Proyecto: análisis completo con PyMC

En este proyecto aplicas todo el flujo del curso a un problema real: estimar la probabilidad de que un tumor de mama sea maligno a partir de mediciones de imágenes de biopsia. Vas a construir, diagnosticar, comparar y validar una regresión logística bayesiana, y a entregar predicciones con incertidumbre para pacientes que el modelo no vio. El notebook de referencia adjunto contiene la solución completa: úsalo para comparar al terminar.

Los datos

Usamos el conjunto *Breast Cancer Wisconsin (Diagnostic)*, incluido en scikit-learn, así que no hace falta descargar nada. Tiene 569 casos y 30 mediciones por caso. En scikit-learn, `target = 0` significa maligno, así que definimos la variable respuesta al revés para que 1 sea el evento de interés:

```
import pytensor
pytensor.config.mode = "NUMBA"
from sklearn.datasets import load_breast_cancer
from sklearn.model_selection import train_test_split

datos = load_breast_cancer(as_frame=True).frame
datos["maligno"] = 1 - datos["target"]   # 37.3 % malignos
variables = ["mean radius", "mean texture",
             "mean smoothness", "mean concave points"]

entrena, prueba = train_test_split(
    datos, test_size=0.3, random_state=42,
    stratify=datos["maligno"])
media = entrena[variables].mean()
desv = entrena[variables].std()
```

Estandariza ambos conjuntos con la media y la desviación del conjunto de entrenamiento (398 casos); el de prueba (171 casos) no se toca hasta el final.

Lo que debes entregar

1. **Predictiva a priori.** Define $\alpha \sim \mathcal{N}(0, 1.5)$ y $\beta_j \sim \mathcal{N}(0, 1)$, simula del prior y muestra que las probabilidades simuladas no se amontonan en 0 y 1.

2. **Ajuste con NUTS.** Cuatro cadenas de 1,000 sorteos con 1,000 de calentamiento y `random_seed=42`. Reporta divergencias, $\hat{R}$ y ESS, y explica si confías en el resultado.
3. **Interpretación.** Tabla de coeficientes con media y HDI del 94 %, y su traducción a razones de momios por cada desviación estándar. ¿Qué variable tiene el efecto más fuerte?
4. **NUTS frente a ADVI.** Ajusta el mismo modelo con `pm.fit(30000, method="advi")`, antes de cualquier predictiva posterior. Compara medias y desviaciones estándar y explica las diferencias.
5. **Comparación de modelos.** Ajusta un modelo reducido solo con `mean radius` y `mean texture`. Calcula la log-verosimilitud, corre `az.loo` en ambos y compáralos con `az.compare`. Revisa los valores de Pareto k.
6. **Validación cruzada K-fold.** Con `StratifiedKFold(5, shuffle=True, random_state=42)` sobre el conjunto de entrenamiento, calcula el elpd fuera de muestra del modelo completo y compáralo con el elpd de LOO.
7. **Evaluación en prueba.** Para cada paciente de prueba calcula la probabilidad posterior media y su intervalo del 94 %. Reporta exactitud, AUC, puntuación de Brier y una tabla de calibración por grupos de probabilidad.
8. **Conclusión.** Un párrafo para un equipo médico no técnico: qué tan confiable es el modelo y por qué el intervalo de cada paciente importa tanto como la probabilidad media.

Resultados de referencia

Si sigues las semillas y particiones indicadas, deberías obtener valores cercanos a estos:

- NUTS sin divergencias; coeficientes: radio 2.598, textura 1.421, suavidad 0.853 y puntos cóncavos 2.450.
- ADVI con medias casi iguales, pero desviaciones menores (suavidad: 0.243 frente a 0.315 de NUTS).
- elpd_loo del modelo completo: −59.76 (SE 7.79); el reducido queda 49.97 puntos por debajo (dse 8.65). Todos los Pareto k menores que 0.7.
- K-fold de 5 grupos: elpd −60.34, muy cerca del valor de LOO.
- En prueba: exactitud 0.936, AUC 0.987, Brier 0.047. El ancho medio del intervalo del 94 % por paciente es 0.076, pero el más ancho llega a 0.469.

Pequeñas diferencias en el tercer decimal son normales entre versiones y sistemas operativos.

Criterios de evaluación

- **Corrección técnica (40 %):** el modelo, la estandarización sin fuga de información y los diagnósticos son correctos.
- **Validación (30 %):** LOO, K-fold y métricas de prueba bien calculadas e interpretadas.
- **Interpretación (20 %):** razones de momios, comparación NUTS-ADVI y lectura correcta de los intervalos.
- **Comunicación (10 %):** la conclusión es clara para una audiencia no técnica.

Trampas comunes

- Estandarizar con todo el conjunto antes de separar: filtra información del conjunto de prueba.
- Olvidar que en scikit-learn `target = 0` es maligno: los signos de los coeficientes saldrían invertidos.
- Correr `pm.fit` después de `sample_posterior_predictive` en el mismo modelo (falla en PyMC 5.28).
- Reportar solo la exactitud: en un problema médico, la calibración y el intervalo por paciente pesan más.