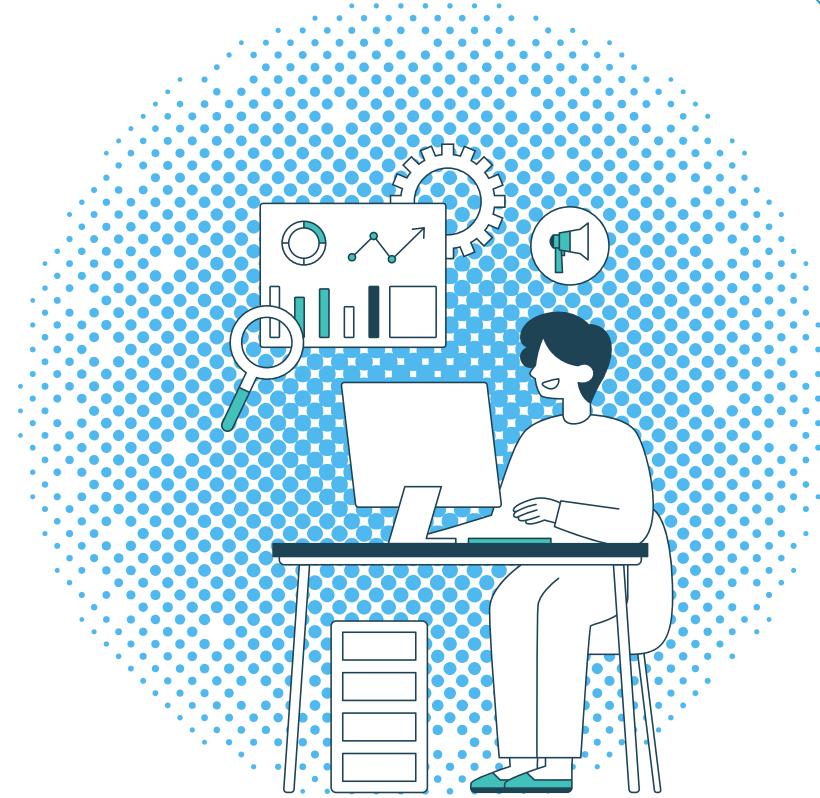


# Análisis de datos

Carrera de Especialización en  
Inteligencia Artificial



# Docentes



Esp. Lic. María Carina Roldán  
[macroldan@fi.uba.ar](mailto:macroldan@fi.uba.ar)

Esp. Ing. Ariadna Garmendia  
[arigarmendia@gmail.com](mailto:arigarmendia@gmail.com)

# Programa de la materia

## 1 Introducción al análisis de datos

Flujo de trabajo típico. Aplicaciones. Configuración del entorno. Introducción a las principales bibliotecas.

## 2 Análisis exploratorio de datos

EDA, estadística descriptiva y visualización.

## 3 Preprocesamiento de datos y Feature Engineering

Tipos de variables. Codificación. Manejo de datos faltantes. Creación de nuevos features.

## 4 Reducción de dimensionalidad

Métodos para reducir la dimensionalidad y técnicas para selección de features.

## 5 Pruebas estadísticas y validación de hipótesis

Pruebas de normalidad y de correlación. ANOVA y test de dependencia.

## 6 Taller práctico

Análisis de datos completo de un dataset. Buenas prácticas. Discusión grupal.

## 7 Presentación de trabajos finales

Exposición por grupos y devolución de los docentes.

## 8 Automatización del análisis de datos

Herramientas para automatizar el EDA y otras tareas del flujo de trabajo.

# Trabajo final integrador



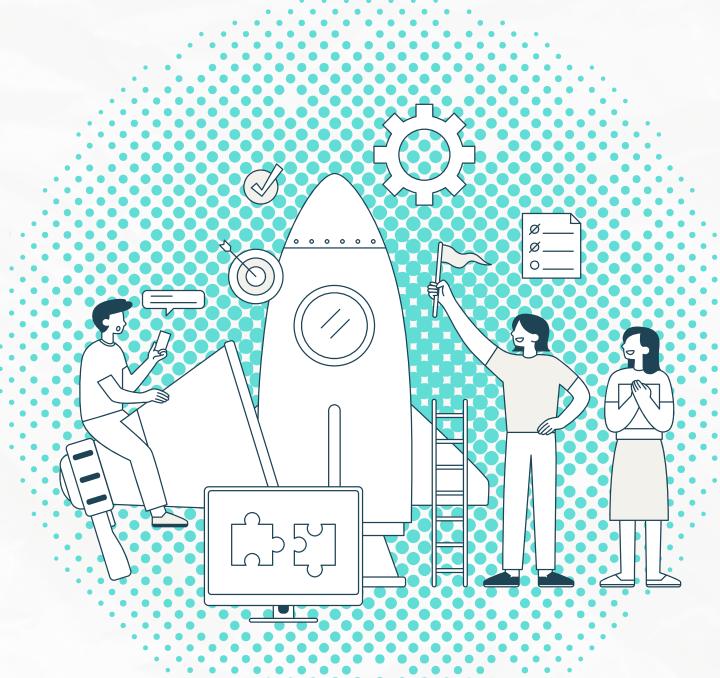
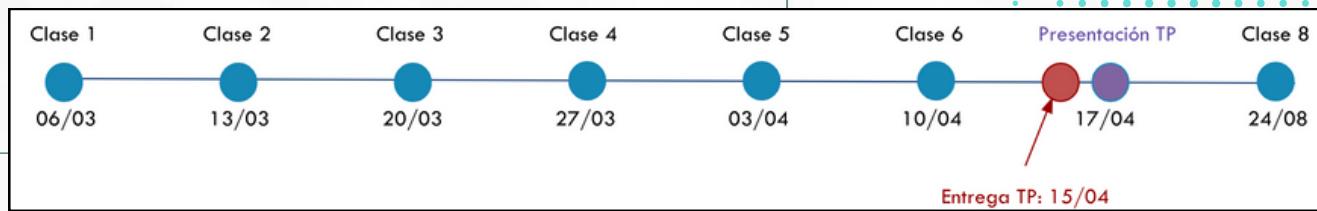
- Es grupal, con 2 alumnos por grupo (3 como excepción).
- Consiste en el análisis de un dataset. Se debe desarrollar una notebook y armar una presentación.
- El trabajo se entrega 2 días antes de la clase 7.
- Cada grupo expone su trabajo en la clase 7.
- Se recomienda hacer avances semanales, incorporando los conceptos vistos en cada clase.
- Las indicaciones del trabajo se encuentran aquí.

# Datos útiles

1 Material de clase: [Campus postgrado](#)

2 Notebooks: [Repositorio de la materia](#)

3 Cronograma:



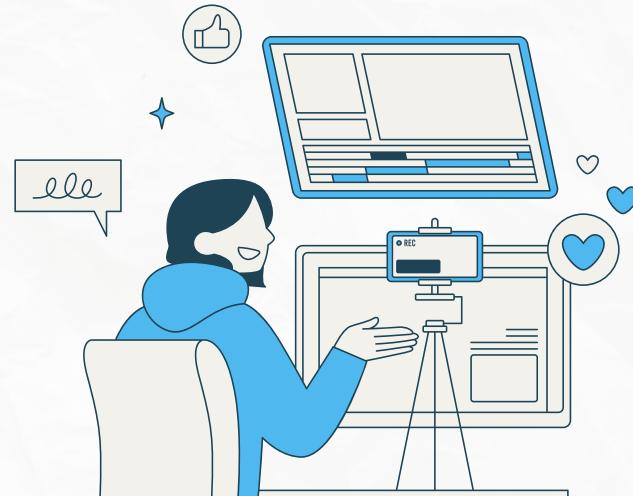
# ¿Qué es el análisis de datos?



Es el proceso de explorar, limpiar y modelar datos para extraer información útil y tomar decisiones.



Es clave para garantizar que los modelos de IA trabajen con datos confiables y adecuados para el problema.





# Tipos de análisis de datos

- 1** Descriptivo: mediante herramientas estadísticas, ayuda a explicar qué ocurrió.
- 2** Diagnóstico: determina por qué ha ocurrido algo.
- 3** Predictivo: predice resultados futuros basándose en estadísticas, modelos y aprendizaje automático.
- 4** Prescriptivo: identifica la mejor recomendación posible a seguir.



**¿Qué casos de uso de análisis de datos conocen?**

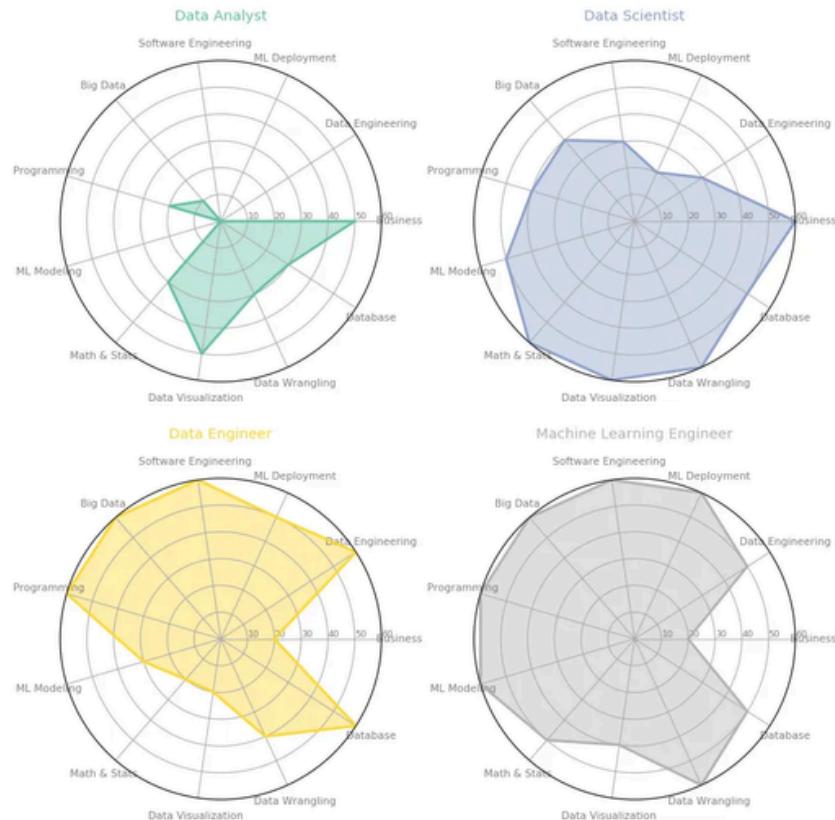
# Aplicaciones



¿Cuál es la diferencia entre resolver problemas mediante análisis de datos y hacerlo con ML?

- 1 Detección de fraudes: análisis de transacciones y detección de anomalías.
- 2 Salud: identificar tendencias en enfermedades y mejorar tratamientos.
- 3 E-commerce y retail: analizar hábitos de compra para recomendar productos.
- 4 Manufactura: identificar productos defectuosos en una línea de producción.
- 5 Redes sociales: entender opiniones de los usuarios y evaluar sus interacciones con el contenido.

# Análisis de datos vs. ML



DATAKADEMAY



## ¿Quiénes trabajan en IA?

Quienes son las personas que trabajan en este campo, sus roles y un poco de sus personalidades.

Medium / Jan 10, 2024

# Flujo de trabajo del análisis de datos

- 1 Identificar el problema



- 2 Recopilar los datos.



- 4 Analizarlos  
(estadística y visualización)



- 3 Limpiar y transformar los datos



- 5 Interpretar los resultados

# 1. Identificar el problema

1

Definir el objetivo a alcanzar.

2

Caracterizar la situación actual.

3

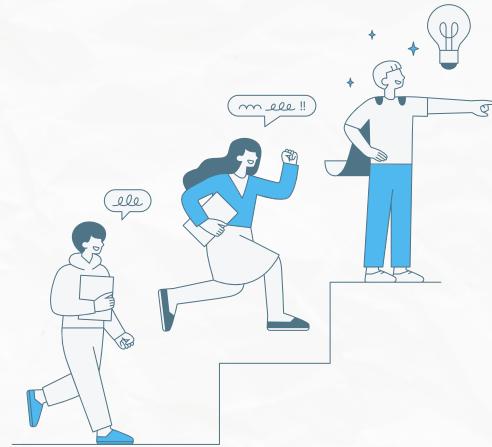
Formular la pregunta o hipótesis que se desea probar con el análisis.

4

Definir el alcance del problema

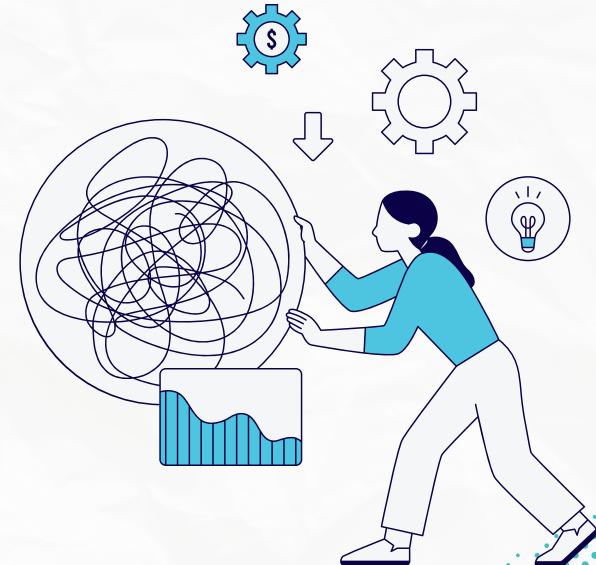
5

Establecer los entregables



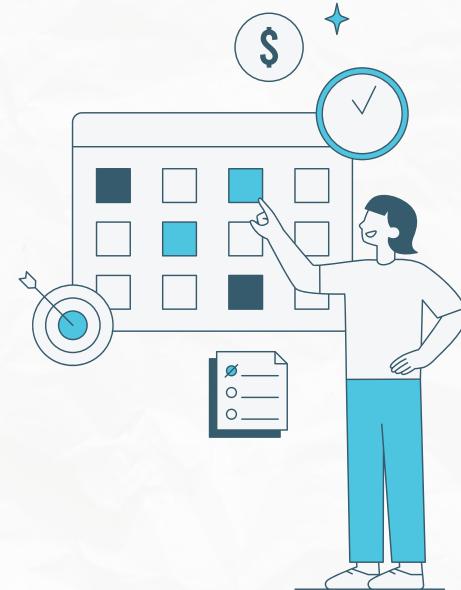
## 2. Recopilar los datos

- 1** Definir una estrategia para recolectar y combinar los datos:  
¿Qué datos se necesitan?
  
- 2** Identificar fuentes internas (por ejemplo del software de gestión de la empresa) o externas (de otras compañías, servicios de consultoría, de entidades gubernamentales, etc.)
  
- 3** Gestionar desafíos comunes: inconsistencia, errores tipográficos, falta de estandarización y valores faltantes, entre otros.



# 3. Limpiar y transformar los datos

- 1 Remover información duplicada o irrelevante. Corregir errores tipográficos e inconsistencias.
- 2 Implementar una estrategia para el manejo de datos faltantes.
- 3 Cambiar la estructura o el formato.
- 4 Remover o enmascarar PII (Personal Identifiable Information) y generalizar atributos sensibles.



## 4. Analizar los datos

- 1 Aplicar herramientas estadísticas descriptivas (Ej: media, mediana) y pruebas inferenciales (t-tests, ANOVA) para descubrir patrones, tendencias y relaciones.
- 2 Crear gráficos (de barras, scatterplots) por medio de bibliotecas como Matplotlib o Seaborn para explorar los datos y comunicar resultados efectivamente.
- 3 Desafíos: manejar outliers engañosos, interpretar estadísticas complejas correctamente y evitar tergiversar los datos al visualizarlos.



# 5. Interpretar los resultados



1 Traducir los resultados estadísticos y visualizaciones en respuestas accionables, conectándolos al problema original o hipótesis.

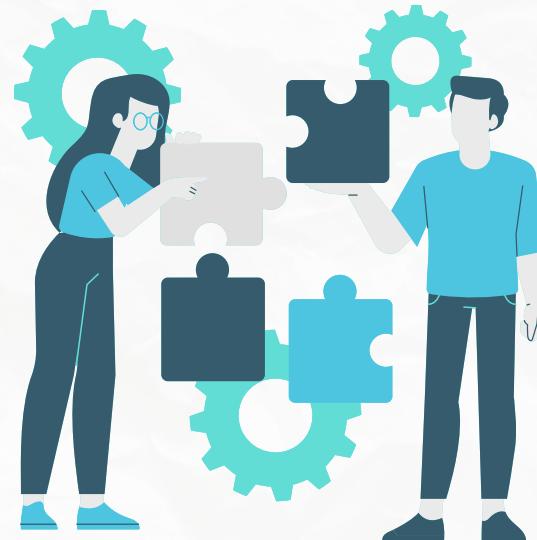
2 Evaluar el significado práctico y estadístico de los resultados (Ej: intervalo de confianza) para determinar su confiabilidad e importancia para el problema en cuestión.

3 Presentar los resultados a los interesados. Ajustar el mensaje a la audiencia.

4 Evitar generalizar, y considerar posibles sesgos y limitaciones en los datos o el análisis. Gestionar interpretaciones conflictivas o incertidumbre.

# Análisis de datos e IA responsable

- 1 Bias en IA: se refiere a sistemas de IA/ML que reproducen y continúan sesgos humanos.
- 2 Tiene consecuencias negativas en la equidad y la responsabilidad social, la credibilidad y responsabilidad legal de las empresas e impacto al negocio y la satisfacción del cliente.
- 3 Los datos son la fuente principal de sesgo en los sistemas de IA.
- 4 Afortunadamente, existen diversas estrategias para mitigar este problema.



# **¿Cómo se relaciona el flujo de trabajo con el TP final?**



# Introducción a bibliotecas relevantes



Recolección de datos

**BeautifulSoup**

**SQLAlchemy**

EDA, preprocesamiento, limpieza



Análisis estadístico, reducción de dimensionalidad



Visualización y análisis

**matplotlib**

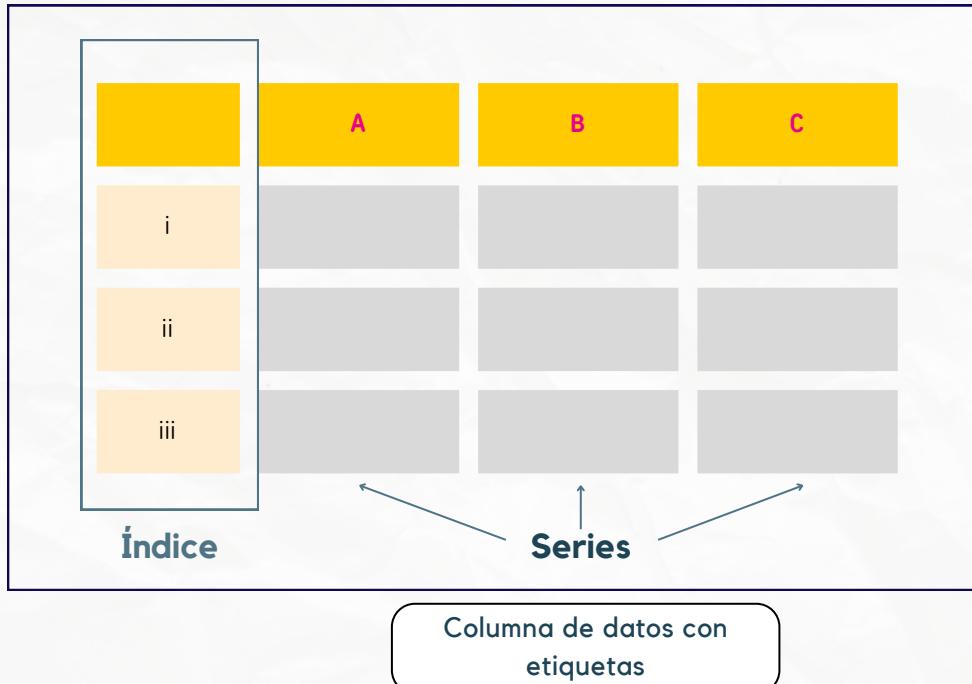


**statsmodels**



# Pandas. Terminología básica

Etiqueta única para  
identificar filas  
(DataFrame) o elementos  
(Series)





# Pandas. Documentación

## Sitio oficial

<https://pandas.pydata.org/pandas-docs/stable/index.html>

**pandas**

pandas is a fast, powerful, flexible and easy to use open source data analysis and manipulation tool, built on top of the Python programming language.

[Install pandas now!](#)

**Latest version: 2.2.3**

- What's new in 2.2.3
- Release date: Sep 20, 2024
- Documentation (web)
- Download source code

**Follow us**

**Recommended books**

Python for Data Analysis

Effective Pandas 2

**Getting started**

- Install pandas
- Getting started

**Documentation**

- User guide
- API reference
- Contributing to pandas
- Ask a question
- Ecosystem

**Community**

- About pandas
- Ask a question
- Ecosystem

**With the support of:**

## Cheatsheet

**Data Wrangling** with pandas Cheat Sheet <http://pandas.pydata.org>

**Tidy Data – A foundation for wrangling in pandas**

In a tidy data set:

- Each variable is saved in its own column
- Each observation is saved in its own row

Tidy data complements pandas' **vectorized operations**; pandas will automatically preserve observations as you manipulate variables. No other format works as intuitively with pandas.

**Creating DataFrames**

```
df = pd.DataFrame({
    "a": [4, 5, 6],
    "b": [7, 8, 9],
    "c": [10, 11, 12],
    index = [1, 2, 3]
})
```

Specify values for each column.

**Reshaping Data – Change layout, sorting, reindexing, renaming**

**Subset Observations - rows**

**Subset Variables - columns**

**Subsets - rows and columns**

**Method Chaining**

Most pandas methods return a DataFrame so that another pandas method can be applied to the result. This improves readability of code.

```
df = (pd.DataFrame(
        ...).columns['var']
        .value='val'
        .query('val > 200')
    )
```

**Logic in Python (and pandas)**

< less than	> greater than	= equals	<> not equal to
>= greater than or equal	<= less than or equal	!= not equal	~ matches
== equals	!= not equals	is None	contains
<< less than or equal	>> greater than or equal	is not None	not contains
>> greater than or equal	<< less than or equal	is Null	startswith

**Regular Expression Examples**

^/ matches strings containing a period ('.)	Length* matches strings ending with word 'length'
* / matches strings beginning with the word 'Sequel'	Starts* matches strings beginning with 'Sequel'
^abc\$ matches strings beginning with 'abc' and ending with 'abc'	^abc\$ matches strings beginning with 'abc' and ending with 'abc'
^((?!spec)).*	^((?!spec)).* matches strings except the string 'spec'

# Ejemplos prácticos en Jupyter



# **Armado de grupos para el trabajo final**

