

**IMPORTANTE:** Si podés, no leas estas respuestas sin primero leer varias veces el código e intentar responder las preguntas por tu cuenta.

## ClassEncoder

Q1: por qué no hace falta definir un `class_to_name` para el mapeo inverso?

Porque el mapeo de clase a nombre (*int* a *string*) ya se puede accediendo al índice correspondiente del vector `self.names`.

Q2: por que hace falta un reshape?

Porque como NumPy no tiene una función que aplique elemento a elemento una función escalar, la aplicación se debe hacer vía una list comprehension sobre el arreglo flatteneado. Al hacer el flatten se pierde la dimensionalidad original, por lo que es necesario un reshape para recuperarla.

## BaseBayesianClassifier

Q3: para qué sirve bincount?

Sirve para contar las frecuencias absolutas de aparición de cada clase. Al dividir por `y.size`, se recuperan las frecuencias relativas, que son la estimación de las probabilidades a priori de cada clase.

Q4: por qué el `_fit_params` va al final? no se puede mover a, por ejemplo, antes de la `priori`?

Repasemos qué ocurre antes:

1. Se encodean las clases para pasarlas a valores numéricos entre 0 y  $k-1$ .
2. Se obtienen las probabilidades a priori (impuestas o estimadas). Notar que si no se encodean previamente las clases, el `bincount` de `_estimate_a_priori` rompe.

El método `_fit_params` requiere que las clases estén encodeadas y utiliza el atributo `self.log_a_priori`, por lo que rompe si alguno de los dos pasos anteriores no es realizado.

## QDA

Q5: por qué hace falta el flatten y no se puede directamente `X[:, y==idx]`?

Porque `y` en realidad está modelado como una matriz para poder ser un vector columna, y NumPy arroja error si se intenta indexar una dimensión (la de las columnas de `X`) con un array de dos dimensiones, incluso si una de esas dos es de tamaño 1.

Q6: por qué se usa `bias=True` en vez del default `bias=False`?

Porque `bias=True` divide por  $n$  en vez de por  $n-1$ , que es el valor default, y es lo que queremos porque ese es el estimador de máxima verosimilitud.

Q7: qué hace `axis=1`? por que no `axis=0`?

Porque en nuestro esquema de observaciones como vectores columna, los datos se encuentran en formato (features, observaciones), donde una matriz `X` de dimensiones  $(n, m)$  indica que tiene  $n$  features y hay  $m$  observaciones. Calcular la media sobre las columnas es promediar sobre las observaciones, hacerlo sobre las features (`axis=0`) no tendría sentido.

