

# Algoritmo: $\alpha$ Distance Borderline-ADASYN-SMOTE

## Algoritmo: $\alpha$ Distance Borderline-ADASYN-SMOTE

**Referencia:** Feng & Li, 2021.

### 1. División de vecinos:

Para cada muestra minoritaria  $p_i$ , obtener sus  $m$  vecinos más cercanos y dividirlos en:

- Vecinos de clase minoritaria: cantidad  $pnum$
- Vecinos de clase mayoritaria: cantidad  $nnum$

### 2. Cálculo de pesos inversos:

Para cada vecino  $p_j$  de  $p_i$ , calcular el peso basado en la distancia:

$$\alpha_j = \frac{1}{\text{dist}(p_i, p_j)}$$

donde  $p_j$  es uno de los  $m$  vecinos más cercanos de  $p_i$ .

### 3. Suma de pesos por clase:

$$\alpha'_p = \sum \alpha_j \text{ (de vecinos minoritarios)}, \quad \alpha'_n = \sum \alpha_j \text{ (de vecinos mayoritarios)}$$

### 4. Identificación de muestras peligrosas:

Si  $\alpha'_n > \alpha'_p$ , entonces  $p_i$  se considera una muestra *peligrosa*.

### 5. Cálculo total de ejemplos sintéticos:

$$G = (N - n) \cdot \beta$$

Donde:

- $N$ : número de muestras de la clase mayoritaria
- $n$ : número de muestras de la clase minoritaria
- $\beta \in [0, 1]$ : proporción deseada de balance. Ejemplos:
  - $\beta = 1$ : balance total (igualar ambas clases)
  - $\beta = 0.4$ : generación del 40% de la diferencia

## 6. Distribución proporcional del total:

Para cada muestra peligrosa  $p_i$ , calcular:

$$r_i = \frac{\Delta_i}{m}, \quad \hat{r}_i = \frac{r_i}{\sum r_i}, \quad g_i = \hat{r}_i \cdot G$$

Donde:

- $\Delta_i$ : número de vecinos mayoritarios de  $p_i$
- $m$ : cantidad total de vecinos
- $r_i$ : proporción de vecinos mayoritarios para  $p_i$
- $\hat{r}_i$ : proporción normalizada
- $g_i$ : cantidad de muestras sintéticas a generar para  $p_i$

## 7. Generación de muestras sintéticas:

Para cada  $p_i$ , generar  $g_i$  muestras sintéticas mediante interpolación:

$$s = p_i + \lambda \cdot (p_z - p_i), \quad \lambda \in [0, 1]$$

Donde  $p_z$  es un vecino minoritario aleatorio de  $p_i$ .