

Hiperparámetros que se pueden ajustar en los modelos de ML vistos en clase

1. Regresión Lineal

- `fit_intercept` : Indica si se debe ajustar el intercepto o no.
 - Valores: `True` o `False` .
- `normalize` : Si se debe normalizar las características antes de ajustarlas.
 - Valores: `True` o `False` (disponible en versiones anteriores de `scikit-learn` , pero ahora se hace a través de `Pipeline`).
- `n_jobs` : Número de núcleos de CPU a usar para el ajuste.
 - Valores: Entero (e.g., `1` , `1` para usar todos los núcleos disponibles).
- `positive` : Si las coeficientes de regresión deben ser estrictamente positivas.
 - Valores: `True` o `False` .

Nota: En su forma básica, la regresión lineal no tiene muchos hiperparámetros ajustables, pero si se utiliza con regularización (Lasso o Ridge), los hiperparámetros más relevantes son los relacionados con la regularización:

- `alpha` : El parámetro de regularización en Lasso y Ridge.
 - Valores: Flotante positivo.

2. Regresión Logística

- `penalty` : Tipo de regularización aplicada.
 - Valores: `'l1'` , `'l2'` , `'elasticnet'` , o `'none'` .
- `c` : Parámetro inverso de regularización (menores valores implican mayor regularización).
 - Valores: Flotante positivo.

- **solver** : Algoritmo de optimización utilizado para encontrar los coeficientes.
 - Valores: `'newton-cg'`, `'lbfgs'`, `'liblinear'`, `'sag'`, `'saga'`.
- **max_iter** : Número máximo de iteraciones en el proceso de optimización.
 - Valores: Entero positivo (e.g., `100`, `200`).
- **class_weight** : Peso asociado a cada clase, útil cuando hay un desbalance de clases.
 - Valores: `'balanced'` o diccionario con pesos.
- **fit_intercept** : Si se debe ajustar el intercepto en el modelo.
 - Valores: `True` o `False`.
- **tol** : Tolerancia para la convergencia.
 - Valores: Flotante positivo (e.g., `1e-4`).

3. Árboles de Decisión

- **criterion** : Función para medir la calidad de la división.
 - Valores: `'gini'` (para clasificación), `'entropy'` (para clasificación), `'squared_error'` (para regresión), `'absolute_error'` (para regresión).
- **max_depth** : Profundidad máxima del árbol.
 - Valores: Entero positivo (e.g., `10`, `20`) o `None` (sin límite).
- **min_samples_split** : Número mínimo de muestras necesarias para dividir un nodo.
 - Valores: Entero positivo (e.g., `2`, `10`) o flotante (proporción de muestras).
- **min_samples_leaf** : Número mínimo de muestras que debe tener un nodo hoja.
 - Valores: Entero positivo o flotante (proporción de muestras).
- **max_features** : Número de características a considerar al buscar la mejor división.
 - Valores: `'auto'`, `'sqrt'`, `'log2'`, entero positivo o flotante.
- **max_leaf_nodes** : Número máximo de nodos hoja en el árbol.
 - Valores: Entero positivo o `None`.

- `min_weight_fraction_leaf` : Proporción mínima de la suma de los pesos de las muestras que debe estar en un nodo hoja.
 - Valores: Flotante entre 0 y 1.
- `splitter` : Estrategia usada para dividir en cada nodo.
 - Valores: `'best'` (el mejor split) o `'random'` (split aleatorio).
- `ccp_alpha` : Parámetro de complejidad para la poda mínima coste-complejidad.
 - Valores: Flotante positivo (e.g., `0.01`).

4. k-Nearest Neighbors (kNN)

- `n_neighbors` : Número de vecinos a considerar para la clasificación o regresión.
 - Valores: Entero positivo (e.g., `3` , `5` , `7`).
- `weights` : Función de ponderación utilizada en la predicción.
 - Valores: `'uniform'` (todos los vecinos tienen igual peso), `'distance'` (los vecinos más cercanos tienen más peso) o una función definida por el usuario.
- `algorithm` : Algoritmo utilizado para calcular los vecinos más cercanos.
 - Valores: `'auto'` , `'ball_tree'` , `'kd_tree'` , `'brute'` .
- `leaf_size` : Tamaño de las hojas en los algoritmos de árboles (`ball_tree` o `kd_tree`).
 - Valores: Entero positivo (e.g., `30`).
- `p` : Parámetro para la métrica de distancia. Cuando `p=1` , se usa la distancia de Manhattan; cuando `p=2` , se usa la distancia euclidiana.
 - Valores: Entero positivo (`1` , `2` , etc.).
- `metric` : Métrica utilizada para calcular la distancia entre los puntos.
 - Valores: `'minkowski'` , `'euclidean'` , `'manhattan'` , o cualquier otra métrica definida por el usuario.