

Regresión Lineal

En el mercado de smartphones, los dispositivos con mayor capacidad de almacenamiento suelen tener baterías más duraderas. Modelar estos datos podría ayudar a estimar la duración de la batería en función de su capacidad de almacenamiento, algo útil para los consumidores a la hora de elegir un nuevo dispositivo. Sea X la capacidad de almacenamiento de los smartphones (en TB) e Y la duración de su batería (en días), con densidad de probabilidad conjunta de la forma:

$$p_{XY}(x, y) = \frac{3}{4} \cdot \mathbf{1} \{0 < y < 1 + x^2, 0 < x < 1\}$$

(a) *Soluciones óptimas:*

1. Calcular la esperanza condicional.
2. Calcular el error bayesiano.
3. Calcular la recta de regresión.

🔗: Las únicas integrales que debe resolver son con respecto a la marginal $p_X(x)$. El resto de los cálculos debe hacerse utilizando propiedades

(b) *Simulación:* Utilizando `random.uniform` (numpy), generar un dataset de 500 pares de muestras. 🔗: Truncada de uniforme es uniforme y simular truncadas es intuitivo.

(c) *Regresión Lineal:*

1. Implementar una regresión lineal (matricial) a partir de los datos generados previamente. El código debe estar estructurado de la siguiente manera:

```
class regresion_lineal:
    # Opcional, para inicializar atributos o declarar hiperparámetros
    def __init__(self, ...

    # Etapa de entrenamiento
    def fit(self, X, y):

    # Etapa de testeo
    def predict(self, X):
```

A su vez, debe poder extraer los atributos `reglin.w_` y `reglin.b_` una vez que el regresor fue entrenado.

2. Entrenar la regresión lineal con el dataset generado anteriormente.
3. Utilizar el regresor para predecir la duración de una batería de 256GB.

(d) *Comparación:* Graficar en una misma figura (pyplot) el soporte, la esperanza condicional, la recta de regresión (teórica) y la regresión lineal hallada.

(e) Repetir el inciso (c) utilizando gradiente descendente.