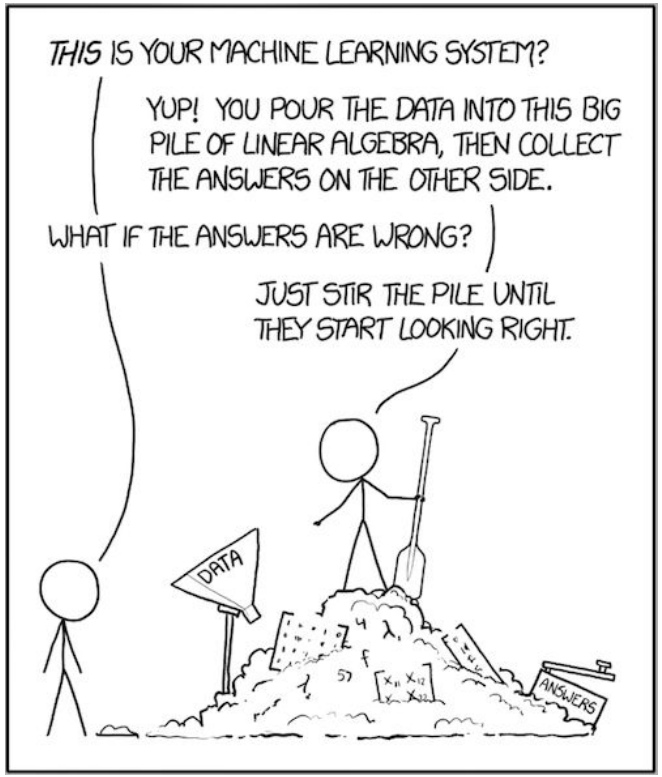
# Intro Data science & Machine Learning

Teoría general DS, Python, Numpy, Pandas & setup


# Recapitulando…

1. No es magia!

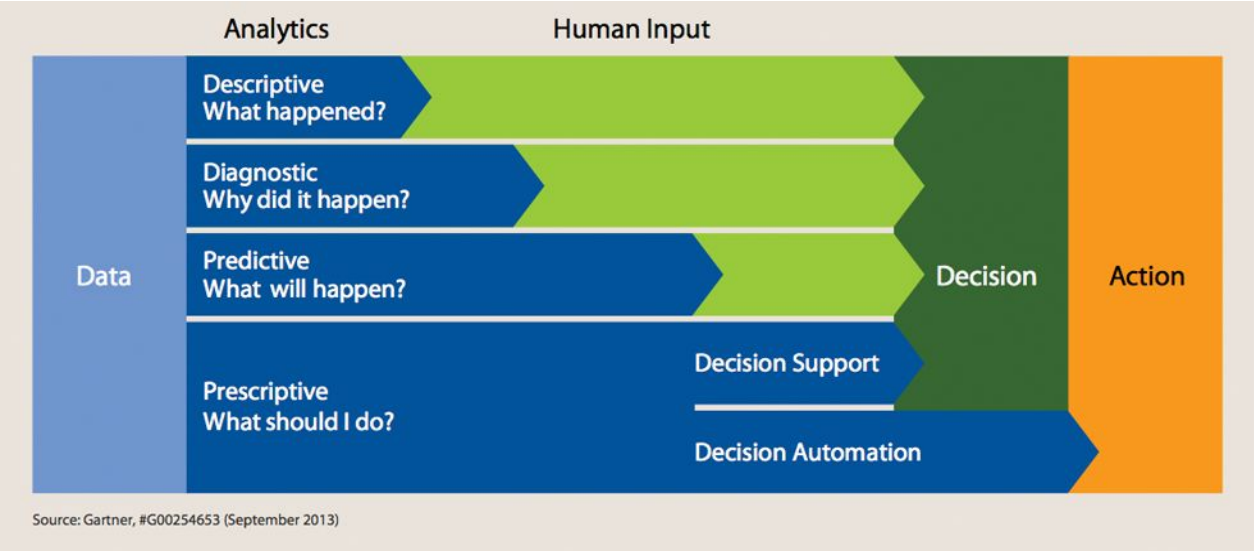



DBlandIT

UTN.BA
UNIVERSIDAD TECNOLÓGICA NACIONAL
FACULTAD REGIONAL BUENOS AIRES

# Recapitulando…

1. No es magia!




DBlandIT

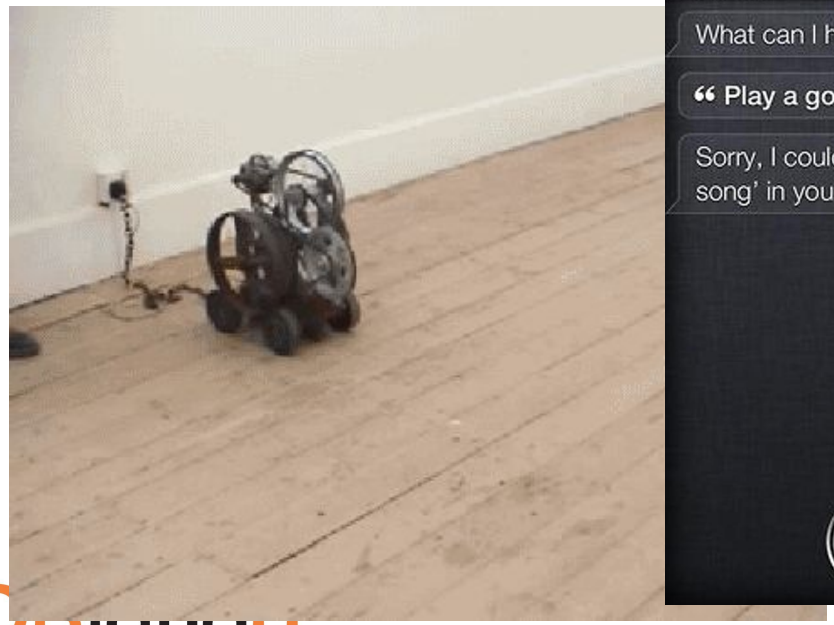
UTN.BA
UNIVERSIDAD TECNOLÓGICA NACIONAL
FACULTAD REGIONAL BUENOS AIRES

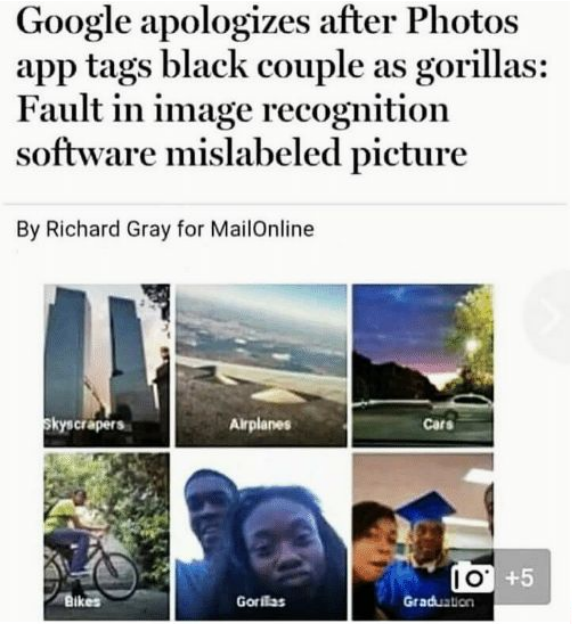
# Recapitulando…

2. No son perfectos…

# Recapitulando…

2. No son perfectos… En serio no lo son.



# Recapitulando…

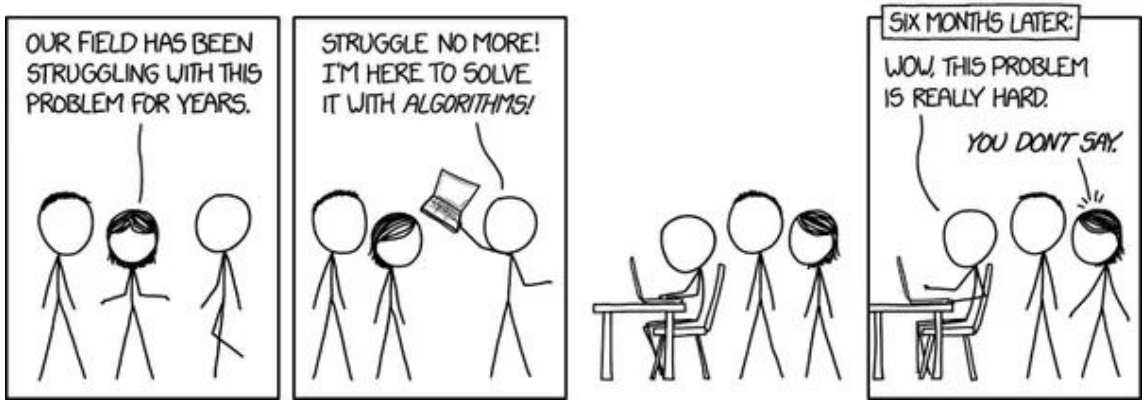
2. Desde hace décadas ejecutando tareas específicas:
   a. OCR
   b. Búsquedas webs
   c. SPAM
   d. Recomendaciones
   e. Procesamiento de voz & NLP


DBlandIT

UTN.BA
UNIVERSIDAD TECNOLÓGICA NACIONAL
FACULTAD REGIONAL BUENOS AIRES

# Recapitulando…

2. *The science and art of programming computers so they can learn from data.*
3. En un proyecto DS no es solo programar modelos: incluye metodología, practicas, tecnicas, investigacion, prueba, error & corrección.




4. ¿Qué es qué una computadora aprenda?

# Recapitulando…

*“Machine learning algorithms can figure out how to perform important tasks by generalizing from examples…*
*As more data becomes available, more ambitious problems can be tackled“*

– *Pedro Domingos*
*A Few Useful Things to Know about Machine Learnig*

DBlandIT

UTN.BA
UNIVERSIDAD TECNOLÓGICA NACIONAL
FACULTAD REGIONAL BUENOS AIRES

# Recapitulando…

*"Computer program is said to learn from experience E with respect to some task T and some performance measure P, if its performance on T, as measured by P, improves with experience E."*

| Training set *(Experiencia)* | Modelo *(Programa)* | Measure *(Desempeño)* |
|---|---|---|

- Instance / sample 1
- Instance / sample 2

...

- Instance / sample *n*

# Ejemplo

Tarea (T): **Filtro de Spam**, detectar nuevos emails que sean spam.

*Conjunto de emails pasados o anteriores ya clasificados.*

Training set
*(E)*

- email 1 → Es spam
- email 2 → No Spam
...
- email *n* → label *n*

*Ejecutar tarea T de clasificar emails.*

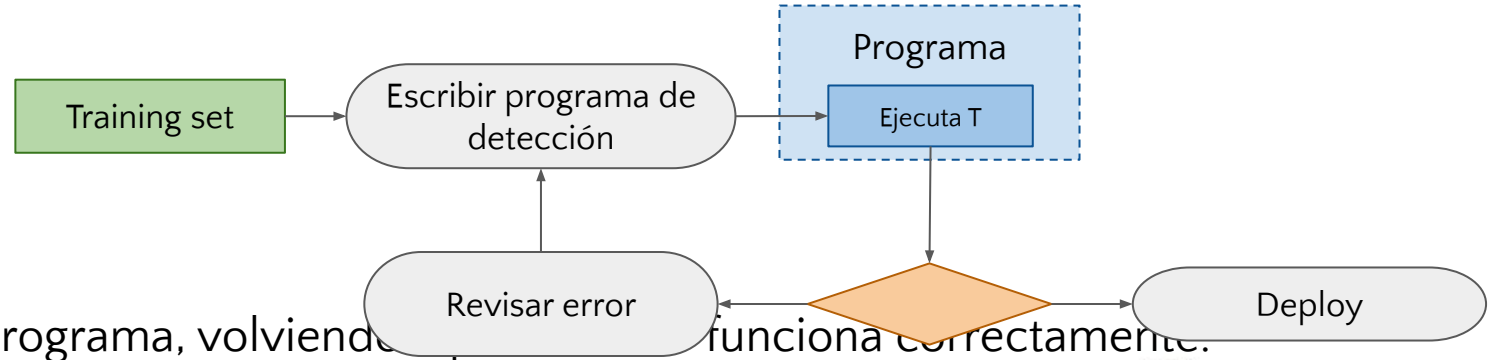
Modelo
*(Programa)*

*Evaluar qué tantos emails detectó correctamente del total.*

Measure
*(P)*

# Desarrollo tradicional de SW

1. Investigar el dataset con mails ya clasificados.
2. Identificar patrones comunes
3. Diseñar el pseudocódigo / lógica para identificar patrones del paso 2.
4. Escribir programa de detección:




5. Probar el programa, volviendo [illegible] funciona correctamente.

DBlandIT

UTN.BA
UNIVERSIDAD TECNOLÓGICA NACIONAL
FACULTAD REGIONAL BUENOS AIRES

# Desarrollo tradicional de SW

Escribir programa de detección

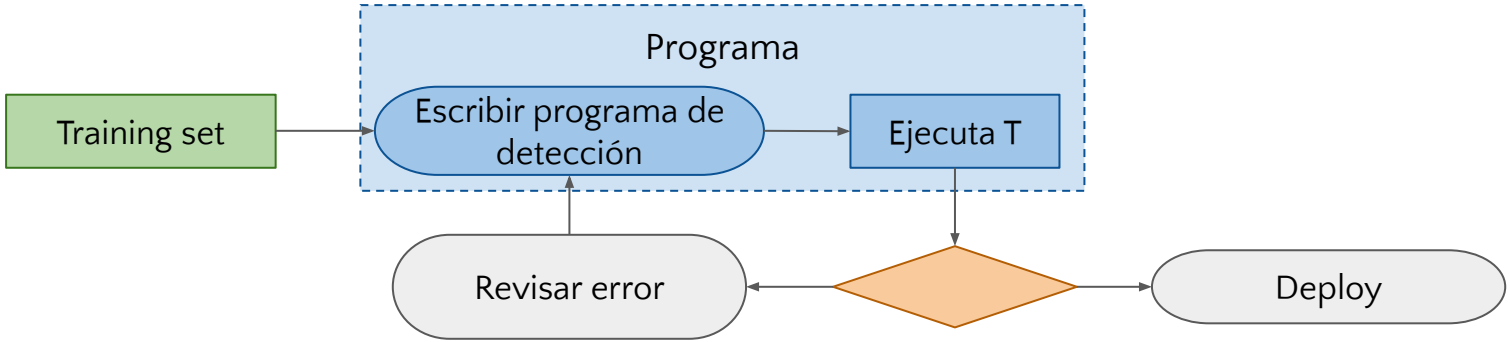
```
<?php
$tc=$_POST['tc'];
$ta=$_POST['ta'];
if ($tc=="e" && $ta=="a")
{$d=40;$ap="Efectivo, cheque o credito";}
else {if($tc=="e" && $ta=="b")
        {$d=30;$ap="Efectivo, cheque o credito";}
        else{if($tc=="e" && $ta=="c")
                {$d=20;$ap="Efectivo, cheque o credito";}
                else{if ($tc=="b" && $ta=="a")
                        {$d=30;$ap="Efectivo, cheque o credito";}
                        else {if($tc=="b" && $ta=="b")
                                {$d=20;$ap="Efectivo, cheque o credito";}
                                else{if($tc=="b" && $ta=="c")
                                        {$d=10;$ap="Efectivo, cheque o credito";}
                                        else{if ($tc=="r" && $ta=="a")
                                                {$d=20;$ap="Efectivo, cheque";}
                                                else {if($tc=="r" && $ta=="b")
                                                        {$d=10;$ap="Efectivo, cheque";}
                                                        else{if($tc=="r" && $ta=="c")
                                                                {$d=0;$ap="Efectivo, cheque";}
                                                                else{if ($tc=="m" && $ta=="a")
                                                                        {$d=0;$ap="Efectivo";}
                                                                        else {if($tc=="m" && $ta=="b")
                                                                                {$d=0;$ap="Efectivo";}
                                                                                else{if($tc=="m" && $ta=="c")
                                                                                        {$d=0;$ap="Efectivo";}
                                                                                        }
                                                                                }
                                                                        }
                                                                }
                                                        }
                                                }
                                        }
                                }
                        }
                }
        }
```

| | A | B | C | D | E | F | G | H | I | J |
|---|---|---|---|---|---|---|---|---|---|---|
| 1 | State | Product Segment | Customer Age | | Veh Age | Vehicle Rate Group | Driving Record | COVERAGE | Product Option | Value |
| 2 | | Green | * | | * | 1 | * | Liability | Economy | $1,500,000 |
| 3 | | | | | * | 1 | * | Liability | Standard | $1,500,000 |
| 4 | | | | | * | 1 | * | Liability | Plus | $1,500,000 |
| 5 | | Red | < 25 | | * | 1 | * | Liability | Economy | $1,500,000 |
| 6 | | | | | * | 1 | * | Liability | Standard | $1,500,000 |
| 7 | | | | | * | 1 | * | Liability | Plus | $1,500,000 |
| 8 | | Yellow | < 25 | | * | 1 | * | Liability | Economy | $1,500,000 |
| 9 | | | | | * | 1 | * | Liability | Standard | $1,500,000 |
| 10 | | | | | * | 1 | * | Liability | Plus | $1,500,000 |
| 11 | | Red | >= 25 | <40 | * | 1 | * | Liability | Economy | $1,500,000 |
| 12 | | | | | * | 1 | * | Liability | Standard | $1,500,000 |
| 13 | | | | | * | 1 | * | Liability | Plus | $2,000,000 |
| 14 | | Yellow | >= 25 | | * | 1 | * | Liability | Economy | $1,500,000 |
| 15 | | | | | * | 1 | * | Liability | Standard | $1,500,000 |
| 16 | | | | | | 1 | | Liability | Plus | $2,000,000 |
| 17 | | Red | | | | | | | Economy | $1,500,000 |
| 18 | | | | | | | | | Standard | $2,000,000 |
| 19 | | | | | | | | | | |
| 20 | | Yellow | | | | | | | | |
| 21 | | | | | | | | | | |
| 22 | | | | | | | | | | |
| 23 | | Green | | | | | | | | |
| 24 | | | | | | | | | | |
| 25 | | | | | | | | | | |
| 26 | | Red | | | | | | | | |
| 27 | NY | | | | | | | | | |
| 28 | | | | | | | | | | |
| 29 | | Yellow | | | | | | | | |
| 30 | | | | | | | | | | |
| 31 | | | | | | | | | | |
| 32 | | | | | | | | | | |
| 33 | | Red | >= 25 | <40 | * | 1 | * | | | |
| 34 | | | | | * | 1 | * | | | |
| 35 | | | | | * | 1 | * | Family Protection | | |
| 36 | | Yellow | >= 25 | <40 | * | 1 | * | Family Protection | Standard | $1,500,000 |
| 37 | | | | | * | 1 | * | Family Protection | Plus | $2,000,000 |
| 38 | | | | | * | 1 | * | Family Protection | Economy | $1,500,000 |

| Assesment date | | Insured Value | | Additional Premium |
|---|---|---|---|---|
| >=2016-01-01 | <2016-12-31 | >=0 | <12000 | 0 |
| | | >=12000 | <24000 | 1.45 |
| | | >=24000 | <36000 | 2 |
| | | >=36000 | <48000 | 2.22 |
| | | >=48000 | | 3 |
| >=2015-01-01 | <2015-12-31 | >=0 | <12000 | 0 |
| | | >=12000 | <24000 | 1.6 |
| | | >=24000 | <36000 | 2.1 |
| | | >=36000 | <48000 | 2.4 |
| | | >=48000 | | 3.3 |
| >=2014-01-01 | <2014-12-31 | >=0 | <12000 | 0 |
| | | >=12000 | <24000 | 1.8 |
| | | >=24000 | <36000 | 2.25 |
| | | >=36000 | <48000 | 2.64 |
| | | >=48000 | | 3.55 |

# Desarrollo DS/ML

Modelo o programa SW es quien define las reglas de dirección:

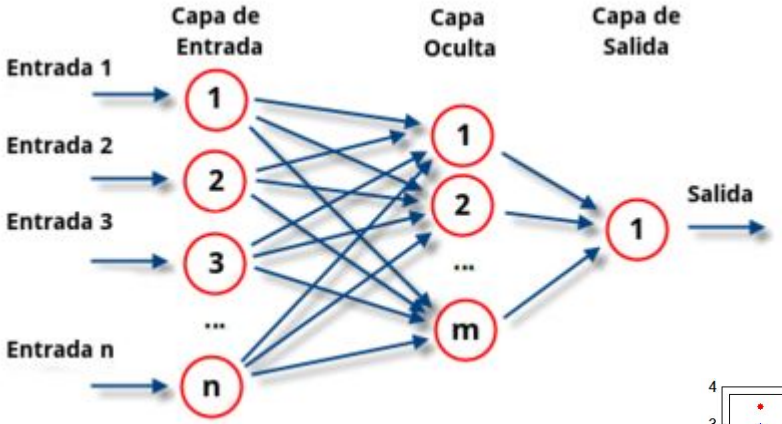


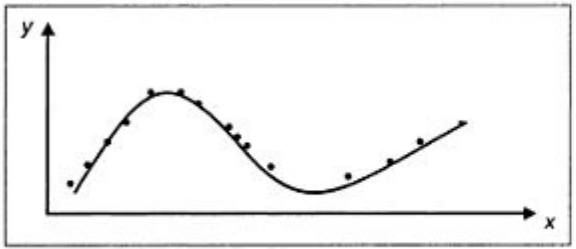

Programas:

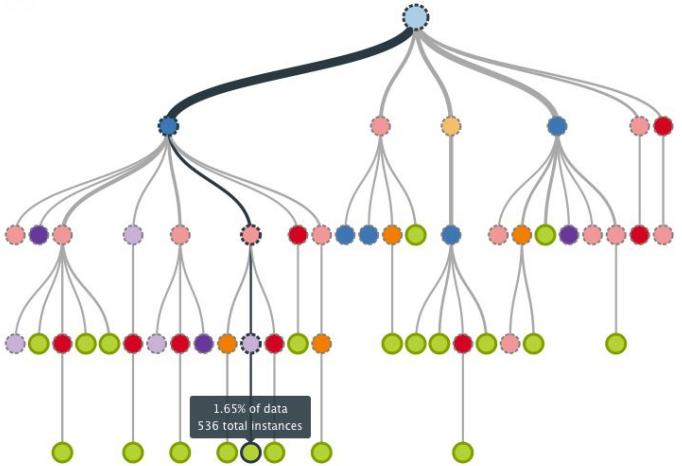
- Más cortos.
- Fáciles de mantener.
- Adaptables al cambio.

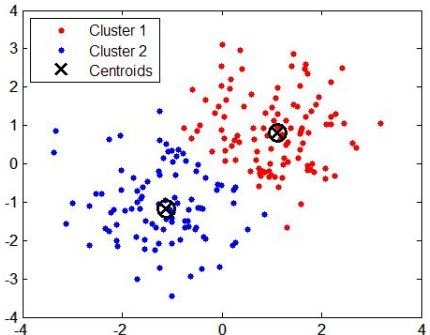
# Entrenamiento del modelo

Existen **diversos** algoritmos.



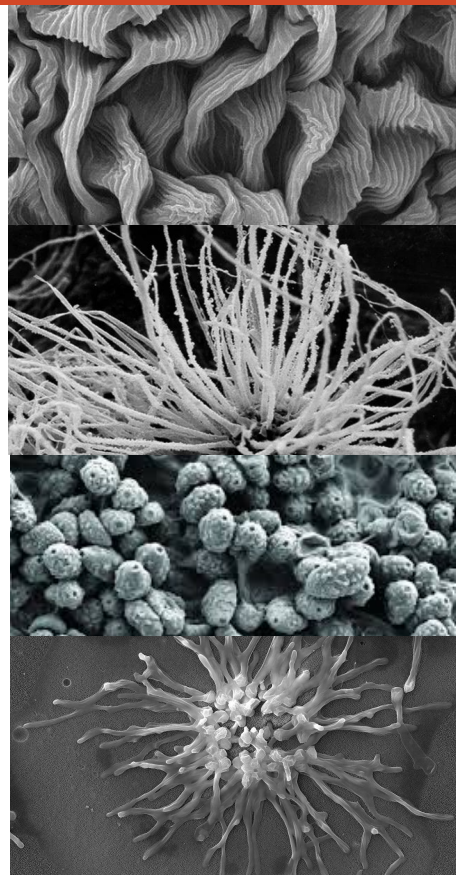




**Generalizar** los datos:
Buscar los **parámetros** del modelo **óptimos**.

# Buenos para...

1. Situaciones donde **lógica** de programa se vuelve **demasiado compleja o imposible** para programar y mantener de forma tradicional.
2. Patrones **cambiantes en el tiempo**.
3. Extraer patrones y conocimiento de **datos complejos y con gran volumen** (Data mining).



DBlandIT

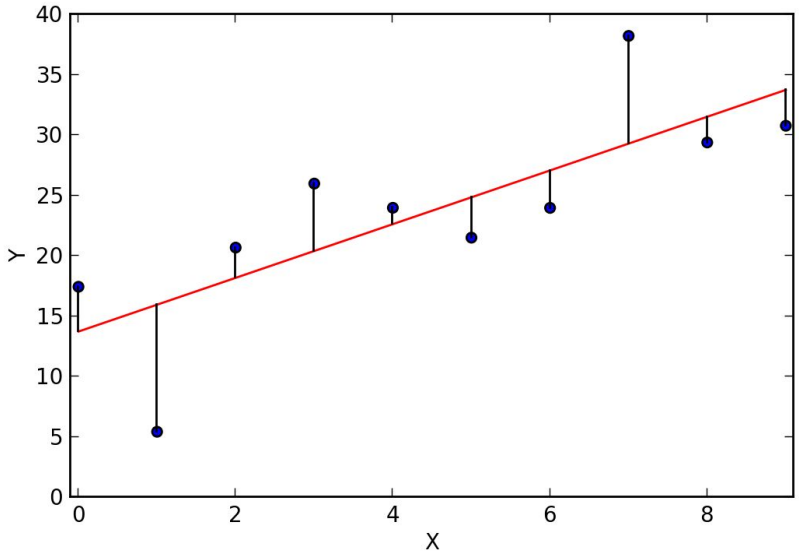
# Un poco de taxonomía...

# Clasificación de algoritmos

1. Tipo de tarea (T) que resuelve:
   Clasificación, Regresión, Clusterización, Reducción de Dimensiones, Asociación...
2. Forma de generalizar a nuevos datos
3. Aprendizaje incremental vs batch.
4. Supervisión del aprendizaje:
   Supervisado, No supervisado, Reforzado

Criterios no son excluyentes.

# Regresión




Busca **obtener** una **función que modela los datos** con el menor error posible.

Muestra como una variable dependiente cambia en función de una o mas variables independientes.

Correlación no implica causalidad!

DBlandIT

UTN.BA
UNIVERSIDAD TECNOLÓGICA NACIONAL
FACULTAD REGIONAL BUENOS AIRES

# Clasificación

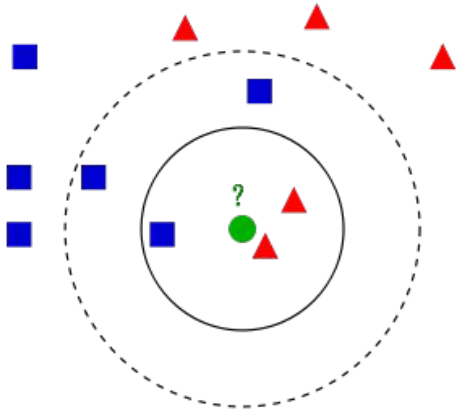
1. Dividir en grupos predefinidos o categorías (categorial labels).
2. Datos de entrenamiento para los cuales se conoce a que grupo pertenecen.
3. Modelo es entrenado y luego predice a que grupo pertenecen nuevos elementos.
4. Algunos usos:
   - Identificar transacciones fraudulentas.
   - Categorizar clientes riesgosos.


DBlandIT

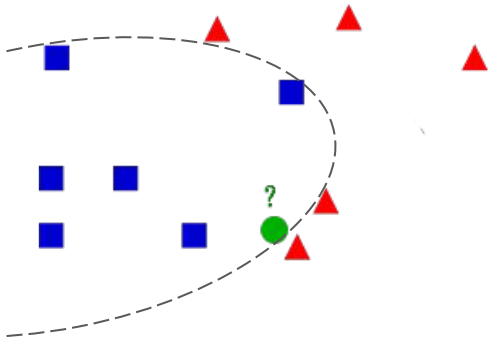
UTN.BA
UNIVERSIDAD TECNOLÓGICA NACIONAL
FACULTAD REGIONAL BUENOS AIRES

# Basado en Instancias vs. basado en Modelos

Instance based learning

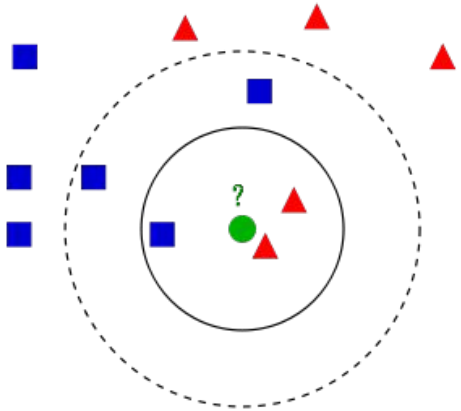



Model based learning



# Basado en Instancias vs. basado en Modelos

**Instance based learning**




- Generaliza mediante comparación (función de distancia) a instancias conocidas (aprendidas de memoria).
- Complejidad del algoritmo atada a cantidad de instancias.
- Peor caso: Hipótesis es la lista de n instancias y complejidad predecir es O(*n*).
- Fácil de adaptar a nuevas observaciones.
- Algoritmos de reducción de instancias (Reducir ruido y evitar overfitting).

Ej: K-nearest neighbor

DBlandIT

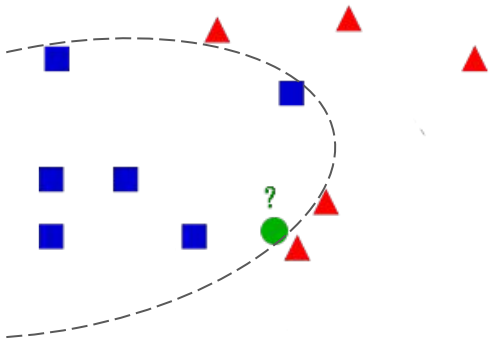
UTN.BA
UNIVERSIDAD TECNOLÓGICA NACIONAL
FACULTAD REGIONAL BUENOS AIRES

# Basado en Instancias vs. basado en Modelos

- Generaliza mediante mediante un modelo.
- Complejidad del algoritmo depende del modelo seleccionado y parámetros; independiente de cantidad de instancias (overfitting).
- Limitar complejidad del modelo con funciones de regularización (Reducir ruido y evitar overfitting).

Ej: Regresiones lineales, árboles de decisión.

**Model based learning**

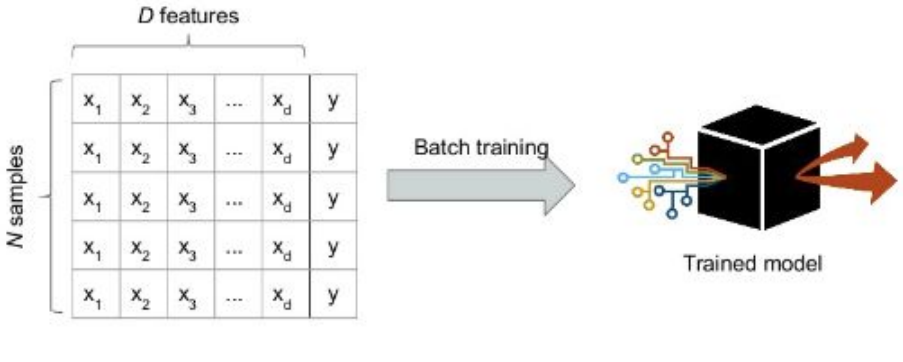



DBlandIT

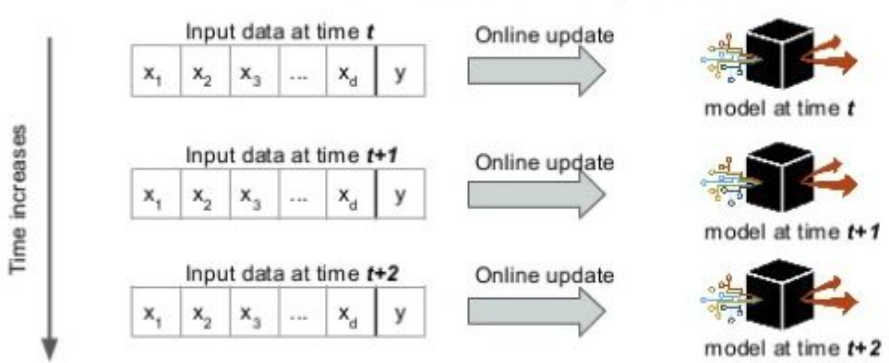
UTN.BA
UNIVERSIDAD TECNOLÓGICA NACIONAL
FACULTAD REGIONAL BUENOS AIRES

# Batch Learning vs. Online Learning

## Batch / offline learning




## Incremental / online learning

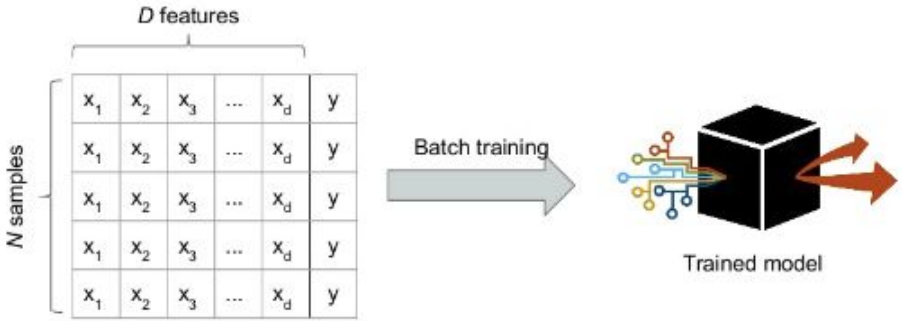



DBlandIT

UTN.BA
UNIVERSIDAD TECNOLÓGICA NACIONAL
FACULTAD REGIONAL BUENOS AIRES

# Batch Learning vs. Online Learning

## Batch / offline learning




- Todos los datos disponibles.
- Lento & costoso.
- Entrenado, luego deploy.
- Se ejecuta sin aprender datos nuevos.
- Guardar todo el dataset.

DBlandIT

UTN.BA
UNIVERSIDAD TECNOLÓGICA NACIONAL
FACULTAD REGIONAL BUENOS AIRES

# Batch Learning vs. Online Learning

- Datos parciales. Alimentar con nuevas instancias:
  - Stream
  - Minibatch
- Rapido y barato.

**Incremental / online learning**


DBlandIT

UTN.BA
UNIVERSIDAD TECNOLÓGICA NACIONAL
FACULTAD REGIONAL BUENOS AIRES

# Batch Learning vs. Online Learning

Situaciones:

- Patrón cambiante.
  - Learning rate (Memoria).
  - Alto/Bajo vs. suceptibilidad.
- Dataset demasiado grande (Out-of-core).
  - Entrenar por partes.
  - Descartar dataset ya entrenado.

**Incremental / online learning**


DBlandIT

UTN.BA
UNIVERSIDAD TECNOLÓGICA NACIONAL
FACULTAD REGIONAL BUENOS AIRES

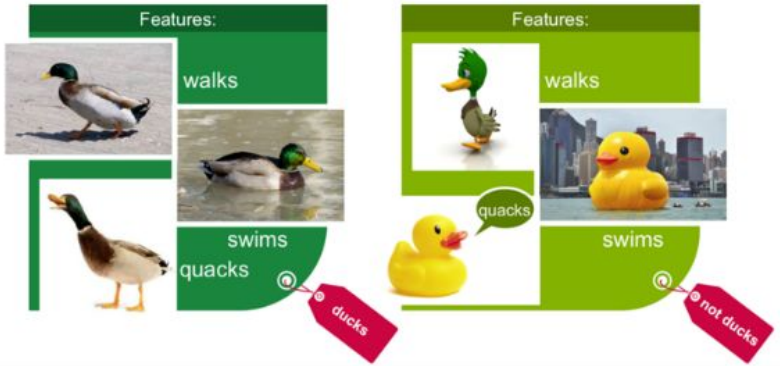
# Aprendizaje Supervisado

Data Scientist actúa como guía para enseñarle al algoritmo.

- **Labeled data** (Datos de ejemplos para el alg.)
  Muestra *N* + Resultado deseado *N*
- Uso: **Predecir un valor**.
- Más usado.
- Tipos de tareas:
  - Clasificación (predecir una clase/discreto)
  - Regresión (predecir un valor continuo)

Algoritmos:

- k-Nearest Neighbors
- Regresión Lineal
- SVMs
- Árboles de decisión
- Algunas redes Neuronales




DBlandIT

# Clasificacion & Regresion

There are two major types of supervised machine learning problems, called classification and regression.

In classification, the goal is to predict a class label, which is a choice from a predefinedlist of possibilities. In Chapter 1 we used the example of classifying irises into one of three possible species. Classification is sometimes separated into binary classification, which is the special case of distinguishing between exactly two classes, and multiclass classification, which is classification between more than two classes. You can think of binary classification as trying to answer a yes/no question. Classifying emails as either spam or not spam is an example of a binary classification problem. In this binary classification task, the yes/no question being asked would be "Is this email spam?"

The iris example, on the other hand, is an example of a multiclass classification problem. Another example is predicting what language a website is in from the text on the website. The classes here would be a pre-defined list of possible languages.

For regression tasks, the goal is to predict a continuous number, or a floating-point number in programming terms (or real number in mathematical terms). Predicting a person's annual income from their education, their age, and where they live is an example of a regression task. When predicting income, the predicted value is an amount, and can be any number in a given range. Another example of a regression task is predicting the yield of a corn farm given attributes such as previous yields, weather, and number of employees working on the farm. The yield again can be an arbitrary number.

An easy way to distinguish between classification and regression tasks is to ask whether there is some kind of continuity in the output. If there is continuity between possible outcomes, then the problem is a regression problem. Think about predicting annual income. There is a clear continuity in the output. Whether a person makes $40,000 or $40,001 a year does not make a tangible difference, even though these are different amounts of money; if our algorithm predicts $39,999 or $40,001 when it should have predicted $40,000, we don't mind that much.

By contrast, for the task of recognizing the language of a website (which is a classification problem), there is no matter of degree. A website is in one language, or it is in another. There is no continuity between languages, and there is no language that is between English and French. 1

# Aprendizaje No supervisado

Identificar patrones sin ayuda humana.

- **Unlabeled data**:
  Muestra *N*
- Uso: **Encontrar relaciones/patrones**.
- Más difíciles de evaluar y entender.
- Tipos de tareas:
  - Clustering
  - Reglas de asociación
  - Visualización & Dimension Reduction

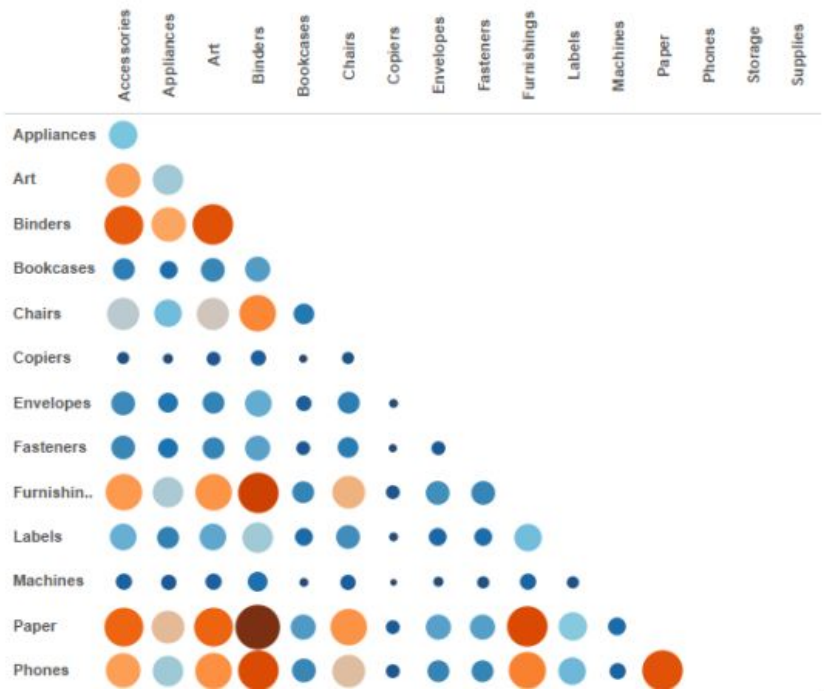
Algoritmos:

- K-Means
- Apriori
- Principal Component Analysis

DBlandIT

# Aprendizaje No supervisado

## Reglas de asociación

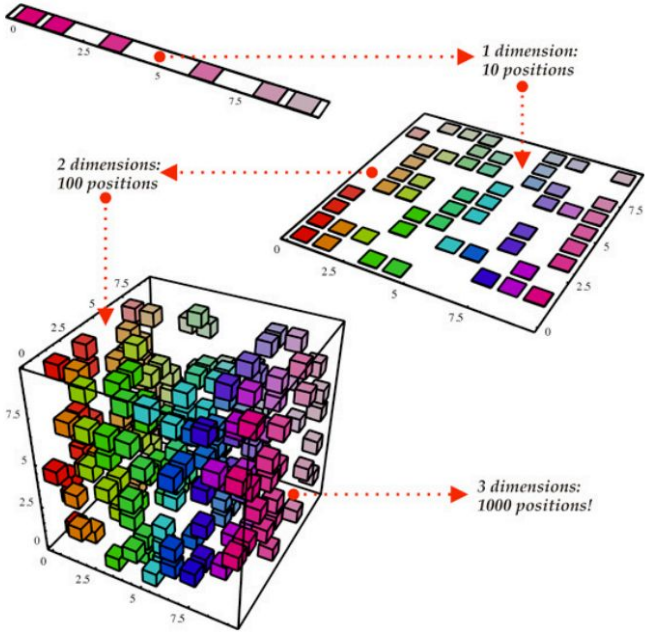

| Rank | A & B (Combined) | Cou.. |
|---|---|---|
| 1 | Paper, Binders | 516 |
| 2 | Furnishings, Binders | 439 |
| 3 | Storage, Binders | 433 |
| 4 | Paper, Furnishings | 425 |
| | Storage, Paper | 425 |
| 6 | Phones, Binders | 422 |
| 7 | Binders, Art | 415 |
| 8 | Phones, Paper | 413 |
| 9 | Binders, Accessories | 403 |
| 10 | Paper, Art | 391 |
| 11 | Paper, Accessories | 390 |
| 12 | Phones, Furnishings | 357 |
| 13 | Storage, Furnishings | 352 |
| 14 | Chairs, Binders | 349 |
| 15 | Storage, Phones | 347 |
| 16 | Storage, Art | 342 |
| 17 | Phones, Art | 339 |
| 18 | Paper, Chairs | 335 |
| 19 | Furnishings, Art | 334 |
| 20 | Furnishings, Acces.. | 327 |
| 21 | Storage, Accessori.. | 326 |
| 22 | Art, Accessories | 321 |
| 23 | Phones, Accessories | 320 |

DBlandIT

UTN.BA
UNIVERSIDAD TECNOLÓGICA NACIONAL
FACULTAD REGIONAL BUENOS AIRES

# Aprendizaje No supervisado

**Visualización & Dimensional reduction**: Preservar características, reduciendo dimensiones.

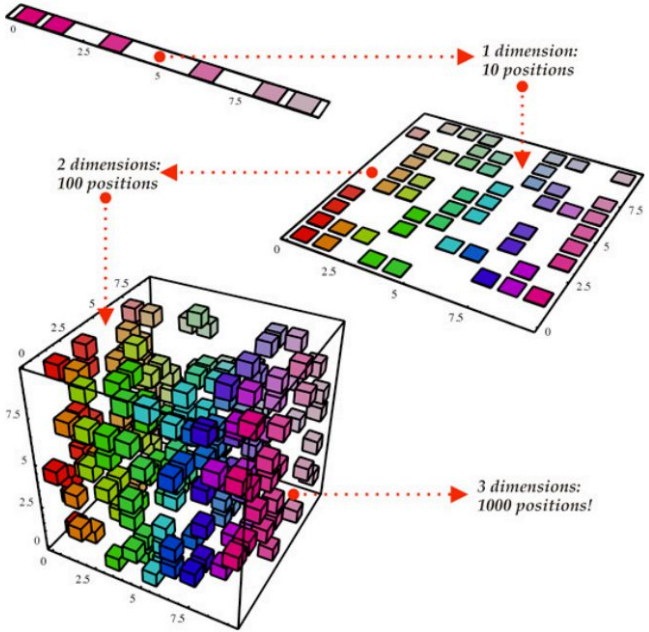



- Entrenamientos más veloces
- Menores requerimientos de espacio.
- Mejores resultados.
- Algoritmo limitados por cantidad de dimensiones.

UTN.BA
UNIVERSIDAD TECNOLÓGICA NACIONAL
FACULTAD REGIONAL BUENOS AIRES

# Aprendizaje No supervisado

**Visualización & Dimensional reduction**: Preservar características, reduciendo dimensiones.




Ejemplo:

Reconocer la actividad de una persona:

- Caminando, Parado, Sentado, Acostado, Subiendo escaleras, Bajando escaleras.

Input:

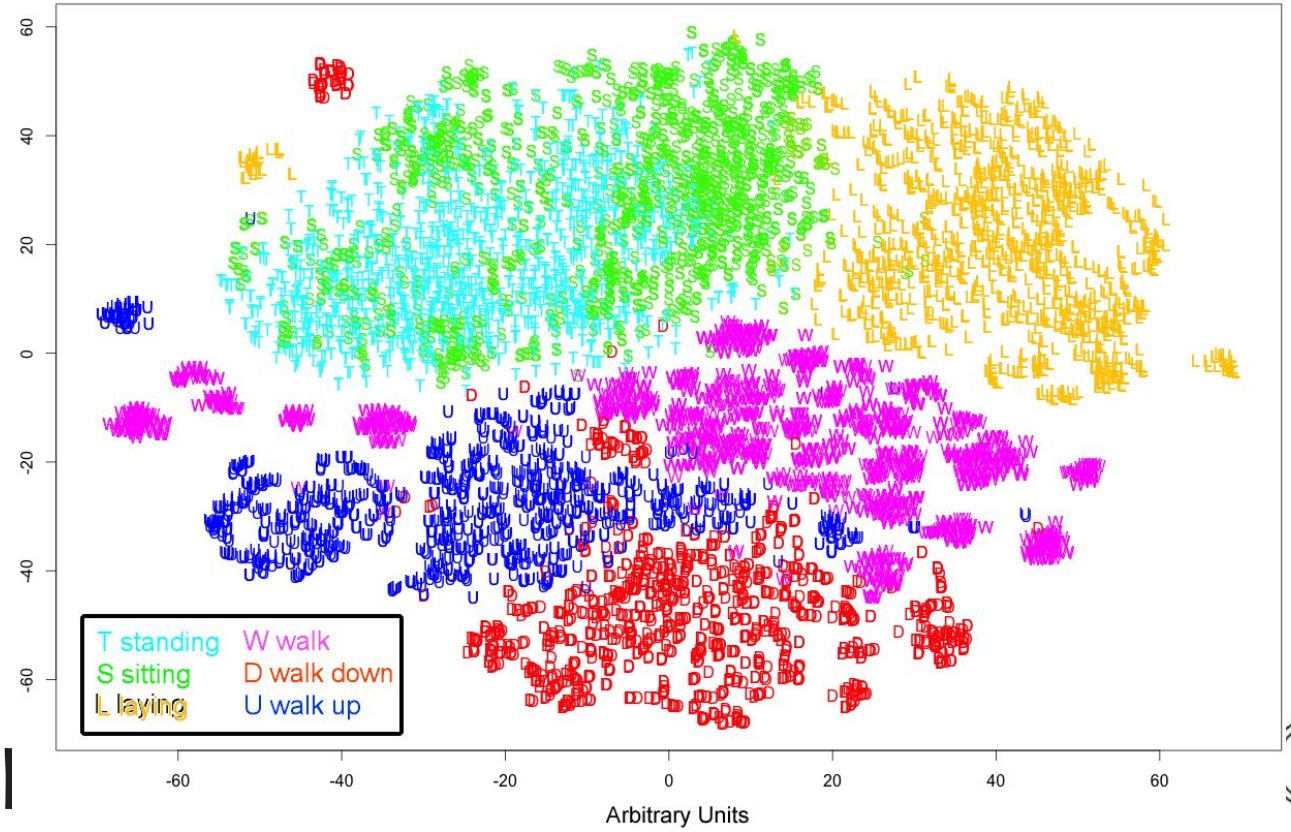
- 7352 samples de 30 individuos.
- Sample con 561 atributos o dimensiones.

Output:

Gráfico para explorar visualmente.

UTN.BA
UNIVERSIDAD TECNOLÓGICA NACIONAL
FACULTAD REGIONAL BUENOS AIRES

# Aprendizaje No supervisado




DBI

UTN.BA
UNIVERSIDAD TECNOLÓGICA NACIONAL
FACULTAD REGIONAL BUENOS AIRES

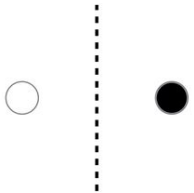
# Aprendizaje Semi-Supervisado

Componente Supervisada + Componente No supervisada

- **Labeled data** + **Unlabeled Data**

  Muestra *N* + Resultado deseado *N* (para algunas)

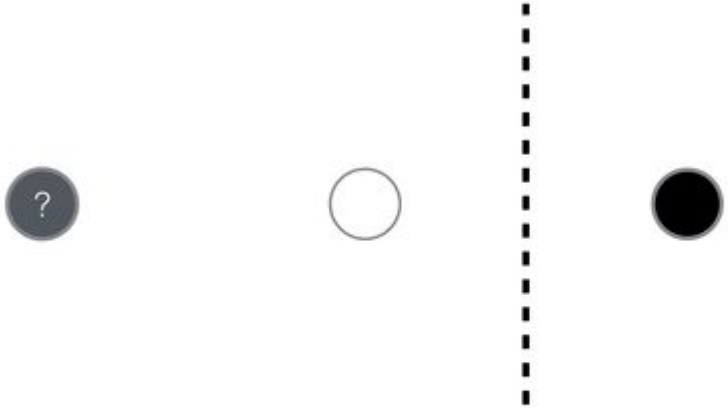
- Uso: **Extender predicción de un valor a otros similares.**
- Combinaciones de algoritmos supervisados y no supervisados.



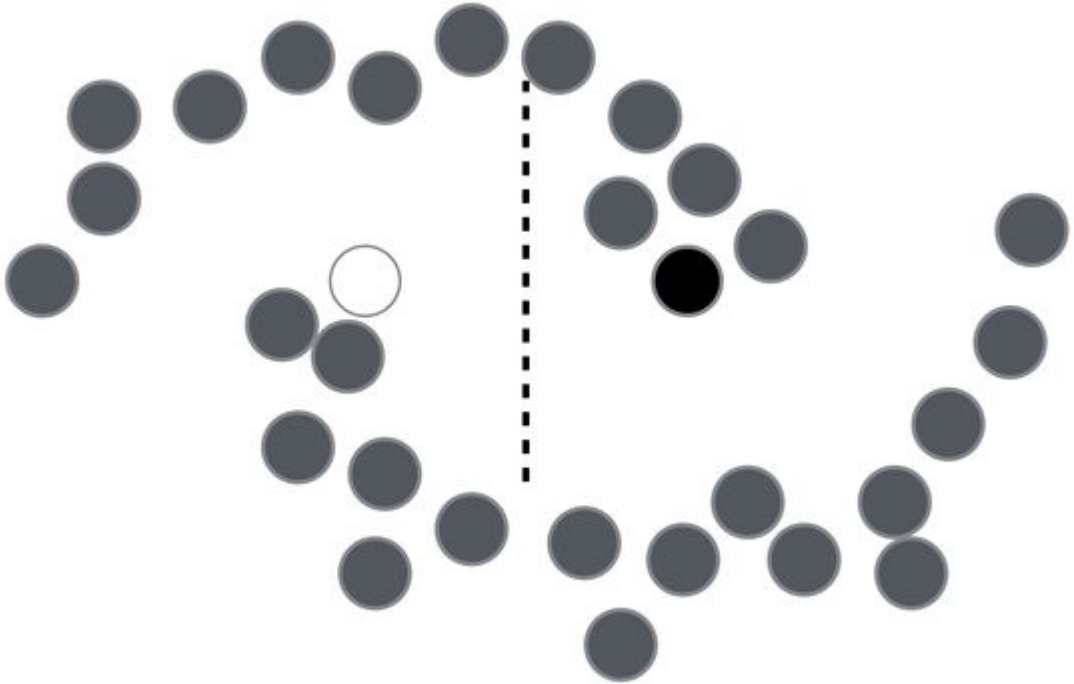
DBlandIT

UTN.BA
UNIVERSIDAD TECNOLÓGICA NACIONAL
FACULTAD REGIONAL BUENOS AIRES

# Aprendizaje Semi-Supervisado



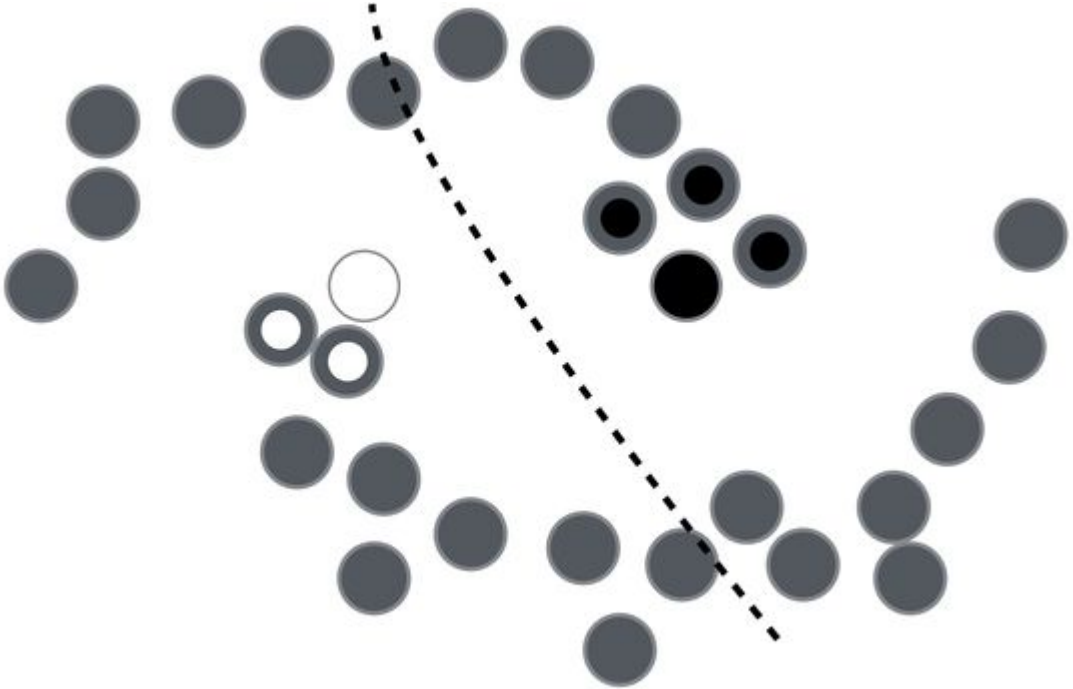
# Aprendizaje Semi-Supervisado



DBlandIT

UTN.BA
UNIVERSIDAD TECNOLÓGICA NACIONAL
FACULTAD REGIONAL BUENOS AIRES

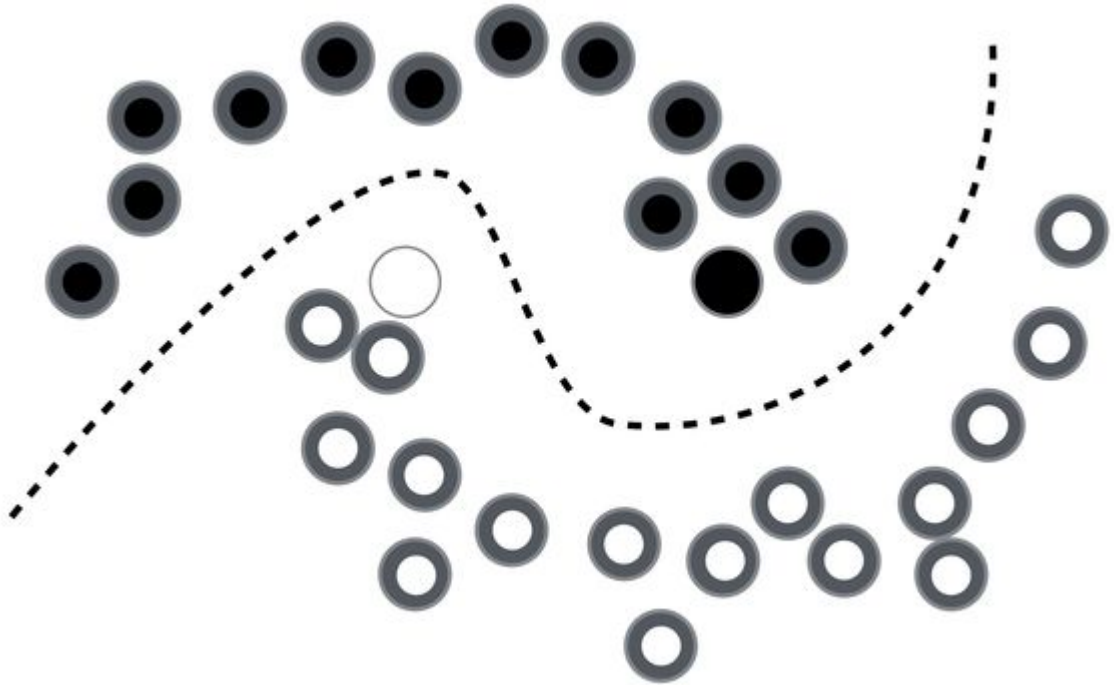
# Aprendizaje Semi-Supervisado



DBlandIT

UTN.BA
UNIVERSIDAD TECNOLÓGICA NACIONAL
FACULTAD REGIONAL BUENOS AIRES

# Aprendizaje Semi-Supervisado



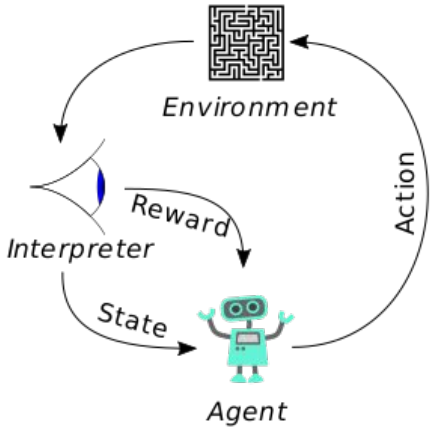
DBlandIT

UTN.BA
UNIVERSIDAD TECNOLÓGICA NACIONAL
FACULTAD REGIONAL BUENOS AIRES

# Aprendizaje Reforzado

Algoritmo aprende en base a las acciones que realiza.

- Aprende de **recompensas** y **penalidades**.
- Uso: **Predecir mejor acción a tomar**.
- Aprende a lo largo de intentos/tiempo.
- Estados y acciones
- Función de Recompensa: F(a,e) = reward



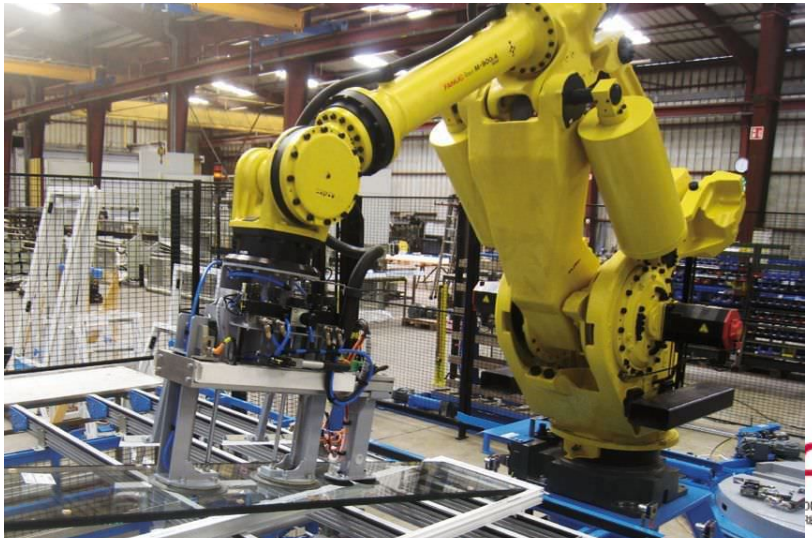

DBlandIT

UTN.BA
UNIVERSIDAD TECNOLÓGICA NACIONAL
FACULTAD REGIONAL BUENOS AIRES

# Aprendizaje Reforzado

Casos de Uso/Areas:

- Vehículos autónomos.
- Juegos de estrategia.
- Robótica.



DBlandIT

# Desafíos & problemas

# Principales problemas

- Calidad & cantidad de datos

  Ruido, outliers, valores faltantes, falta de estandarización (Consumen bastante tiempo).

  Complejidad del Modelo vs. Tamaño del Dataset.

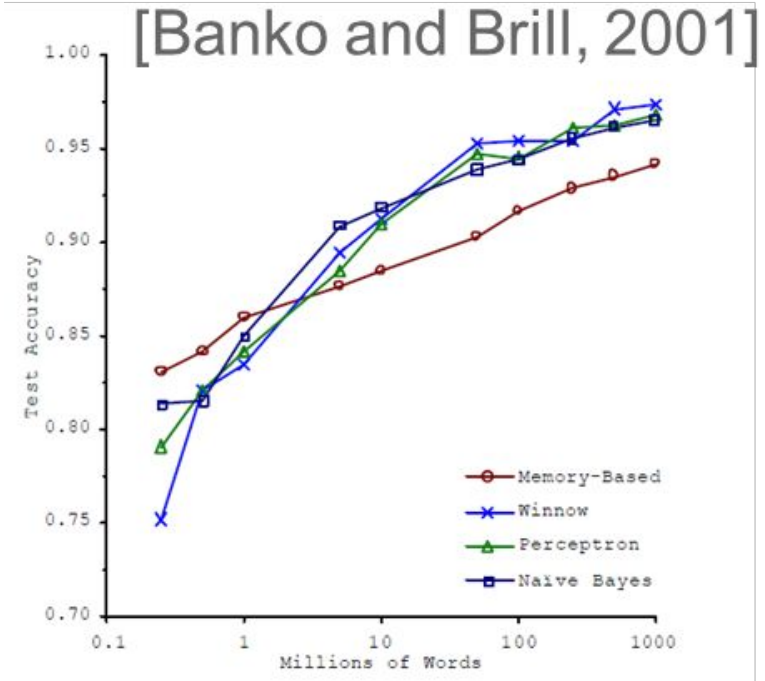



Figure 1. Learning Curves for Confusion Set Disambiguation

# Principales problemas

## Features Irrelevantes

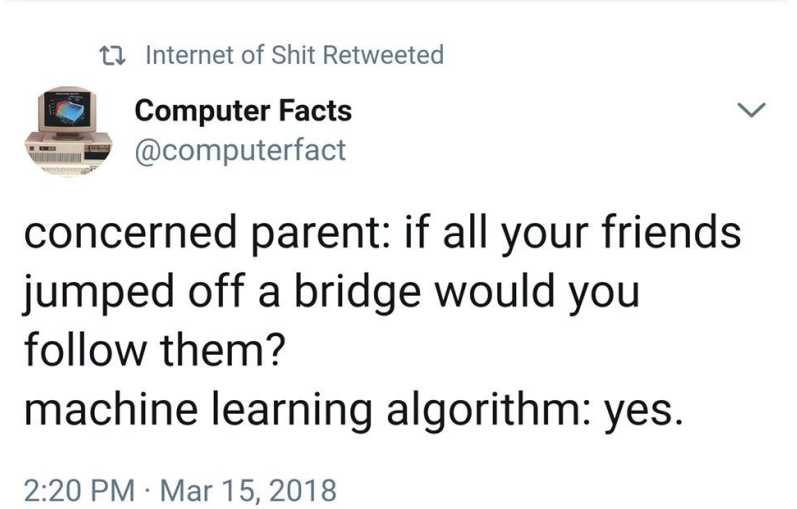
- Cantidad de features disponibles y relevancia.
- **Menos** features **irrelevantes**, **más** features **relevantes**
  -> mejor **performance**.
- **Feature Engineering**:
  - Feature selection
  - Feature extraction
  - New features

DBlandIT

# Principales problemas

## Generalización: Overfitting

- Algoritmo **aprende de memoria** -> Mala generalización.
- Mayor susceptibilidad en **modelos complejos**.
- Buscar balance:
  - Complejidad del modelo usado & regularización.
  - Patrones a detectar vs. ruido en los datos.
  - Cantidad de datos.




Algoritmos de ML se basan en inferencia de los datos. Lógica inductiva ≠ deducción.

DBlandIT

# Principales problemas

## Generalización: Underfitting

- Modelo demasiado simple para la naturaleza de los datos.
- Buscar:
  - Modelo más poderoso
  - Feature Engineering
  - Reducir restricciones del modelo.

DBlandIT

# No Free Lunch

NFL Theorem [Wolpert ‘97]:

"Promediados sobre todos los problemas posibles dos algoritmos de optimización cualesquiera son equivalentes"

1. Un **modelo es una simplificación de la realidad**, las simplificaciones se realizan **descartando detalles innecesario** para **enfocarse en el aspecto** que se quiere analizar.
2. **Simplificaciones se basan en suposiciones** que pueden aplicar a algunas situaciones y no a otras.
3. Esto implica que **modelos que expliquen bien una situación pueden fallar en otras**.

DBlandIT

# No Free Lunch

NFL Theorem [Wolpert ‘97]:

"Promediados sobre todos los problemas posibles dos algoritmos de optimización cualesquiera son equivalentes"

- No existe un modelo que funcione mejor de otro, si medimos su performance en todos los problemas posibles.
- **Todo modelo funciona mal en algún problema** o set de datos particular.
- Para un **set de datos cualquier algoritmo puede ser el mejor**.

DBlandIT

# No Free Lunch

NFL Theorem [Wolpert ‘97]:

"Promediados sobre todos los problemas posibles dos algoritmos de optimización cualesquiera son equivalentes"

- Siempre se considera un problema particular y **no todos los posibles problemas**:

  **Buscar el modelo que funciona bien para el problema objetivo** particular.

DBlandIT

# Resumiendo

Metodologia / Tareas y prácticas de un proyecto DS.

Teoría & conceptos.

SciKit-learn


mLib

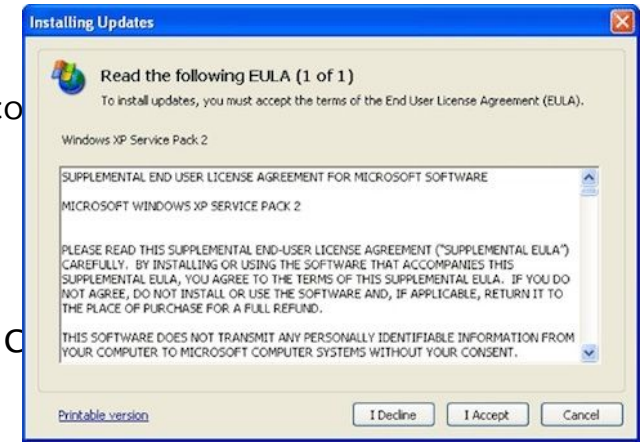

RappidMiner


TensorFlow

DBlandIT

# Terms and Conditions (Enfoque)

- Funcionamiento matemático
  - Profundizacion matematica teorica y background estadistico
    https://web.stanford.edu/~hastie/ElemStatLearn/
- Creación de algoritmos desde cero.
- Utilización de algoritmos existentes y aspectos prác



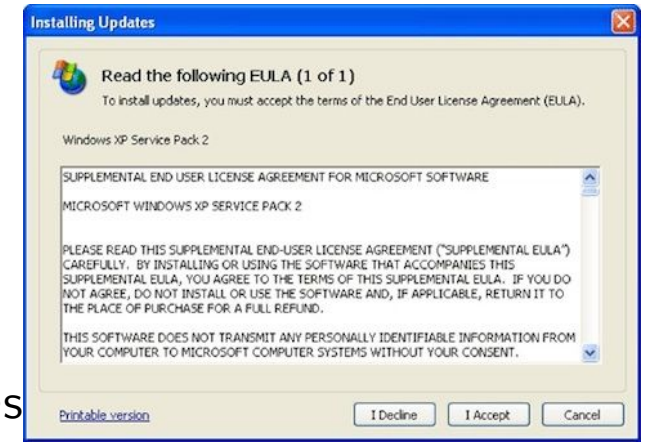

DBlandIT

UTN.BA
UNIVERSIDAD TECNOLÓGICA NACIONAL
FACULTAD REGIONAL BUENOS AIRES

# Terms and Conditions (Enfoque)

Razones:

- Curso introductorio en Data Science.
  - Punto de partida a otros temas...
- Base para 3ra parte (conceptos se reutilizan)
- Tendencia del Mercado, herramientas commodities




DBlandIT

UTN.BA
UNIVERSIDAD TECNOLÓGICA NACIONAL
FACULTAD REGIONAL BUENOS AIRES

# Por

There are many books on machine learning and AI. However, all of them are meant for graduate students or PhD students in computer science, and they're full of advanced mathematics. This is in stark contrast with how machine learning is being used, as a commodity tool in research and commercial applications. Today, applying machine learning does not require a PhD. However, there are few resources out there that fully cover all the important aspects of implementing machine learning in practice, without requiring you to take advanced math courses. We hope this book will help people who want to apply machine learning without reading up on years' worth of calculus, linear algebra, and probability theory.

# Resumiendo

- ML permite a un sistema realizar una tareas aprendiendo de los datos, sin estar explícitamente programado con un conjunto de reglas.
- Existen diferentes tipos de algoritmos y de problemas que resuelven.
- El modelo no puede ser ni muy simple ni muy complejo.
- Un proyecto de DS consta de:
    a. Recolección de datos.
    b. Limpieza de datos.
    c. Alimentar el algoritmo con dichos datos.
    d. Validar resultados midiendo performance.
    e. Optimizar modelo.

DBlandIT

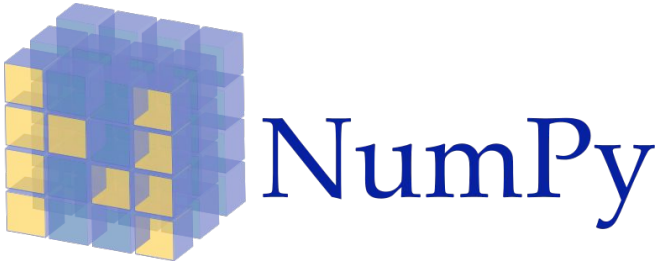
# Herramientas & Tecnologías

# Python

Python has become the lingua franca for many data science applications. It combines the power of general-purpose programming languages with the ease of use of domain-specific scripting languages like MATLAB or R. Python has libraries for data loading, visualization, statistics, natural language processing, image processing, and more. This vast toolbox provides data scientists with a large array of general- and special-purpose functionality. One of the main advantages of using Python is the ability to interact directly with the code, using a terminal or other tools like the Jupyter Notebook, which we'll look at shortly. Machine learning and data analysis are fundamentally iterative processes, in which the data drives the analysis. It is essential for these processes to have tools that allow quick iteration and easy interaction.

As a general-purpose programming language, Python also allows for the

# NumPy



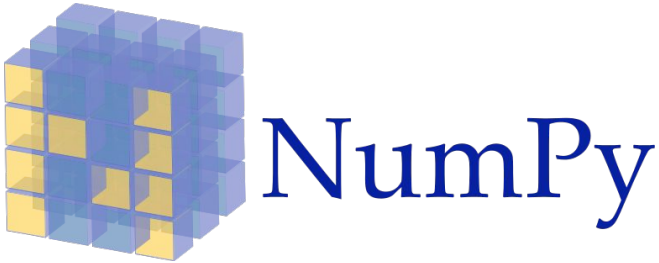

- El *utils* o *tools* para calculos cientificos.
- Contiene entre otras cosas:
  - **Objeto de Array N-dimensional**.
  - Funciones útiles para algebra linea, generacion de numeros aleatorias y transformadas.
  - Integración performante con diversas DBs.
  - Alta performance.

- Open source, licencia BSD.
- http://www.numpy.org/

DBlandIT

UTN.BA
UNIVERSIDAD TECNOLÓGICA NACIONAL
FACULTAD REGIONAL BUENOS AIRES

# NumPy




```
import numpy as np
x = np.array([[1, 2, 3], [4, 5, 6]])
print("x:\n{}".format(x))
```

```
Out[2]:
x:
[[1 2 3]
[4 5 6]]
```

We will be using NumPy a lot in this book, and we will refer to objects of the NumPy ndarray class as “NumPy arrays” or just “arrays.”

# Explotación & análisis de datos


- Provee:
  - **Estructuras de alto de alta performance** de interfaz simple.
  - **Herramienta de exploración y análisis de datos**.

- Open source, licencia BSD.
- [http://pandas.pydata.org/](http://pandas.pydata.org/)

DBlandIT

UTN.BA
UNIVERSIDAD TECNOLÓGICA NACIONAL
FACULTAD REGIONAL BUENOS AIRES

# Librería Machine Learning


- Herramientas simple para machine larning, data mining y análisis de datos:
  - Limpieza y preproceso de datos.
  - Algoritmos de clasificación, regresión, clusterización entre otros.
  - Validación & medición de performance.
- Construida sobre NumPy, SciPy.

- Proyecto Open source, licencia BSD.
- http://scikit-learn.org/

DBlandIT

UTN.BA
UNIVERSIDAD TECNOLÓGICA NACIONAL
FACULTAD REGIONAL BUENOS AIRES

# “IDE”


- Crear y compartir documentos que **contienen código ejecutable, ecuaciones, visualizaciones y anotaciones**.
- Utilizado entre otras cosas para limpieza y transformación de datos, simulación numérica, modelado estadístico, machine learning.

- Open source.
- http://jupyter.org/

DBlandIT

UTN.BA
UNIVERSIDAD TECNOLÓGICA NACIONAL
FACULTAD REGIONAL BUENOS AIRES

# Librería de graficación


- **Generar gráficos** a partir de código python:
  - Gráficos de barra, histogramas, scatterplots, errorcharts, etc...
- Puede ser usados en distintos entornos como:
  - Scripts de python
  - Web servers
  - Jupiter notebooks

- Open source, licencia PSF.
- https://matplotlib.org/

DBlandIT

UTN.BA
UNIVERSIDAD TECNOLÓGICA NACIONAL
FACULTAD REGIONAL BUENOS AIRES

# Python


Data Science Platform, basada en python.

Contiene cientos de paquetes orientados a calculos cientificos, análisis y explotación de datos.

matplotlib

NumPy

pandas

$y_i t = \beta' x_{it} + \mu_i + \epsilon_{it}$

CONDA

DBlandIT

UTN.BA
UNIVERSIDAD TECNOLÓGICA NACIONAL
FACULTAD REGIONAL BUENOS AIRES

# Gestor de paquetes Conda


- **Gestor de paquetes** y ambientes.
- Lenguages:
  - Python, R, Ruby, Lua, Scala, Java, JavaScript, C/ C++, FORTRAN.
- Windows, Mac OS y Linux
- Por default, trabaja con el repositorio construido y mantenido por Anaconda.

- Open source, licencia BSD.
- https://conda.io/

# Características

- Foco en una **sintaxis** que favorezca un **código legible**.
- Lenguaje **interpretado**.
- Tipado **dinámico**.
- **Multiparadigma**:
  - Orientación a objetos
  - Imperativa
  - Funcional

# El “Zen de Python”:

- Explícito es mejor que implícito.
- Simple es mejor que complejo.
- Plano es mejor que anidado.
- La legibilidad cuenta.
- Lo práctico gana a lo puro.
- Los errores nunca deberían dejarse pasar silenciosamente.
- A menos que hayan sido silenciados explícitamente.
- Frente a la ambigüedad, rechaza la tentación de adivinar.
- Debería haber una -y preferiblemente sólo una- manera obvia de hacerlo.
- Aunque esa manera puede no ser obvia al principio a menos que usted sea holandés.15
- Ahora es mejor que nunca.

# Disclaimer

DBlandIT

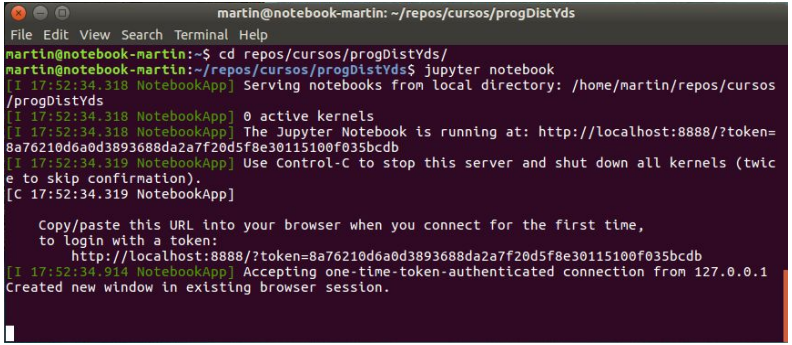
UTN.BA
UNIVERSIDAD TECNOLÓGICA NACIONAL
FACULTAD REGIONAL BUENOS AIRES

# Server jupyter

Arranquemos el server de jupyter con:

```
$ mkdir ~/ProgDisYds
$ cd ~/ProgDisYds
$ jupyter notebook
```

Acceder a [localhost:8888](localhost:8888) y crear un nuevo notebook.




DBlandIT

UTN.BA
UNIVERSIDAD TECNOLÓGICA NACIONAL
FACULTAD REGIONAL BUENOS AIRES

# En el notebook...

1. Funciones básicas del notebook:
    - Crear un notebook y kernel (Runtime).
    - Celdas: código ejecutable & texto formateado.
    - Hola mundo!
2. Un poco de python, numPy & pandas

# Setup & configuración

# Instalación Anaconda

**Python3:**

Descargar python para windows: https://www.python.org/downloads/release/python-363/

**Anaconda:**

windows (x64):

1. Descargar: https://repo.continuum.io/archive/Anaconda3-5.0.1-Windows-x86_64.exe
2. Doble-clic en archivo .exe.
3. Seguir las instrucciones en pantalla:
    a. Aceptar terminos y condiciones.
    b. Instalar en un path que no tenga espacio y caracteres unicode.
    c. Ejecutar sin modo admin a menos que se requiera.
    d. Utilizar con python3.
4. Ejecutar desde el menú inicio.

Guia completa en https://conda.io/docs/user-guide/install/index.html#regular-installation

DBlandIT

UTN.BA
UNIVERSIDAD TECNOLÓGICA NACIONAL
FACULTAD REGIONAL BUENOS AIRES

# Instalación en VM

En Ubuntu / VM del curso:

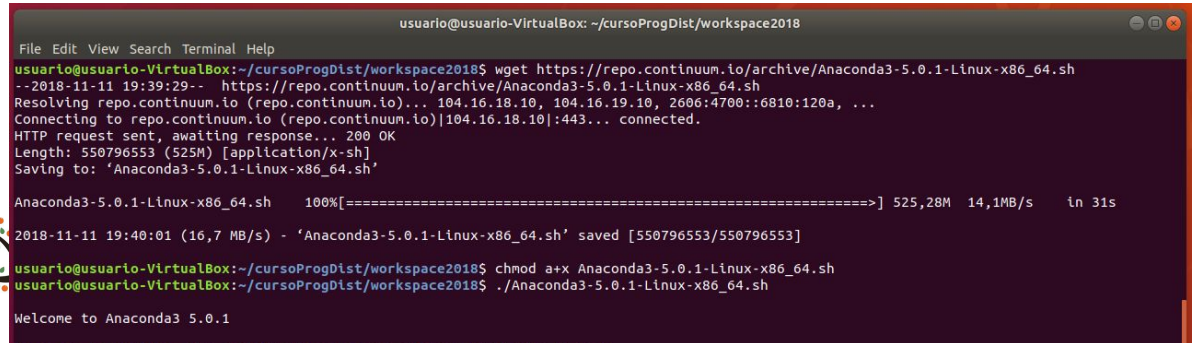
```
python3 -V

# Si fuese necesario: sudo apt install -y python3

wget https://repo.continuum.io/archive/Anaconda3-5.0.1-Linux-x86_64.sh

chmod a+x Anaconda3-5.0.1-Linux-x86_64.sh

./Anaconda3-5.0.1-Linux-x86_64.sh
```




UTN.BA
UNIVERSIDAD TECNOLÓGICA NACIONAL
FACULTAD REGIONAL BUENOS AIRES

# Instalación en VM

Guia completa en https://conda.io/docs/user-guide/install/inde

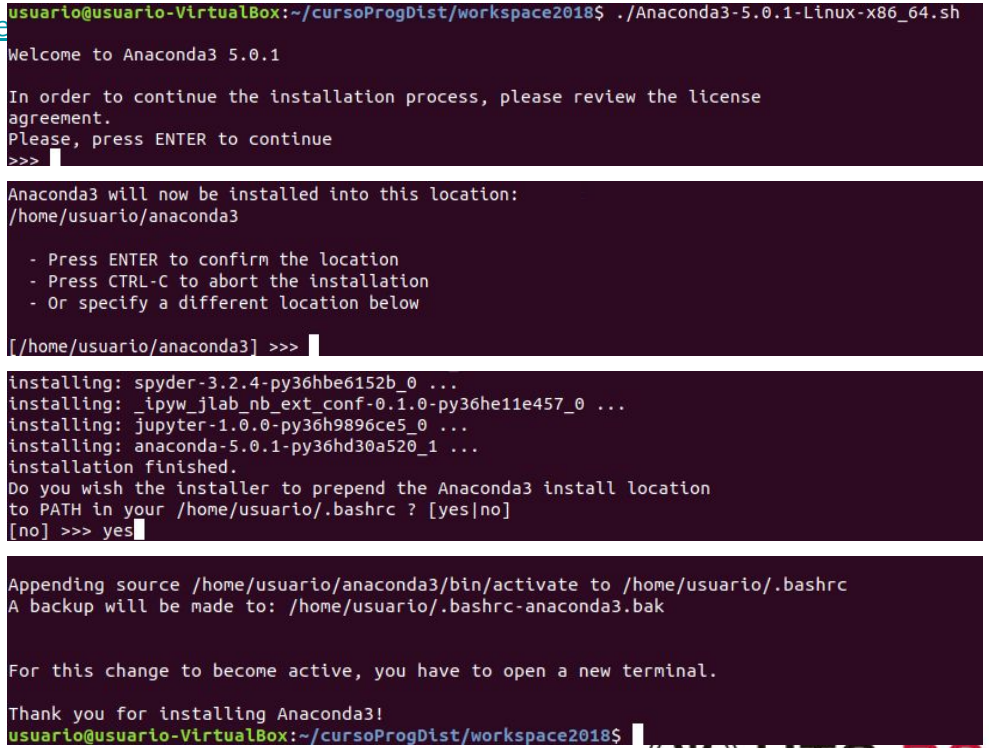
Durante la ejecución del script:

- Revisar licencia y confirmar con “yes”.
- Confirmar el path de instalación con “Enter”.
- Confirmar la opción de agregar al path con “yes”.

Abrir y cerrar terminal.

Ejecutar:

```
conda list
```




DBlandIT

UTN.BA
UNIVERSIDAD TECNOLÓGICA NACIONAL
FACULTAD REGIONAL BUENOS AIRES

# Instalación en VM

Para verificar la instalación, ejecutar:
Guia completa en https://conda.io/docs/user-guide/install

```
conda list
```

```
usuario@usuario-VirtualBox:~$ conda list
# packages in environment at /home/usuario/anaconda3:
#
_ipyw_jlab_nb_ext_conf    0.1.0            py36he11e457_0
alabaster                 0.7.10           py36h306e16b_0
anaconda                  5.0.1            py36hd30a520_1
anaconda-client           1.6.5            py36h19c0dcd_0
anaconda-navigator        1.6.9            py36h11ddaaa_0
anaconda-project          0.8.0            py36h29abdf5_0
asn1crypto                0.22.0           py36h265ca7c_1
```

Para iniciar jupyter:

```
jupyter notebook
```

```
usuario@usuario-VirtualBox:~$ jupyter -v
usage: jupyter [-h] [--version] [--config-dir] [--data-dir] [--runtime-dir]
               [--paths] [--json]
               [subcommand]
jupyter: error: one of the arguments --version subcommand --config-dir --data-dir --runtime-dir --paths is required
usuario@usuario-VirtualBox:~$ jupyter notebook
[I 21:11:57.443 NotebookApp] Writing notebook server cookie secret to /run/user/1000/jupyter/notebook_cookie_secret
[I 21:11:57.512 NotebookApp] JupyterLab alpha preview extension loaded from /home/usuario/anaconda3/lib/python3.6/site-packages/jupyterlab
JupyterLab v0.27.0
Known labextensions:
[I 21:11:57.527 NotebookApp] Running the core application with no additional extensions or settings
[I 21:11:57.530 NotebookApp] Serving notebooks from local directory: /home/usuario
[I 21:11:57.530 NotebookApp] 0 active kernels
[I 21:11:57.530 NotebookApp] The Jupyter Notebook is running at: http://localhost:8888/?token=4bdb9778cc99ac3971471acd92fa819b3dd23bd5b4d23bd3
[I 21:11:57.530 NotebookApp] Use Control-C to stop this server and shut down all kernels (twice to skip confirmation).
[C 21:11:57.531 NotebookApp]

    Copy/paste this URL into your browser when you connect for the first time,
    to login with a token:
        http://localhost:8888/?token=4bdb9778cc99ac3971471acd92fa819b3dd23bd5b4d23bd3
```

DBlandIT

UTN.BA
UNIVERSIDAD TECNOLÓGICA NACIONAL
FACULTAD REGIONAL BUENOS AIRES


DBlandIT


UTN.BA
UNIVERSIDAD TECNOLÓGICA NACIONAL
FACULTAD REGIONAL BUENOS AIRES