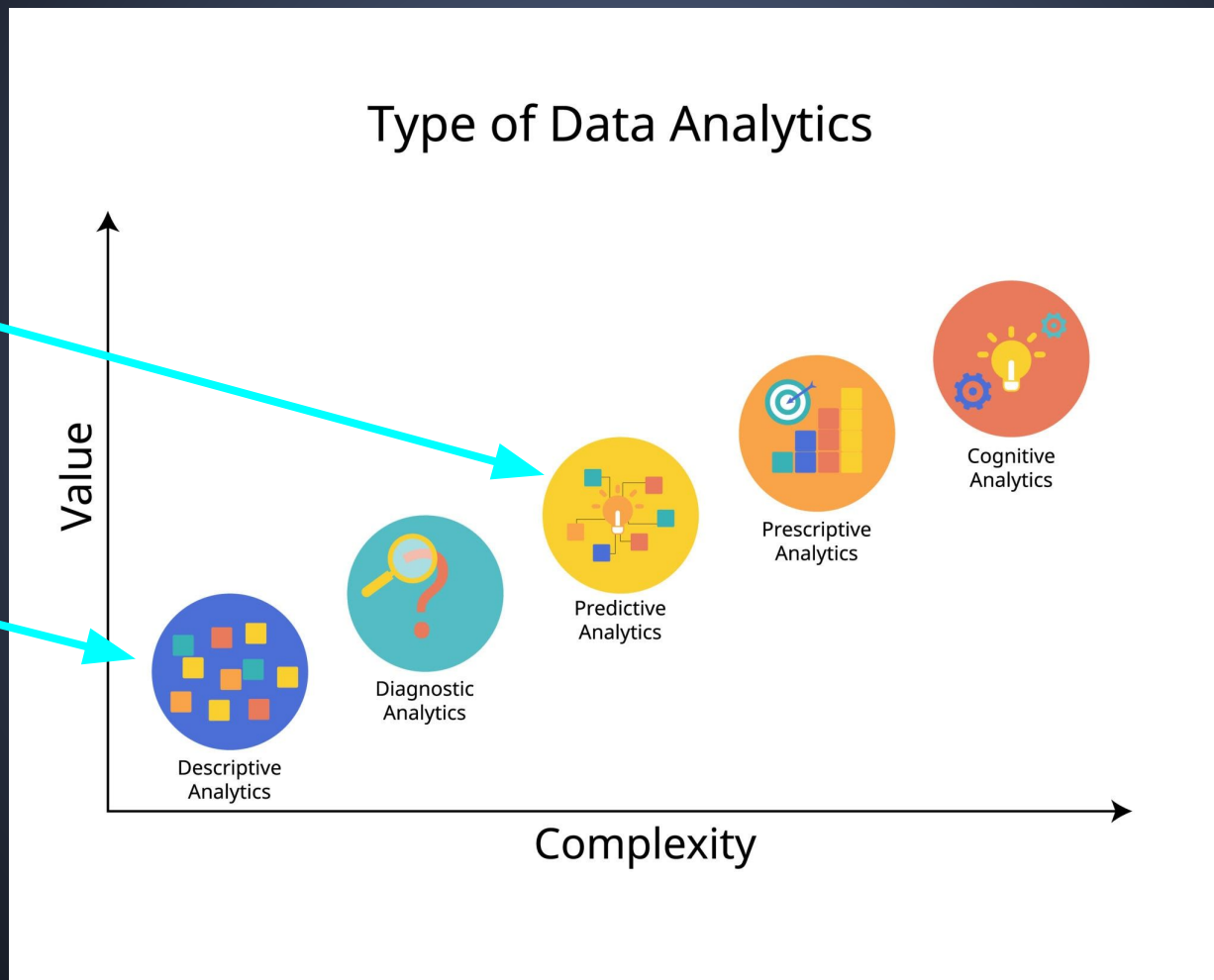




Aprendizaje Automático y Minería de Datos

Introducción
Datos y modelos

Mapa general del Análisis de Datos

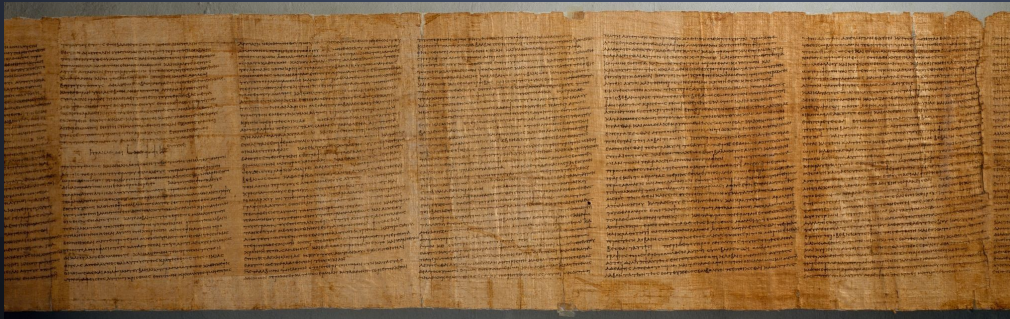
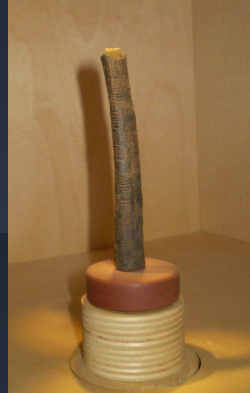


Análisis
de Datos
con MD
y con AA
=

Ciencia
de Datos

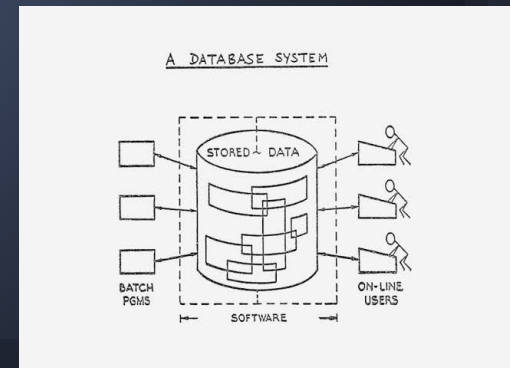
Historia

- La Humanidad usa las matemáticas para *contabilizar* desde al menos el 20 000 a.C.
- En Sumeria y Egipto (3000 a.C.) ya se usaban *tablillas de arcilla* para inventariar
- Los *estados* maduros como Grecia y Roma (V a.C.) tenían *censos*, *catastros* e *impuestos* en papiros bien estructurados
 - Era “*Estad-ística*” aunque no lo llamasen así



Historia

- Nace la Estadística (siglos **XVI** y **XIX**)
 - Primeros análisis demográficos, Teorema de **Bayes**, analítica espacial, etc.
 - Teoría de la Probabilidad (**Pascal y Fermat, 1654**), para repartir bien el dinero en *juegos de azar* inacabados
 - Tarjetas perforadas en EEUU (¡el germen de **IBM!**)
- La Informática irrumpe en el siglo **XX**
 - Fundamentos de la inferencia estadística (**30-50's**)
 - El Análisis de Datos se independiza de las Matemáticas... y nace el AA (**50-60's**)
 - Edgar F. Codd (**1970**) inventa el modelo relacional de las Bases de Datos
 - Llega la Minería de Datos...
y renace el AA, como ya vimos (**80-90's**)



La catadora de té



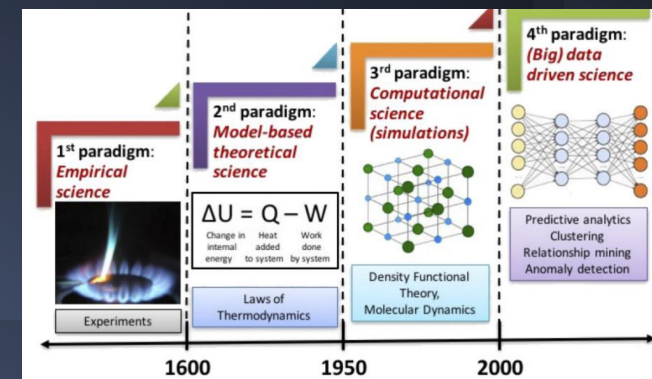
- En el **té con leche**, ¿es mejor echar primero la **leche** o el **té**?
- **Ronald Fisher (1935)** diseñó el *experimento modelo* de la Ciencia de Datos
 1. Plantea una **afirmación comprobable** (**Hipótesis alternativa** / H_a / H_n si hay varias): *“Muriel Bristol puede apreciar si se echa la leche primero y después el té” (Of course!)*
 2. Supón que esa afirmación es *falsa* (**Hipotesis nula**, H_0): *En realidad lo echa a suertes*
 3. Diseña un **experimento controlado y aleatorizado** para comprobarlo: *Ronald prepara 8 tazas, 4 con leche primero (y Muriel lo sabe) pero en orden aleatorio (ella no puede conocerlo)... deja que las pruebe todas ¡y a ver cuantas acierta!*
 4. Realiza el experimento y calcula bien las **probabilidades**: *Hay 70 combinaciones posibles de 4 tazas y sólo 1 es correcta. Acertar las 4 sería SORPRENDENTE (<2% de posibilidades echándolo a suertes, un **p-valor** -redondeando- de 0.02), de hecho para Ronald <5% ya es estadísticamente significativo (0.05 es su **a**, nivel de significación elegido). Acertar 3 estaría bien (<25%) pero NO sorprendería (queda por encima de **a**), y acertar 2 sería casi lo esperado (≈50%) a cara o cruz. Pues Muriel... ¡ACIERTA LAS 4!*
 5. Finalmente, si **p-valor** < **a**, **RECHAZA** la hipótesis **nula** y quédate con la **alternativa**; y en caso contrario, al revés: *Como 0.02 < 0.05, Ronald RECHAZA la idea de que Muriel lo esté echando a suertes... vaya, ¡quizá SÍ puede apreciar el buen té con leche!*

Ejemplo



- Típica **prueba A/B** de un **City Builder**
 1. Plantea una afirmación comprobable (Hipótesis alternativa / H_1 / H_a): **“Rebajar la curva de dificultad del tutorial aumenta la retención de los jugadores a Día 1”**
 2. Supón que esa afirmación es *falsa* (Hipótesis nula, H_0): **En realidad no afecta en nada**
 3. Diseña un experimento controlado y aleatorizado para comprobarlo: **El servidor del juego asigna automáticamente una mitad de los jugadores que se conectan al grupo de control (A), que tienen el tutorial normal, y la otra al grupo de tratamiento (B), que tienen el tutorial con la curva de dificultad rebajada**
 4. Realiza el experimento y calcula bien las probabilidades: **Con telemetría se marcan los jugadores que SÍ vuelven al juego tras el Día 1 y los que NO. Se calculan ambas proporciones para miles de jugadores y se comparan, viendo lo probable que es esa desviación entre todas las posibles en una distribución normal estándar, dando un p-valor -redondeando- de 0.001. PREVIAMENTE el equipo de análisis había decidido que <1% ya es estadísticamente significativo (0.01 es su α , nivel de significación elegido).**
 5. Finalmente, si $p\text{-valor} < \alpha$, RECHAZA la hipótesis nula y quédate con la alternativa; y en caso contrario, al revés: **Como $0.001 < 0.01$, el equipo de análisis rechaza la idea de que el tutorial no afecte en nada... vaya, ¡quizá SÍ aumenta la retención que sea fácil!**

Historia



- Ciencia de Datos, el paradigma del siglo XXI
 - En 2001 William S. Cleveland propone el concepto y Jim Gray habla del **cuarto paradigma investigador**
 - **Empírico, Teórico, Computacional e Intensivo en Datos**
 - BIG DATA** ○ A partir de **2005** se habla de **Datos Masivos**, cuando la escala requiere nuevas metodologías
 - Almacenamiento distribuido y procesamiento paralelo
 - Y desde **2012**, **Aprendizaje Profundo** y la actual **IA Generativa**

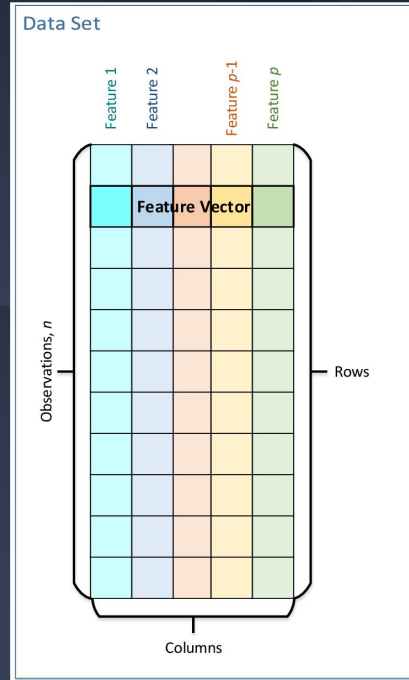


Datos

“Todos hemos conocido a ese diseñador que se niega siquiera a considerar los hallazgos de los datos, porque “Picasso no necesitó analíticas”. Pero el arte y los datos no son opuestos, y luchar con los ojos vendados es tan insensato como poético”

- David Helgason (Unity Technologies)

- Representaciones **cuantitativas** o **cualitativas** de atributos de entidades reales
 - **Numéricos** (discretos o continuos)
 - **Categoricos** (nominales u ordinales)
- Para trabajarlos se estructuran mediante **ingeniería** en **vectores** que ocupan un **espacio** multidimensional **de características**
 - Vienen en **conjuntos de datos** DATASET
 - No son perfectos, pueden tener **ruido** o excesivo **desequilibrio** o **variabilidad**
 - La unidad mínima es la **muestra**, que puede verse como **vector de características** o **fila** de datos de una tabla u hoja de cálculo



FEATURE ENGINEERING / VECTORS / SPACE

NOISE
IMBALANCE
VARIABILITY

SAMPLE

#	Age	Has Job	Own House	Credit Rating	Class
1	Young	false	false	fair	No
2	Young	false	false	good	No
3	Young	true	false	good	Yes
4	Young	true	true	fair	Yes
5	Young	false	false	fair	No
6	middle	false	false	fair	No
7	middle	false	false	good	No
8	middle	true	true	good	Yes
9	middle	false	true	excellent	Yes
10	middle	false	true	excellent	Yes
11	old	false	true	excellent	Yes
12	old	false	true	good	Yes
13	old	true	false	good	Yes
14	old	true	false	excellent	Yes
15	old	false	false	fair	No

Modelos



- **Simplificaciones** de fenómenos reales en base a supuestos **patrones** *subyacentes*
 - **Paramétricos**, inmutables y rápidos
 - **No paramétricos**, van *adaptándose* a los datos

“Todos los modelos son falsos, pero algunos son útiles”
- George E. P. Box (1976)

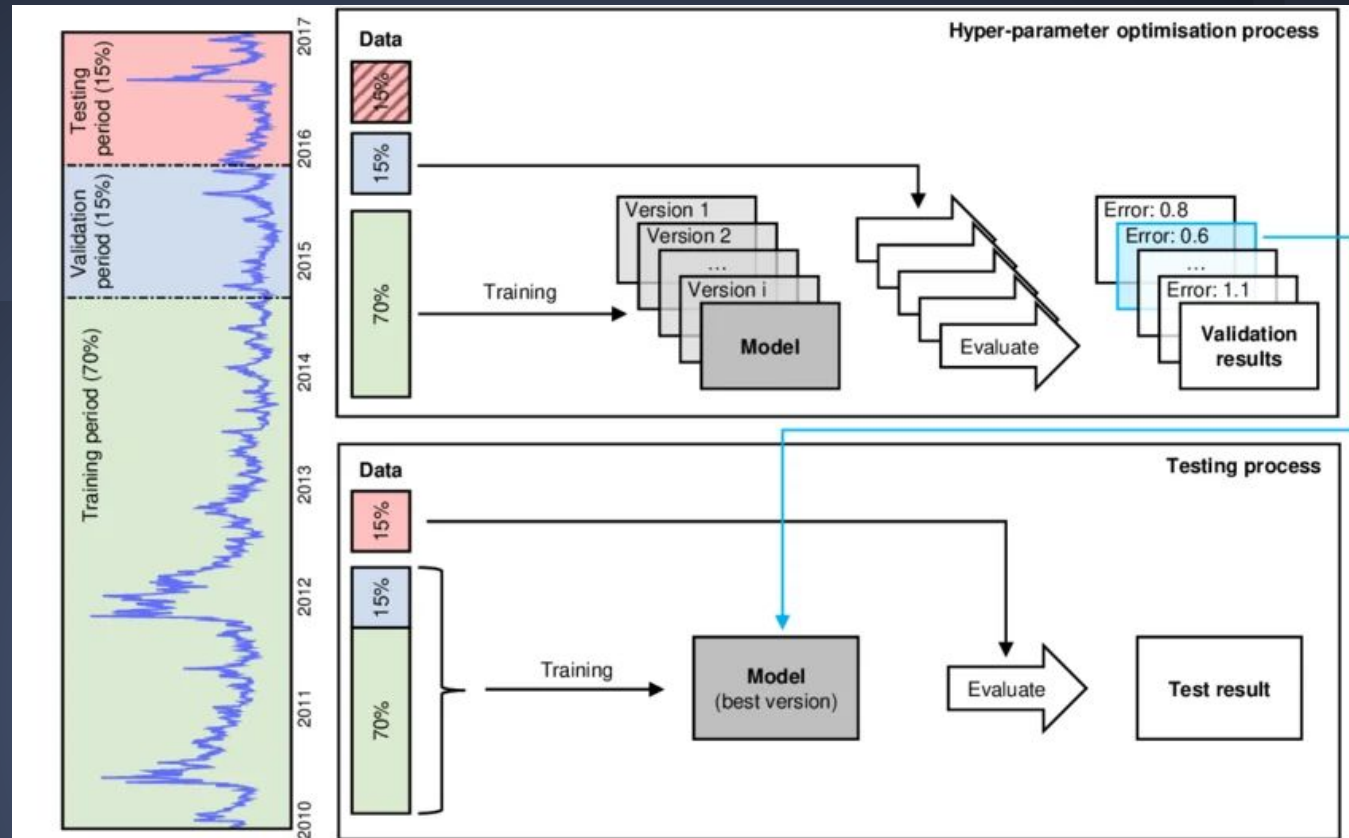
- Su calidad está acotada por la *calidad de los datos* usados para construirlos
- Hay distintos tipos, según sus objetivos
 - **Descriptivos**, describen esos patrones hallados en los datos
 - **Predictivos**, predicen datos futuros
 - **Prescriptivos**, prescriben acciones a realizar
 - **Generativos**, generan nuevos conjuntos de datos

Usos y particiones

- Los modelos no se desarrollan “de golpe” sino que se “cultivan” o *entrenan* hasta estar listos
- Para el AA lo ideal es hacer **3 particiones de datos**

- TRAINING ○ **Entrenamiento**, datos para entrenar, que consiste en ir ajustando los **parámetros del modelo**
- VALIDATION ○ **Validación**, datos para *validar el entrenamiento*, teniendo que ajustar los **parámetros del sistema** de AA (= **hiperparámetros**) y *reentrenar* el modelo si el resultado no es suficientemente bueno
- TEST ○ **Prueba**, datos que únicamente se usan al final, para medir la calidad del sistema de AA ya terminado
- DATA LEAKAGE ■ *Reentrenado* con los datos de entrenamiento + validación
 - ¡Tenerlos en cuenta es **fuga de datos**, voluntaria o no!

Ejemplo



- El **error** puede medirse al final de cada etapa
 - **Entrenamiento**, ¿qué tal funciona el modelo para los propios datos de partida?
 - **Validación**, ¿qué tal funciona el proceso para crear un modelo general?
 - **Prueba**, ¿qué tal funciona el sistema resultante?

Participación

- ¿Qué tipo de modelo se diseña para acertar en **La Quiniela** (resultados futbolísticos)?
 - A. Descriptivo
 - B. Predictivo
 - C. Prescriptivo
 - D. Deportivo
 - E. Generativo



Recursos y herramientas



Hugging Face

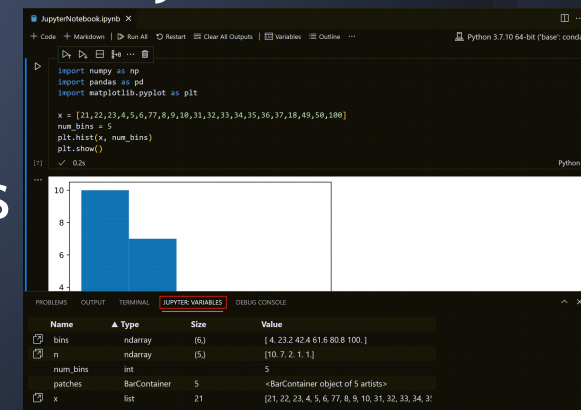
- **Fuentes de datos** (en formato **CSV**, **JSON** o **Parquet**) y herramientas **de extracción** (aparte de recurrir a **APIs REST** y **Webhooks**)
 - **Kaggle**, la plataforma de referencia para competir
 - **Hugging Face Datasets**, fundamental para texto, audio, imágenes, etc.
 - **UCI Machine Learning Repository**, conjuntos de datos limpios ideales para hacer prácticas:
 - **Zenodo** y **Dataverse**, datos científicos en abierto
 - **BeautifulSoup** y **Request**, para parsear HTML estático
 - **Playwright** o **Selenium** para simular navegación
 - **Scrapy**, rastrea la Web a gran escala

BeautifulSoup

Recursos y herramientas



- **Entornos de ejecución y lenguajes de programación** (ideales para MD y AA)
 - **Python** y su ecosistema, sin duda
 - Lenguaje **interpretado de alto nivel** y **tipado dinámico**
 - Con bibliotecas (muchas en **C/C++**) como **numpy**, **pandas**, **matplotlib**, **seaborn**, **plotly**, **scikit-learn**, **xgboost**, **pytorch**...
 - **Conda** o **Mamba** para gestionar entornos y paquetes, aislando dependencias
 - **Visual Studio Code** o **Cursor**, IDEs que admiten cuadernos interactivos **IPYNB** y asistentes de IA
 - **JupyterLab**, editor más ligero
 - **Google Colab** funciona en la nube



Ejemplo

```
# Un simple programa de Python
import random as rnd # Importación de módulos/librerías con alias

# Definición de variables
# En Python no se declara el tipo de dato (tipado dinámico e inferido)
USUARIO = "Ana" # Tipo str (cadena de texto)
OPCIONES = ["Hola", "¿Qué tal?", "Adiós"] # Tipo list (lista de elementos)

# Las funciones se definen con la palabra clave 'def'
# Los parámetros tampoco llevan tipo explícito y se usa ':' para iniciar el bloque
def saludar(nombre):
    # ¡LA INDENTACIÓN ES OBLIGATORIA! (4 espacios)
    # Reemplaza a las llaves {} de otros lenguajes para delimitar el bloque
    saludo = rnd.choice(OPCIONES)

    # Las f-strings permiten incrustar variables dentro de cadenas usando {}
    print(f"{saludo}, {nombre}.")

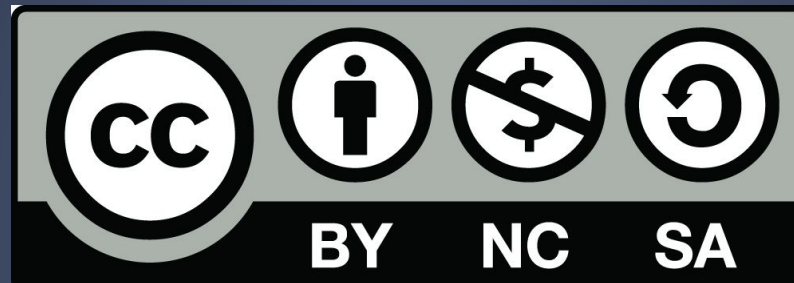
# Punto de entrada principal (comprueba si el script se ejecuta directamente)
if __name__ == "__main__":
    # El bucle 'for' itera directamente sobre un rango o una secuencia
    for i in range(3):
        # Llamada a la función pasándole la variable como argumento
        saludar(USUARIO)
```

Recursos y herramientas

- **Despliegue y mantenimiento en producción** (DevOps... en el caso del AA, **MLOps**)
 - Amazon S3 o Google Cloud Storage para los datos
 - Amazon Redshift, Google BigQuery o Snowflake para ejecutar SQL sobre datos masivos
 - MLflow compara modelos
 - Airflow automatiza reentrenos
 - FastAPI o ONNX para convertir tu modelo en aplicación real
 - ...



Críticas, dudas, sugerencias...



* Excepto el contenido multimedia de terceros autores

Federico Peinado (2026)

www.federicopeinado.es

