



Módulo 3

Sesión N° 1



ACTIVIDAD:



Modelado de Regresión Lineal con Álgebra Matricial en Python

- Objetivo: Implementar un sistema en Python que utilice matrices y vectores para resolver un problema de regresión lineal simple utilizando álgebra matricial.



Instrucciones:

1. Diseño e Implementación

- Implementar en Python una solución de regresión lineal basada en álgebra matricial, aplicando el modelo:

$$\beta = (X^T X)^{-1} X^T y$$

Donde:

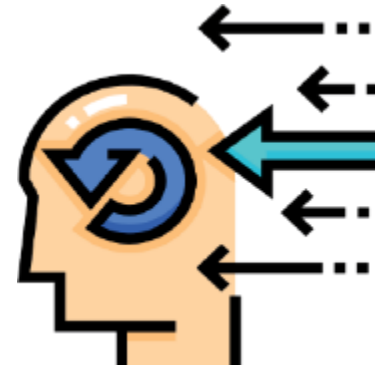
- X : Matriz de diseño (columna de unos + columna de características).
- y : Vector de valores observados (etiquetas).
- β : Vector de parámetros ajustados.
- El sistema debe incluir:
 - Generación de datos sintéticos (x, y) con ruido gaussiano.
 - Creación de la matriz X e implementación del cálculo matricial paso a paso usando NumPy.
 - Predicción de valores y graficación de resultados.
- Incluir en el sistema ejemplos de uso de:
 - Vectores y Matrices: Operaciones como suma, transpuesta, producto punto y multiplicación.
 - Inversa Matricial: Cálculo de $(X^T X)^{-1}$ utilizando `np.linalg.inv`.



2. Funciones Específicas

- `generar_datos(n=100)`
Crea dos arrays "x" e "y" que representen una relación lineal con ruido:
- `x = np.random.rand(n, 1)`
- `y = 4 + 3 * x + np.random.randn(n, 1)`
- `ajustar_modelo(x, y)`
Calcula el vector de parámetros β utilizando álgebra matricial:
- `X = np.hstack([np.ones((n, 1)), x])`
- `beta = np.linalg.inv(X.T @ X) @ X.T @ y`
- `graficar_resultado(x, y, beta)`
Grafica los datos originales y la recta ajustada.
- Se recomienda imprimir la forma (shape) de cada matriz y resultado intermedio para reforzar el aprendizaje conceptual.





3. Documentación y Comentarios

- Cada función debe estar documentada con un docstring que describa sus entradas, salidas y propósito.
- El archivo debe incluir comentarios explicativos del flujo del modelo y por qué cada operación es necesaria desde la perspectiva de Machine Learning.
- Incluir un README breve explicando cómo el álgebra matricial resuelve la regresión lineal y por qué esta técnica es la base de muchos algoritmos en ML.

4. Tiempo Estimado para la Realización de la Actividad

- Duración estimada: 2 a 3 horas

5. Formato de Ejecución

Formato de Trabajo: Trabajo en parejas

Anexo:

Parámetros Fijos y Generación de Datos

1. Semilla Aleatoria
 - Todos los estudiantes deben utilizar `np.random.seed(42)` para garantizar la reproducibilidad de los resultados.
2. Número de Muestras
 - Definir $n = 100$ puntos de datos.
3. Generación de la variable independiente x
 - Generar x como $x = \text{np.random.rand}(n, 1)$
Esto produce valores entre 0 y 1 con forma $(100, 1)$.
4. Función Real y Ruido
 - Usar la siguiente relación para obtener y :

$$y = 4 + 3x + N(0,1)$$

