# 🧪 Laboratorio: Predicción de abandono de clientes (churn) usando regresión logística en Python 🧪

## Objetivo

Construir y analizar un modelo de **regresión logística con regularización** para predecir si un cliente abandonará (churn = 1) o no.

---

## 1. Preparación del conjunto de datos

- Carga el archivo churn.xlsx usando pandas.
- Separa la variable objetivo (churn) del resto de las variables explicativas.
- Divide los datos en conjuntos de entrenamiento (80%) y prueba (20%) usando train_test_split, con random_state=1337.

---

## 2. Entrenamiento del modelo

- Crea un modelo de regresión logística con regularización L2 (penalty='l2') usando sklearn.
- Usa validación cruzada para probar distintos valores de C: [0.01, 0.1, 1, 10, 100].
- Selecciona el mejor modelo según la métrica **accuracy**.
- Entrena el modelo óptimo sobre el conjunto de entrenamiento completo.

---

## 3. Evaluación del modelo

- Calcula y muestra:
  - Accuracy
  - Matriz de confusión
  - F1 Score
  - Precisión y recall
- Comenta cómo interpretas estos resultados.

---

## 4. Análisis del modelo

### a. Coeficientes de regresión

- Extrae los coeficientes del modelo y relaciónalos con los nombres de las variables.

### b. Umbral de decisión

- Calcula las probabilidades predichas (predict_proba).
- Prueba varios umbrales de decisión distintos de 0.5 (ej. 0.3, 0.6).
- Para cada umbral, evalúa cómo cambian las métricas (precision, recall, F1).
- Comenta cómo el umbral afecta la clasificación.

---

## 5. Visualizaciones

### a. Curva de decisión

- Genera una gráfica de precision, recall y F1 score en función del umbral de decisión.

### b. Curva Lift

- Ordena los datos de test según probabilidad predicha de churn.
- Calcula y grafica la curva lift.
- Incluye las líneas del modelo aleatorio y del modelo óptimo (si fuera posible).

### c. Gráfico de densidad

- Grafica las distribuciones de las probabilidades predichas para los casos con churn y sin churn.

---

## 6. Análisis tipo “what-if”

- Identifica el coeficiente correspondiente a la variable age.
- Usa las probabilidades predichas para un caso con churn probable bajo (<10%).
- Modifica la edad del cliente en +2 unidades y observa cómo cambia la probabilidad.
- Verifica si el cambio aproximado es proporcional a 1 + 2 * beta_age.

---

## 7. Predicciones

- Muestra un dataframe con:
  - proba_0 (probabilidad de no churn)
  - proba_1 (probabilidad de churn)
  - prediction (basada en un umbral dado)
  - prediction_correct (comparando con el valor real)
- Cambia el umbral y observa cómo cambian las columnas mencionadas.

## 8. Simulación de negocio

- Supón que tienes presupuesto para enviar cupones de retención a 10 clientes del conjunto de prueba.
- Elige los 10 clientes con mayor probabilidad de churn.
- Revisa cuántos de esos realmente hicieron churn (churn = 1).
- Evalúa si tu decisión fue acertada en retrospectiva.

```python
# Importar librerías necesarias
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns

from sklearn.model_selection import train_test_split, GridSearchCV
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import (
    accuracy_score,
    confusion_matrix,
    classification_report,
    f1_score,
    precision_recall_curve
)

# Cargar el dataset (asegúrate de tener 'churn.xlsx' en tu carpeta de trabajo)
df = pd.read_excel("churn.xlsx")

# Mostrar las primeras filas
df.head()
```

- **Desde aquí os toca a vosotros ☺**