



INTELIGENCIA ARTIFICIAL CON APLICACIONES EN ECONOMÍA I

Lina María Castro Córdoba
Clase 19 de agosto de 2025

MACHINE LEARNING

APRENDIZAJE SUPERVISADO

Algoritmos que permiten inferir, a partir de ejemplos, una función o fórmula matemática que **relacione un conjunto de atributos con una variable de respuesta**. El **objetivo** es **predecir** (generalizar) **la respuesta** ante futuras observaciones de los atributos.

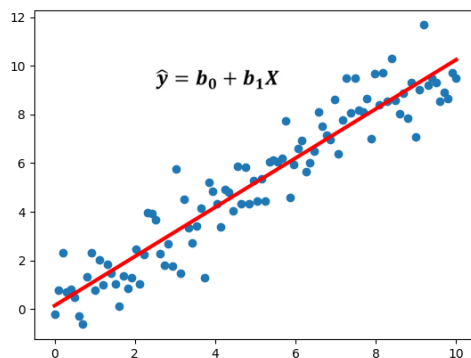
APRENDIZAJE NO SUPERVISADO

Algoritmos que permiten describir cómo están organizados o agrupados un conjunto de datos que **no tienen una variable de respuesta**. El **objetivo** de este método es **conseguir agrupaciones de datos** no detectables a simple vista, **con base en las variables** que describen cada uno de los ejemplos de la muestra.

APRENDIZAJE SUPERVISADO

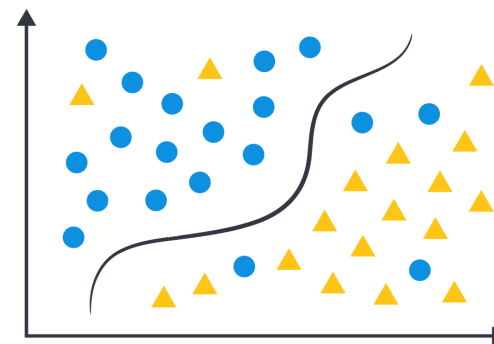
REGRESIÓN

Salario	Nivel de estudios	Sexo	Años de experiencia
\$1.000	Bachillerato	M	2
\$5.500	Doctorado	F	5
\$3.600	Maestría	M	3
\$2.800	Pregrado	F	4



CLASIFICACIÓN

Pobre	Nivel de estudios	Sexo	Salario
SI	Bachillerato	M	\$1.000
NO	Doctorado	F	\$5.500
NO	Maestría	M	\$3.600
SI	Primaria	F	\$500



ALGUNOS ALGORITMOS DE APRENDIZAJE SUPERVISADO

REGRESIÓN LINEAL

Permite predecir el valor de una variable continua como una función lineal de las variables de entrada.

REGRESIÓN LOGÍSTICA

Permite estimar la probabilidad de que ocurra (o no) un evento como función de las variables de entrada.

KNN

Clasifica puntos según la mayoría de votos de sus K vecinos más cercanos teniendo en cuenta las características o variables de entrada.

ÁRBOLES DE DECISIÓN

Estructura en forma de árbol que toma decisiones basadas en reglas if-then.

RANDOM FOREST

Conjunto de árboles de decisión que se entrenan de manera independiente y cuyas predicciones se combinan.

SUPPORT VECTOR MACHINES

Algoritmo de clasificación que encuentra el hiperplano óptimo que separa las clases teniendo en cuenta las características.

GRADIENT BOOSTING MACHINES

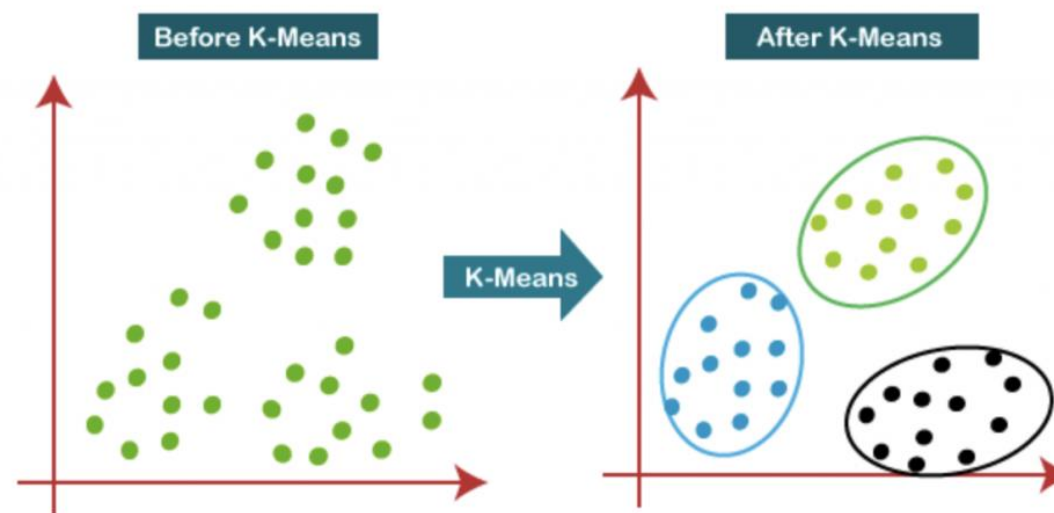
Modelo que construye iterativamente árboles de decisión en serie, cada uno corrigiendo los errores de los anteriores, mejorando el rendimiento en cada iteración.

XGBOOST

Es una implementación avanzada y eficiente de Gradient Boosting, optimizada para mayor velocidad y rendimiento.

APRENDIZAJE NO SUPERVISADO

Cliente	Edad	Sexo	Compras mensuales
Fernando Mora	35	M	\$1.000
Andrea Parra	20	F	\$5.500
Cristina Ramírez	50	F	\$3.600
Juan Gómez	65	M	\$2.800



ALGUNOS ALGORITMOS DE APRENDIZAJE NO SUPERVISADO

K-MEANS

Algoritmo de clustering que agrupa un conjunto de datos en K grupos distintos basados en la distancia. Cada punto de datos se asigna al clúster cuyo centroide está más cercano, y los centroides se recalculan iterativamente hasta que las asignaciones de los puntos no cambian.

Aplicaciones: Segmentación de clientes, análisis de patrones.

PCA

Método de reducción de dimensionalidad que transforma los datos a un nuevo sistema de coordenadas donde las nuevas variables (componentes principales) son combinaciones lineales de las variables originales, ordenadas por la varianza que explican. Se utiliza para simplificar los datos mientras se retiene la mayor variabilidad posible.

Aplicaciones: Visualización de datos, compresión de datos, preprocesamiento para otros algoritmos.

APRIORI

Algoritmo de reglas de asociación que se utiliza para descubrir relaciones entre variables en grandes conjuntos de datos. Busca conjuntos frecuentes de ítems y genera reglas que describen relaciones entre los ítems.

Aplicaciones: Análisis de cestas de compra, detección de patrones en transacciones, recomendadores de productos.

ESTRUCTURAS DE DATOS

SECUENCIALES

Son colecciones de elementos organizados en un orden específico, donde cada elemento tiene una posición determinada (índice) y se puede acceder a ellos siguiendo ese orden.

Ejemplos:

- Listas
- Tuplas
- Cadenas de texto / strings

NO SECUENCIALES

Son aquellas donde los elementos no están almacenados en un orden fijo o consecutivo, y se accede a ellos mediante claves o relaciones, no por posición.

Ejemplos:

- Conjuntos
- Diccionarios

PRÓXIMAS EVALUACIONES Y FECHAS

Corte		Evaluación	Peso	Fecha de entrega
Primer corte	Nota de clase	Lectura 1 artículo boletín Economía (OPCIONAL)	10%	31-Jul-2025
		Quiz		14-Ago-2025
		Tarea fork curso		16-Ago-2025
		Taller tipos de objetos y estructuras en Python (OPCIONAL)		26-Ago-2025
	Parcial	Primer parcial (todo lo visto clases 1-5)	10%	21-Ago-2025
	Proyecto	Primer avance del proyecto (presentación y repositorio)	10%	26-Ago-2025

IMPORTANTE: Del 28 de agosto en adelante habrá quiz al inicio de la clase de lo visto en la sesión anterior.

PRIMER AVANCE DEL PROYECTO

1. Presentación en clase sobre:

- Título del proyecto
- Integrantes
- Objetivo
- Descripción
- Desafíos
- Entrega de valor
- Stakeholders*
- Técnicas que se utilizarán
- Fuentes de datos
- Variables

Duración: Max. 10 minutos de presentación + 5 minutos de comentarios / preguntas del resto de compañeros y docente.



* En español, “partes interesadas” o “grupos de interés”. Son todas las personas, grupos u organizaciones que tienen interés o se ven afectados por las actividades, decisiones o resultados de un proyecto, empresa o política pública.

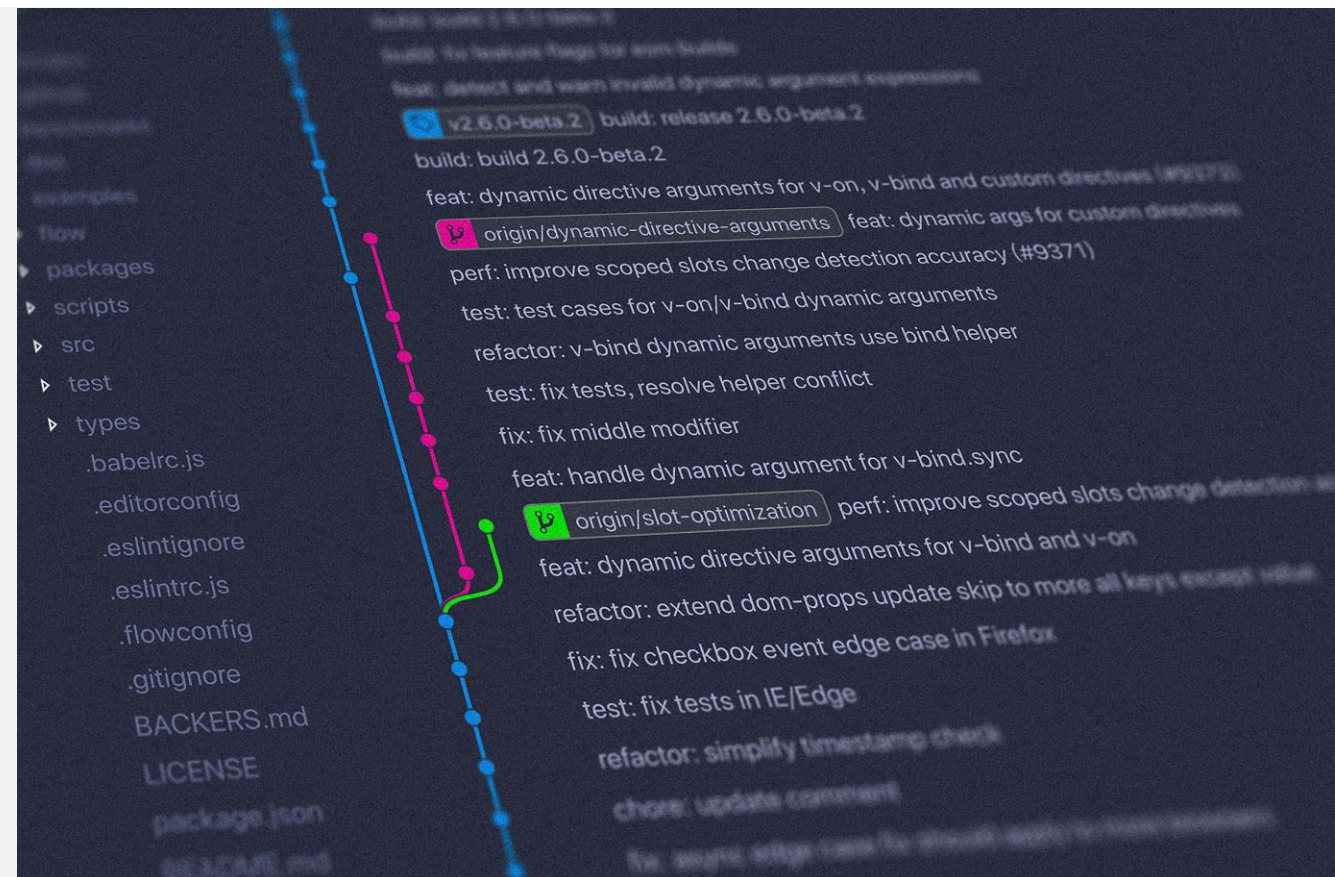
PRIMER AVANCE DEL PROYECTO



2. Creación de repositorio en Github para el proyecto.

Debe contener:

- Archivo requirements.txt
- Archivo README con la redacción del planteamiento del proyecto
- El repositorio debe ser público



EJEMPLO DE FIND EN UN STRING

Es útil, por ejemplo, si estás analizando un reporte económico largo y quieres detectar rápidamente si aparecen los temas que son de tu interés, sin tener que leer todo el documento.

```
informe = """
```

```
El PIB de Colombia creció un 3.2% en 2024.
```

```
La inflación se redujo al 7.5% gracias a la política monetaria.
```

```
Las exportaciones aumentaron, especialmente en el sector agrícola.
```

```
"""
```

```
# Lista de palabras clave a buscar
```

```
palabras_clave = ["PIB", "inflación", "exportaciones", "inversión"]
```

```
# Buscar posiciones
```

```
for palabra in palabras_clave:
```

```
    posicion = informe.find(palabra)
```

```
    if posicion != -1:
```

```
        print(f"Se encontró la palabra '{palabra}' y comienza en la posición  
              {posicion}.")
```

```
    else:
```

```
        print(f"La palabra '{palabra}' no se encontró en el texto.")
```