# Clasificación de Ingresos con Spark ML

La empresa **DataPros** está interesada en construir un modelo que le permita predecir si una persona gana **más de 50K al año** o no, a partir de ciertas características demográficas y laborales.

Se te ha entregado un archivo CSV llamado adult_income_sample.csv, que contiene **2000 registros simulados** con la siguiente información:

| Columna | Descripción |
|---|---|
| age | Edad de la persona (años) |
| sex | Género (Male, Female) |
| workclass | Tipo de empleo (Private, Self-emp, Gov) |
| fnlwgt | Peso estadístico asociado al registro |
| education | Nivel educativo (Bachelors, HS-grad, 11th, Masters, etc.) |
| hours_per_week | Horas trabajadas por semana |
| label | Clase objetivo: >50K o <=50K |

## Objetivo

Construir un modelo de **clasificación binaria** con **Spark ML** utilizando **Logistic Regression** para predecir si una persona pertenece a la clase >50K o <=50K.

## Tareas a realizar

1. **Carga de datos**
    - Leer el archivo CSV en un DataFrame de Spark.
    - Inspeccionar el esquema y mostrar algunos registros para entender los datos.
2. **Preprocesamiento de variables categóricas**
    - Usar `StringIndexer` para transformar las columnas categóricas (`sex`, `workclass`, `education`, `label`).
    - Aplicar `OneHotEncoder` para convertir esas variables en vectores binarios y evitar interpretaciones de orden.

3. **Ensamblaje de características**
    - Construir un vector de características (`features`) con las columnas: `age`, `fnlwgt`, `hours_per_week`, más las variables categóricas codificadas.
4. **Definición y entrenamiento del modelo**
    - Configurar un modelo de **Logistic Regression** con Spark ML.
    - Usar un **Pipeline** para encadenar todo el flujo: indexación, codificación, ensamblaje y entrenamiento.
5. **Evaluación del modelo**
    - Entrenar el modelo con los 2000 registros del archivo.
    - Mostrar las predicciones junto con las probabilidades y la etiqueta real (`label`).
    - Reflexiona: ¿Qué observas sobre los resultados?
6. **Predicción con nuevos datos**
    - Construir un DataFrame con al menos **9 registros nuevos** (creados por ustedes mismos).
    - Aplicar el modelo entrenado para predecir si esas personas ganan `>50K` o `<=50K`.

## Elementos clave que puedes considerar

- Usa las clases de Spark ML:
    - `StringIndexer`, `OneHotEncoder`, `VectorAssembler`, `LogisticRegression`, `Pipeline`.
- Revisa los métodos `.fit()` y `.transform()` para entrenar el modelo y hacer predicciones.
- Usa `.show(truncate=False)` para visualizar bien las predicciones.

## Entregables

1. Script en PySpark con el desarrollo de todo el flujo.
2. Evidencia de ejecución: capturas de pantalla o salida del notebook mostrando predicciones.
3. (Opcional) Presentación visual de los resultados