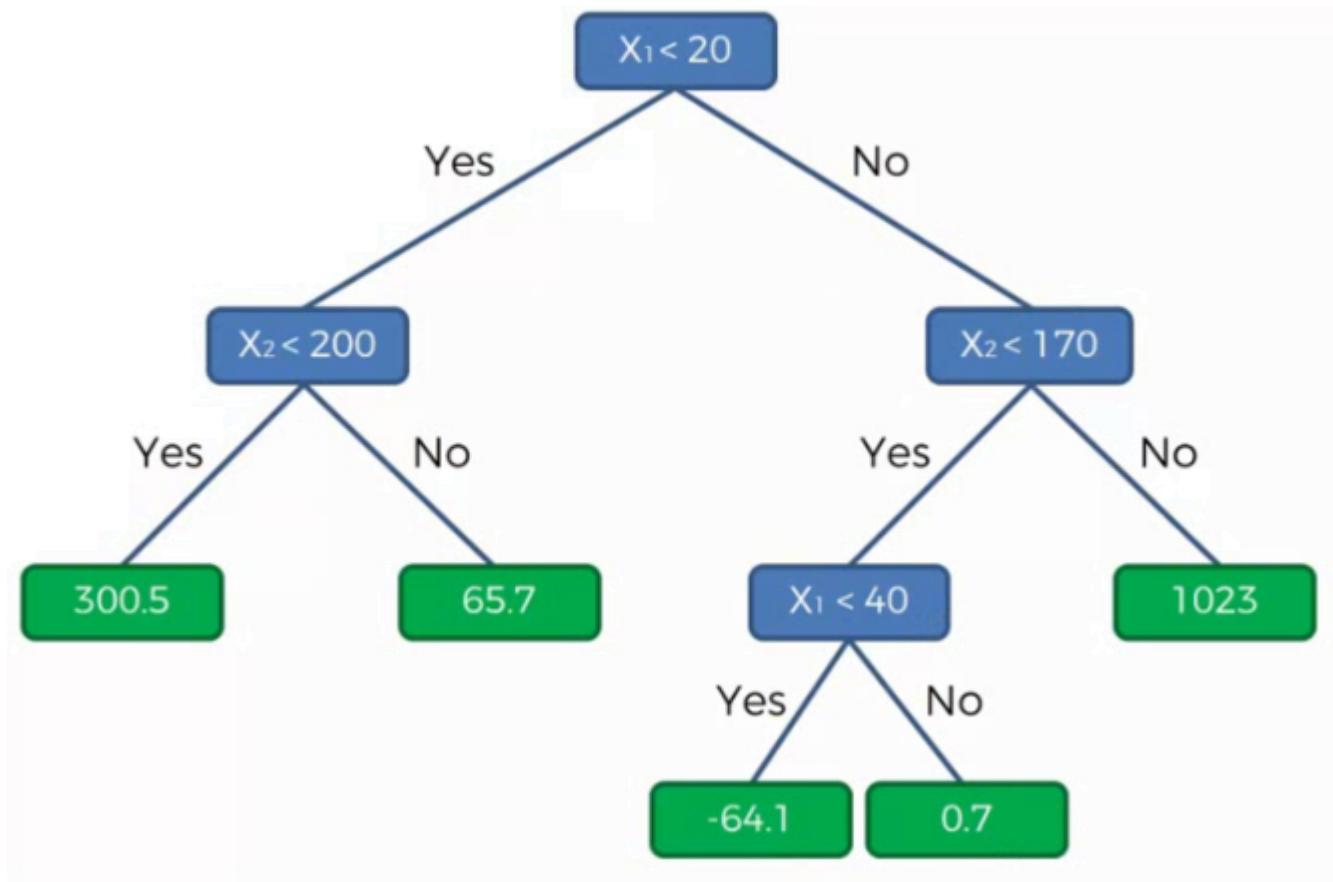
# Decision Tree Regression



# Decision Tree Regression

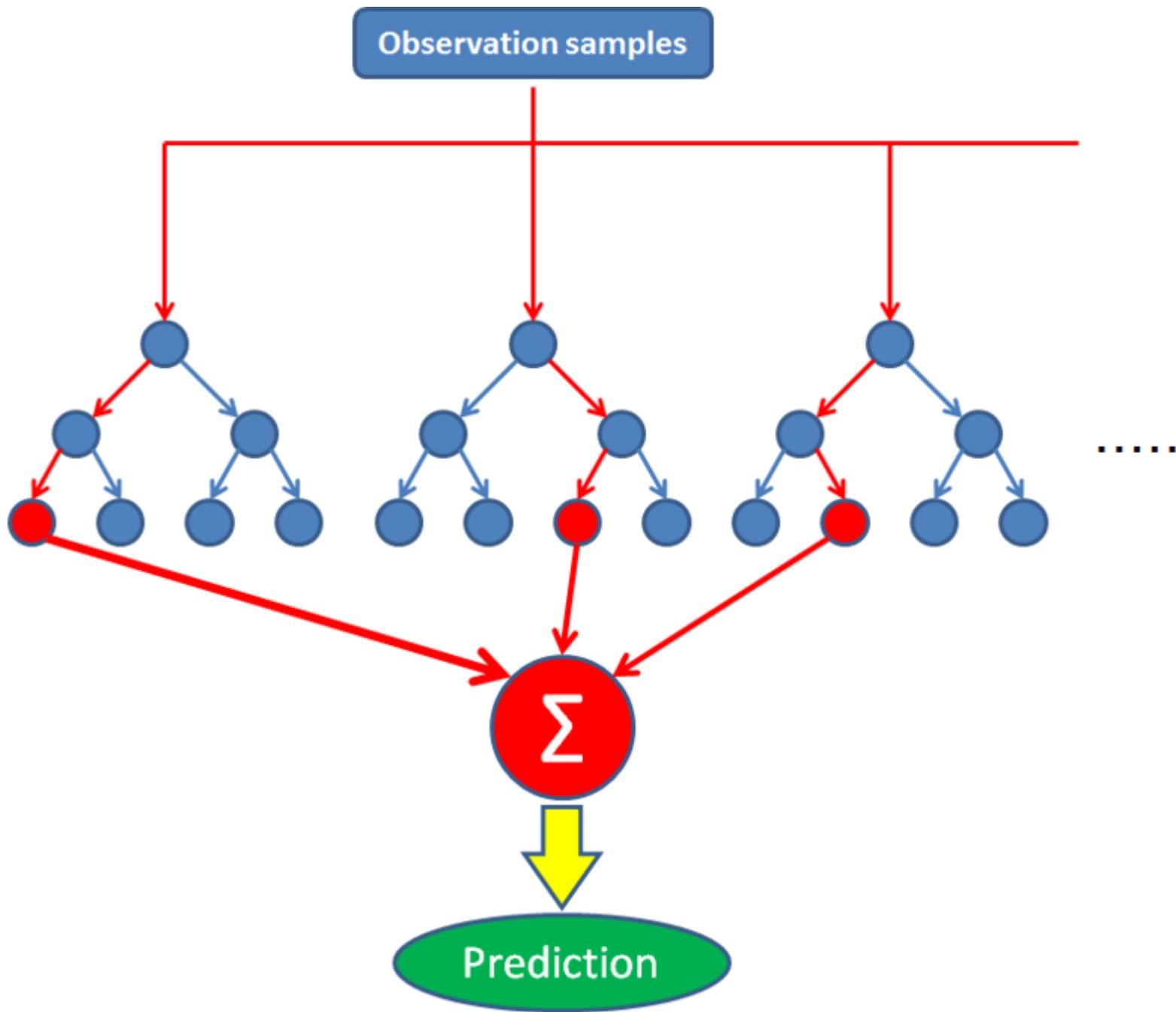


# Decision Tree Regression



# Random Forest Regression

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



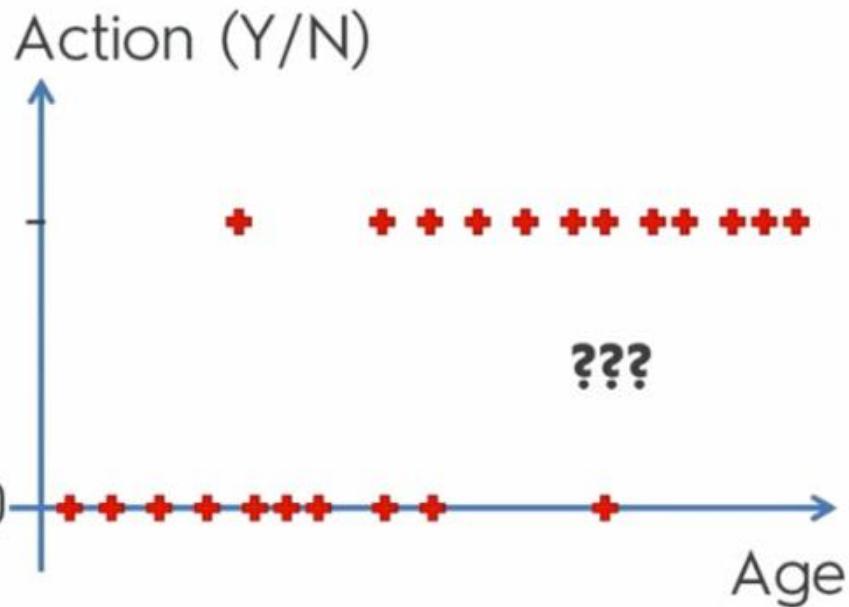
# Classification

# Confusion Matrix

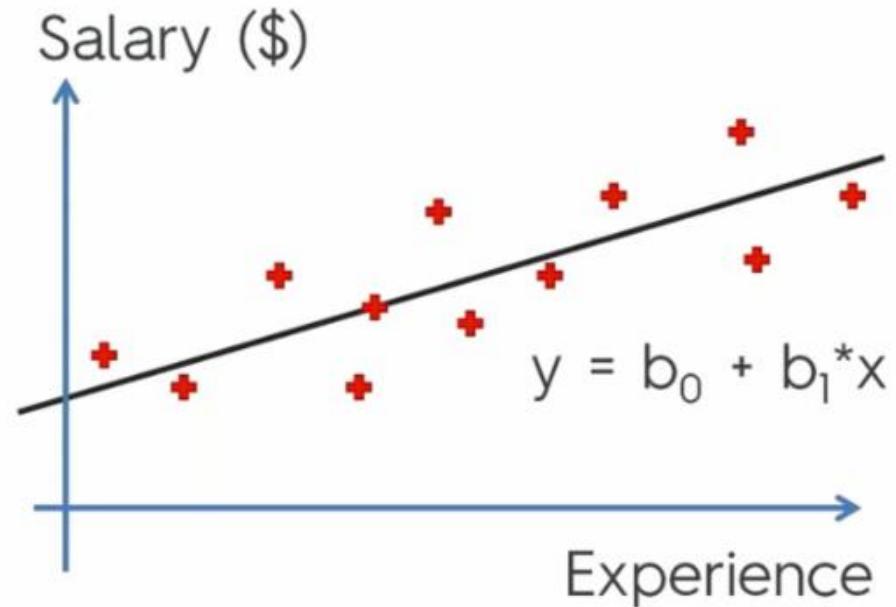
		Predicted: NO	Predicted: YES	
n=165				
Actual: NO	TN = 50	FP = 10	60	
Actual: YES	FN = 5	TP = 100	105	
	55	110		

# Logistic Regression

This is new:

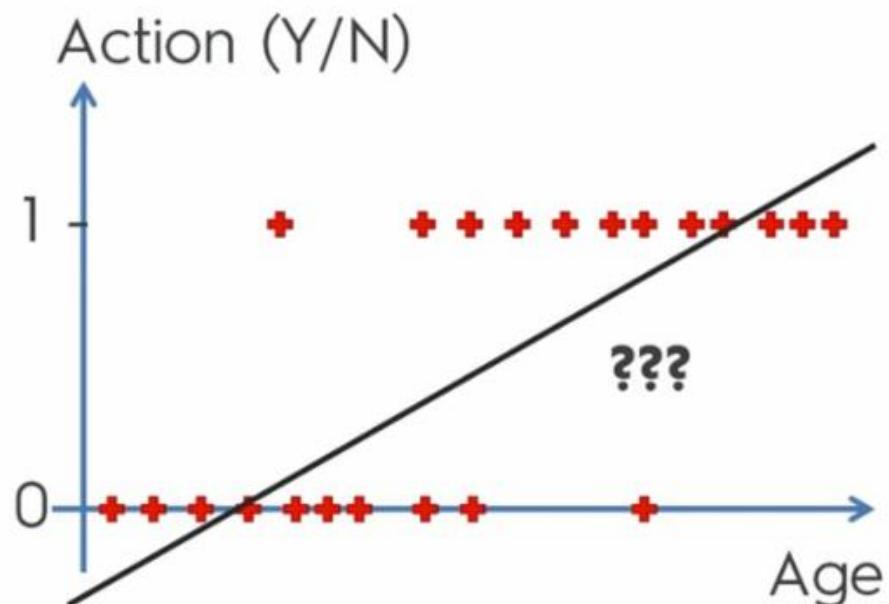


We know this:

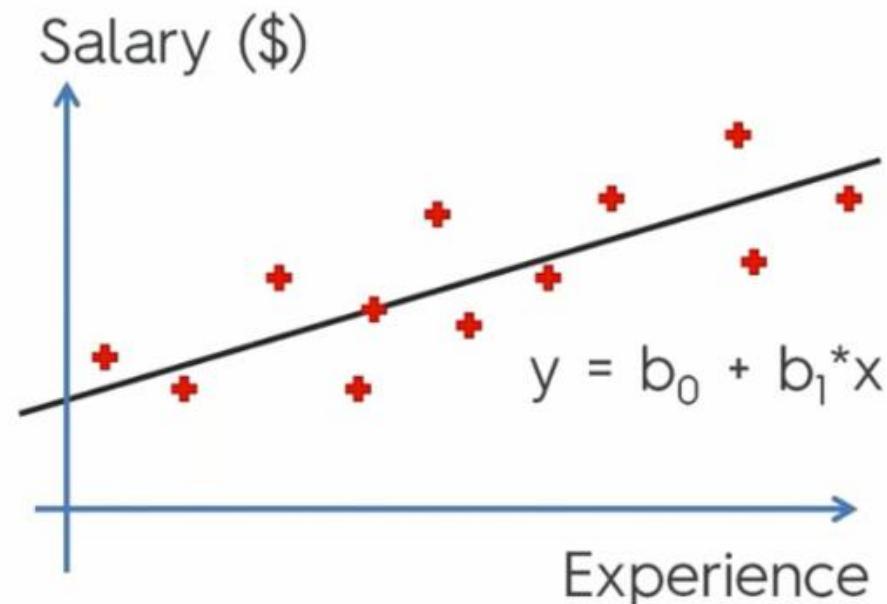


# Logistic Regression

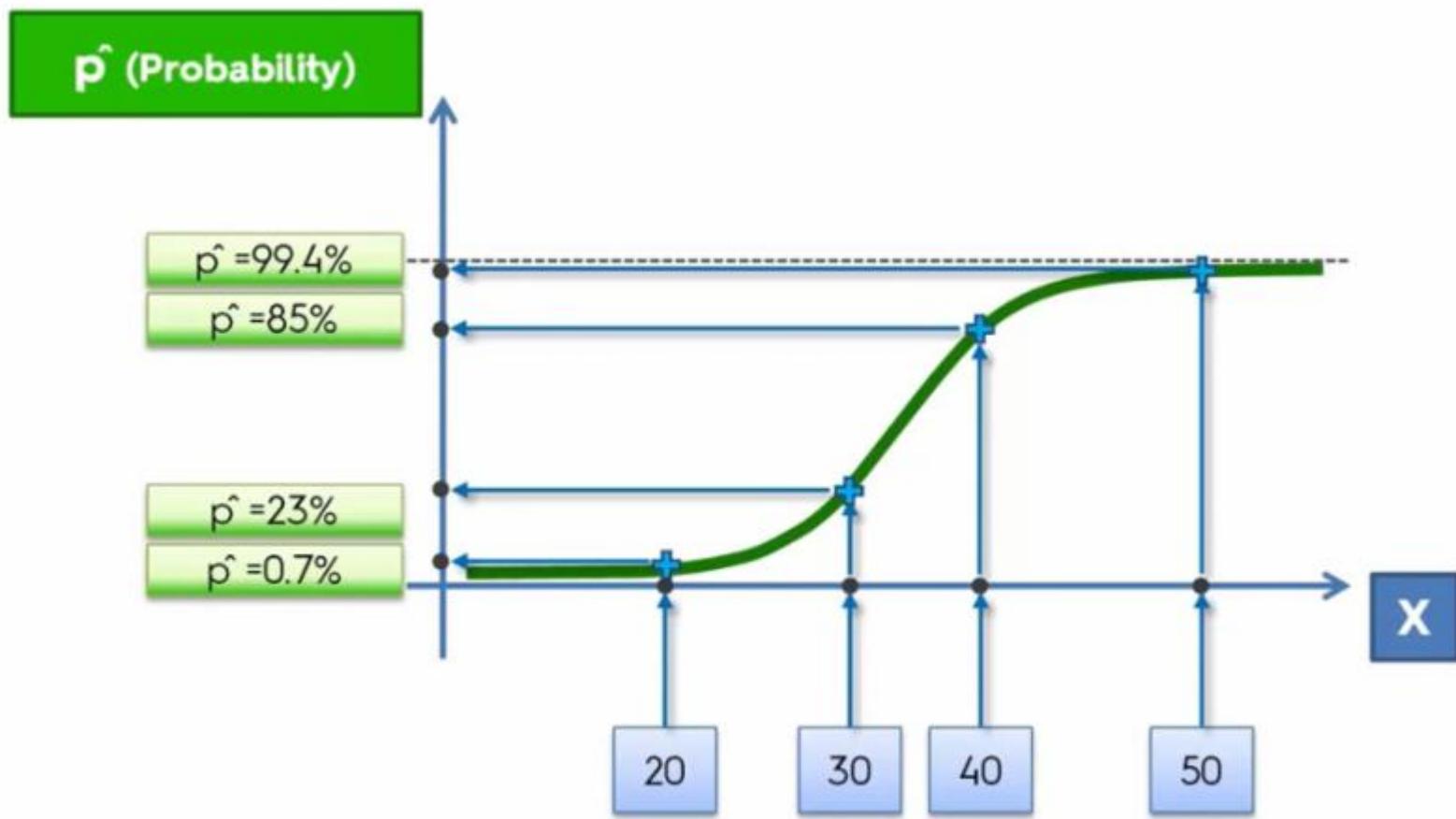
This is new:



We know this:



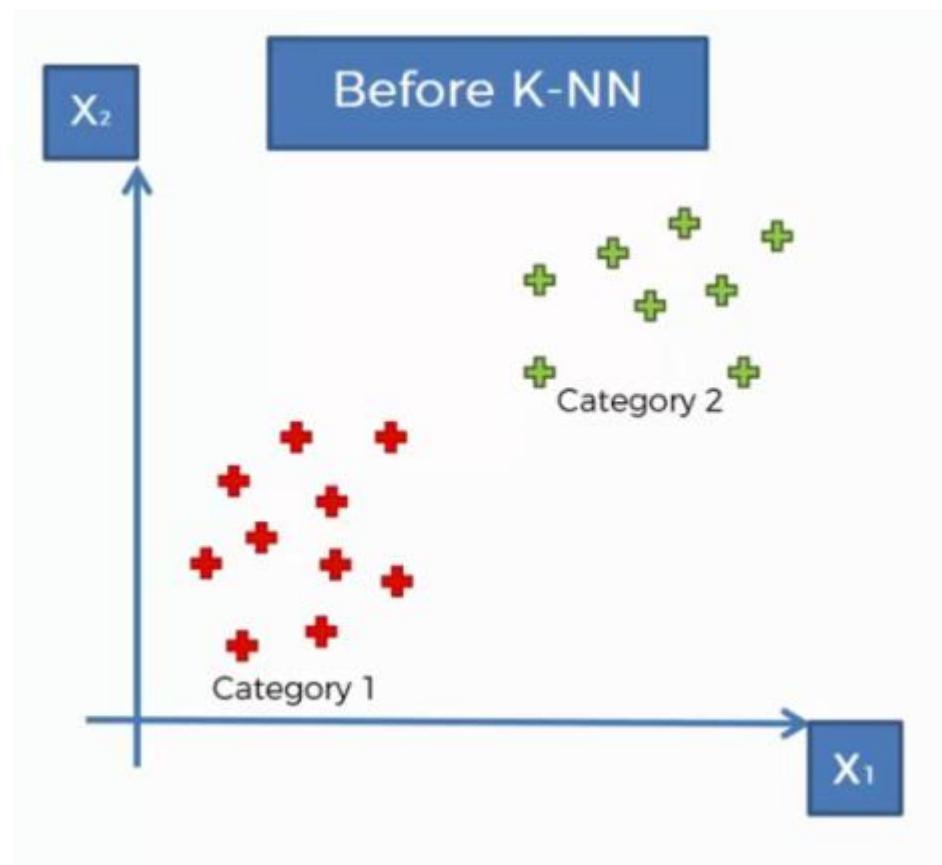
# Logistic Regression



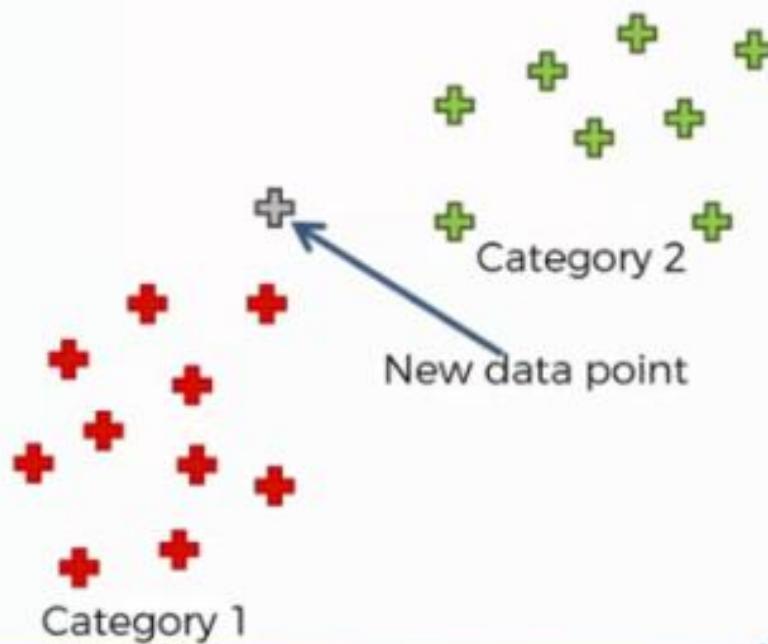
# *Feature Scaling*

- 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

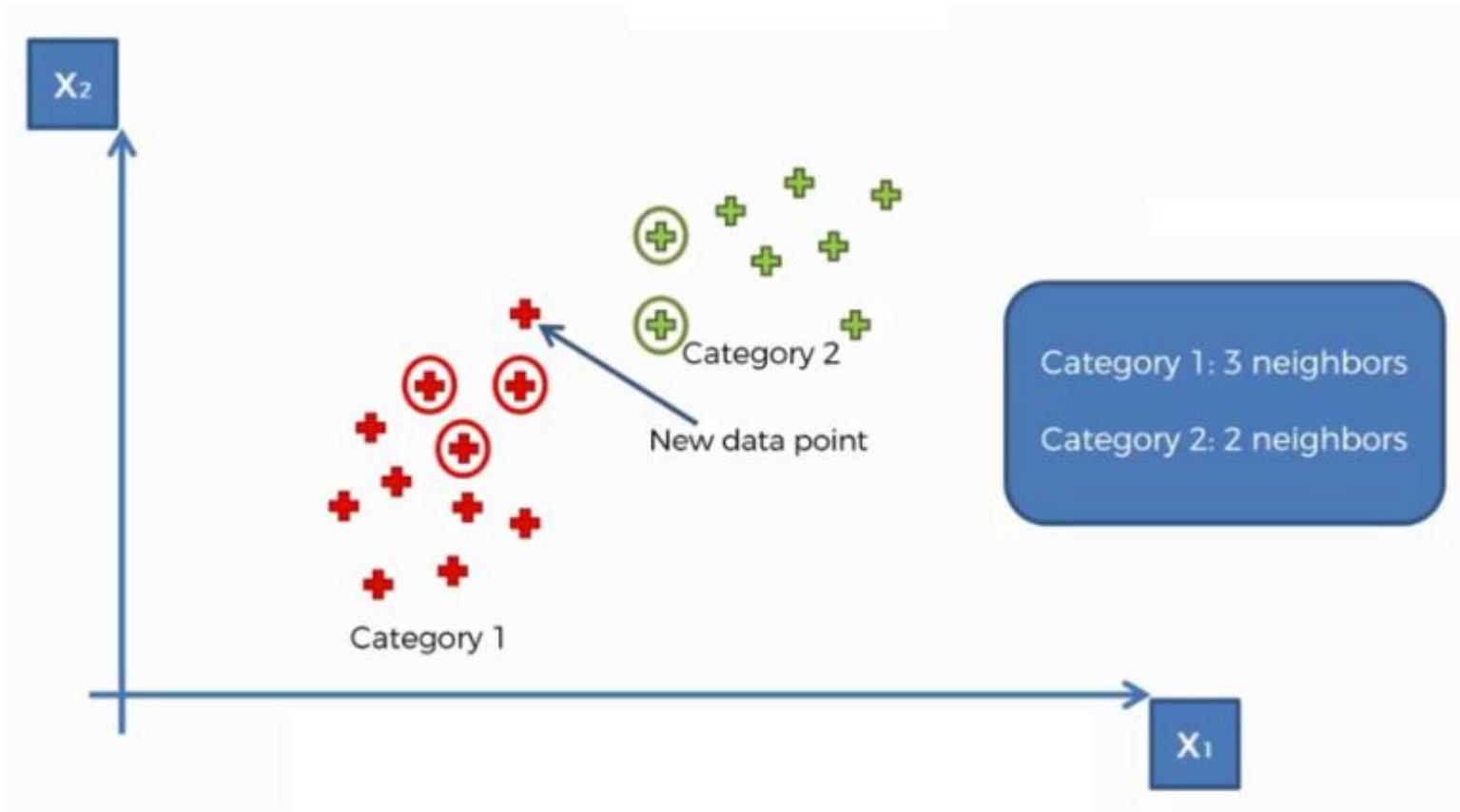


## Before K-NN

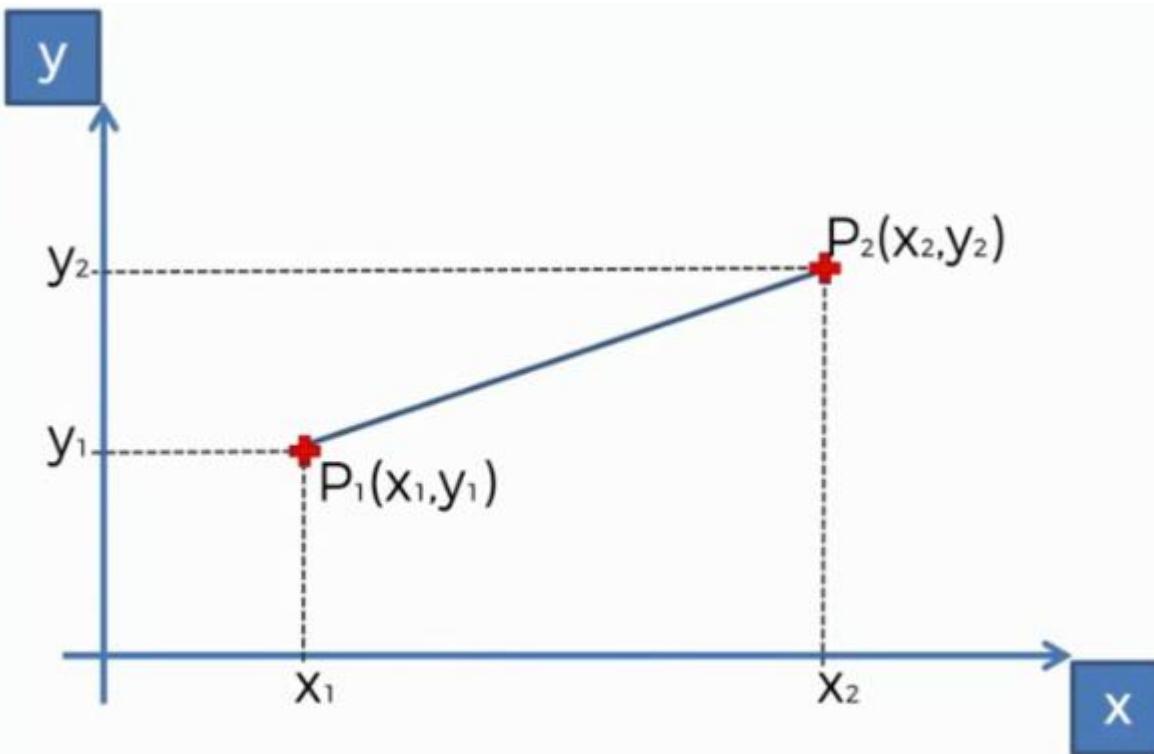


$X_1$



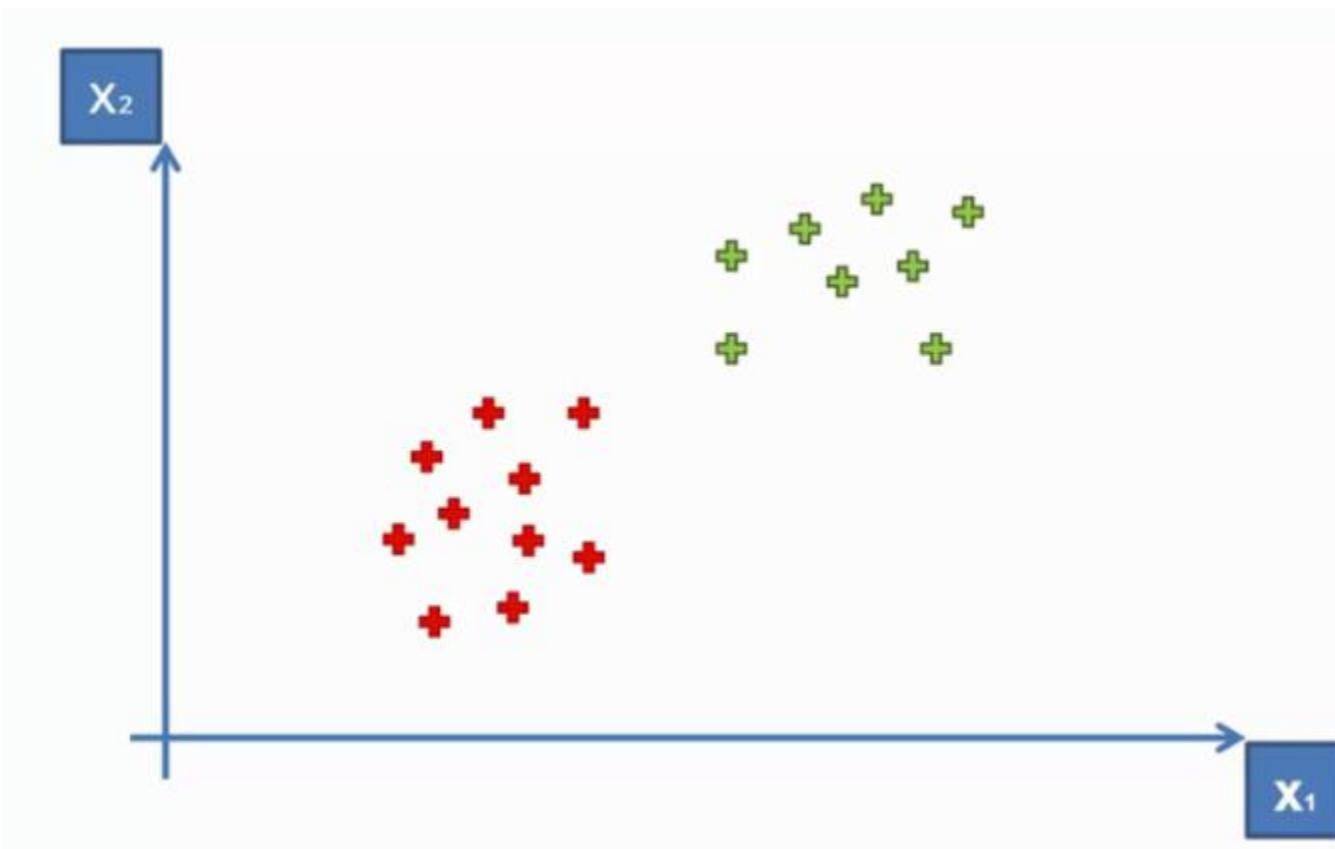


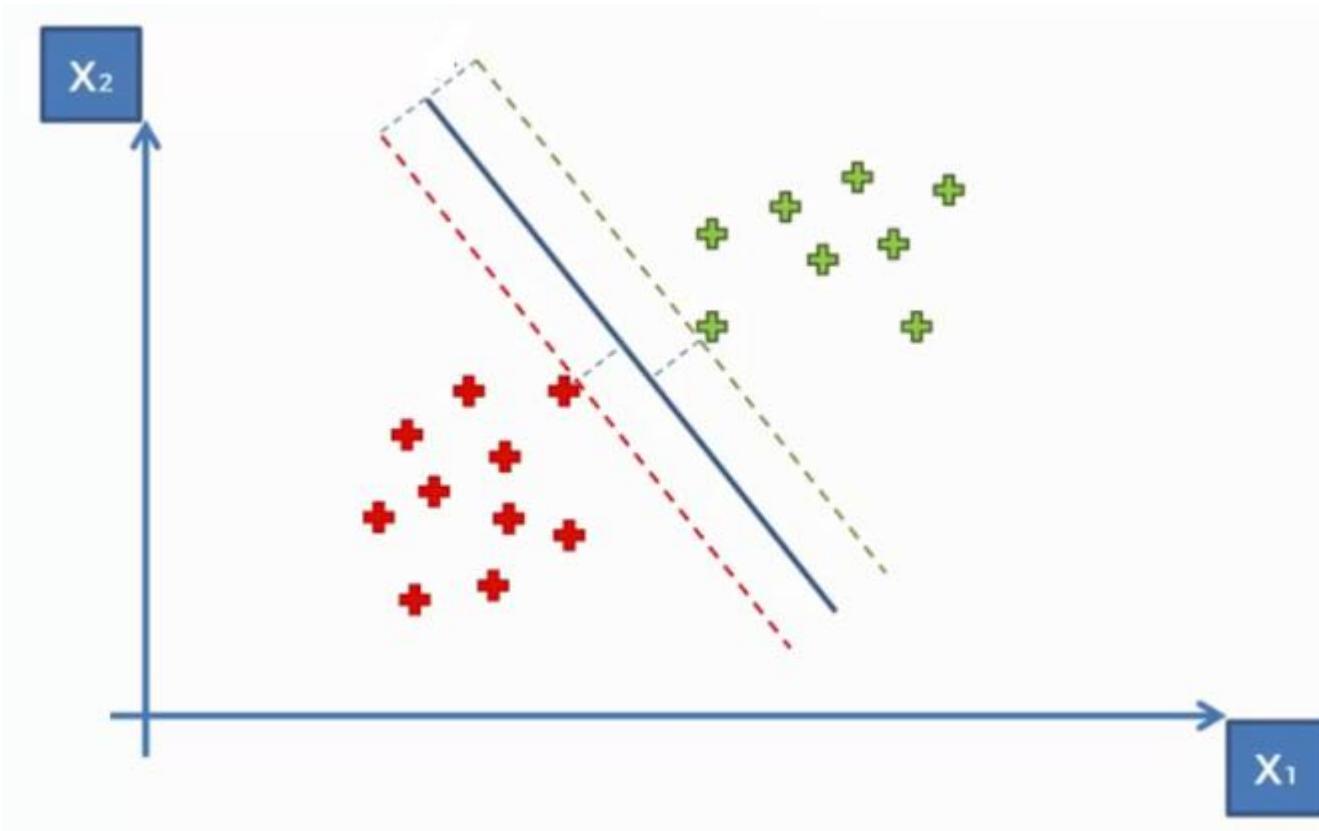
# Euclidean Distance

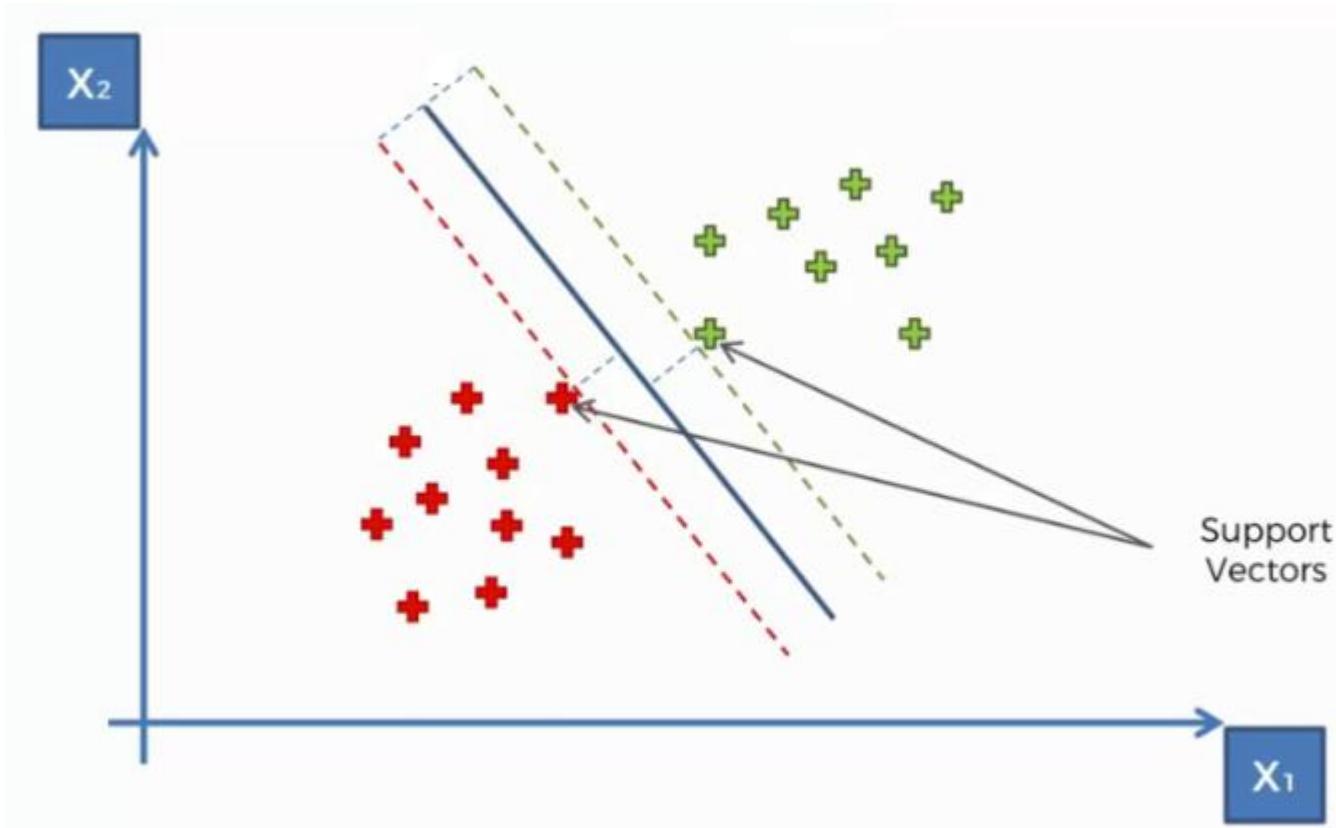


$$\text{Euclidean Distance between } P_1 \text{ and } P_2 = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$$

# Support Vector Machine

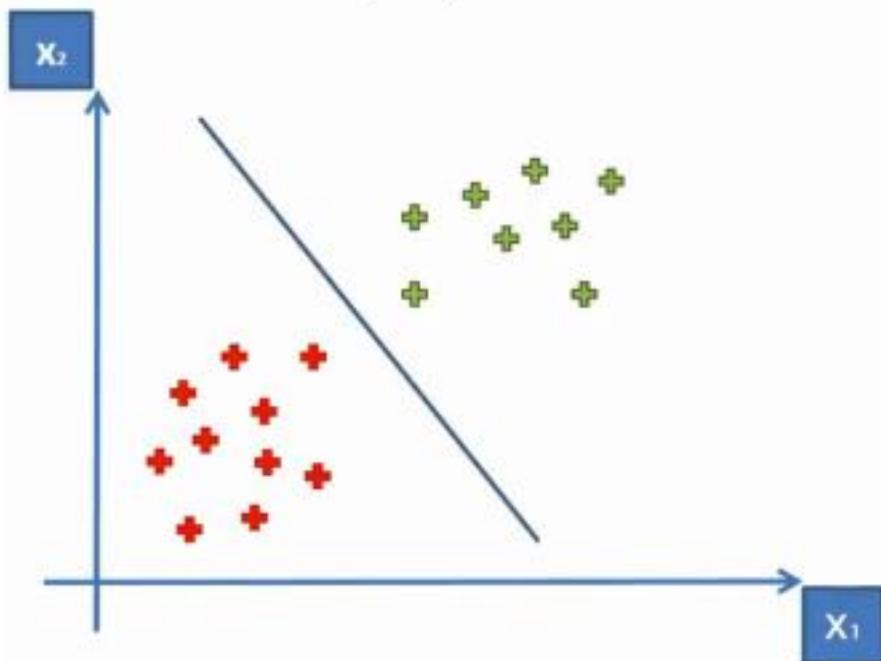




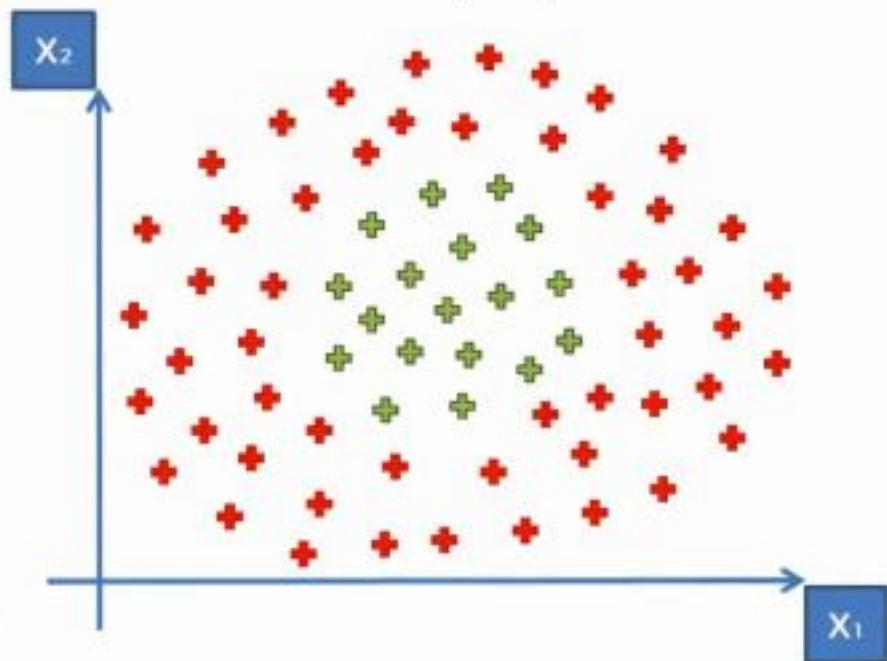


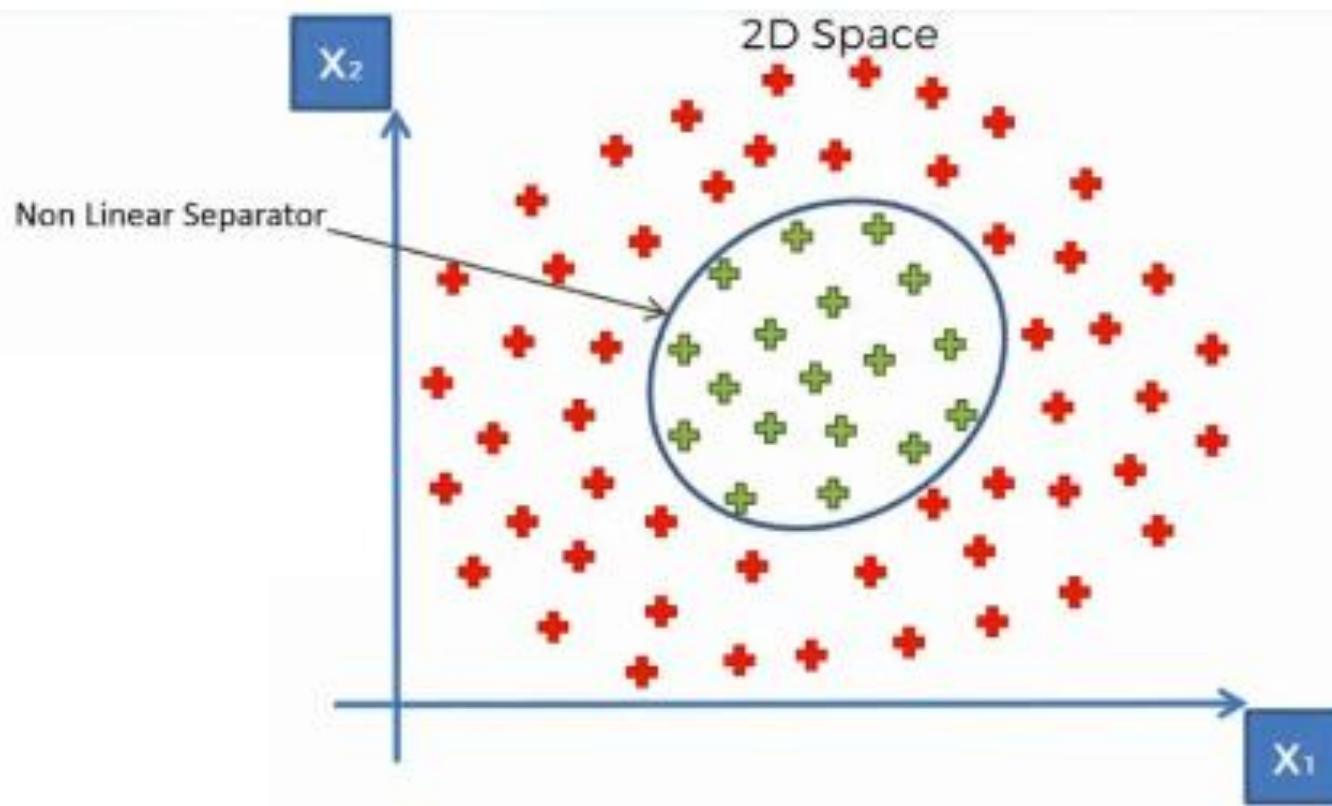
# Kernel SVM

Linearly Separable



Not Linearly Separable





# Naive Bayes

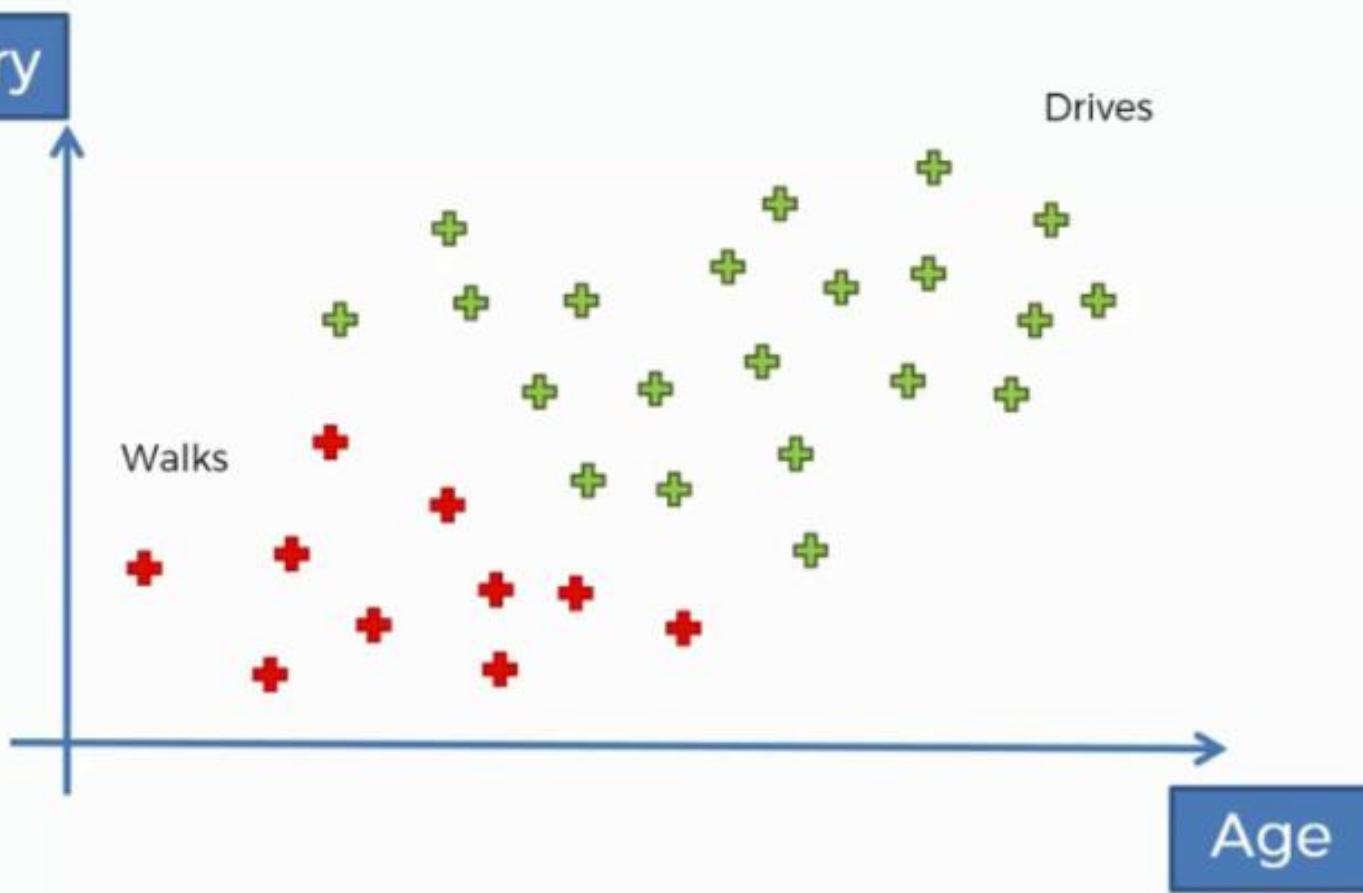
$$P(A|B) = \frac{P(B|A) * P(A)}{P(B)}$$

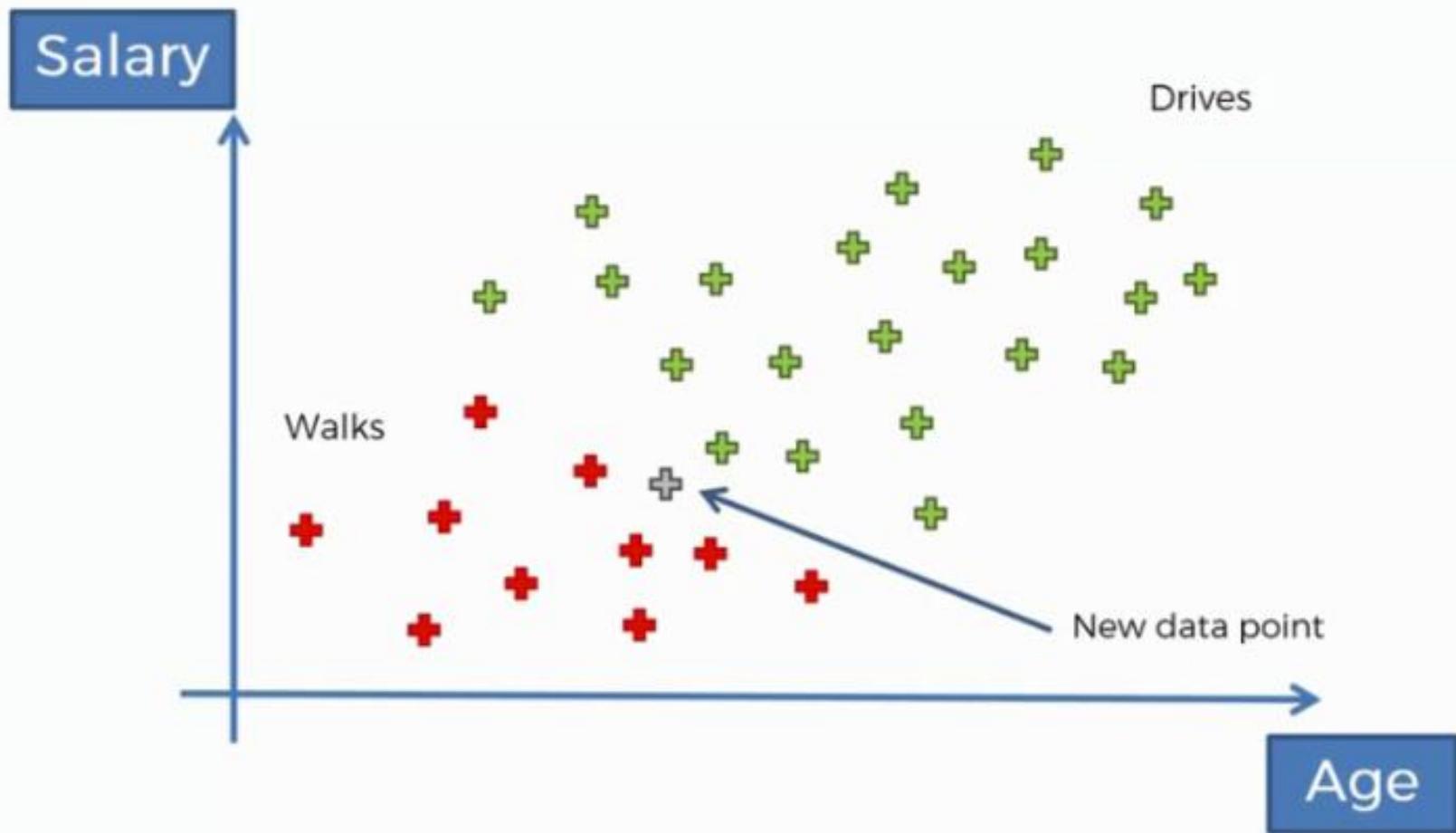
Salary

Drives

Walks

Age





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

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

#4

Posterior Probability

#3

Likelihood

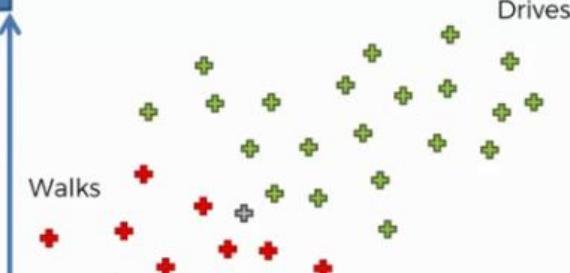
#1

Prior Probability

#2

Marginal Likelihood

Salary



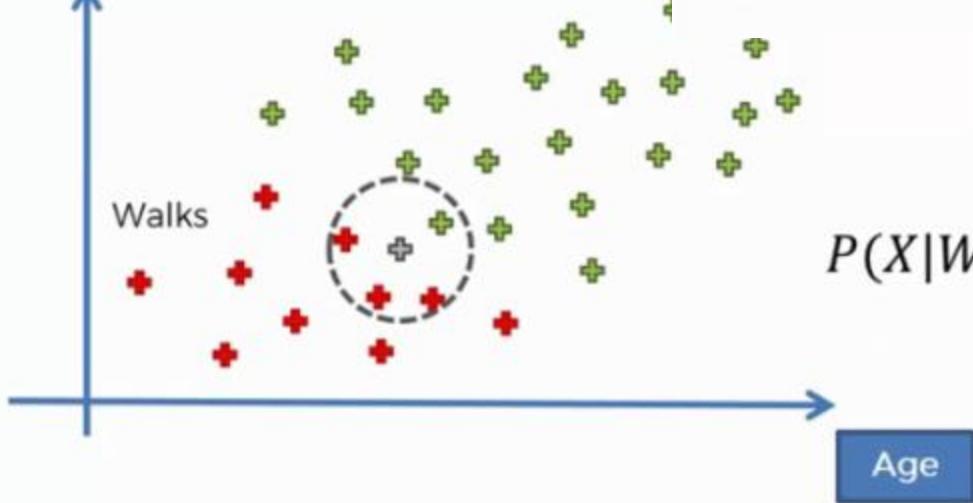
$$P(\text{Walks}) = \frac{10}{30}$$

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

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

Salary

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



*Number of Similar  
Observations*

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

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

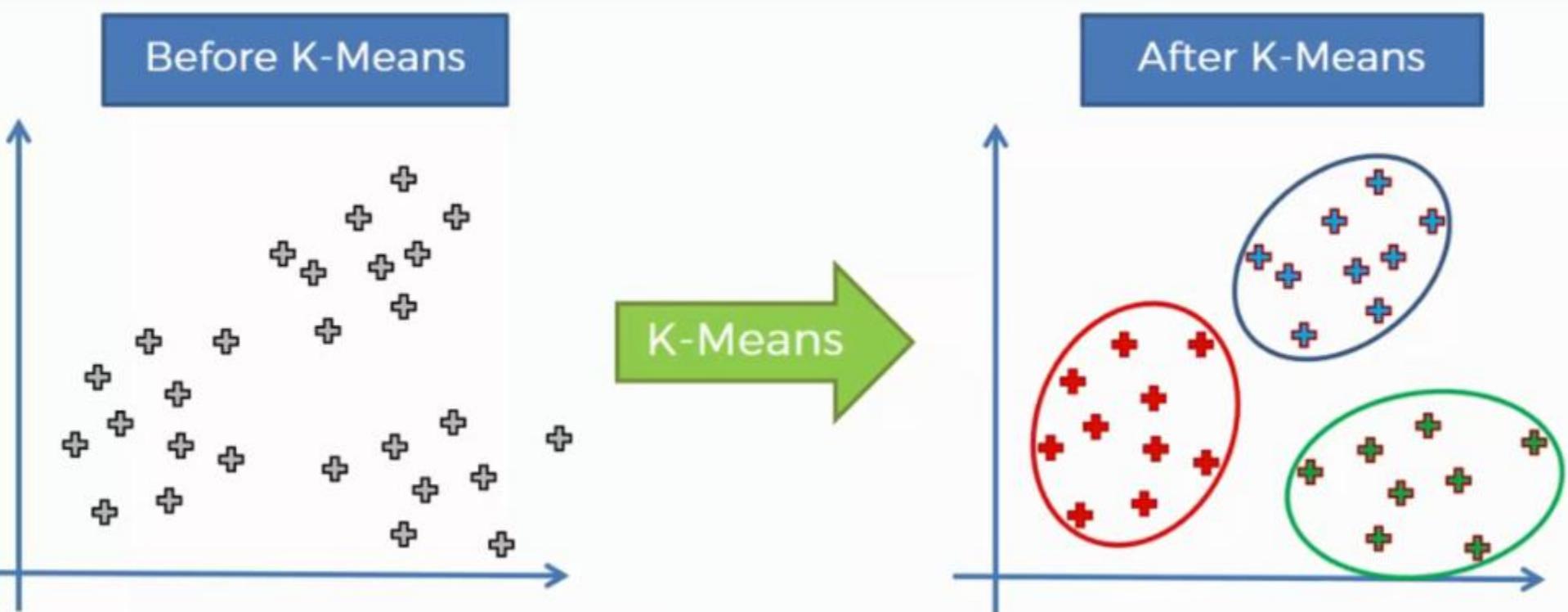
$P(Walks|X)$  v.s.  $P(Drives|X)$

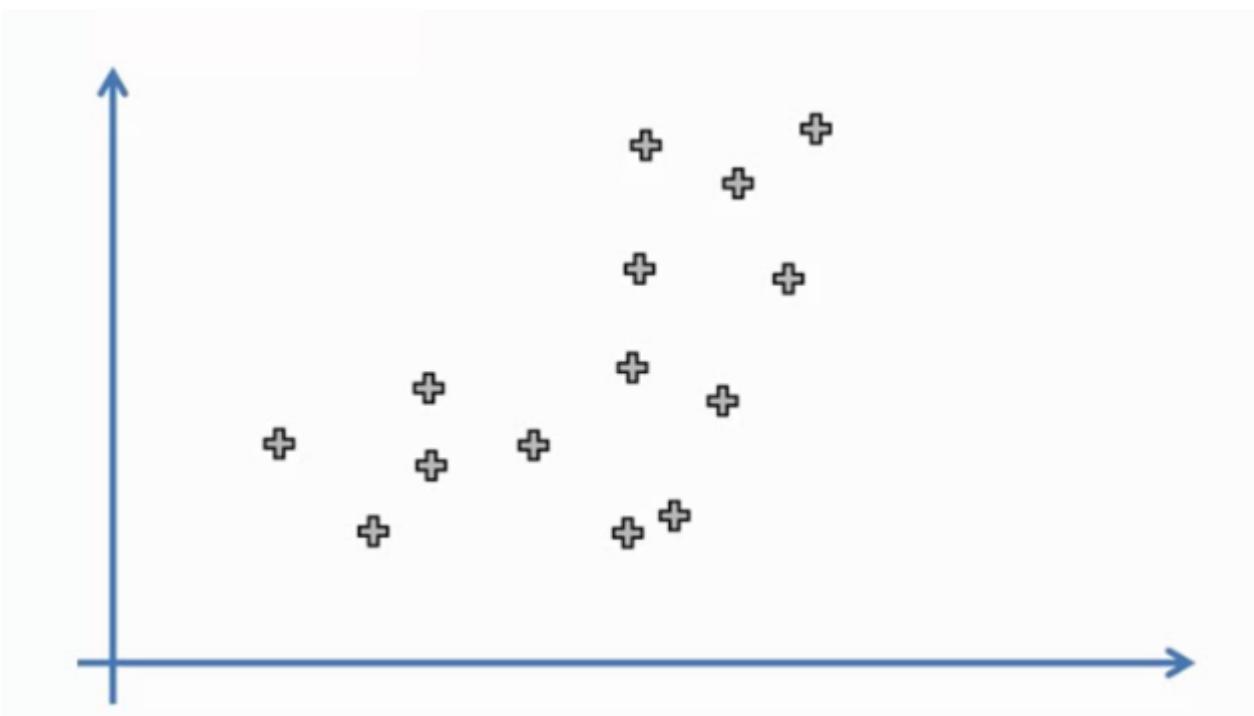
$$P(Drives|X) = \frac{\frac{1}{20} * \frac{20}{30}}{\frac{4}{30}} = 0.25$$

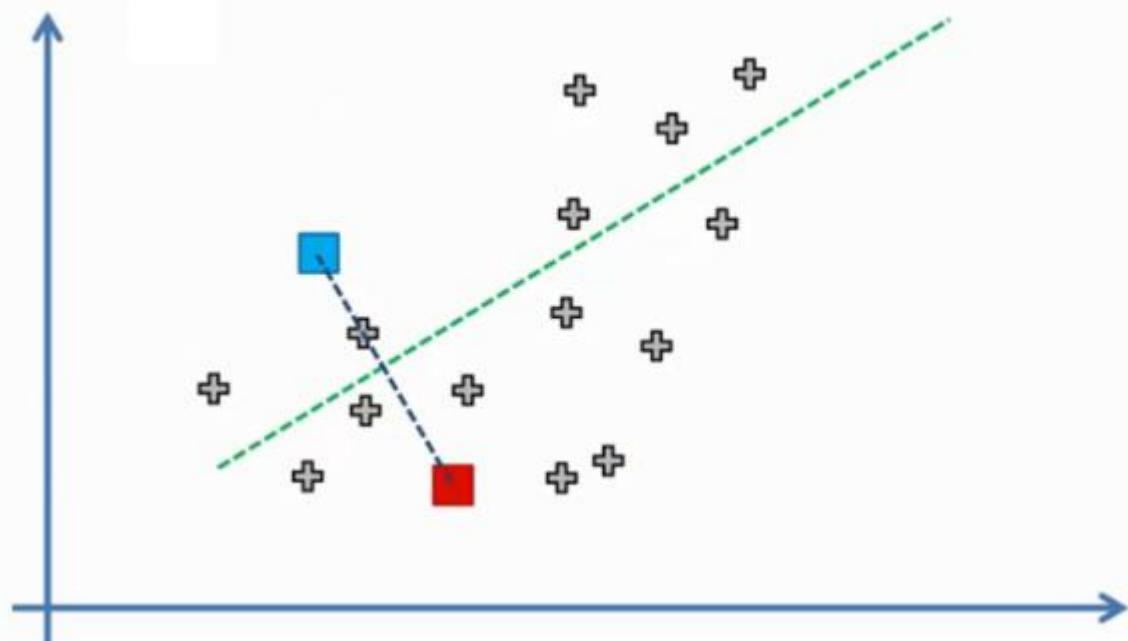
# Clustering

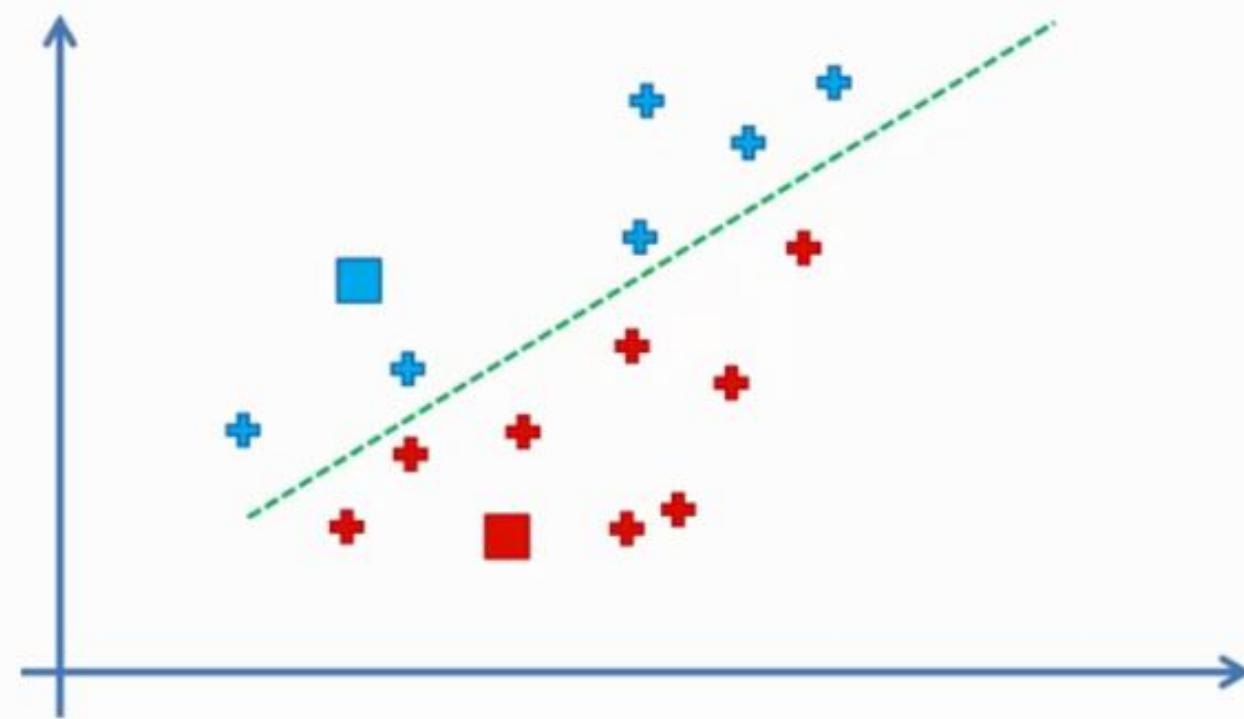
## Unsupervised

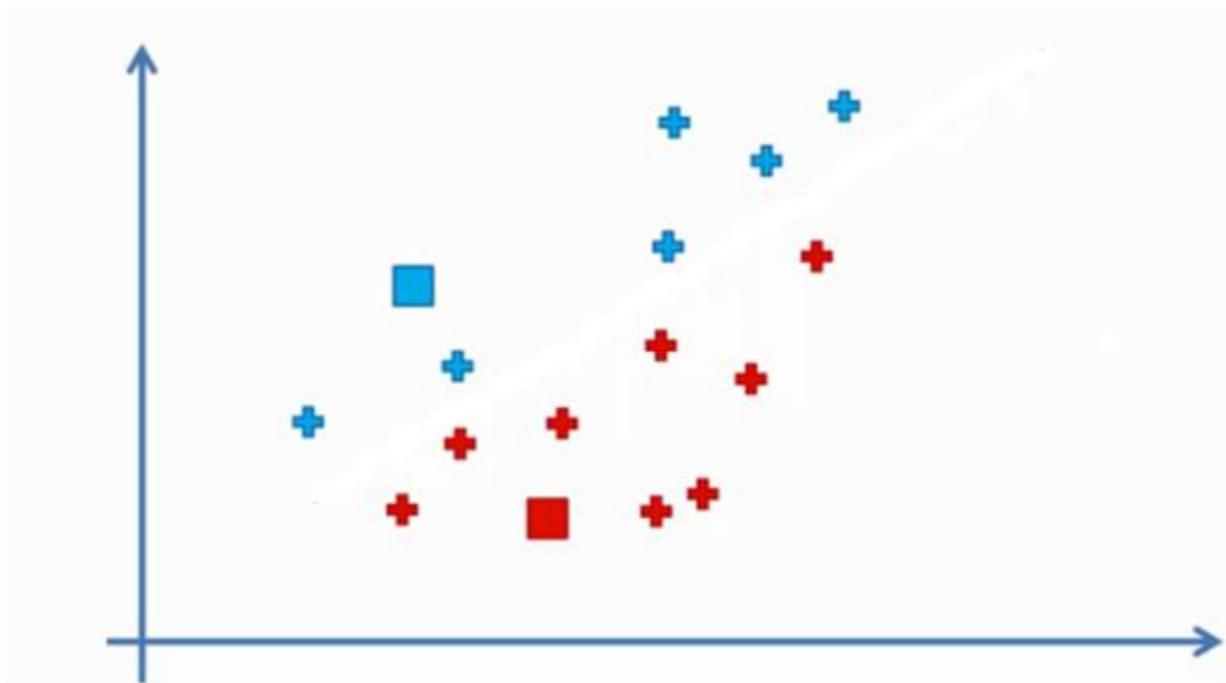
# K-means

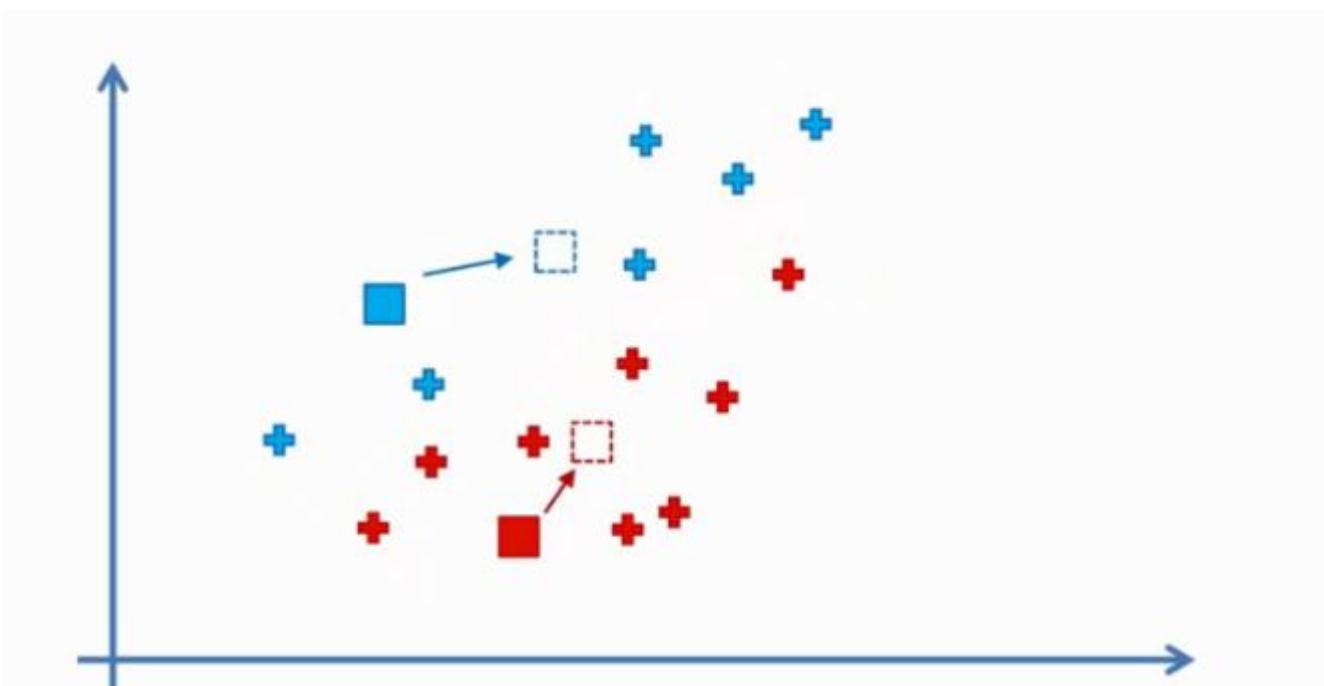


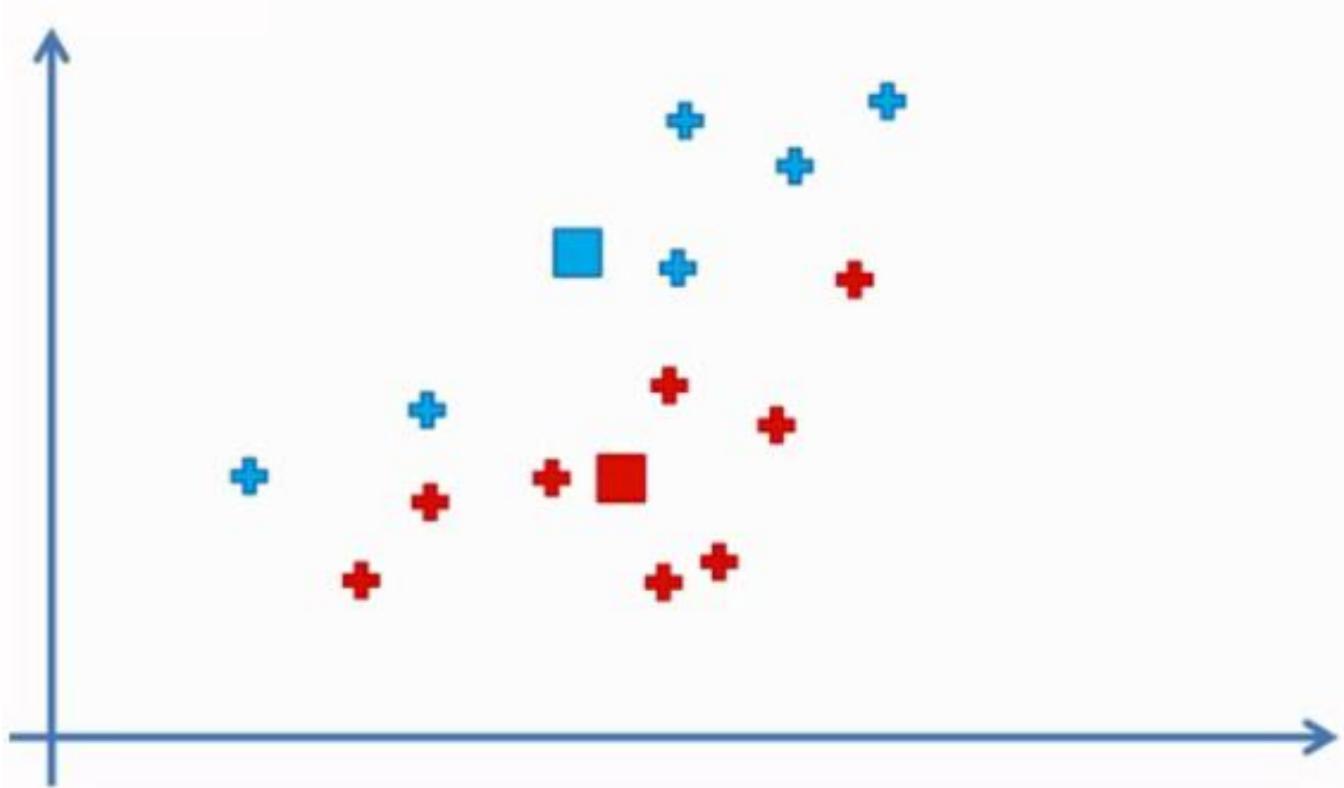


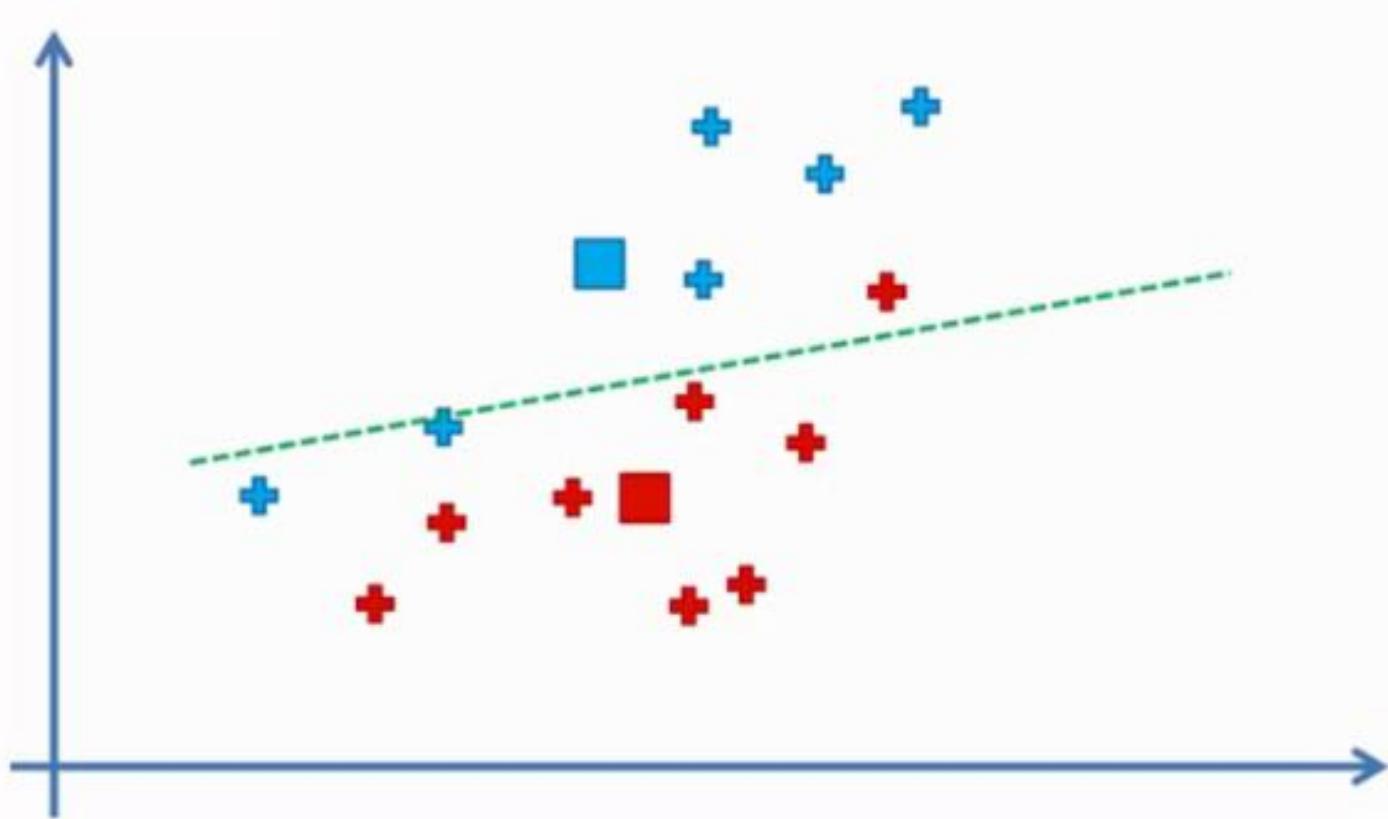


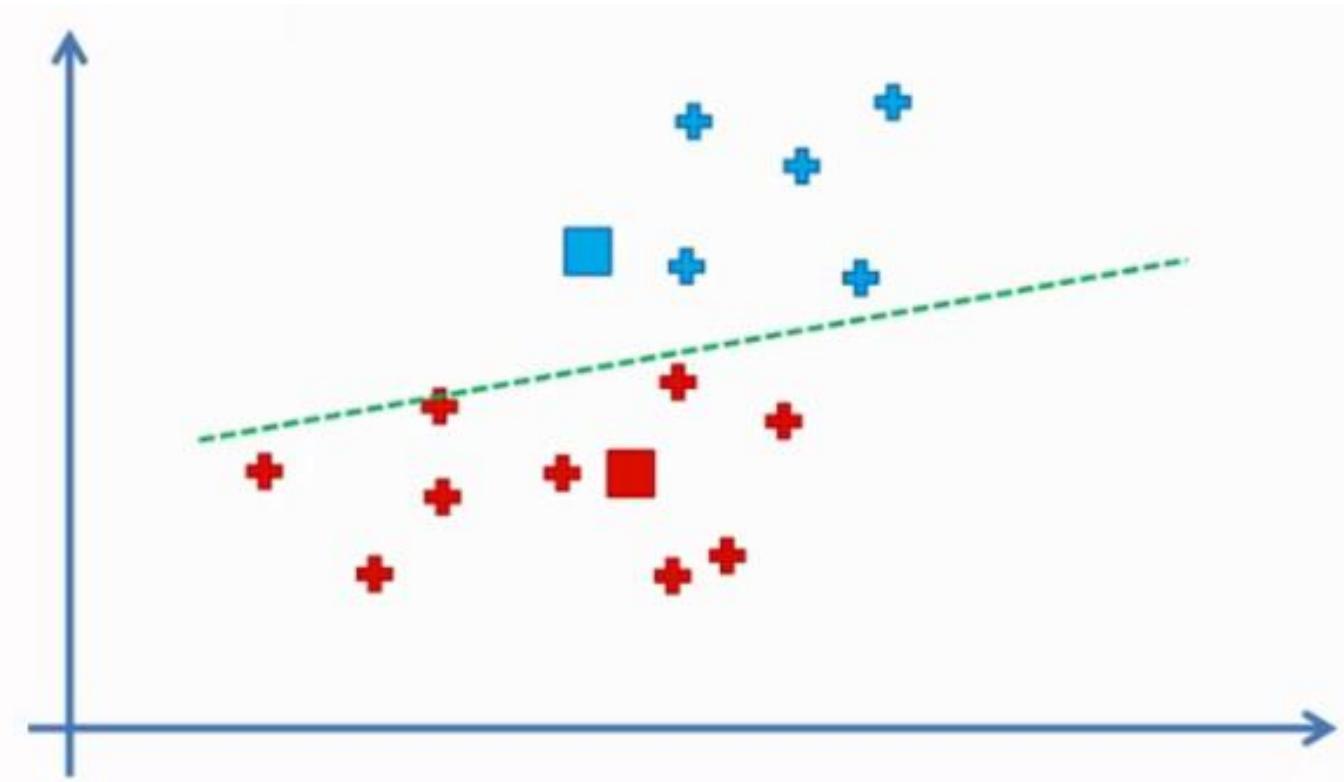


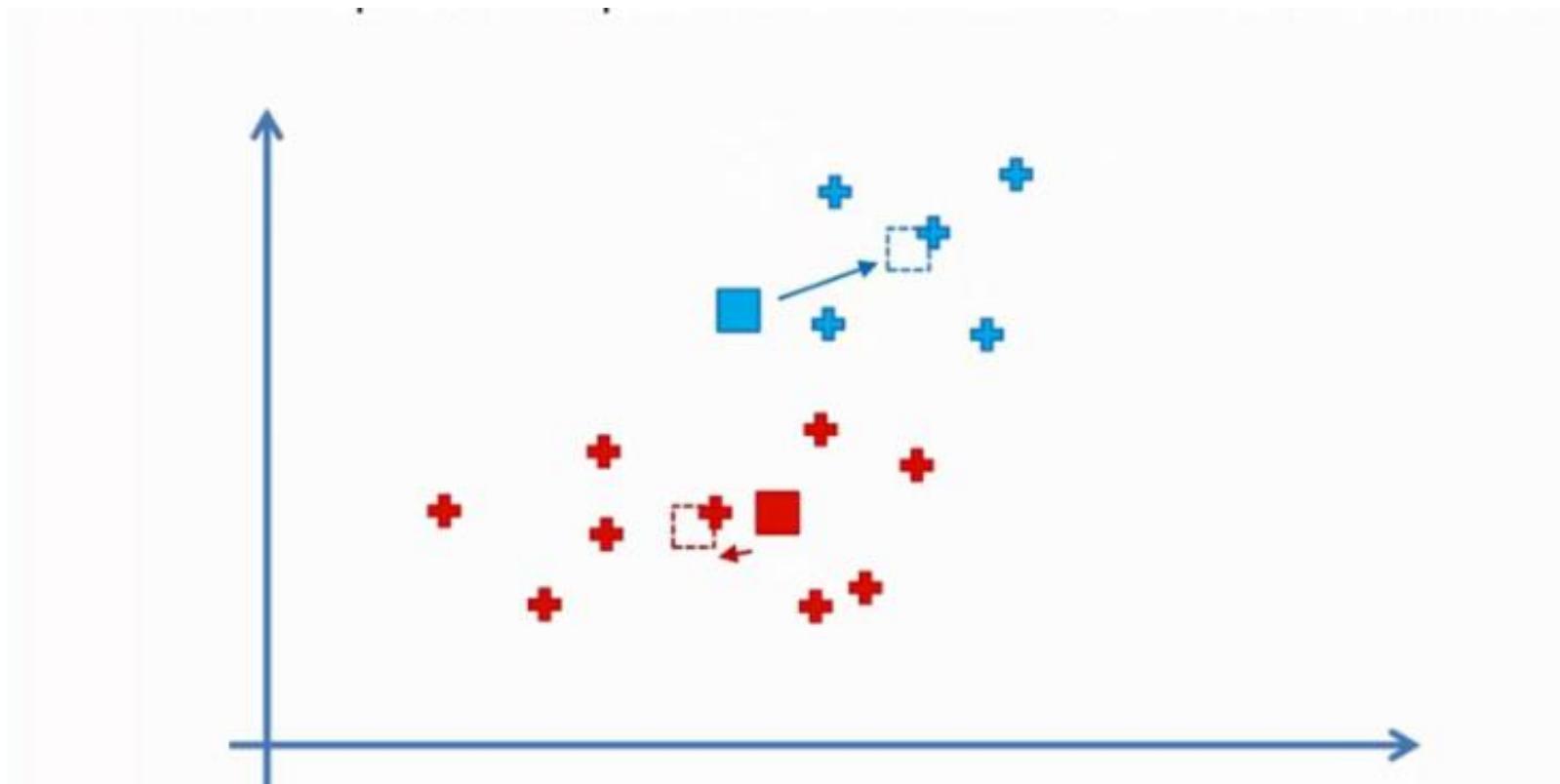


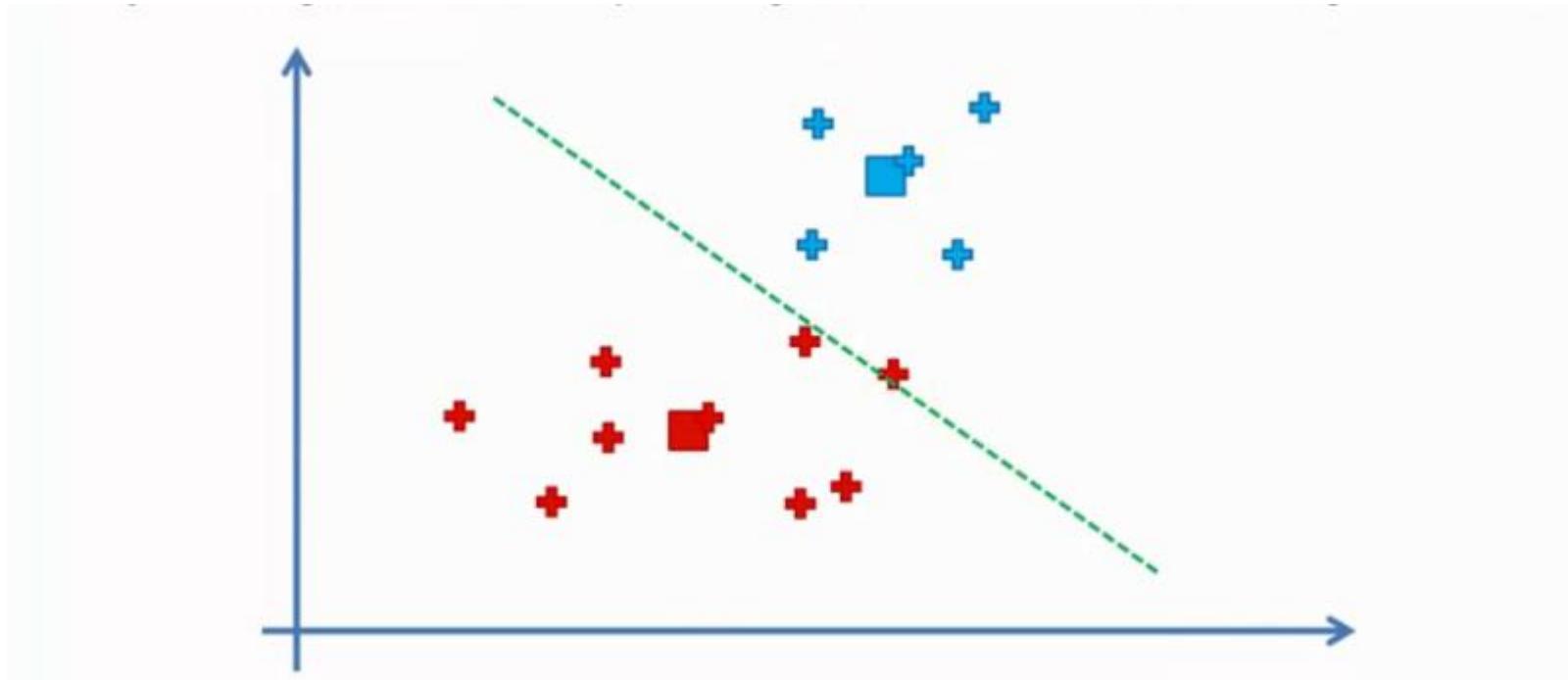


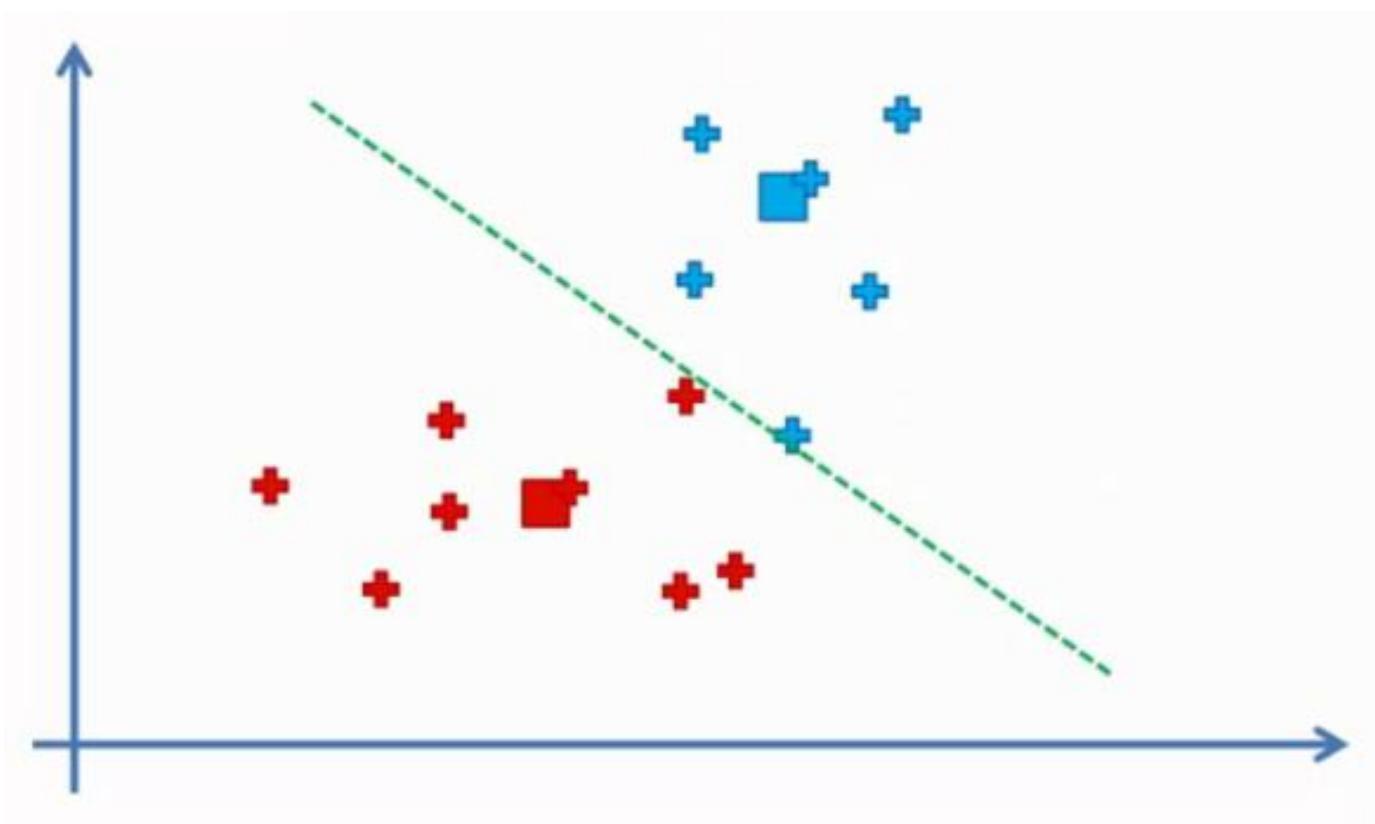


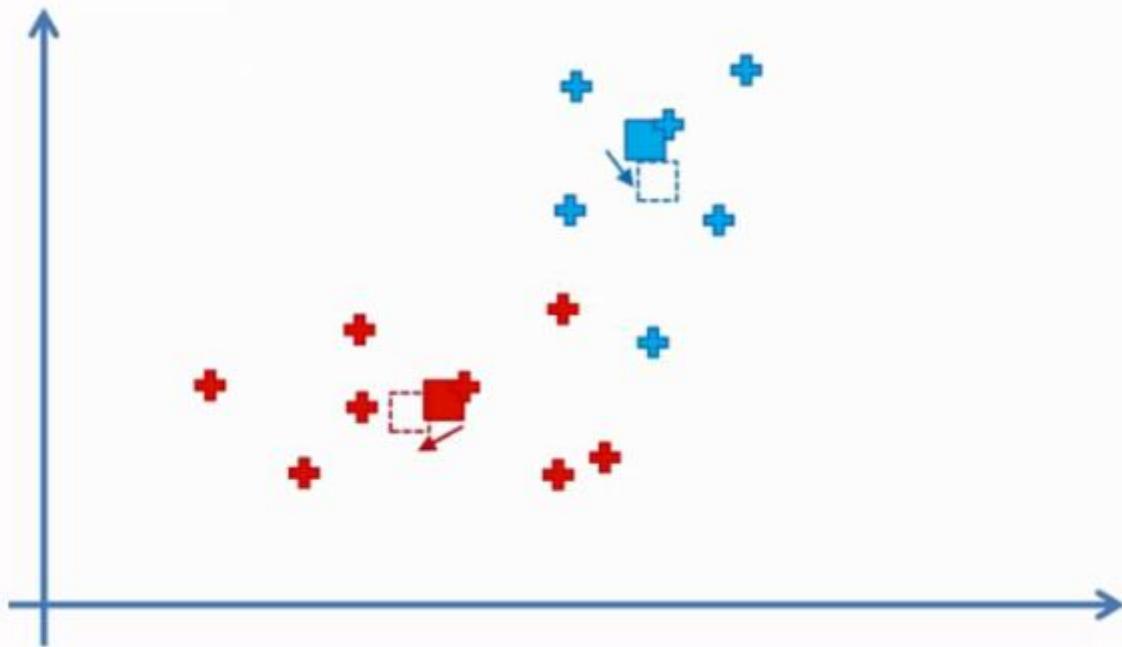


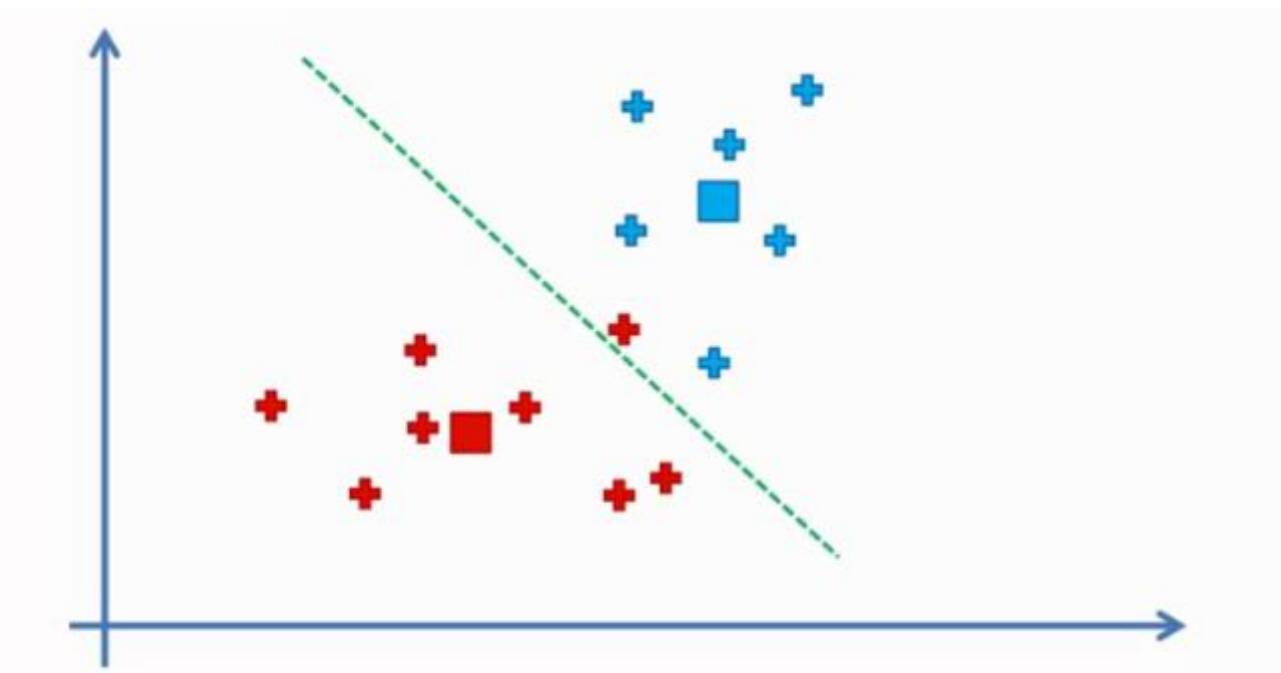


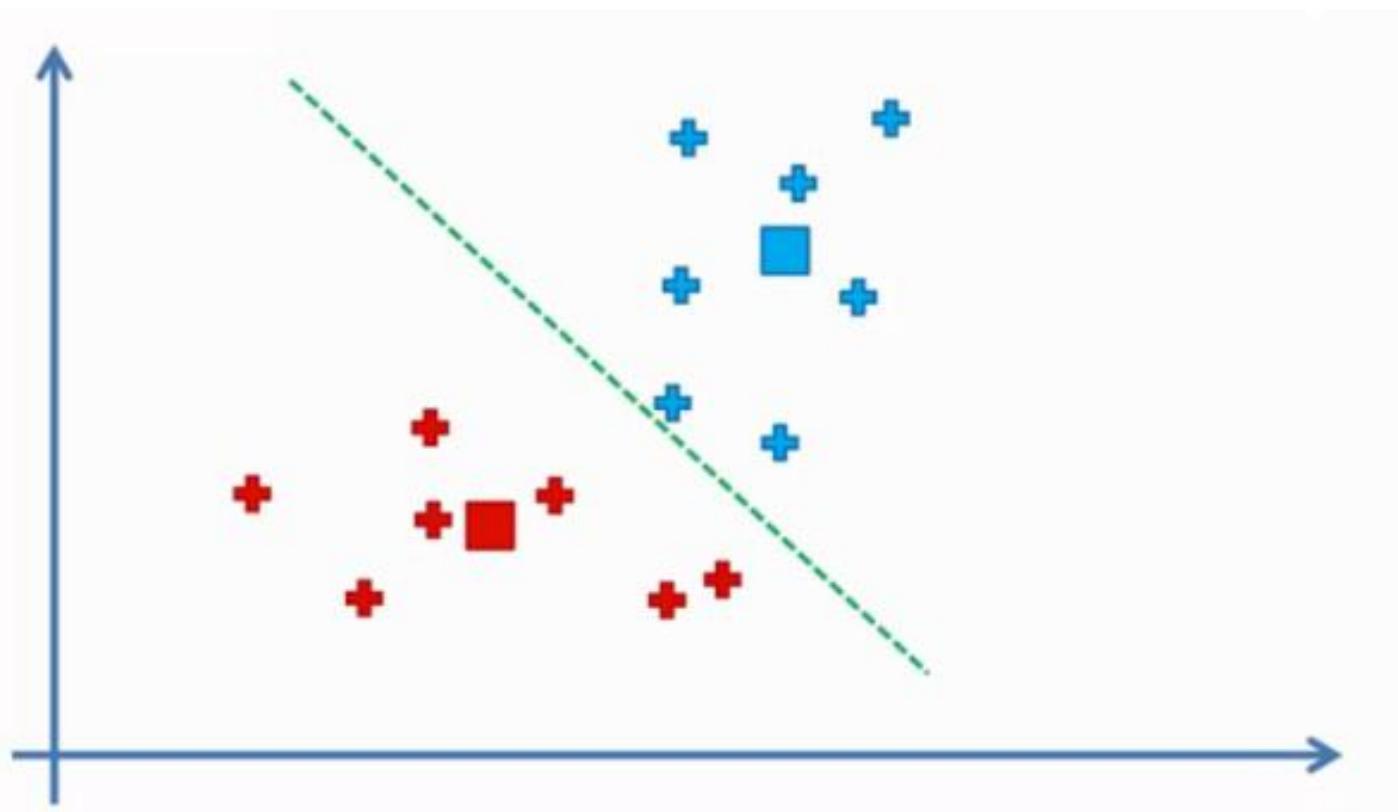


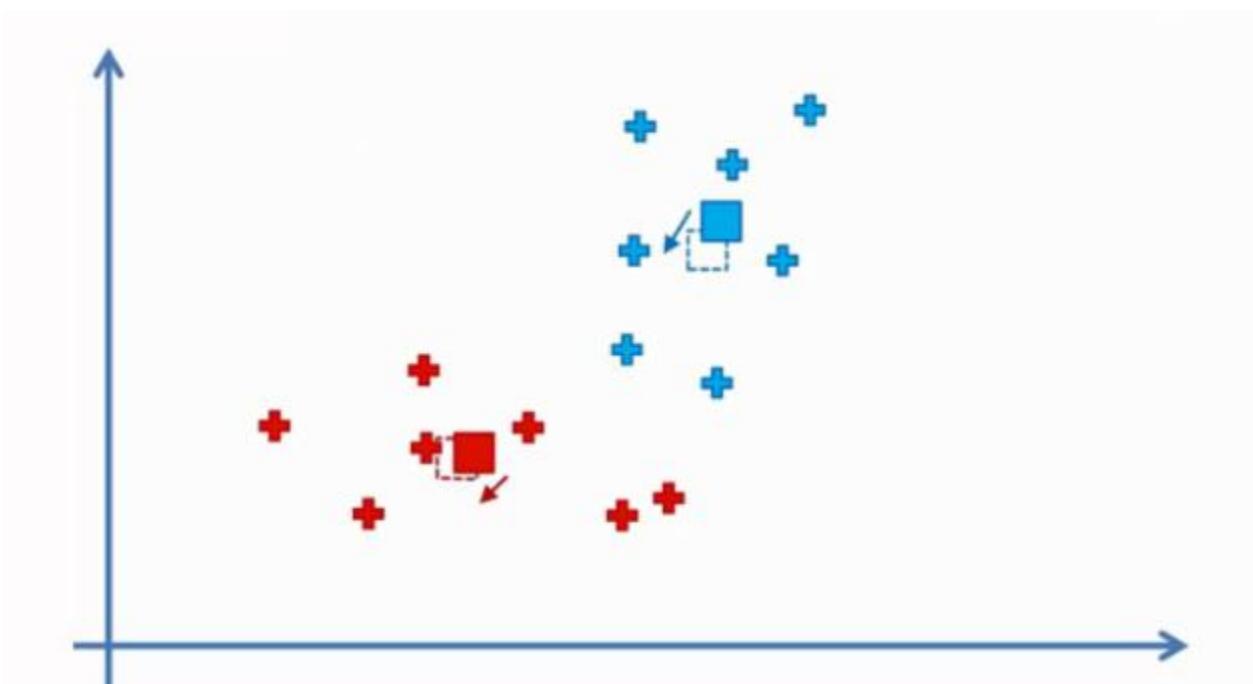


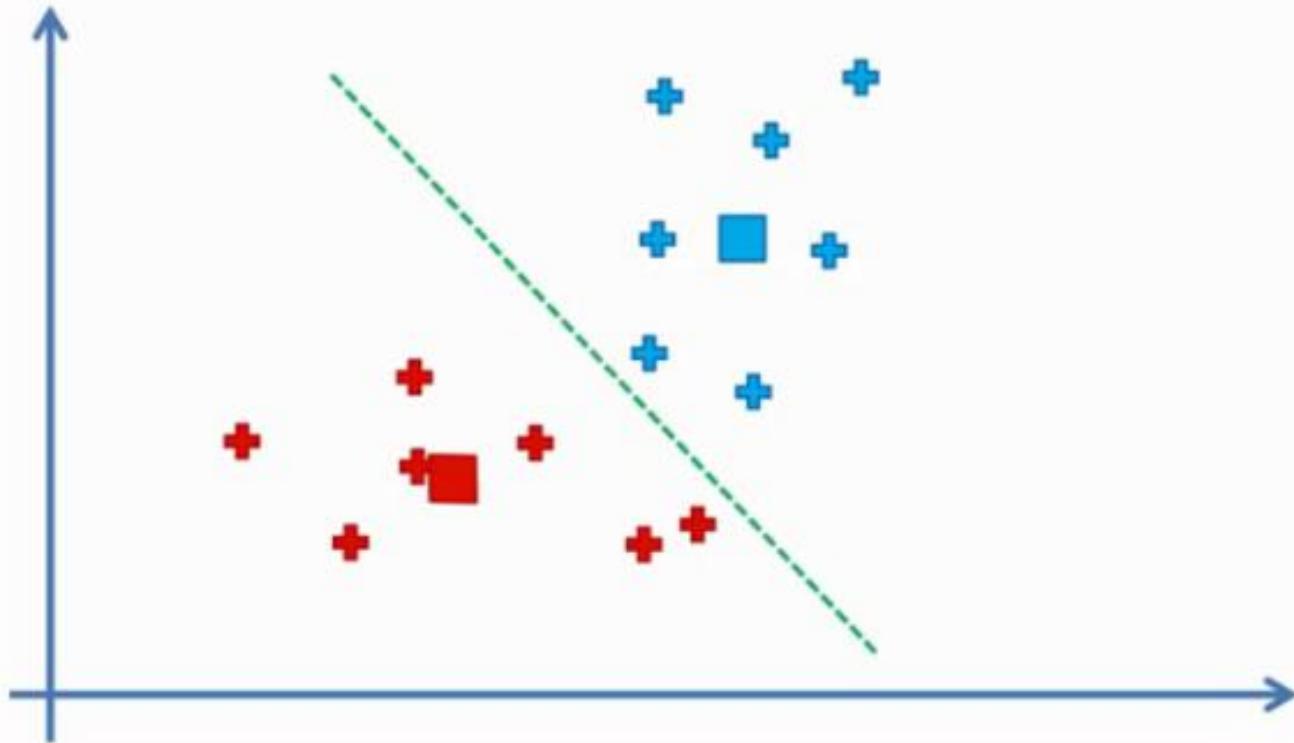


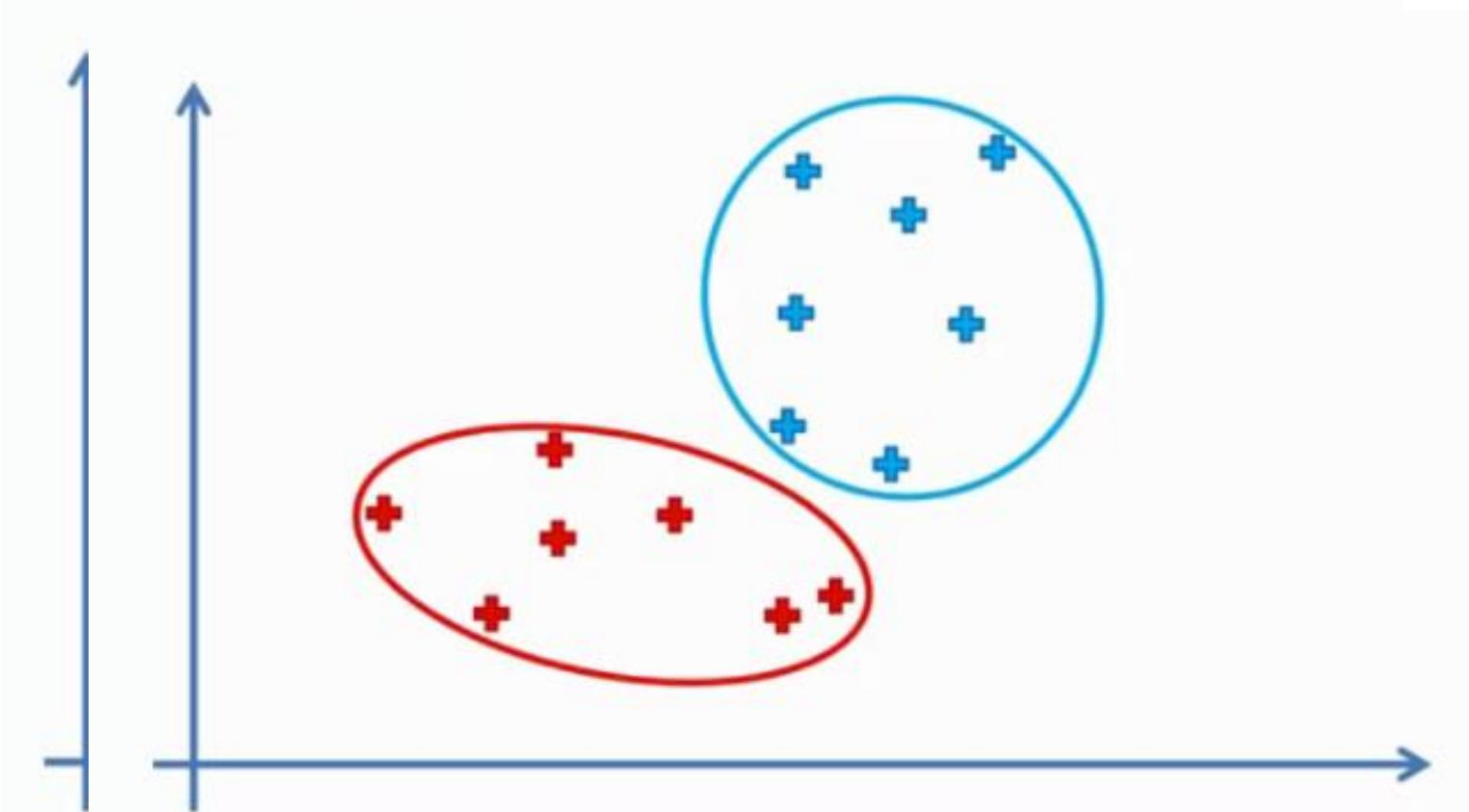


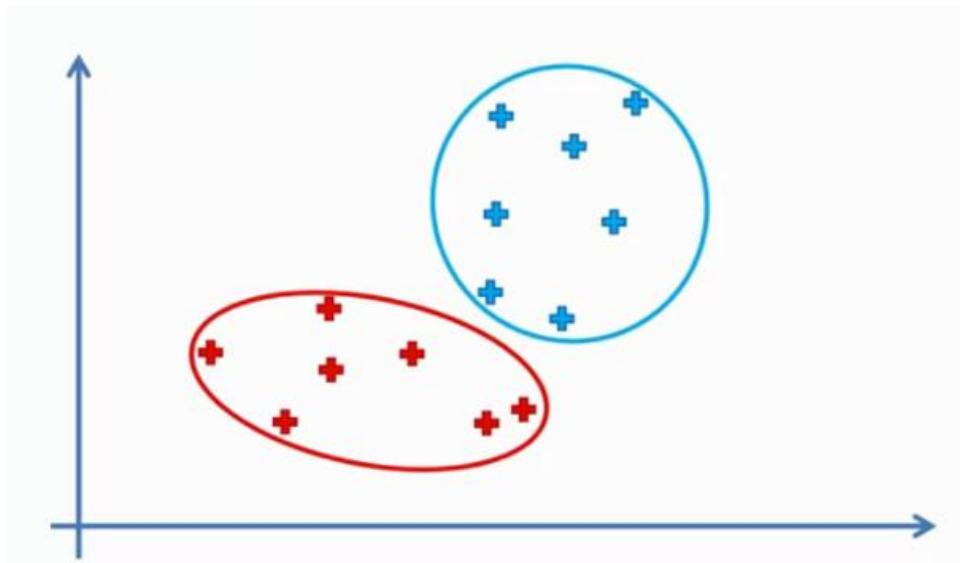
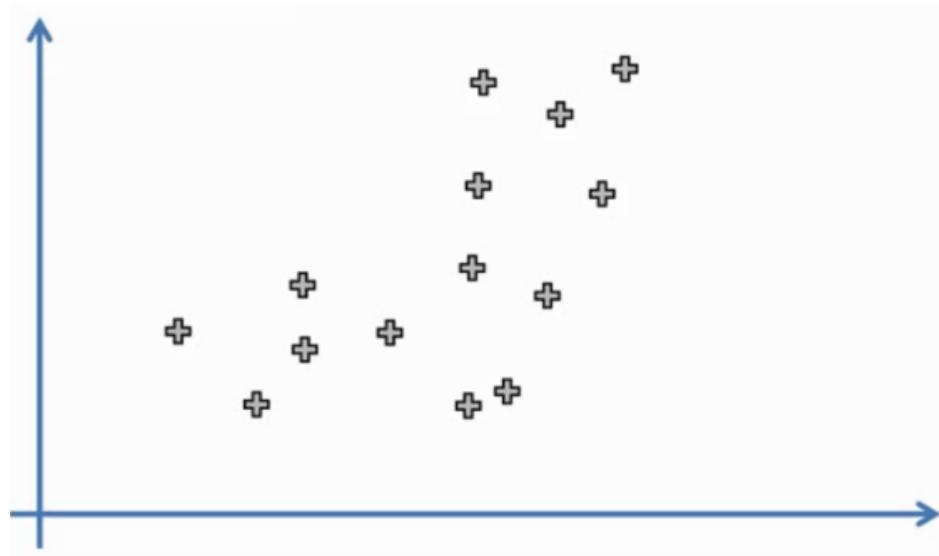












# Upper Bound Confidence

- ❖ 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  $n$

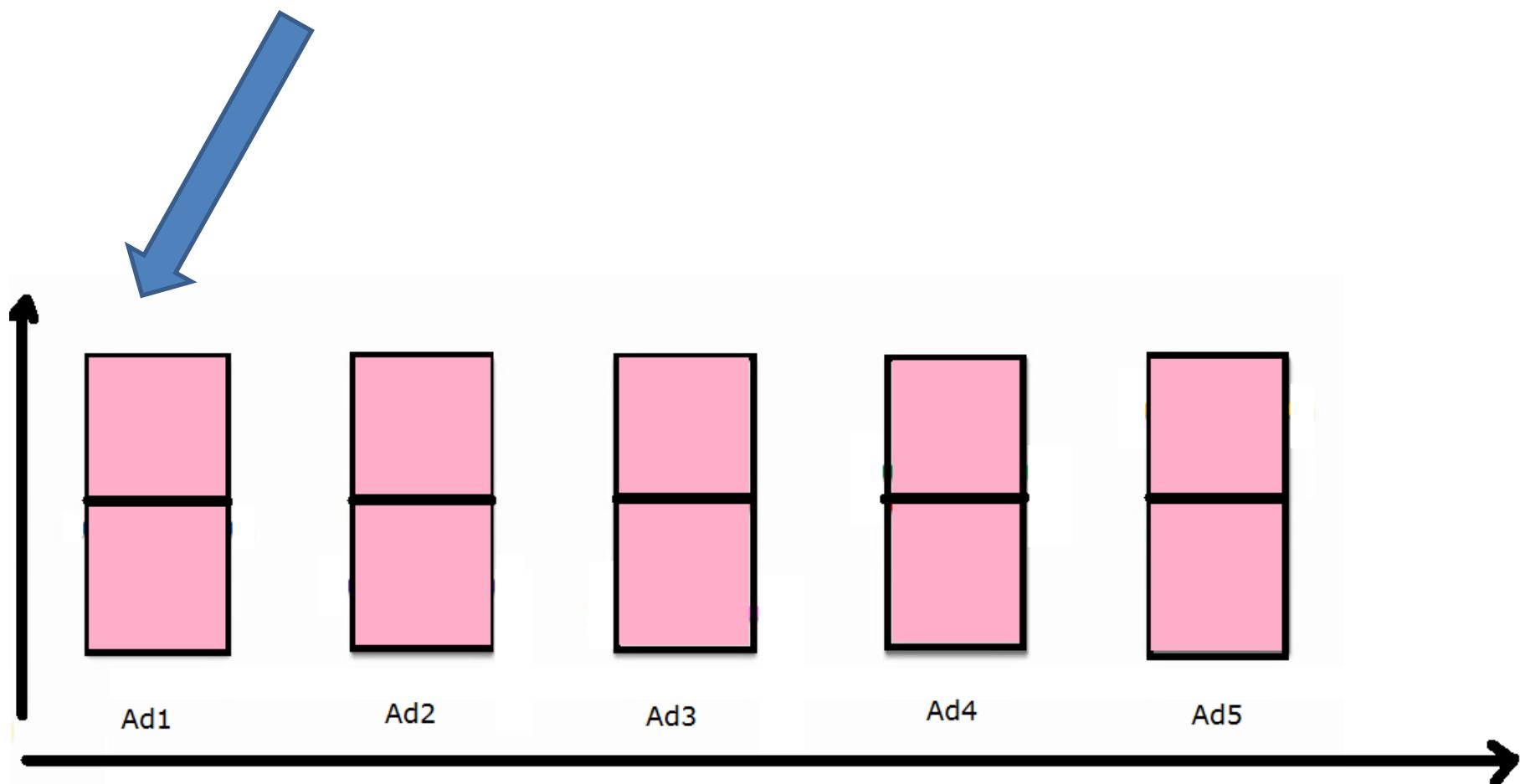
$$\bar{r}_i(n) = \frac{R_i(n)}{N_i(n)}$$

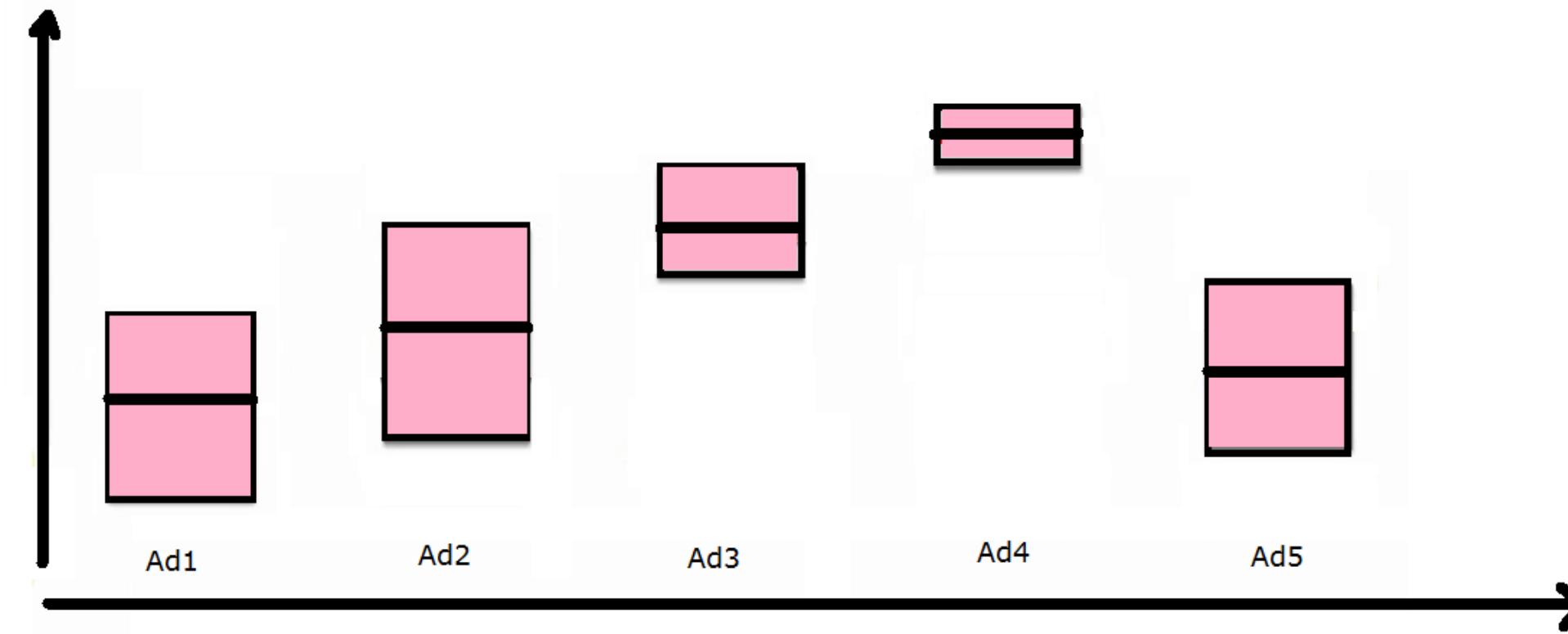
- UCB  $\bar{r}_i(n) + \Delta_i(n)$

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

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

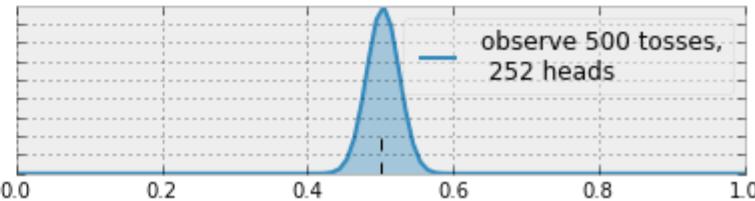
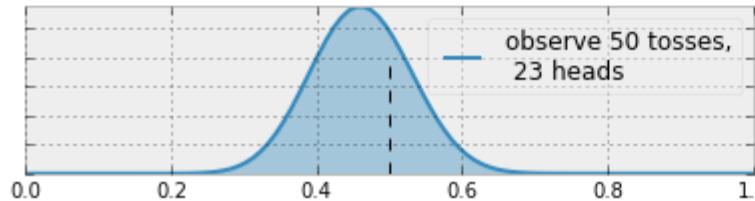
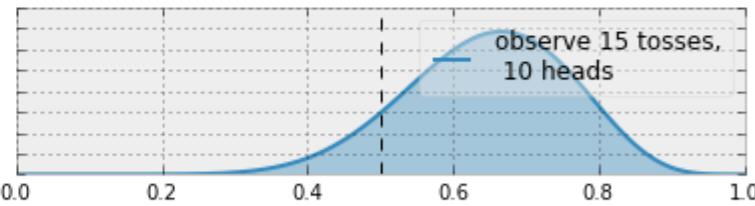
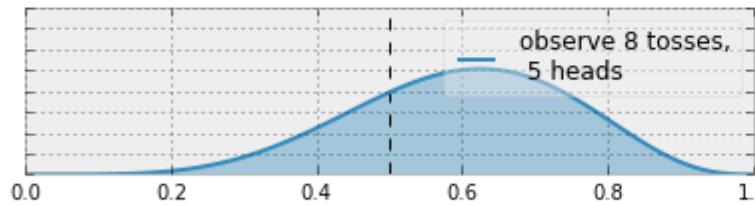
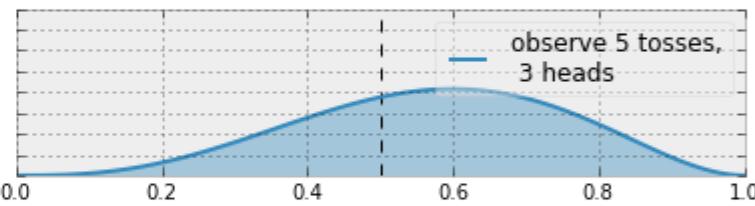
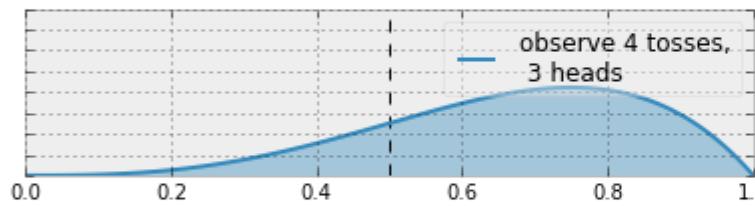
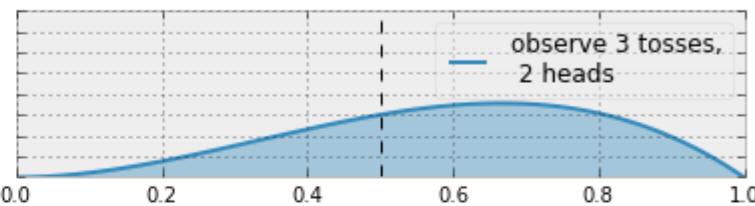
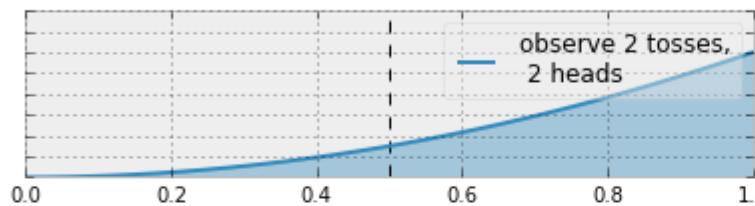
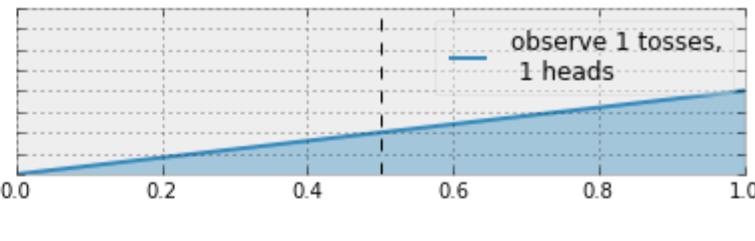
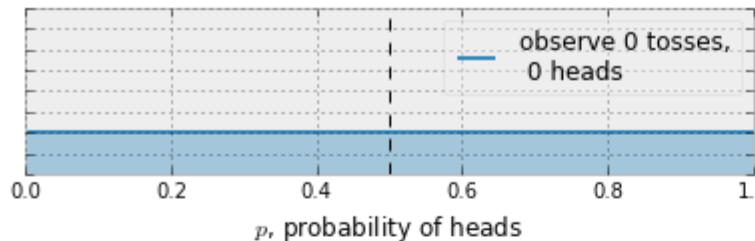
Confidence Band Generated for all Ads





# Thompson Sampling

Bayesian updating of posterior probabilities



# 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



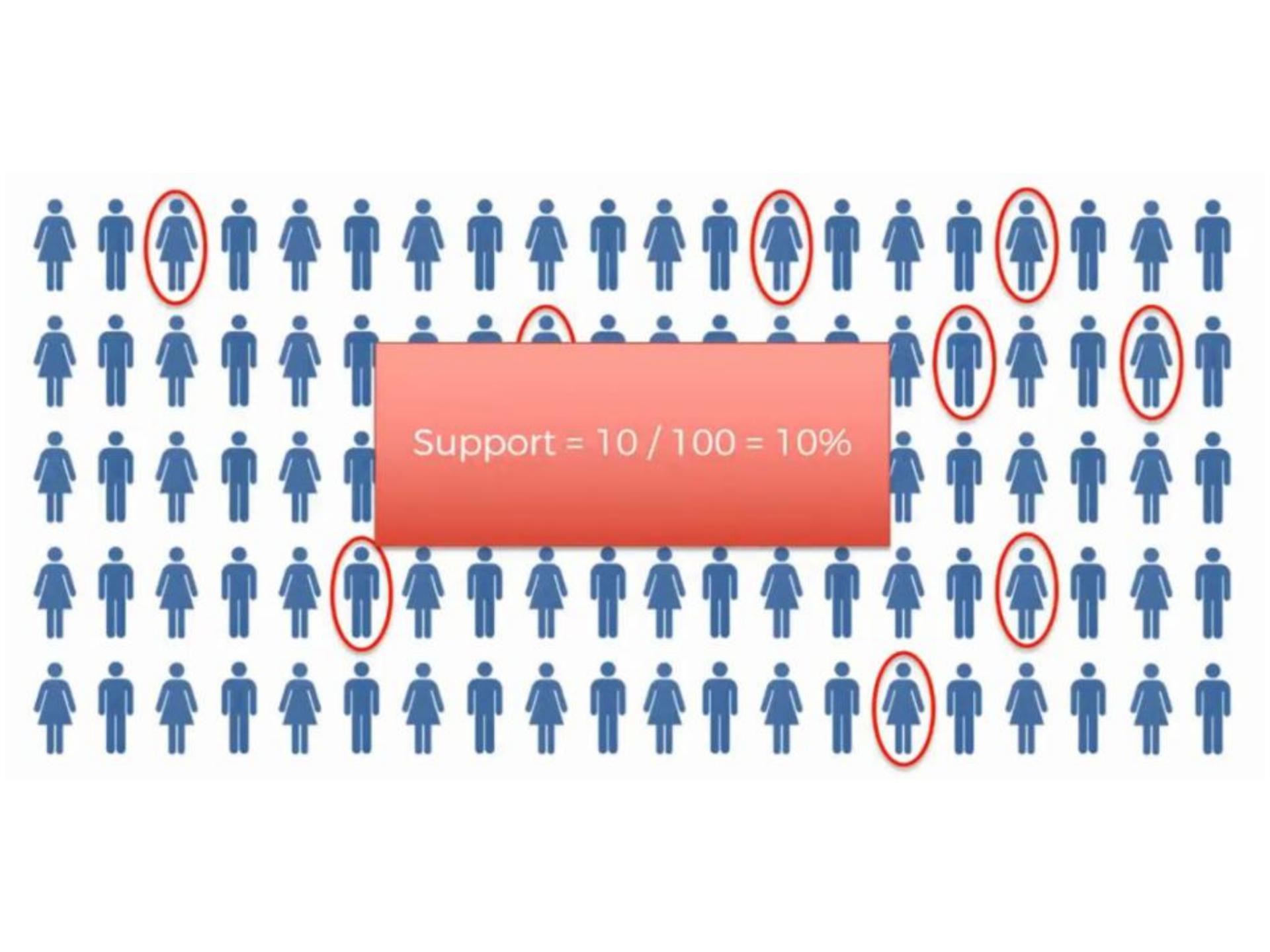
# Support

Movie Recommendation:

$$\text{support}(\mathbf{M}) = \frac{\# \text{ user watchlists containing } \mathbf{M}}{\# \text{ user watchlists}}$$

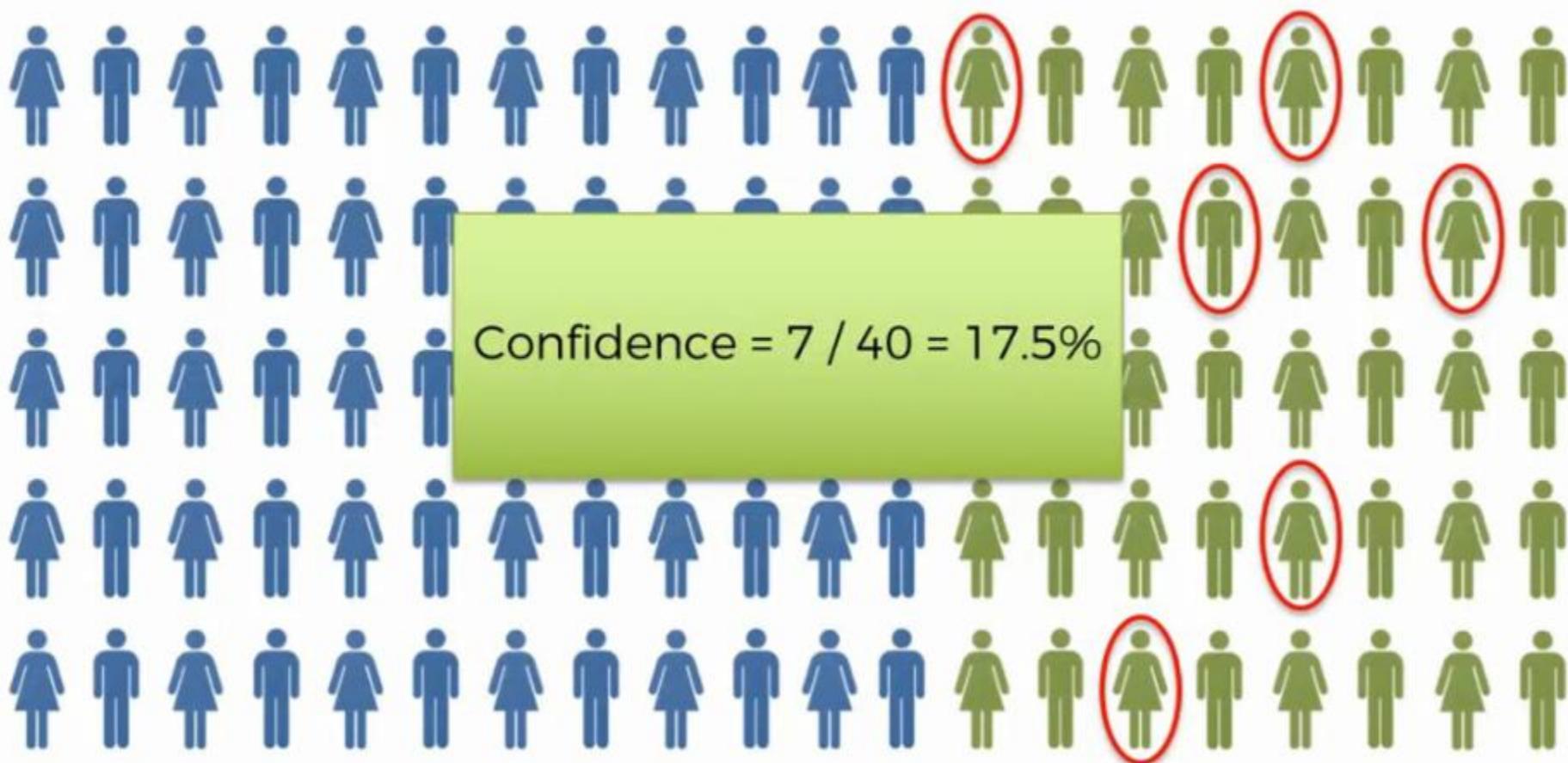
Market Basket Optimisation:

$$\text{support}(\mathbf{I}) = \frac{\# \text{ transactions containing } \mathbf{I}}{\# \text{ transactions}}$$


$$\text{Support} = 10 / 100 = 10\%$$

# Confidence

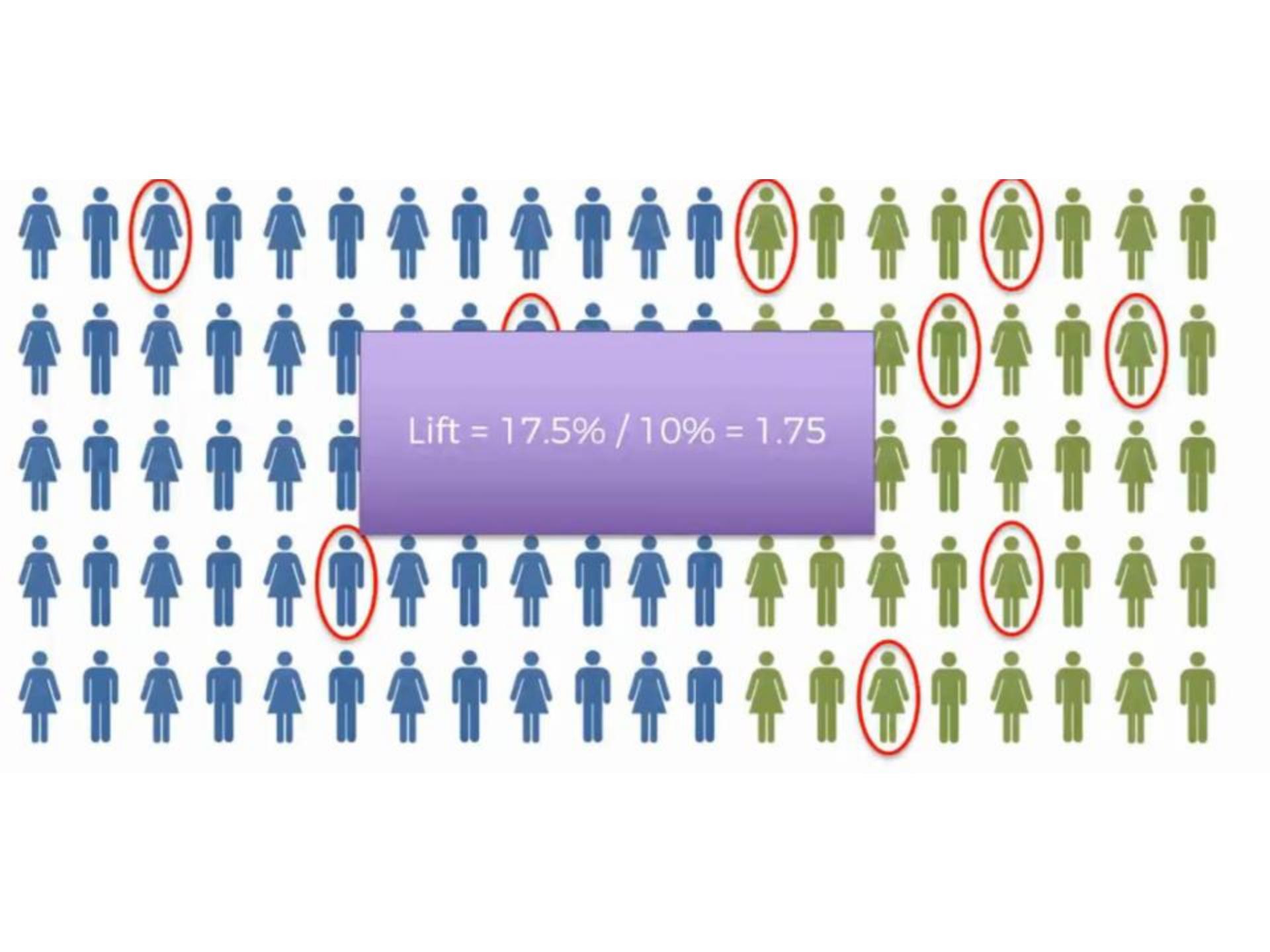
Movie Recommendation:  $\text{confidence}(\mathbf{M}_1 \rightarrow \mathbf{M}_2) = \frac{\# \text{ user watchlists containing } \mathbf{M}_1 \text{ and } \mathbf{M}_2}{\# \text{ user watchlists containing } \mathbf{M}_1}$



# Lift

Movie Recommendation:

$$\text{lift}(\mathcal{M}_1 \rightarrow \mathcal{M}_2) = \frac{\text{confidence}(\mathcal{M}_1 \rightarrow \mathcal{M}_2)}{\text{support}(\mathcal{M}_2)}$$



Lift = 17.5% / 10% = 1.75

# ANN

