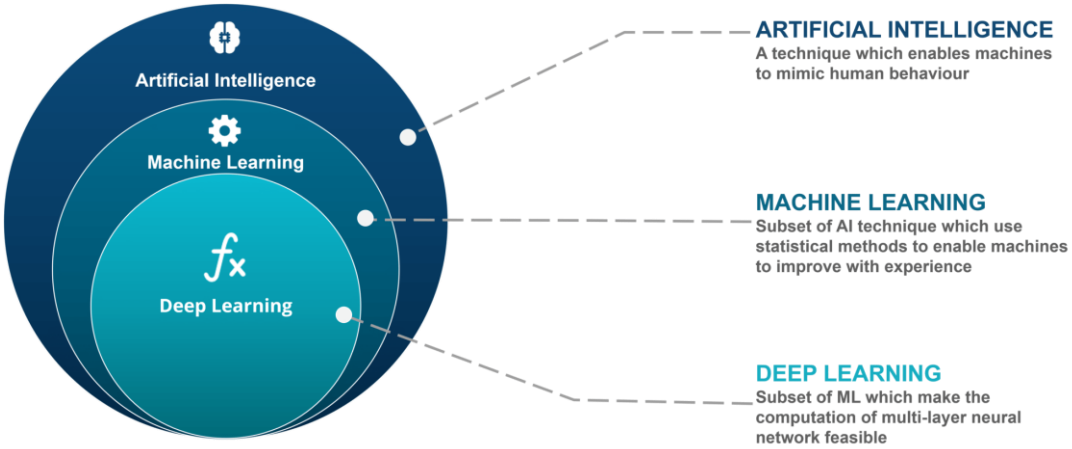
Introducción a *Machine Learning*

IE: Inteligencia Empresarial

Lo que veremos, “en dos palabras”

- Machine Learning



- Python

Objetivos

- Aprender a sacar partido de los datos con métodos estadísticos y de aprendizaje automático
- Aprender a utilizar herramientas del *stack* tecnológico para *Data Mining* en Python
- Aplicar dichas herramientas a diferentes *datasets*

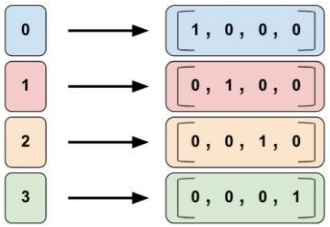
Algunos temas interesantes

Introducción a Python

Lectura de datasets

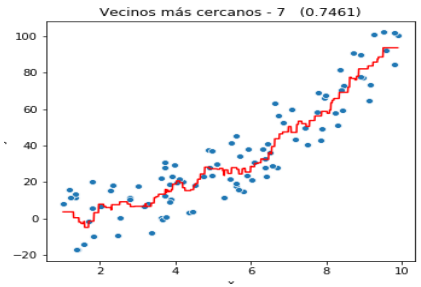
	A	B	C	D
1	datetime	season	holiday	workingday
2	2011-01-01 00:00:00	1	0	0
3	2011-01-01 01:00:00	1	0	0
4	2011-01-01 02:00:00	1	0	0
5	2011-01-01 03:00:00	1	0	0
6	2011-01-01 04:00:00	1	0	0
7	2011-01-01 05:00:00	1	0	0
8	2011-01-01 06:00:00	1	0	0

Preprocesamiento

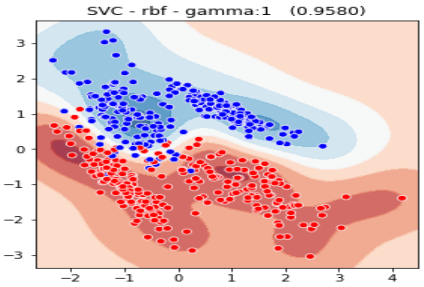


Visualización

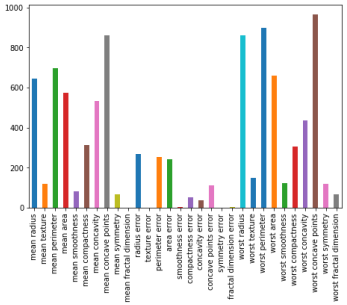
Regresión



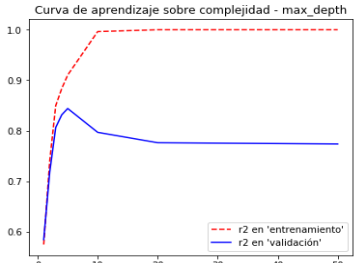
Clasificación



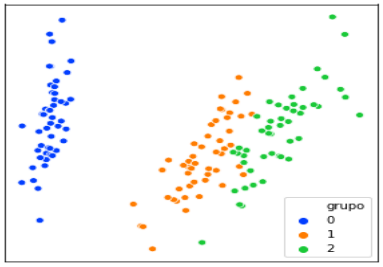
Selección de atributos



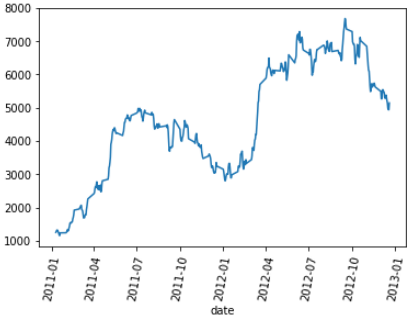
Ajuste de hiperparámetros



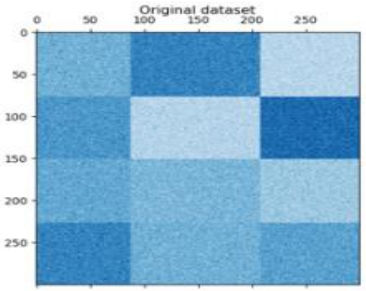
Clustering



Series temporales



Biclustering



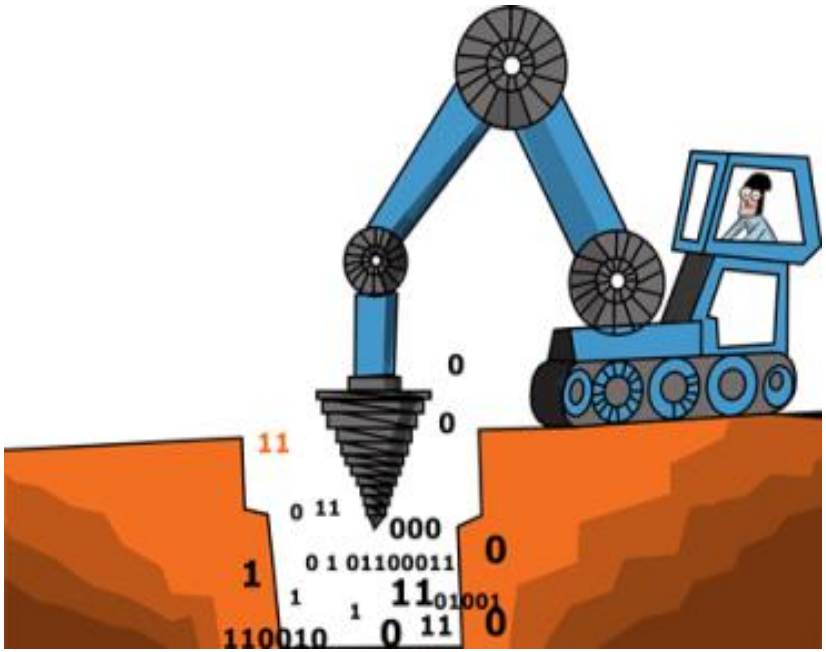
Automated ML

Metodología en clase

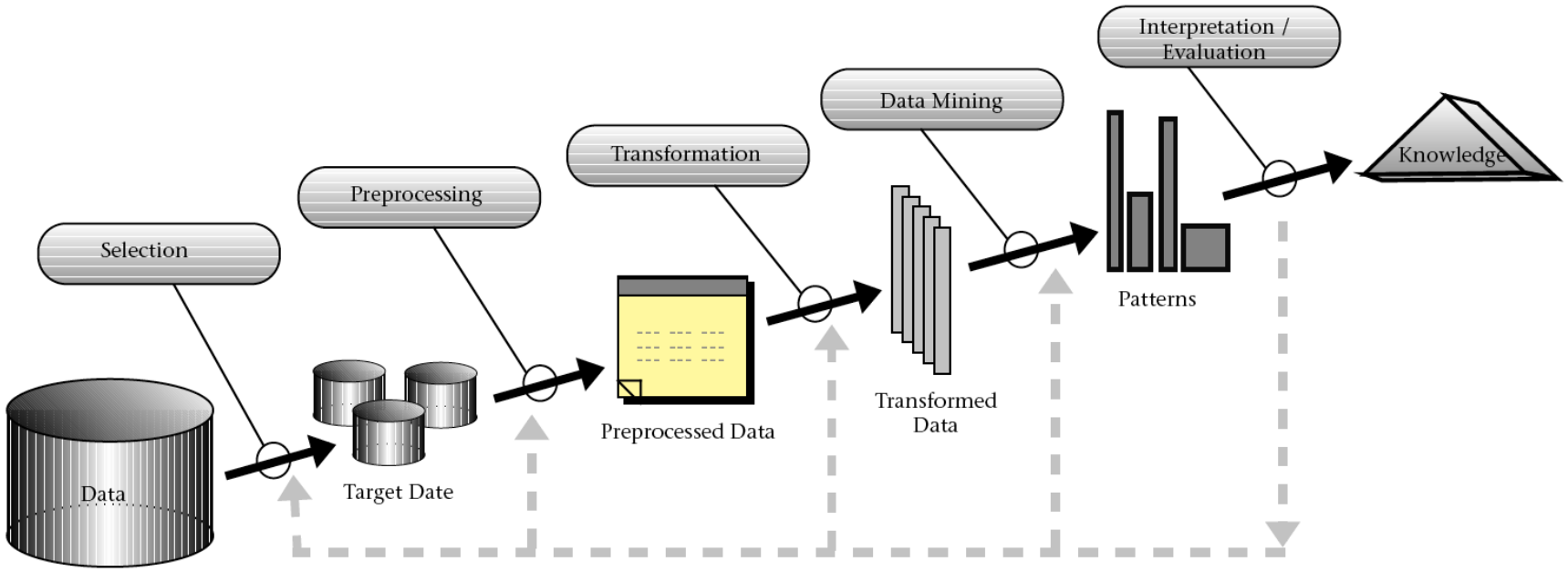
- Hay que instalar:
  - Python
  - Anaconda

- Organización de las sesiones:
  - Presentación de conceptos
  - Búsqueda de información
  - Implementación sobre Notebooks de Jupyter

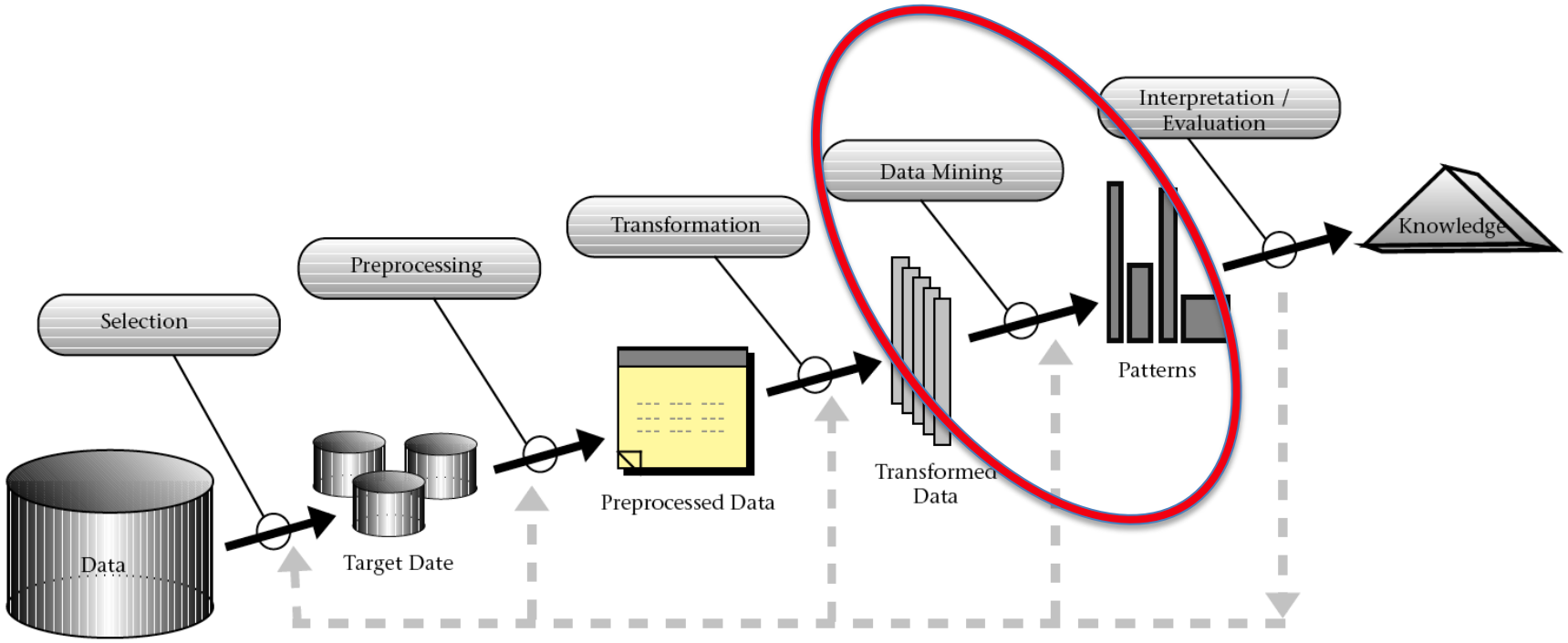
¿Qué es la minería de datos?



KDD Process: Knowledge Discovery in Databases



KDD Process: Knowledge Discovery in Databases



Un ejemplo: demanda de bicis de alquiler

Planteamiento:

- Una pregunta
- Información de entrada
- Datos disponibles

Capital Bikeshare
Washington D.C.

La pregunta

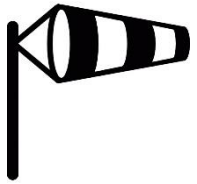
¿Cuál va a ser la demanda de bicicletas en un determinado momento?

- Posibles respuestas:
  - Numérica: 10, 203, 15, …
  - Categórica: tipo de demanda (muy baja, baja, media, alta, muy alta)

Información de entrada

Horas de mayor o menor actividad

Sol, nubes, humedad, …, influyen



El viento influye

Si llueve se cogen menos bicis

Los días de fútbol el patrón cambia

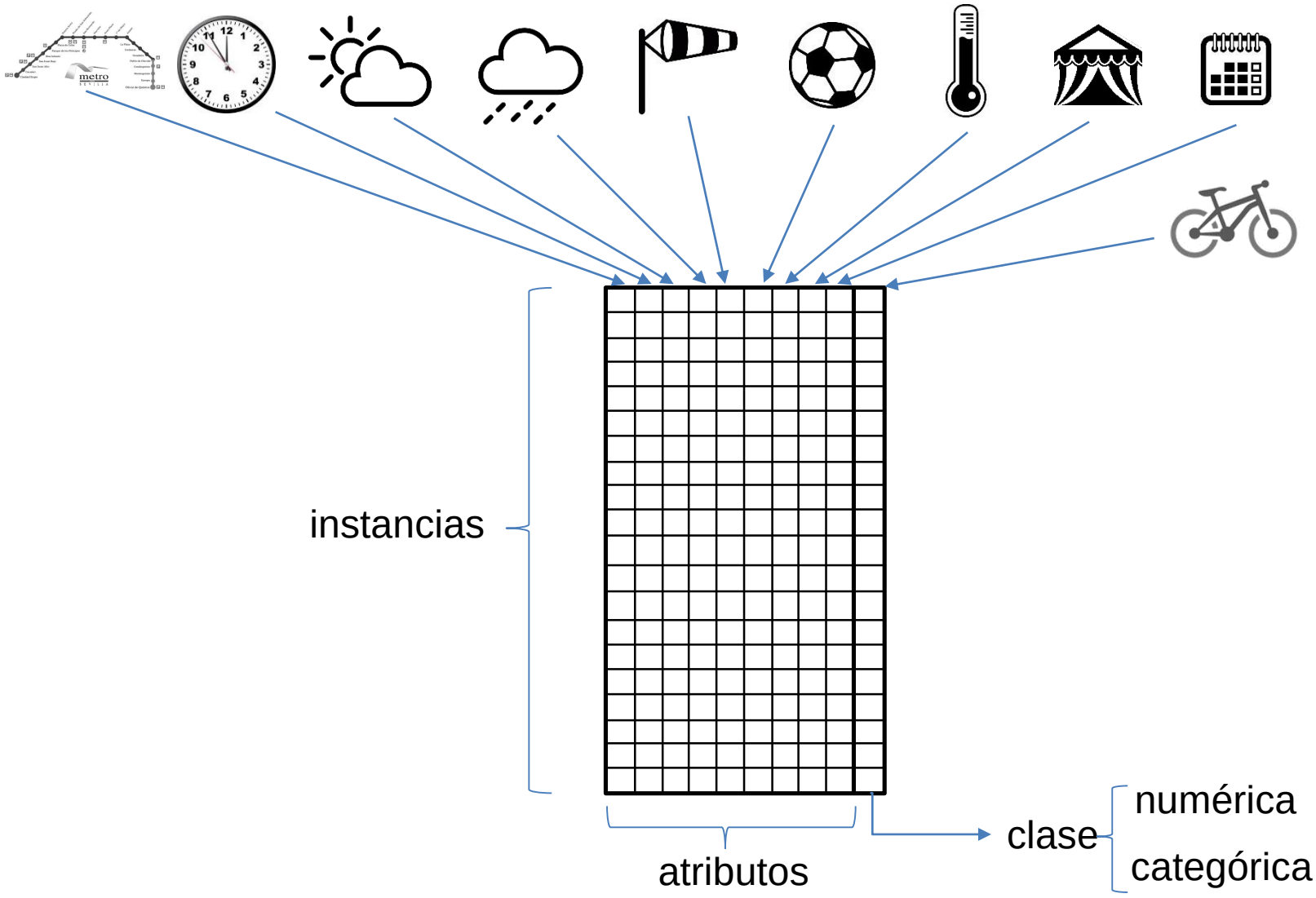
La temperatura influye

Los fines de semana el patrón cambia

Cada zona es distinta

En feria y en semana santa hay otro patrón

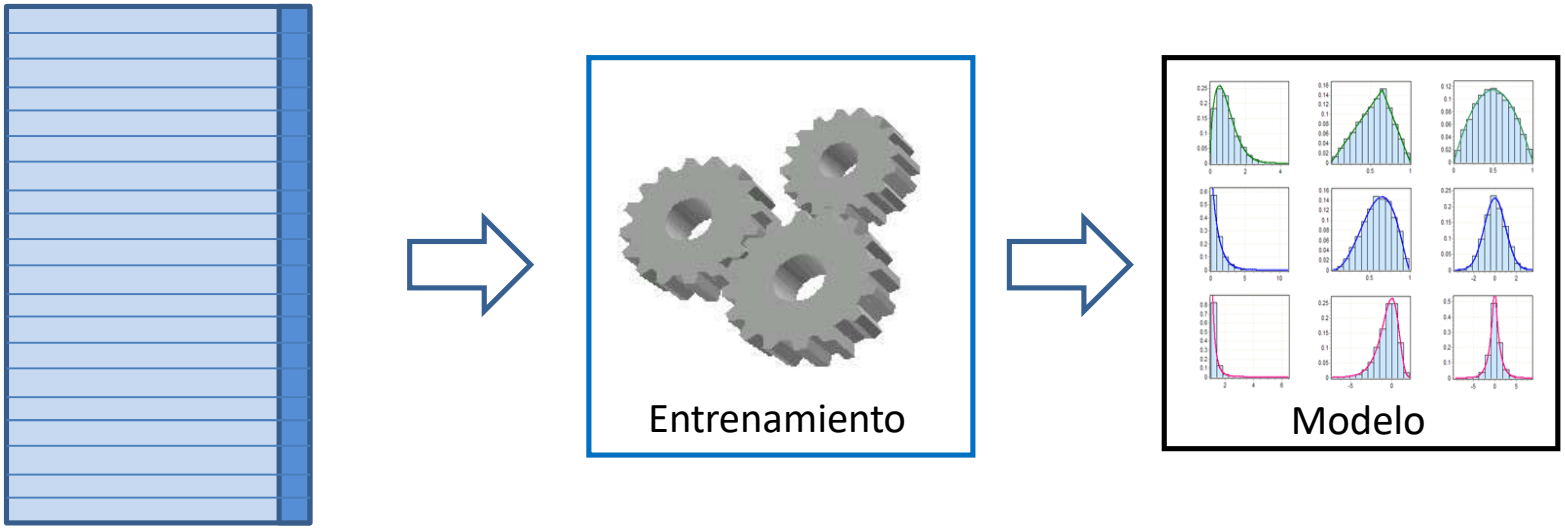
Datos disponibles



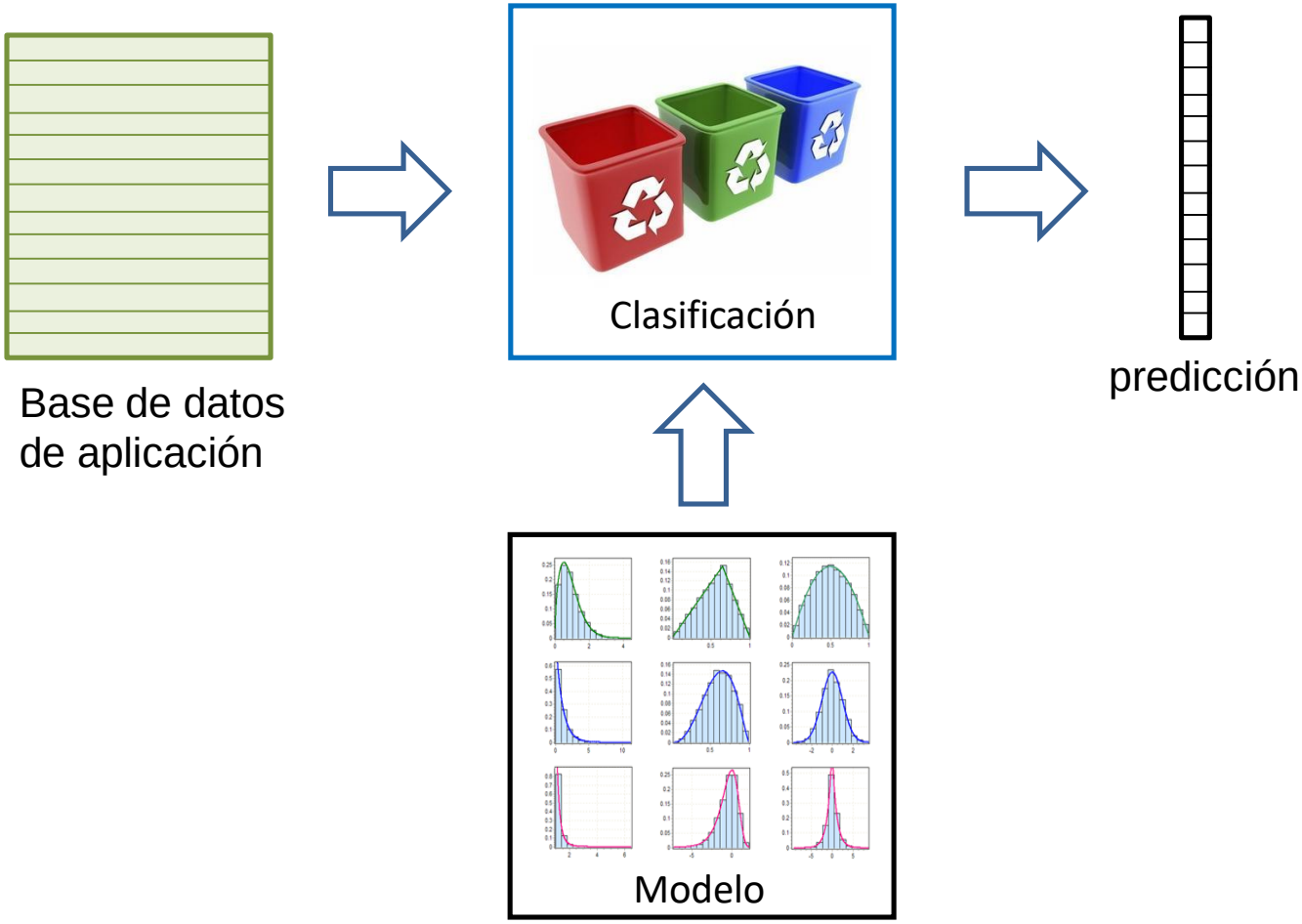
El proceso de “aprendizaje automático”

Fase de entrenamiento

Generan modelos que permiten realizar predicciones

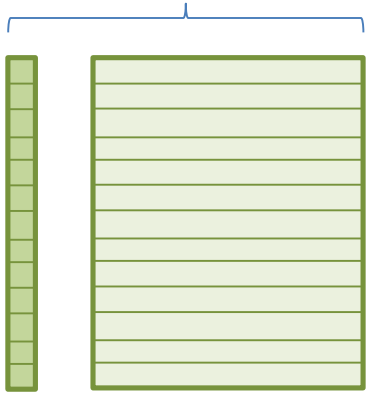


Uso del modelo

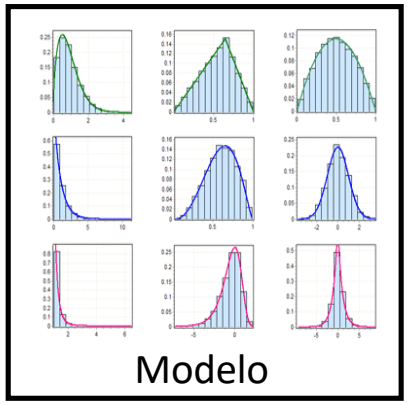


Evaluación

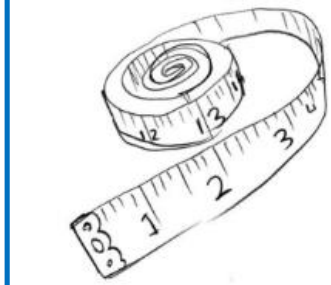
Base de datos de test



Clasificación



Modelo



Evaluación

predicción

También hay algunos modelos (como las reglas de asociación) que dan salidas “interpretables” para las personas.

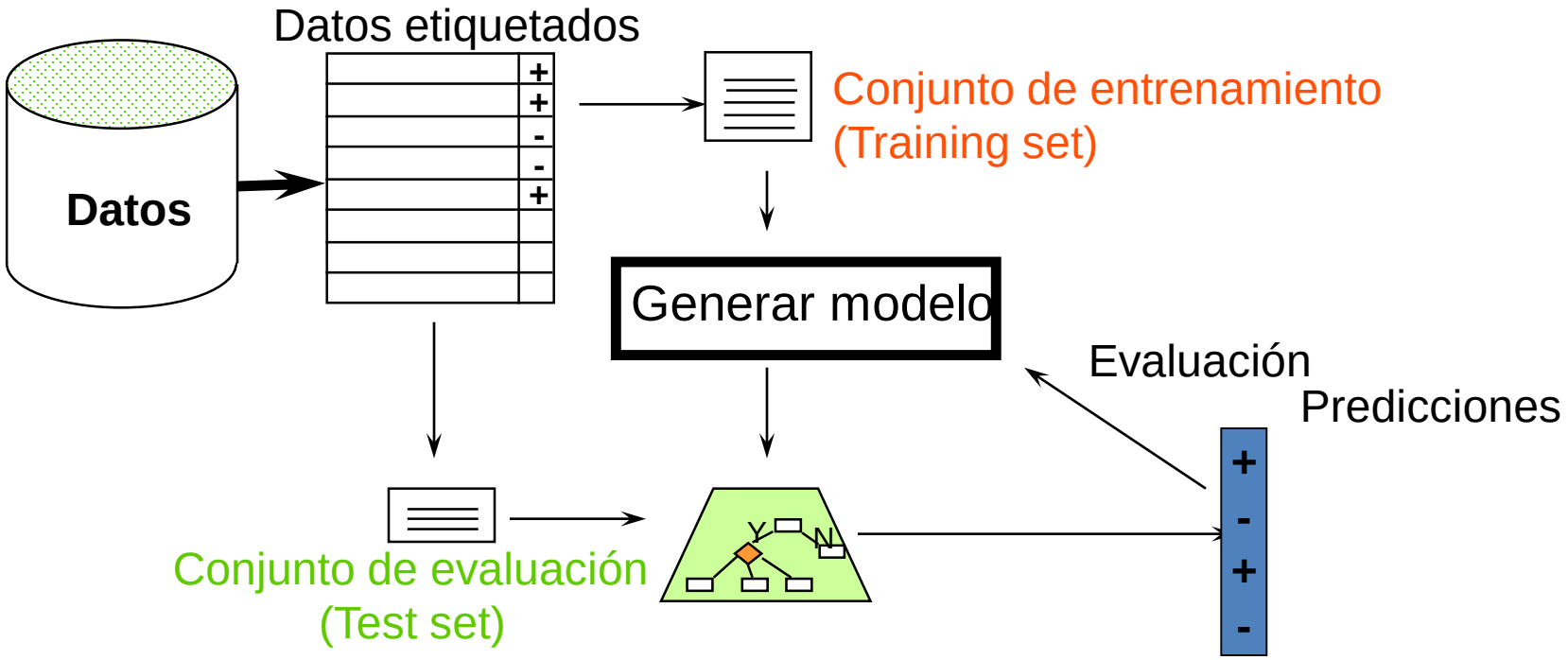
Técnicas de evaluación

- Training/Test sets

- Técnicas de remuestreo
  - K-fold Cross Validation
  - LOOCV
  - Bootstrapping

Training/Test set

- Separar el conjunto de datos en dos
  - Conjunto de entrenamiento (80%)
  - Conjunto de evaluación (20%)

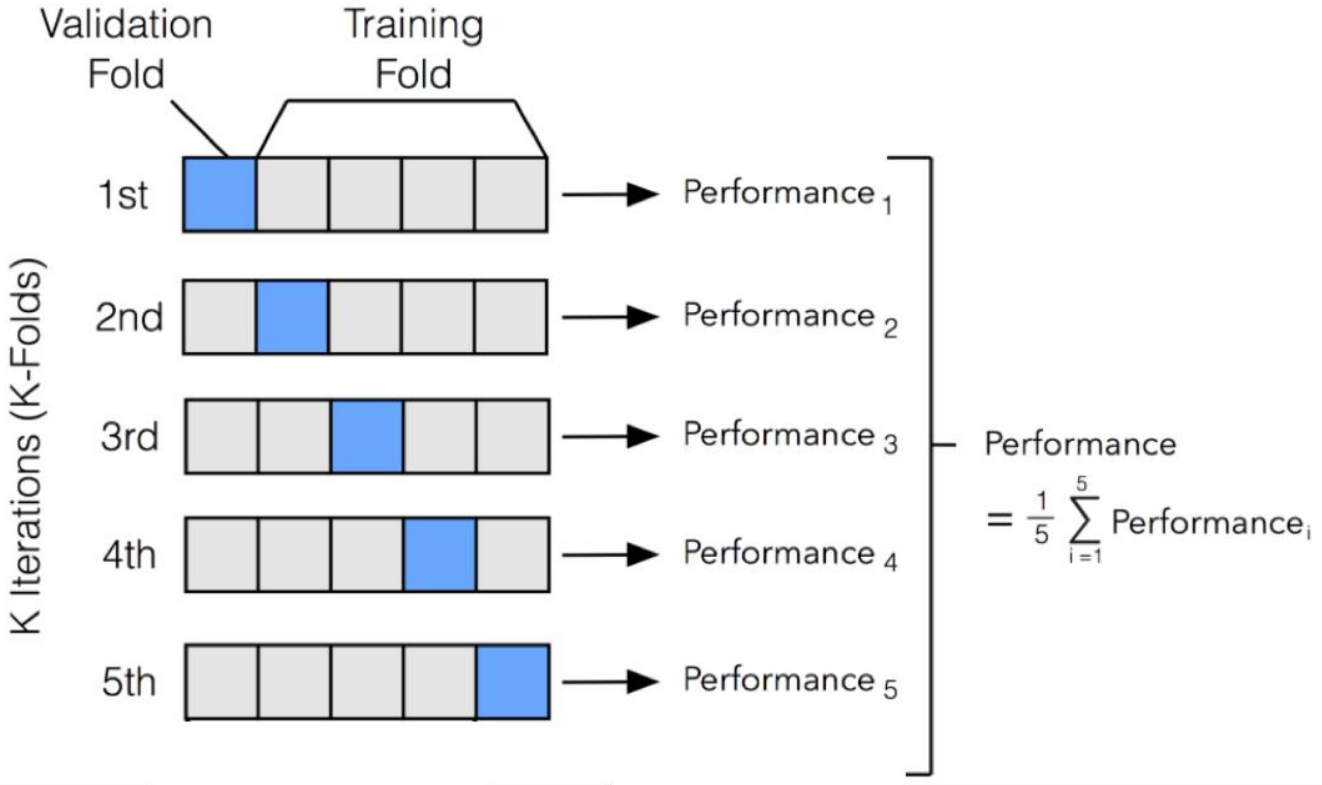


Training/Test sets

- ¿Cuál es el tamaño adecuado de los conjuntos?
  - A mayores conjuntos de entrenamiento y validación, mejor clasificador
  - A mayor conjunto de evaluación, mejor predicción del error
- ¿Qué ocurre cuando tenemos conjuntos de datos “pequeños” o no balanceados?

K-fold Cross Validation

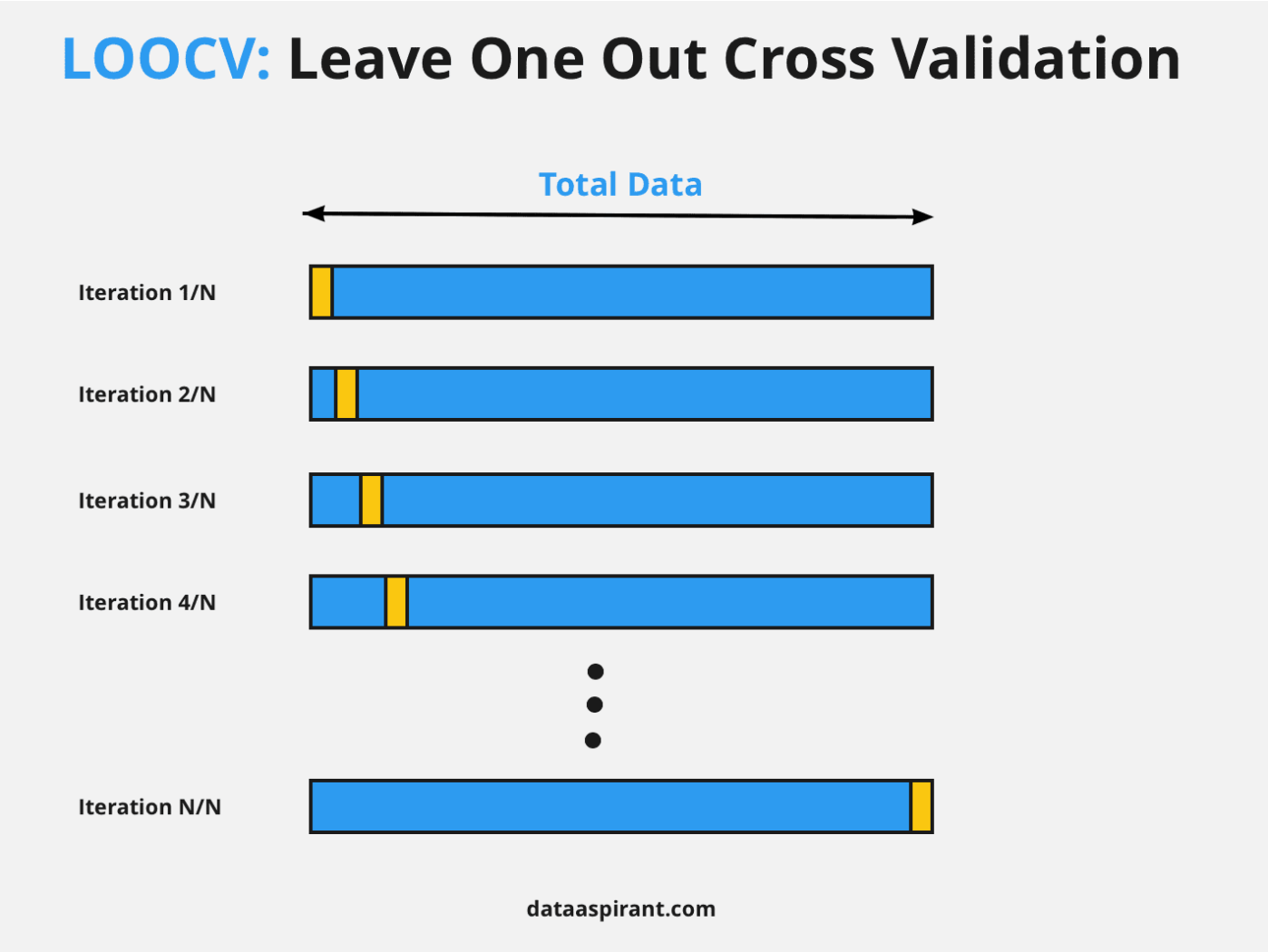
- Repetir el proceso anterior de entrenamiento y evaluación



K-fold Cross Validation

- K-fold Cross Validation:
  - Dividir el conjunto de datos en k subconjuntos de igual tamaño
  - Repetir k veces:
    - Seleccionar un subconjunto de Test
    - Utilizar la unión del resto de subconjuntos (K-1) para entrenamiento

Leave One Out Cross Validation



Leave One Out Cross Validation

- LOOCV: Una forma particular de CV
  - K = n (número de ejemplos o instancias)
  - En cada iteración (n)
    - 1 ejemplo de test
    - n-1 ejemplos forman el conjunto de training
- No implica muestreo aleatorio
- Muy costoso computacionalmente
- No estratificado
  - Solamente hay una instancia en el conjunto de test

Bootstrapping

- Muestreo con reemplazo

Bootstrapping

- CV utiliza muestreo sin reemplazo
  - Cada instancia solamente forma parte de un conjunto de training o test
- Bootstrapping utiliza muestreo con reemplazo para el conjunto de entrenamiento
  - Se realiza un muestreo con reemplazo de las n instancias originales n veces, para formar un conjunto nuevo de n instancias
    - Se utiliza dicho conjunto como conjunto de entrenamiento
  - El conjunto de test estará formado por aquellas instancias que no hayan sido incluidas en el conjunto de entrenamiento
    - OOB: instancias “out-of-bag”

Evaluación de un clasificador

- ¿Cómo de bueno es el modelo obtenido?

- Matriz de confusión

		PREDICCIÓN	
		CLASE POSITIVA	CLASE NEGATIVA
REALIDAD	CLASE POSITIVA	TP	FN
	CLASE NEGATIVA	FP	TN

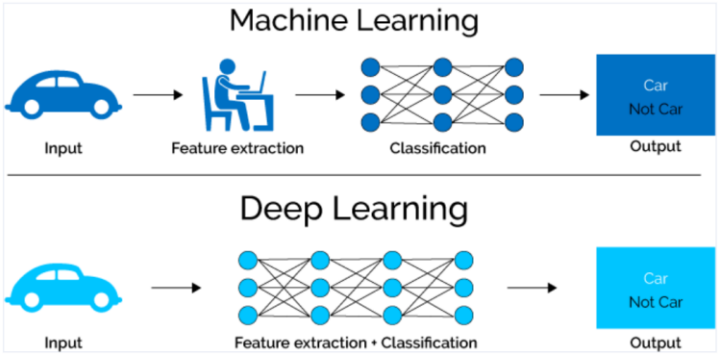
Medidas de evaluación

- Accuracy: (TP + TN) / N
- Pr(P|p) ≈ True Positive Rate: TPR = TP / (TP + FN). (“*recall*” o ”*sensitivity*” o *“positive accuracy”).*
- Pr(N|p) ≈ False Negative Rate: FNR = FN / (TP + FN). (“*positive error*”)
- Pr(N|n) ≈ True Negative Rate: TNR = TN / (TN + FP). (”*specificity*” o ”*negative accuracy*”).
- Pr(P|n) ≈ False Positive Rate: FPR = FP / (TN + FP). (“*negative error*”)
- Pr(p|P) ≈ Positive Predictive Value: PPV = TP / (TP + FP). (”*precision*”).
- Pr(n|N) ≈ Negative Predictive Value: NPV = TN / (TN + FN).
- Macro-average = MEDIA(TPR, TNR). (La media puede ser aritmética, geométrica u otra)
- BREAK-EVEN= (Precision + Recall) / 2 = (PPV + TPR) / 2
- F-MEASURE= (Precision * Recall) / BREAK-EVEN = 2*PPV*TPR / (PPV + TPR)

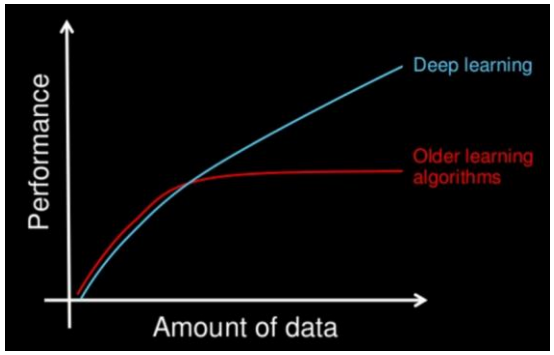
La clasificación es solo una parte

¿Y el *deep learning*?

- Sistemas (redes) compuestos por varias capas de procesadores **no lineales**
- Procesamiento de información no estructurada: ***machine perception***
- Modelos **pre-entrenados**, de forma no supervisada, con grandes volúmenes de datos
- **Ajuste fino** de estos modelos con entrenamiento supervisado de las capas finales



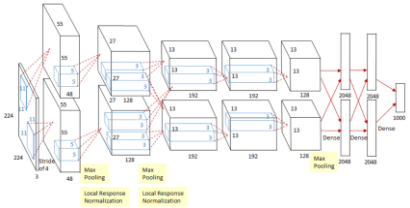
Fuente: Xenonstack



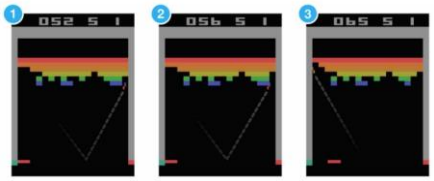
Más volumen = más instancias + más atributos

Una breve historia del *deep learning*

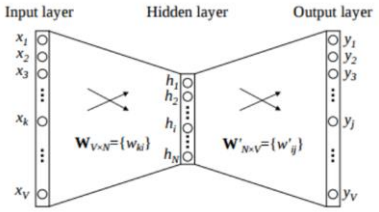
https://dennybritz.com/blog/deep-learning-most-important-ideas



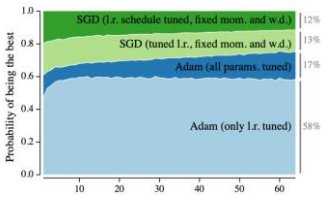
[2012] AlexNet salta la banca de ImageNet (CONV-2D)



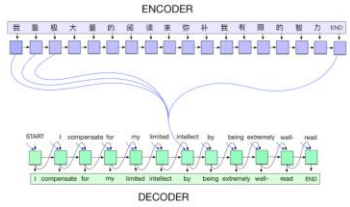
[2013] DeepMind y Deep Reinforcement Learning para jugar a Atari



[2013] Word2Vec y *word embeddings*

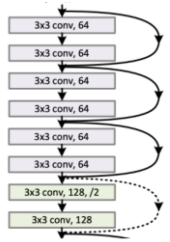


[2014] El método Adam de optimización aligera el ajuste de hiperparámetros



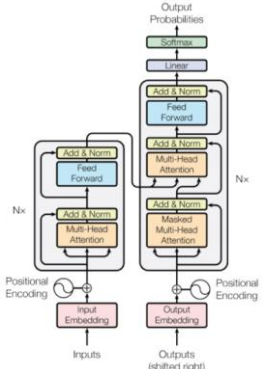
[2014] El mecanismo de atención abre la puerta a tareas PLN

[2015] Generative Adversarial Networks (GANs) para generar datos realistas

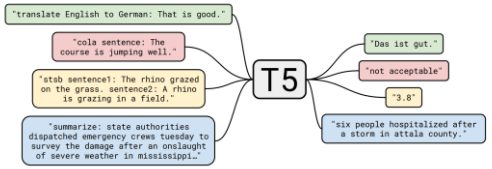


[2015] ResNet: atajos para obtener redes aún más profundas

[2017] DeepMind rompe otra barrera con AlphaGo



[2017] Transformers: auto-atención y paralelización

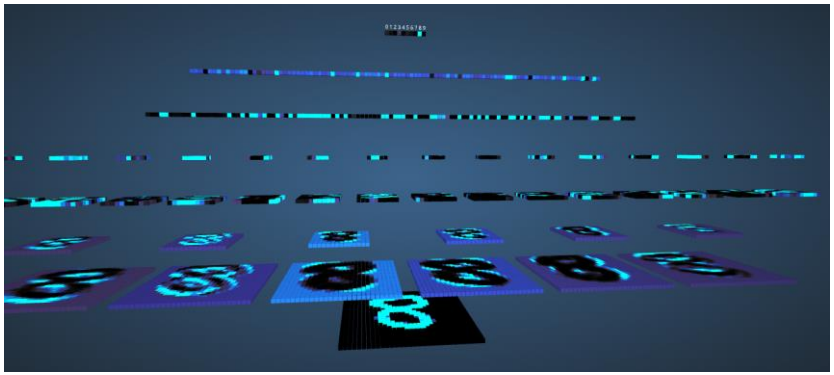


[2018-2020] Modelos text-to-text (BERT, GPT, T5)

Visualización de una CNN

- https://www.cs.ryerson.ca/~aharley/vis/conv/flat.html

- https://www.cs.ryerson.ca/~aharley/vis/conv/



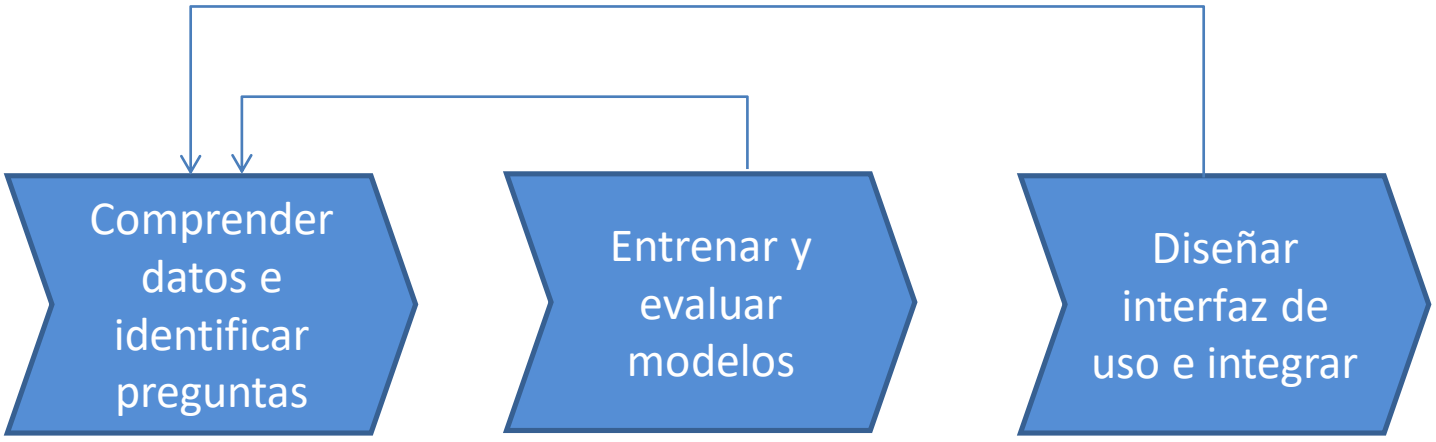
Aplicable a cualquier dominio

- Siempre que:
  - Haya datos
  - Se formulen bien las preguntas
- Por ejemplo:
  - Financiero
  - Energía
  - Salud
  - *Retail*
  - Telefonía
  - Transportes
  - …

Aprendiendo en dos direcciones

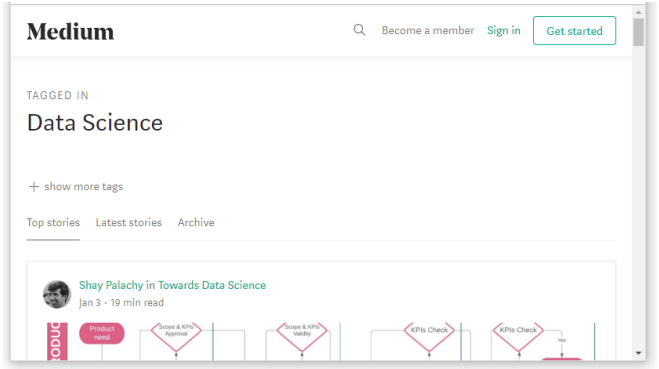
Experto en el dominio

Experto en la tecnología



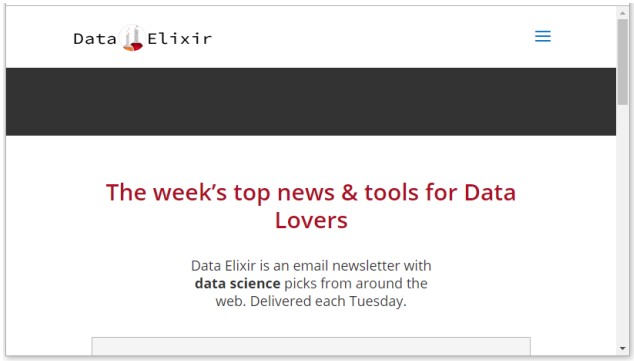
Recursos interesantes

Servicio de publicación de blogs



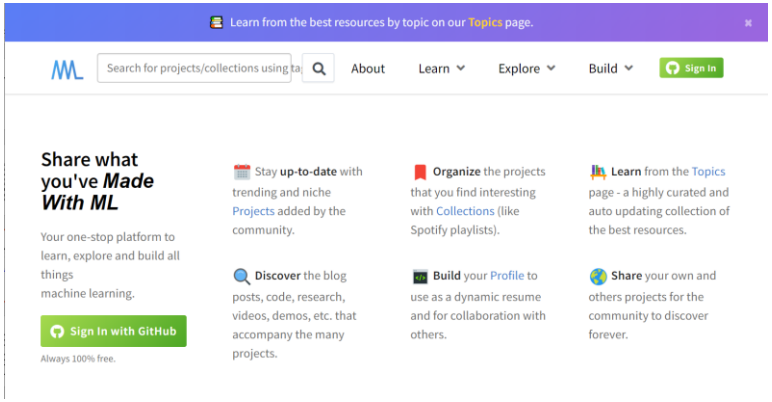
https://medium.com/tag/data-science

Publicación digital por correo electrónico
(acceso a archivo en su web)



https://dataelixir.com/

Recursos, proyectos y posts



https://madewithml.com/

Una buena hoja de ruta

https://towardsdatascience.com/if-i-had-to-start-learning-data-science-again-how-would-i-do-it-78a72b80fd93

Algunos cursos introductorios: https://elvissaravia.substack.com/p/course-recommendations-for-introductory

Tarea

- Instalar Anaconda para Python 3:
  - https://www.anaconda.com/distribution/#download-section