

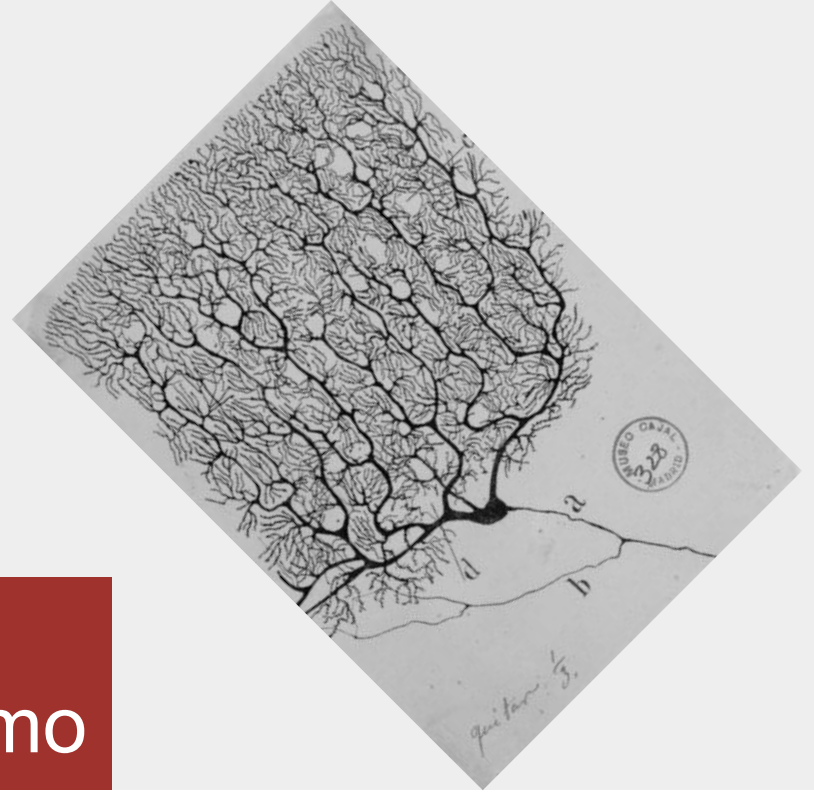


**MDUG**

# Machine Learning & Dynamo

## Un acercamiento

Julio J Garcia Borja



# Sobre mí



# Sobre mi

[Julio J García Borja](#)

- [Arquitecto Técnico](#) @ UPM
- [BIM Manager](#) @ UPC
- [Computational BIM Specialist](#) @ Modelical
- [Associate Proffesor](#) @ UPC/fCIM



# Sobre mi

Formación en ML

- [DataCamp](#)
- ASDM



DataCamp



# Sobre mi

## Formación en ML

- DataCamp
- [ASDM](#)



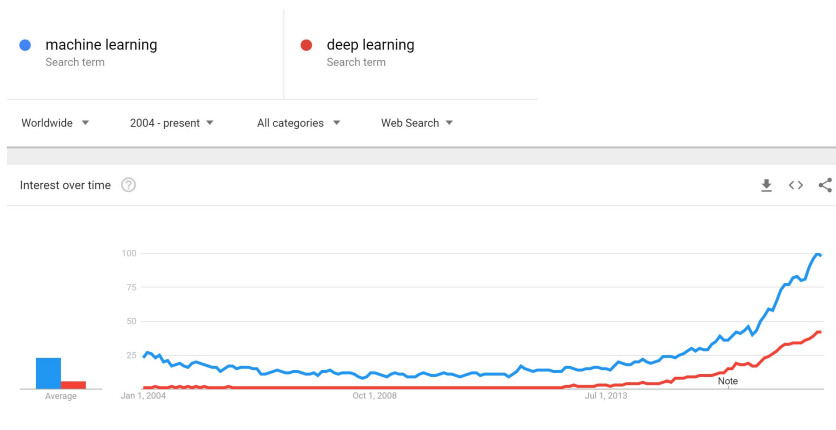
¿Dónde estamos?



# ¿Dónde estamos?

Está de moda debido a 3 factores:

- **Producción académica**
- Reducción de tiempos de computación
- Big Data

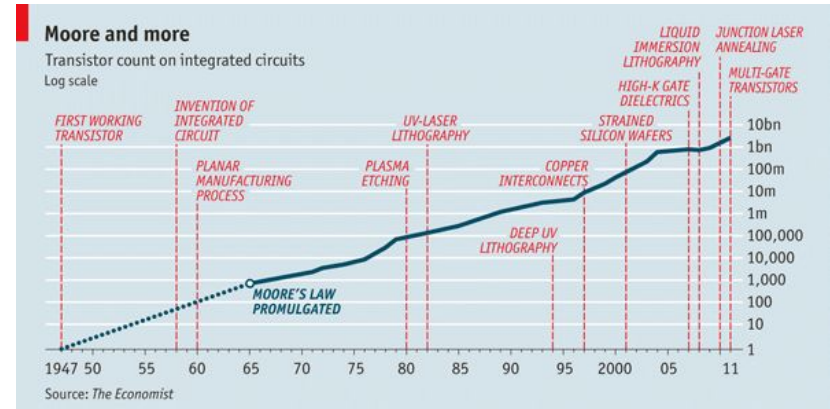




# ¿Dónde estamos?

Está de moda debido a 3 factores:

- Producción académica
- **Reducción de tiempos de computación**
- Big Data

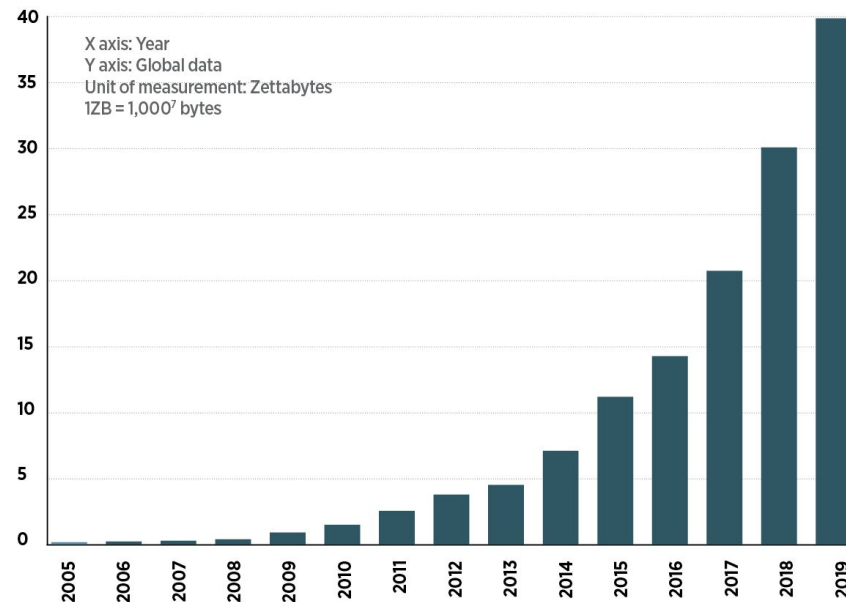


# ¿Dónde estamos?

Está de moda debido a 3 factores:

- Producción académica
- Reducción de tiempos de computación
- **Big Data**

## DATA GROWTH



Note: Post-2013 figures are predicted. Source: UNECE



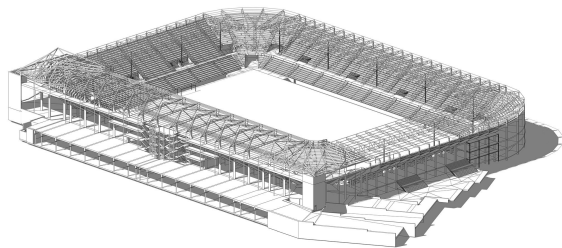
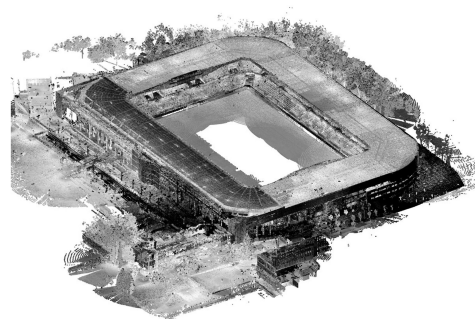
# Del BIM al Big Data



# Del BIM al Big Data

Problemas BIM actuales:

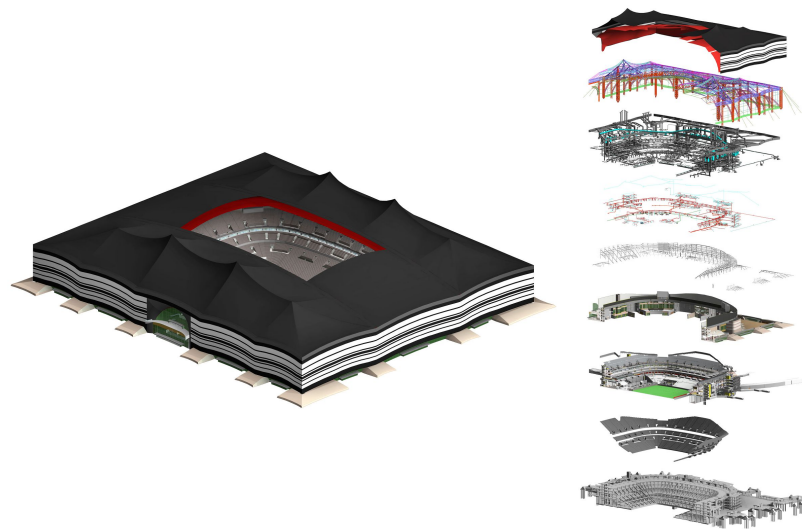
- **Nubes de puntos inmensas que debemos racionalizar**
- Muchos modelos por proyecto
- Muchos problemas de coordinación que gestionar
- Homogeneización de parámetros para FM



# Del BIM al Big Data

Problemas BIM actuales:

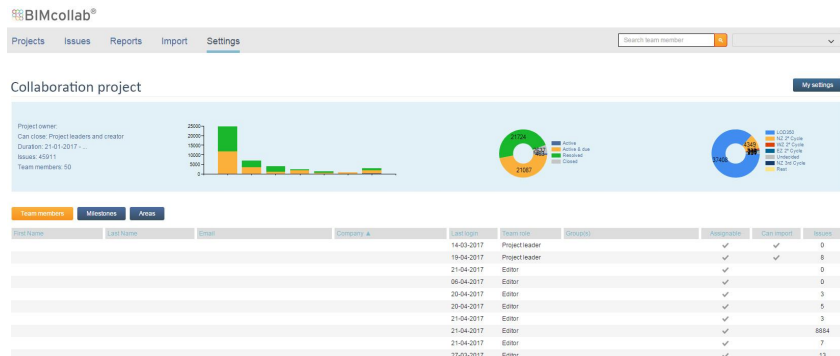
- Nubes de puntos inmensas que debemos modelar
- **Muchos modelos por proyecto**
- Muchos problemas de coordinación que gestionar
- Homogeneización de parámetros para FM



# Del BIM al Big Data

## Problemas BIM actuales:

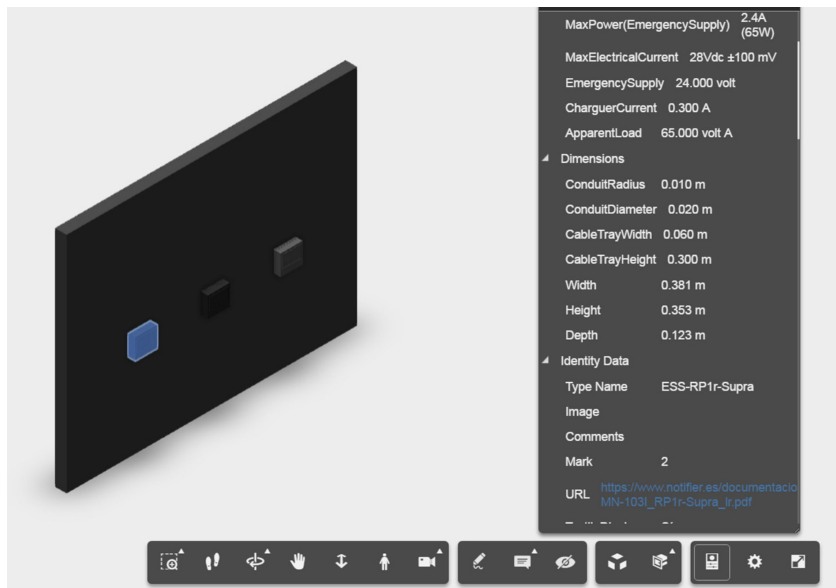
- Nubes de puntos inmensas que debemos modelar
- Muchos modelos por proyecto
- **Muchos problemas de coordinación que gestionar**
- Homogeneización de parámetros para FM



# Del BIM al Big Data

Problemas BIM actuales:

- Nubes de puntos inmensas que debemos modelar
- Muchos modelos por proyecto
- Muchos problemas de coordinación que gestionar
- **Homogeneización de parámetros/modelos para FM**

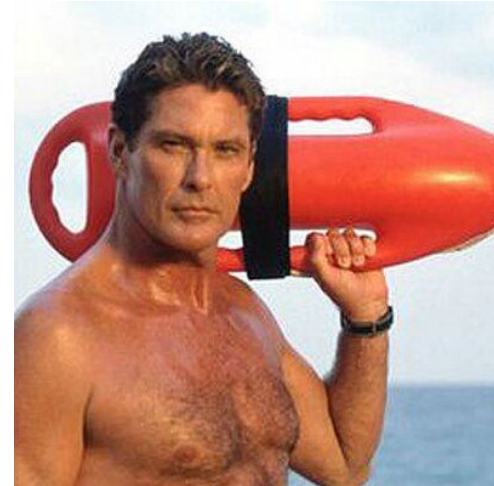


# Del BIM al Big Data

Nacimiento de BIM Dashboards, el nuestro:

- [Baywatch](#)

**bayWatch**  
modelica





# Del BIM al Big Data

Nacimiento de BIM Dashboards, el nuestro:

- [Baywatch](#)

**bayWatch**  
modelical

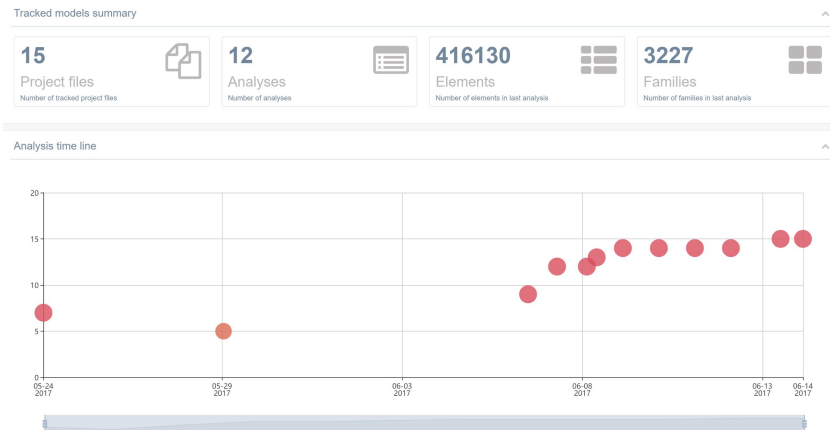


# Del BIM al Big Data

Nacimiento de BIM Dashboards, el nuestro:

- [Baywatch](#)
  - **Monitorización desasistida de modelos**
  - Cuadro de control de proyectos
  - Visualización de métricas BIM
  - Gestión de activos

bayWatch  
modelical

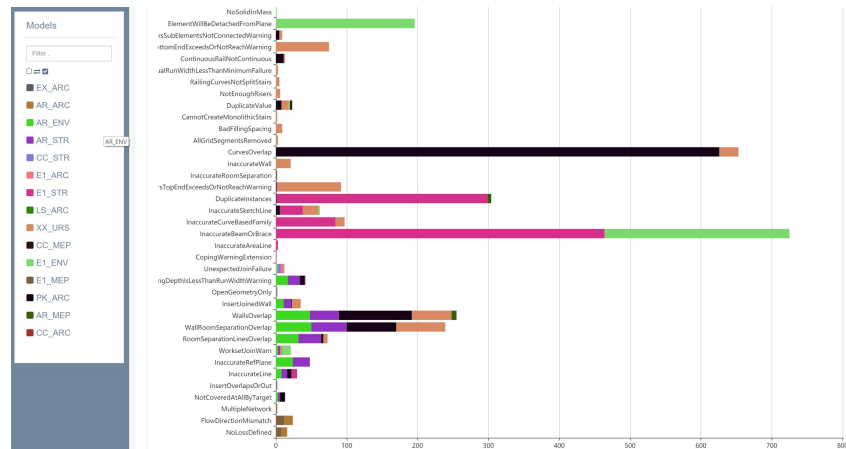


# Del BIM al Big Data

Nacimiento de BIM Dashboards, el nuestro:

- [Baywatch](#)
  - Monitorización desasistida de modelos
  - **Cuadro de control de proyectos**
  - Visualización de métricas BIM
  - Gestión de activos

bayWatch  
modelical



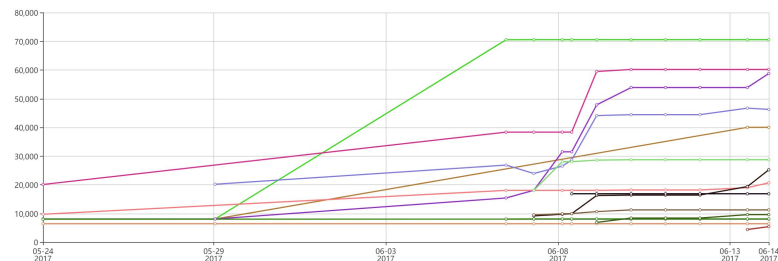
# Del BIM al Big Data

Nacimiento de BIM Dashboards, el nuestro:

- [Baywatch](#)
  - Monitorización desasistida de modelos
  - Cuadro de control de proyectos
  - **Visualización de métricas BIM**
  - Gestión de activos

bayWatch  
modelical

Project evolution for Instances



# Del BIM al Big Data

Nacimiento de BIM Dashboards, el nuestro:

- [Baywatch](#)
  - Monitorización desasistida de modelos
  - Cuadro de control de proyectos
  - Visualización de métricas BIM
  - **Gestión de activos**

bayWatch  
modelical



# Inteligencia Artificial



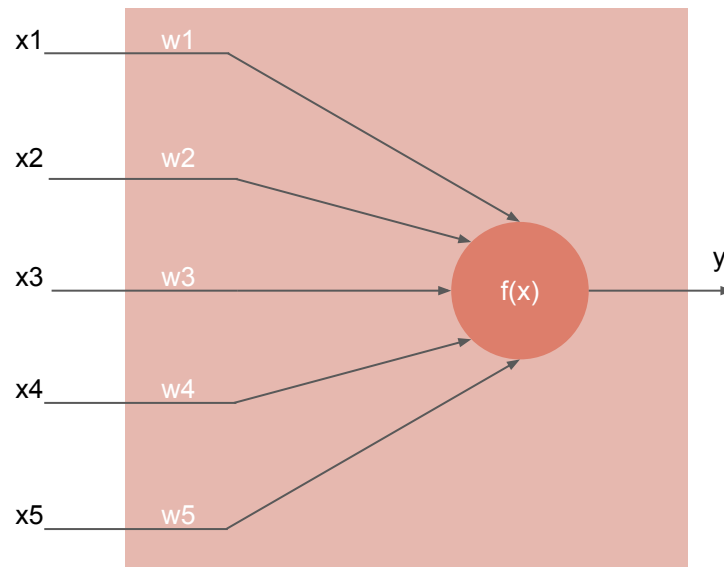
# Top-Down VS Bottom-Up

- Top-Down
  - La estructura del cerebro **no** es primordial para la emergencia de la función cognitiva.
  - Sistema que trata de recoger el máximo conocimiento sobre un tema muy concreto para tratar de emular el comportamiento de un humano avezado en la materia mediante un sistema lógico.
  - Da lugar a sistemas expertos.
  - Popular entre 1950-1970.
  - Ejemplos:
    - MYCIN: Medicina
    - Deep Blue: Ajedrez
    - Watson : Jeopardy!



# Top-Down VS Bottom-Up

- Bottom-Up
  - La estructura del cerebro es primordial para la emergencia de la función cognitiva.
  - Hay que replicar los sistemas naturales para poder hacer que un sistema artificial acceda a las funciones cognitivas.
  - Da lugar a redes neuronales.
  - Popular desde 1985.
  - Ejemplos:
    - [DeepMind: Alpha Go](#)
    - [deepL translator](#)
    - [From image to 3D](#)



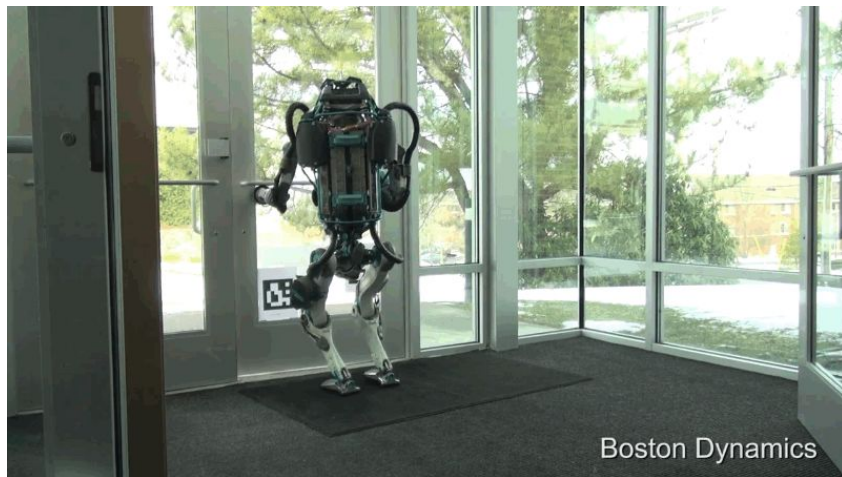


# Inteligencia artificial (IA)

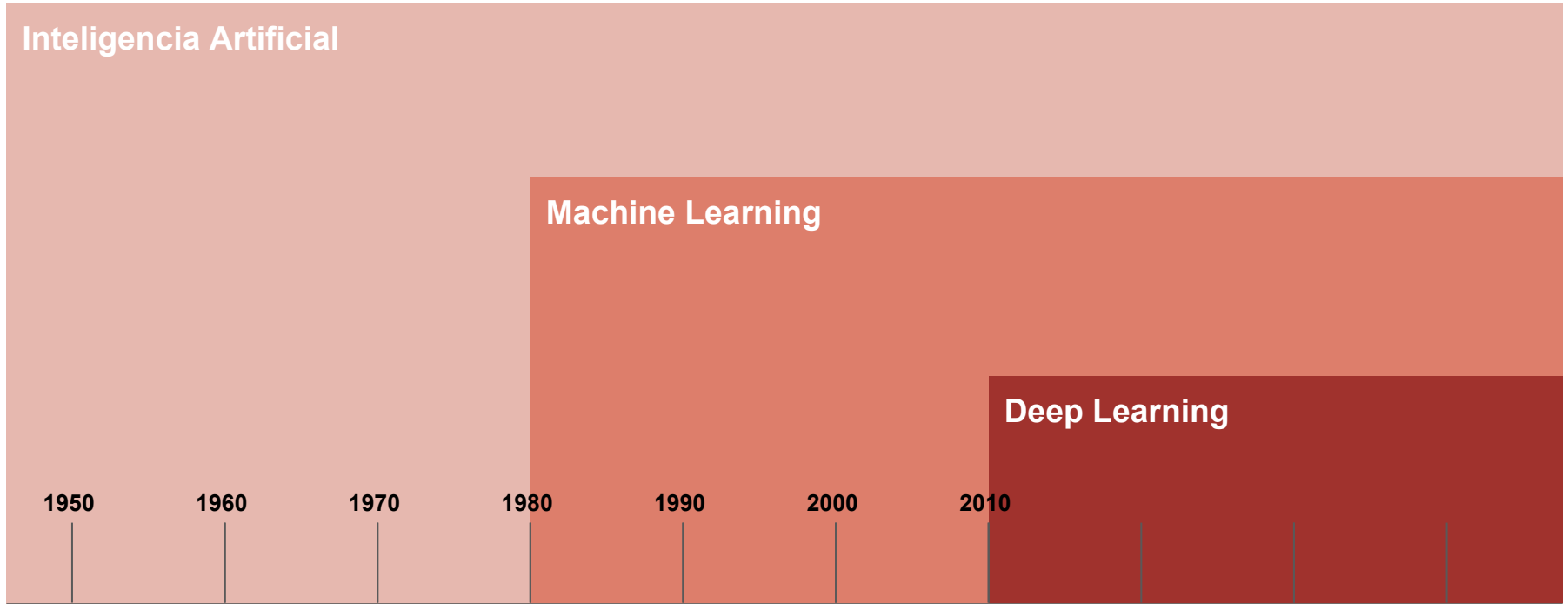
Inteligencia humana exhibida por máquinas

¿Dónde estamos? IA estrecha

- Pueden hacer una o pocas cosas a la vez.  
O reconocen imágenes o lenguaje...
- Necesitan código escrito por humanos



# Inteligencia artificial (IA)



# Machine Learning



# Machine Learning (ML)

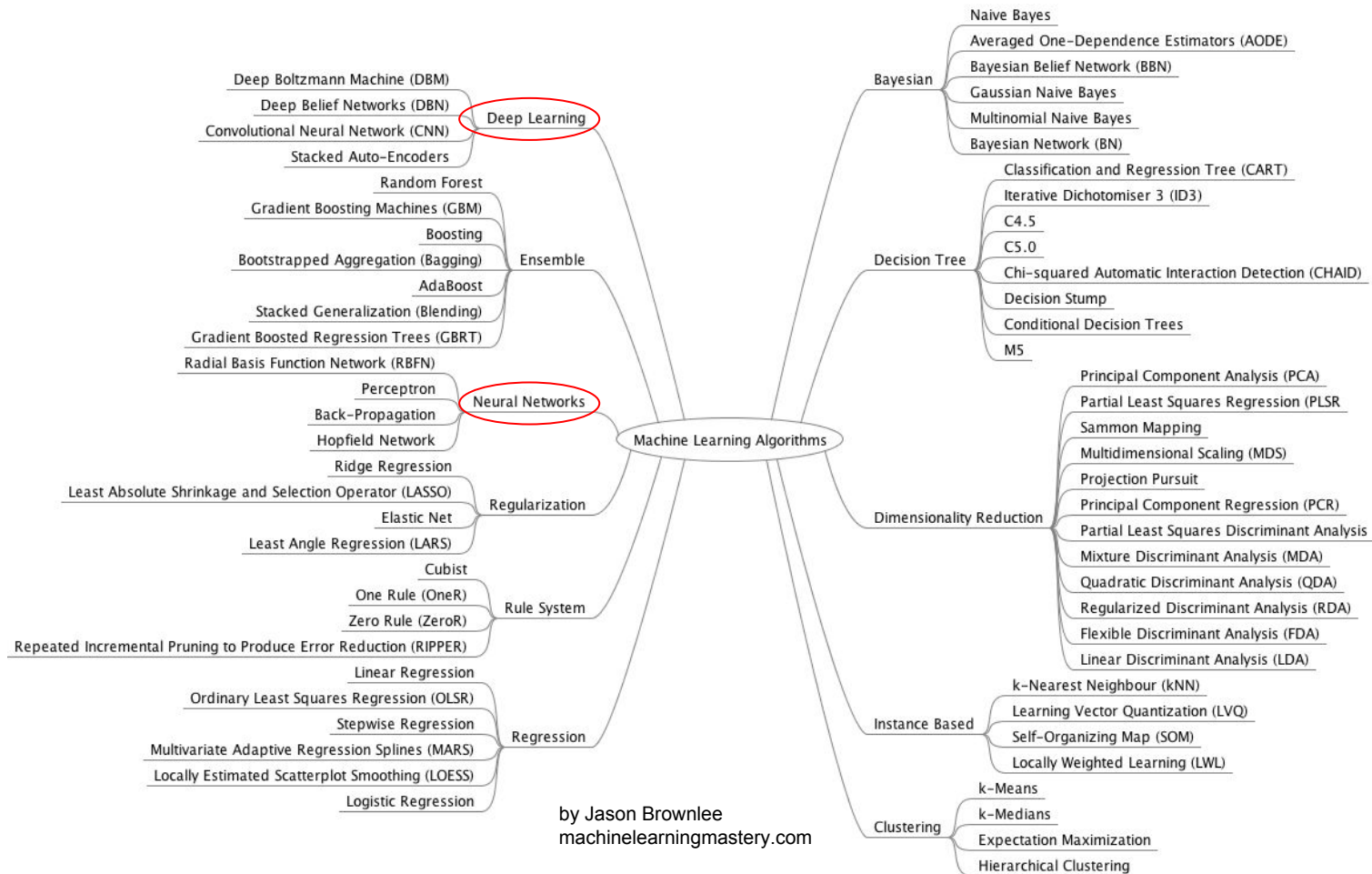
Sistemas de IA que pueden aprender de la experiencia para encontrar patrones en un conjunto de datos en vez de programarlos con unas reglas específicas

¿Cómo funcionan?

- 1. Procesa datos
- 2. Aprende patrones de los datos
- 3. Clasifica nuevos datos que no ha visto

Son técnicas de Inteligencia artificial





by Jason Brownlee  
[machinelearningmastery.com](http://machinelearningmastery.com)



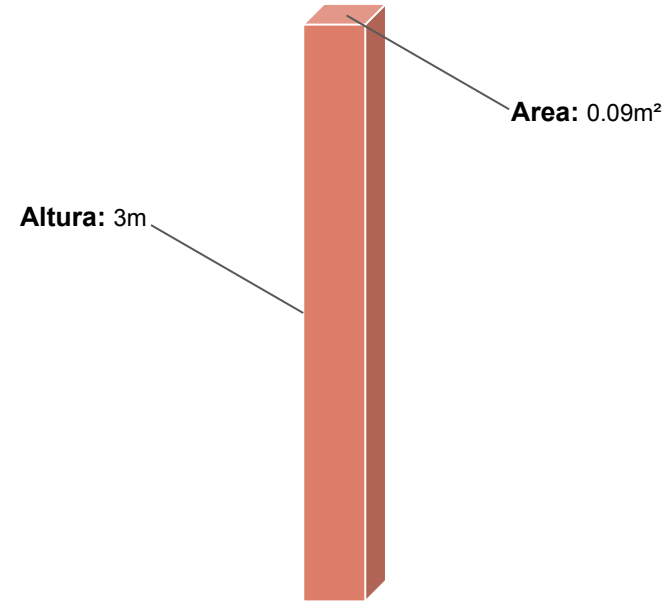
¿Cómo funciona?



# Features (Atributos)

Se usan para entrenar un sistema de ML

Son las propiedades de las cuales estamos intentando aprender



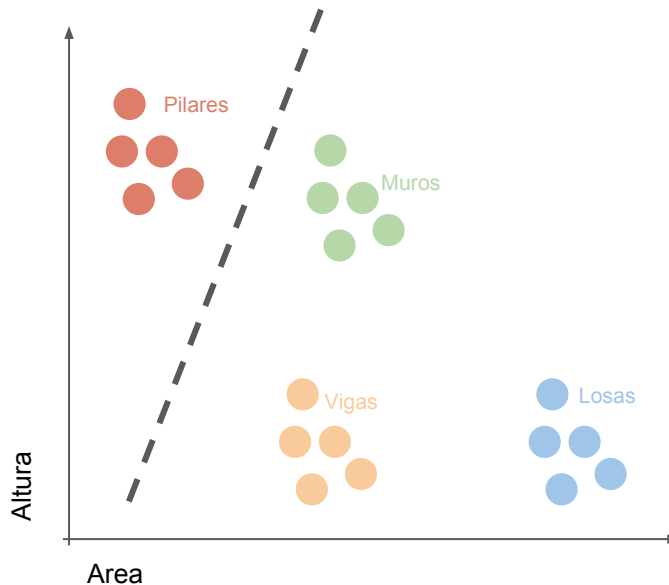
# Features (Atributos)

Cogiendo los elementos estructurales como ejemplo.

2 atributos querrá decir que tengo **2 dimensiones** (2D)

Mi sistema puede **aprender a dividir la gráfica** para discriminar pilares de los demás elementos estructurales

Esto se puede usar para realizar **futuras clasificaciones** que el sistema no haya visto

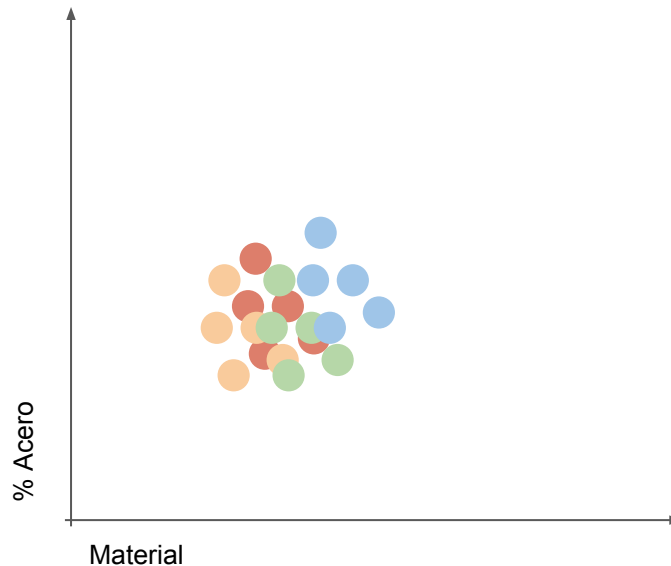




# Features (Atributos)

Escoger **atributos útiles** para la discriminación, por tanto, influye en la calidad de mi sistema.

Si escojo atributos que no sean lo suficientemente diferenciadores tendré un modelo inservible.



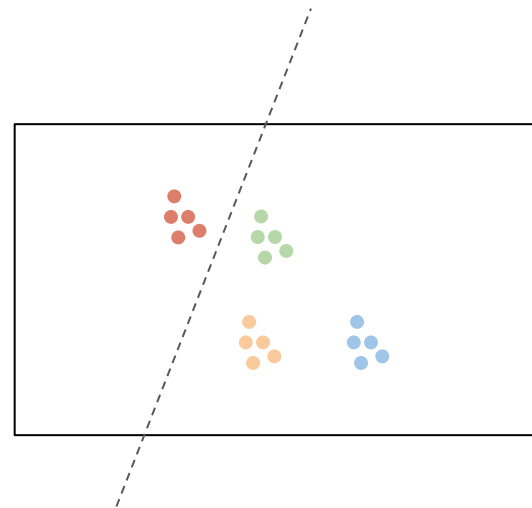
# Información útil

En ocasiones nos bastará con **sistemas 1D**



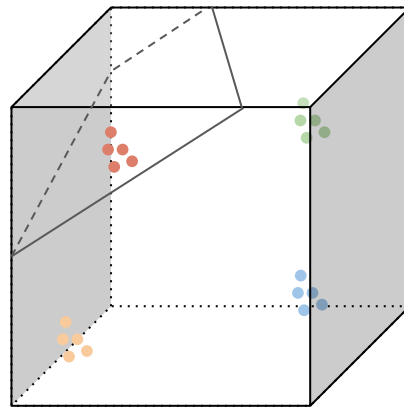
# Información útil

En ocasiones nos bastará con **sistemas 2D**



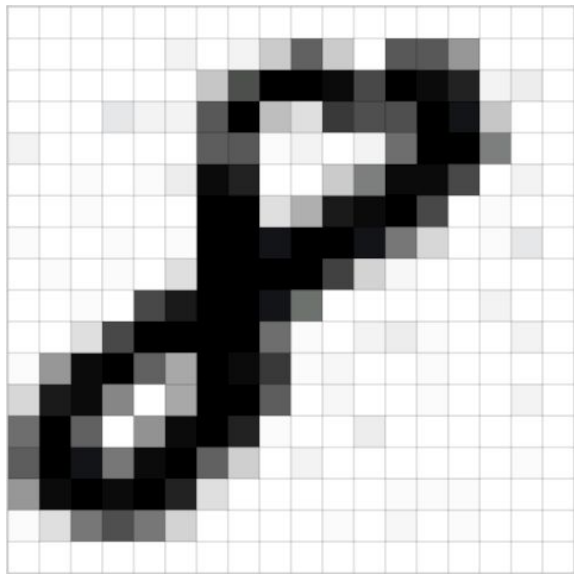
# Información útil

En ocasiones nos bastará con **sistemas 3D**



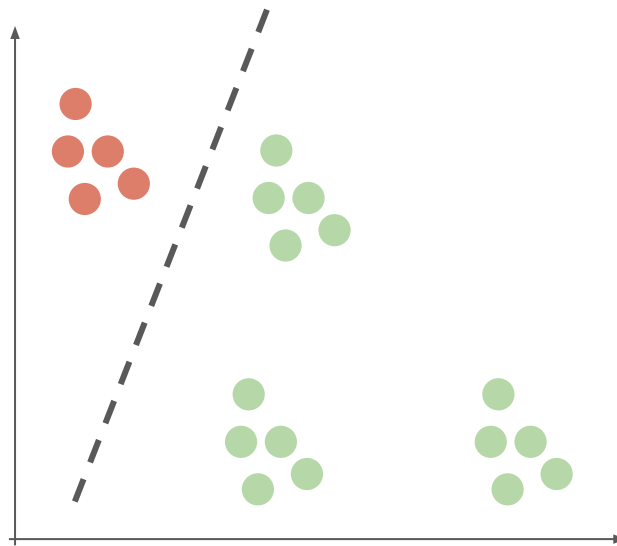
# Información útil

Pero para determinados problemas tendremos que irnos a 20D o **MillonesD** como es el caso del reconocimiento de imágenes en el que cada pixel es un atributo



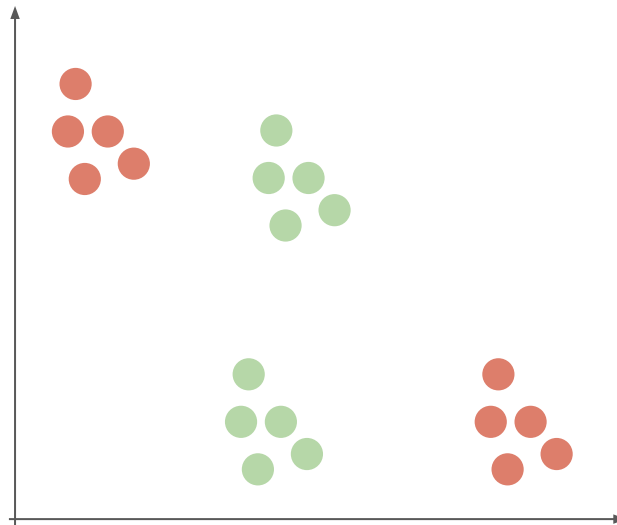
# Problemas lineales / no lineales

- **Lineales:** Son aquellos en los que existe al menos un clasificador lineal para partir todos los casos.
- **No lineales:** Son aquellos en los que no existe un clasificador lineal para partir todos los casos.



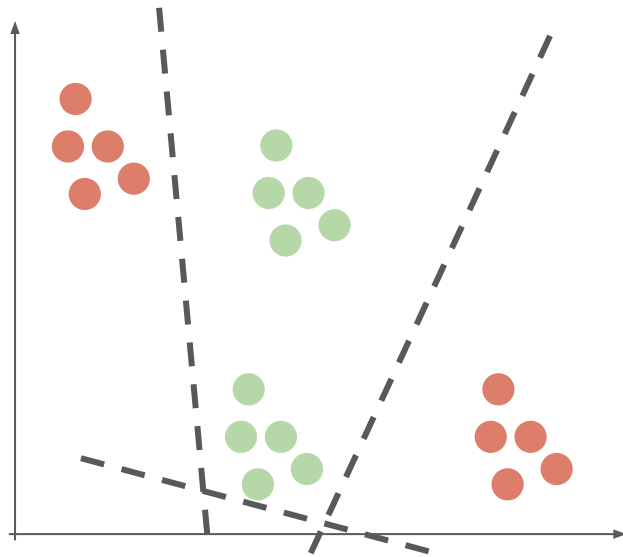
# Problemas lineales / no lineales

- **Lineales:** Son aquellos en los que existe al menos un clasificador lineal para partir todos los casos.
- **No lineales:** Son aquellos en los que no existe un clasificador lineal para partir todos los casos.



# Problemas lineales / no lineales

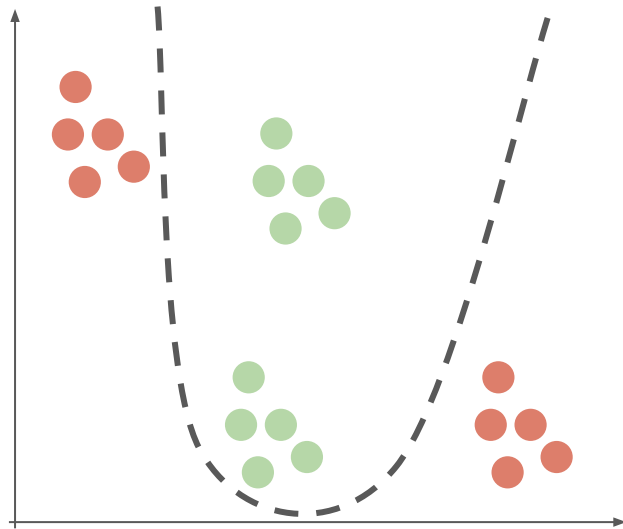
- Lineales: Son aquellos en los que existe al menos un clasificador lineal para partir todos los casos.
- **No lineales:** Son aquellos en los que no existe un clasificador lineal para partir todos los casos.





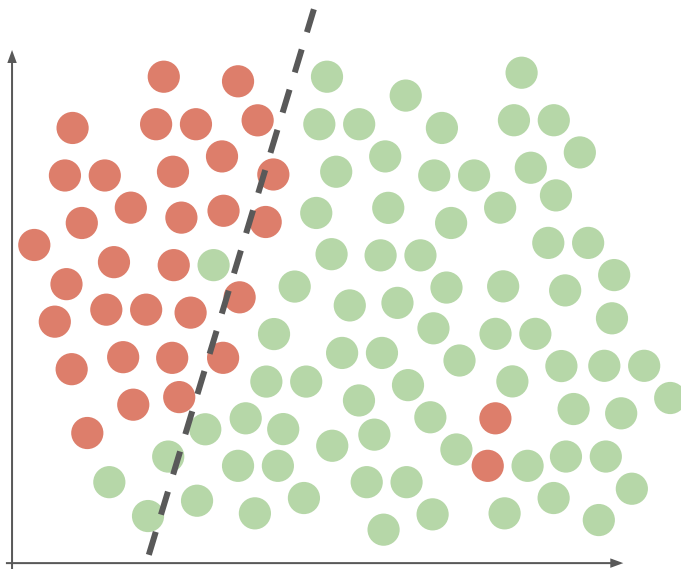
# Problemas lineales / no lineales

- **Lineales:** Son aquellos en los que existe al menos un clasificador lineal para partir todos los casos.
- **No lineales:** Son aquellos en los que no existe un clasificador lineal para partir todos los casos.



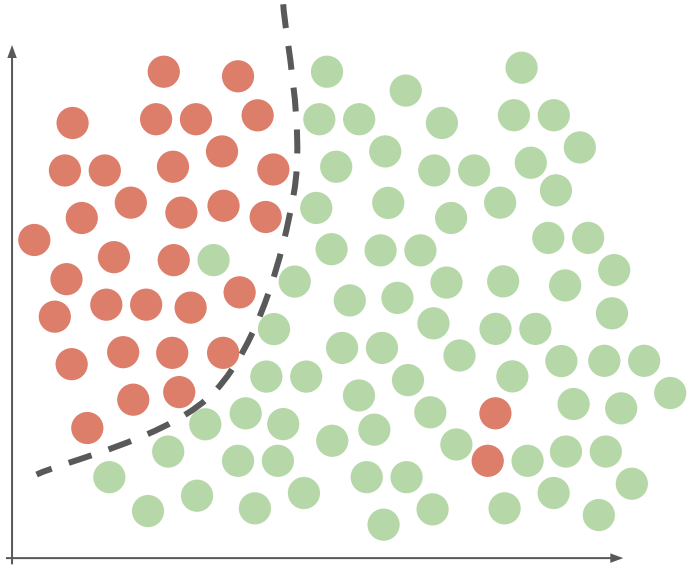
# Underfitting (No ajustado)

Añadir dimensiones puede ayudarnos a resolver problemas complejos pero también podemos hacer que **nuestro sistema se conozca tan bien nuestros datos de entrenamiento que no sea capaz de generalizar** la nueva información y falle en su clasificación.



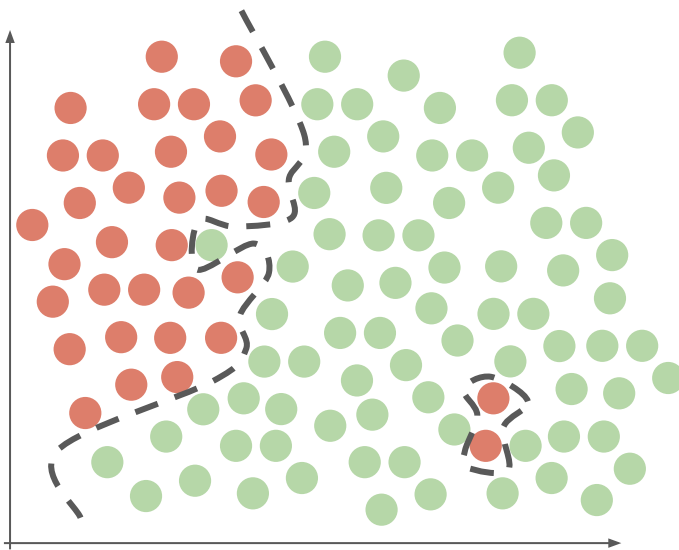
# Well tuned (Bien ajustado)

Añadir dimensiones puede ayudarnos a resolver problemas complejos pero también podemos hacer que **nuestro sistema se conozca tan bien nuestros datos de entrenamiento que no sea capaz de generalizar** la nueva información y falle en su clasificación.



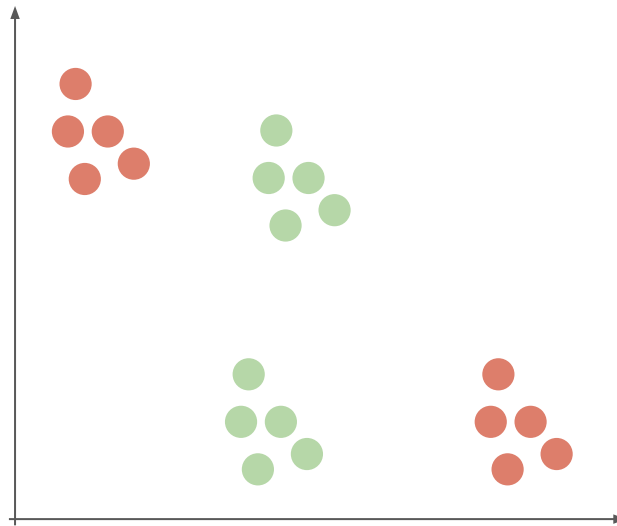
# Overfitting (Sobreentrenamiento)

Añadir dimensiones puede ayudarnos a resolver problemas complejos pero también podemos hacer que **nuestro sistema se conozca tan bien nuestros datos de entrenamiento que no sea capaz de generalizar** la nueva información y falle en su clasificación.



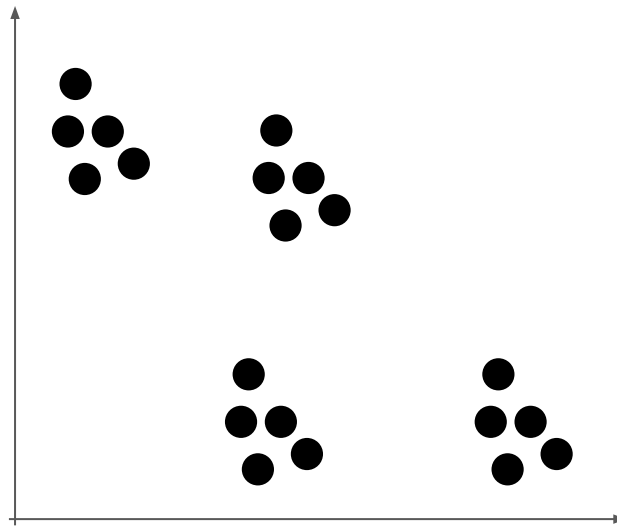
# Aprendizaje supervisado

- Datos de entrenamiento **etiquetados**



# Aprendizaje no supervisado

- Datos de entrenamiento **no etiquetados**



# Aprendizaje supervisado / no supervisado

## Supervisado

- Datos de entrenamiento **etiquetados**
  - 3m, 0.09m<sup>2</sup>, **Pilar**
  - 3.1m, 0.08m<sup>2</sup>, **Pilar**
  - 0.2m, 25m<sup>2</sup>, **Forjado**
- Supervision: Le doy como input el output esperado
- Ejemplos:
  - Problemas de clasificación (Classification)
  - Problemas de regresión (Regression)

## No supervisado

- Datos de entrenamiento **sin etiquetar**
  - 3m, 0.09m<sup>2</sup>
  - 3.1m, 0.08m<sup>2</sup>
  - 0.2m, 25m<sup>2</sup>
- No le doy como input el output esperado
- Hago al sistema que busque patrones
- Ejemplos:
  - Problemas de agrupación (Clustering)



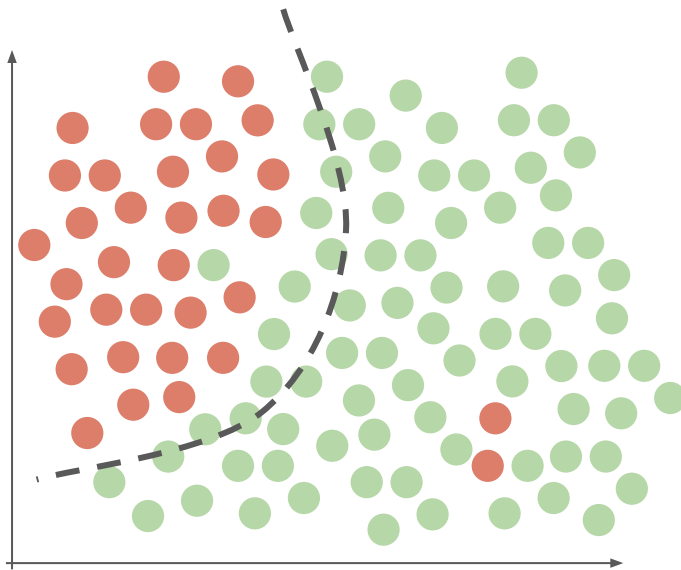
# Aprendizaje por refuerzo

Aprendizaje por ensayo-error

El programa aprende jugando millones de partidas.

Cuando gana le damos una recompensa, cuando pierde, no le damos recompensa (recompensa negativa)

Podemos usar funciones de **optimización** para buscar el resultado deseado





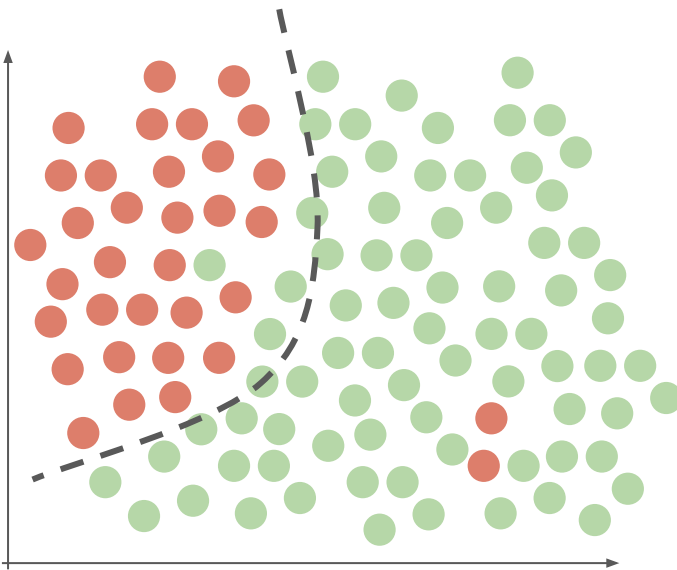
# Aprendizaje por refuerzo

Aprendizaje por ensayo-error

El programa aprende jugando millones de partidas.

Cuando gana le damos una recompensa, cuando pierde, no le damos recompensa (recompensa negativa)

Podemos usar funciones de **optimización** para buscar el resultado deseado



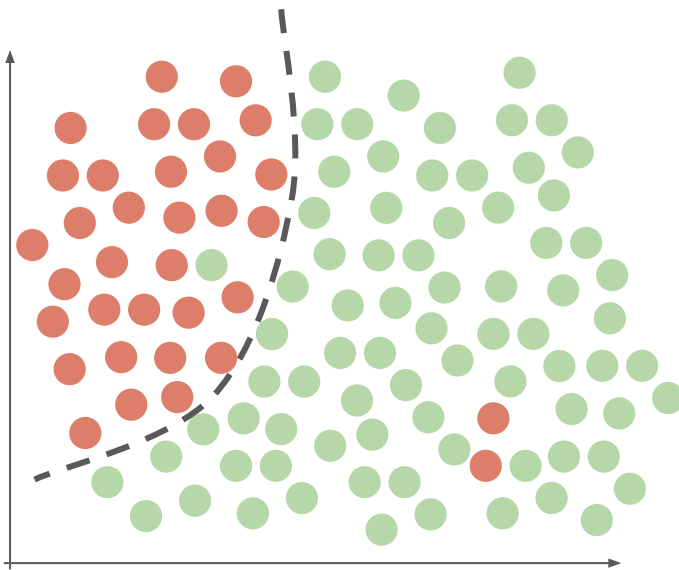
# Aprendizaje por refuerzo

Aprendizaje por ensayo-error

El programa aprende jugando millones de partidas.

Cuando gana le damos una recompensa, cuando pierde, no le damos recompensa (recompensa negativa)

Podemos usar funciones de **optimización** para buscar el resultado deseado

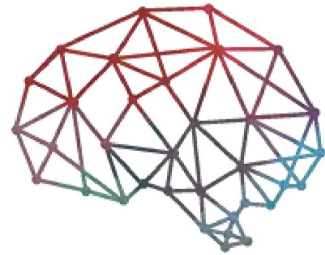


# Ejemplos de uso con Dynamo



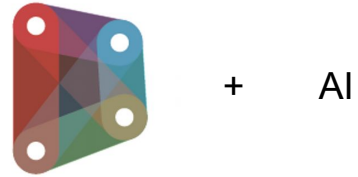
# “AI” package

- By [Radu Gidei](#), [Grimshaw](#)



# “DynamoAI” package

- By [Sharad Jaiswal](#), Autodesk



# “LunchboxML” package

- By [Nathan Miller, Proving Ground](#)

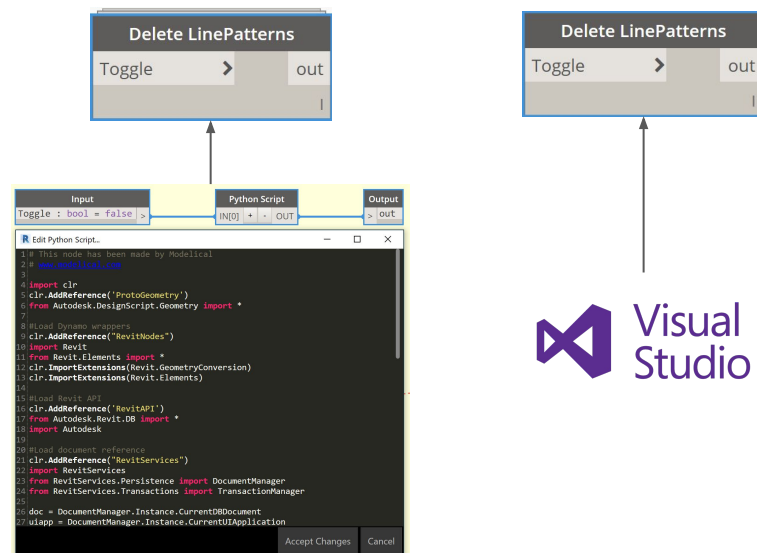


# ZeroTouch Nodes

- Son nodos programados en C#
- Ventajas:
  - C#
  - .NET frameworks (salimos de IronPython!)
  - UIs customizadas
  - Mejor IDE
- Inconvenientes:
  - No se puede acceder al código

¿Dónde empezar? Matteo Cominetti's workshop

<https://github.com/teocomi/dug-dynamo-unchained>



# Accord

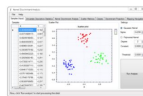
- C# .NET machine learning framework
- Implementable en dynamo via Zero Touch Nodes





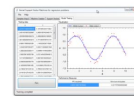
# Agrupación, clasificación y predicción

- Agrupación/Clustering: No supervisado
- Clasificación/Classification: Supervisado
- Regresión/Regression: Supervisado



Classification.

Support Vector Machines , Logistic Regression , Decision Trees , Neural Networks , Deep Learning (Deep Neural Networks) , Levenberg-Marquardt with Bayesian Regularization , Restricted Boltzmann Machines , Sequence Classification , Hidden Markov Classifiers and Hidden Conditional Random Fields .



Regression.

Multiple linear regression , Multivariate linear regression , polynomial regression , **logarithmic regression** , Logistic regression , multinomial logistic regression (softmax) and generalized linear models . L2-regularized L2-loss logistic regression , L2-regularized logistic regression , L1-regularized logistic regression , L2-regularized logistic regression in the dual form and regression support vector machines .



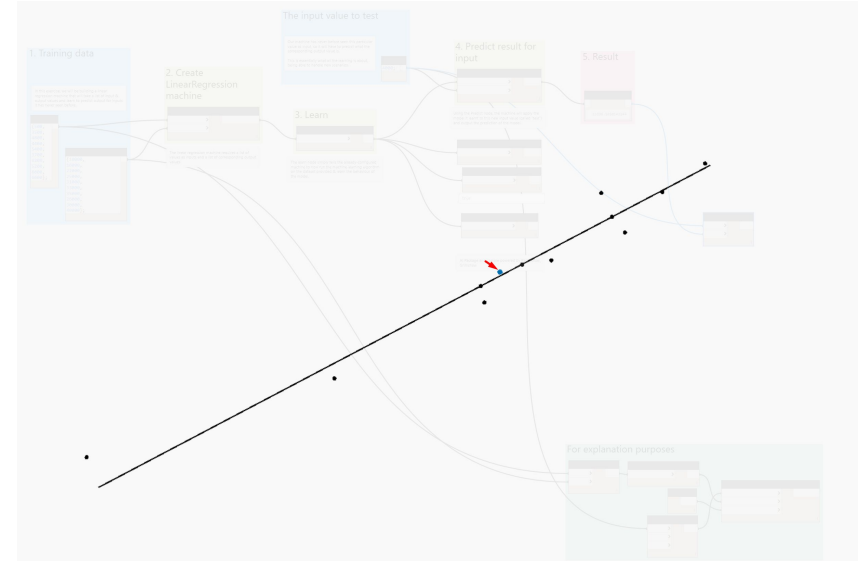
Clustering.

K-Means , K-Modes , Mean-Shift , Gaussian Mixture Models , Binary Split , Deep Belief Networks , Restricted Boltzmann Machines . Clustering algorithms can be applied in arbitrary data , including images , data tables, videos and audio .



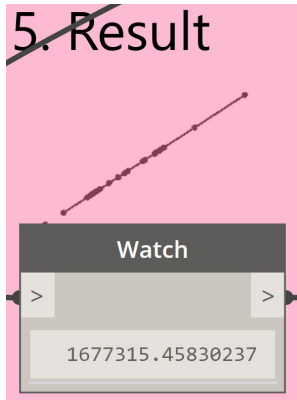
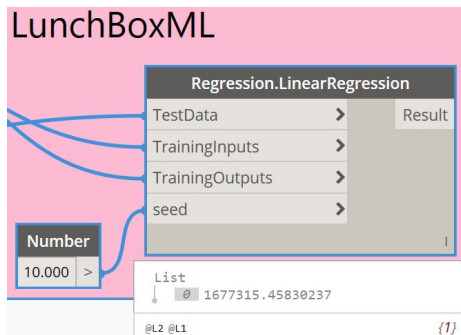
# Regression: Linear Regression (LR)

- Modelo lineal
- Establece **relaciones lineales entre un escalar y una o más variables**



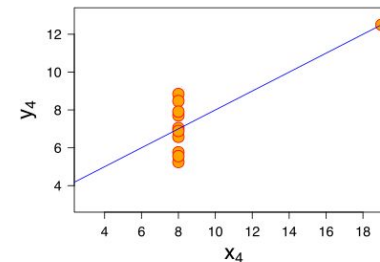
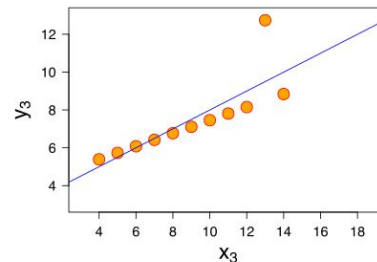
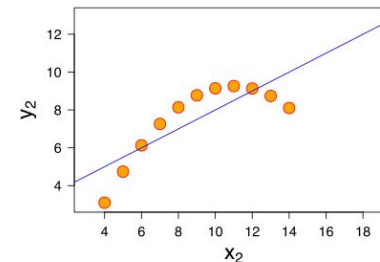
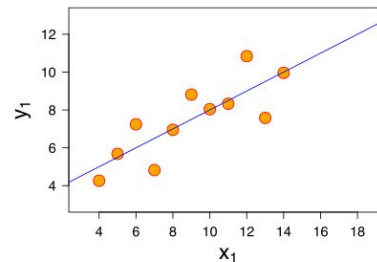
# Regression: Linear Regression (LR)

- Modelo lineal
- Establece **relaciones lineales entre un escalar y una o más variables**
- Como era de esperar, las predicciones del nodo de Miller coinciden con las del de Gidei



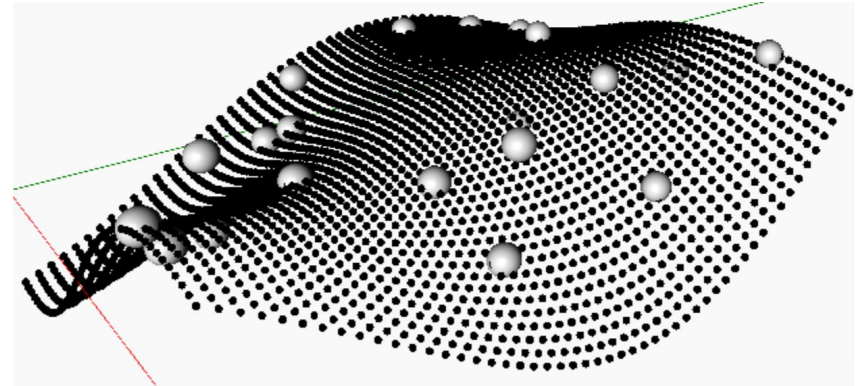
# Regression: Linear Regression (LR)

- Cuarteto de Anscombe
  - Distribuciones gráficamente distintas
  - Mismas métricas estadísticas, misma línea de regresión



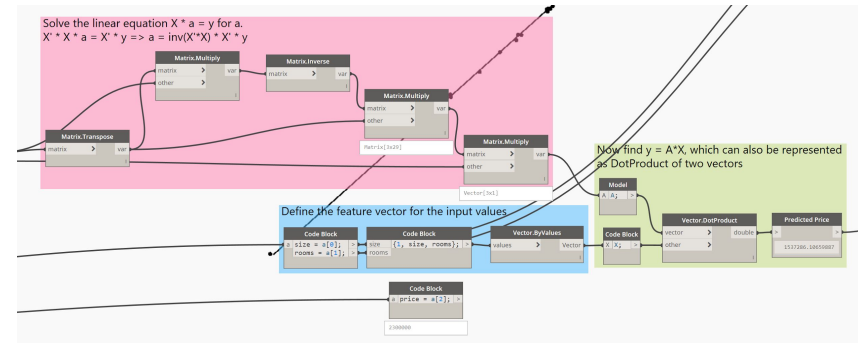
# Regression: Non Linear Regression

- Modelo no lineal
- Establece **relaciones entre un escalar y una o más variables**



# Regression: Polynomial Regression

- Modelo polinomial
- Establece **relaciones polinomiales entre un escalar y una o más variables**
- En el package de DynamoAI se puede seguir las demostraciones matemáticas



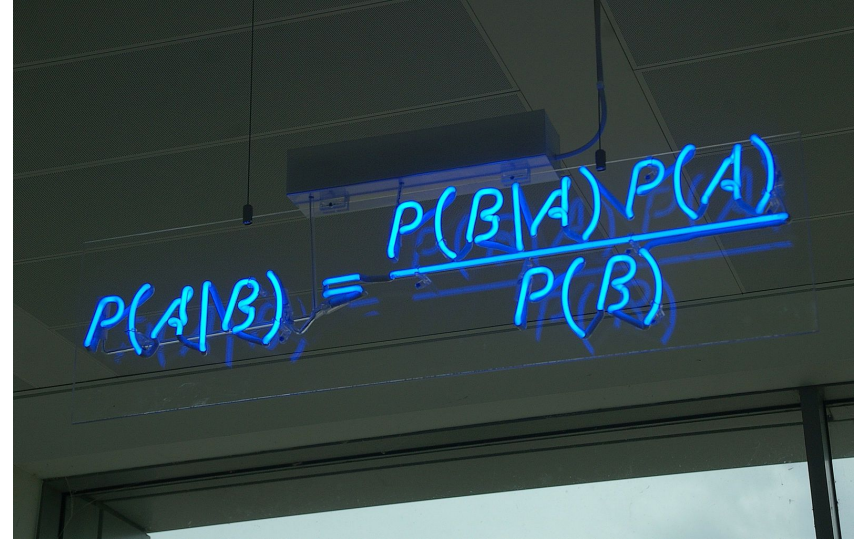
# Clustering: Gaussian Mixture Model (GMM)

- Modelo probabilístico
- Sirve para representar la presencia de **subpoblaciones dentro de una población global**



# Classification: Naive Bayes Classifier

- Clasificador probabilístico
- Distribución normal (gaussiana)

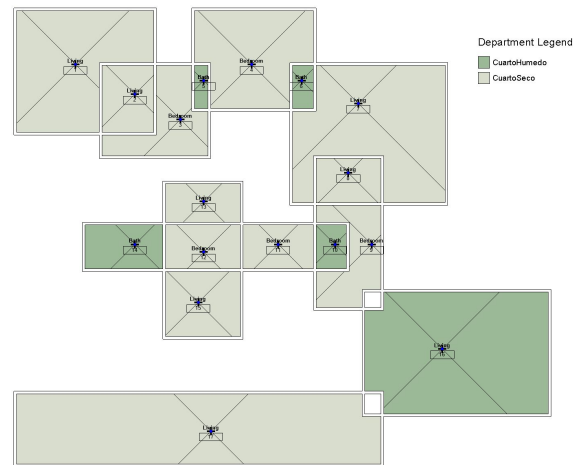

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)}$$





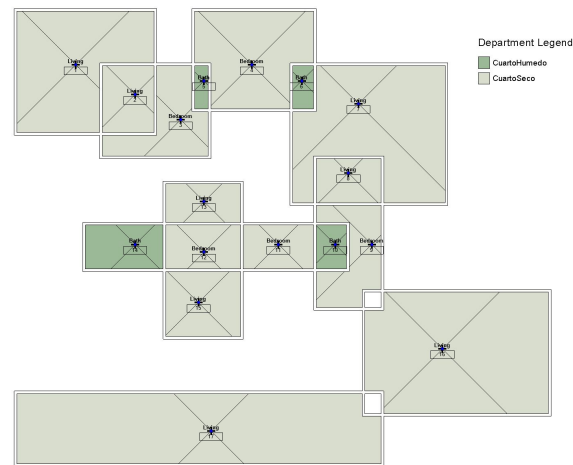
# Classification: Naive Bayes Classifier

- Clasificador probabilístico
- Distribución normal (gaussiana)



# Classification: Naive Bayes Classifier

- Clasificador probabilístico
- Distribución normal (gaussiana)



GRACIAS

# ALGOMAD

**ALGOMAD 2018 se está cocinando!**

<http://www.algomad.org/>





**MDUG**