

# Práctica 2 — Análisis del dataset Adult Income (Python)

\*\*Integrantes\*\*

- NOMBRE APELLIDO (Integrante 1)
- NOMBRE APELLIDO (Integrante 2)

\*\*Repositorio\*\*: PENDIENTE

\*\*Vídeo\*\*: PENDIENTE

Fecha de generación: \*\*2025-12-19\*\*

## 1. Descripción del dataset

El dataset integrado contiene \*\*48,842\*\* registros. La variable objetivo es `income` (<=50K vs >50K).

Distribución de clases: `<=50K` = \*\*37,155\*\* (76.07%), `>50K` = \*\*11,687\*\* (23.93%).

Se observa desbalance aproximado 3:1. Por tanto, además de la accuracy se reportan métricas por clase (precision/recall/F1) y AUC.

## 2. Integración y selección de los datos

Se integran los conjuntos train y test del Adult Income y se conservan las variables estándar del dominio (edad, educación, horas, capital\_gain/capital\_loss y categóricas de contexto).

## 3. Limpieza de los datos

### 3.1 Faltantes y/o valores perdidos

Faltantes reales (NaN) antes de la limpieza (top 5 por columna):

index	missing_count	missing_pct
age	0	0
workclass	0	0
fnlwgt	0	0
education	0	0
education_num	0	0

Faltantes semánticos antes de la limpieza (incluye '?', vacío y equivalentes) (top 5):

col	missing_count	missing_pct
occupation	2809	5.7512
workclass	2799	5.73072
native_country	857	1.75464
marital_status	0	0
education	0	0

Tratamiento aplicado: categóricas imputadas como `Unknown` y numéricas imputadas con mediana.

### 3.2 Tipos de variables y transformaciones

Se normalizan categóricas (strip) y se tipifican numéricas con coerción segura (valores inválidos pasan a NA y se imputan).

### 3.3 Tratamiento de valores extremos

Para `capital\_gain` y `capital\_loss` se aplica winsorización al percentil 99.5% para limitar el impacto de colas extremas en modelos lineales y métricas.

col	cap	n_capped
capital_gain	41310	244

```
| capital_loss | 2258 | 237 |
```

### 3.4 Consideraciones adicionales

Se preserva el tamaño muestral evitando eliminar filas con faltantes; esto reduce riesgo de sesgo por eliminación y mantiene potencia estadística.

## 4. Análisis y métricas

### 4.1 Supervisado y no supervisado

\*\*Modelo supervisado (Regresión logística):\*\* ROC-AUC = \*\*0.9048\*\*, Accuracy = \*\*0.8529\*\*.

Baseline (predecir siempre la clase mayoritaria): \*\*0.7607\*\*.

Para la clase `>50K` (positiva): Precision = \*\*0.736\*\*, Recall = \*\*0.601\*\*, F1 = \*\*0.662\*\*.

Interpretación: AUC alto indica buena discriminación; el recall moderado sugiere que el modelo pierde parte de los casos `>50K`, fenómeno consistente con el desbalance.

Matriz de confusión (test): TN=8658, FP=631, FN=1165, TP=1757.

\*\*No supervisado (PCA+KMeans):\*\* muestra n = \*\*800\*\*, k = \*\*2\*\*, silhouette = \*\*0.4118\*\*.

Interpretación: el clustering es exploratorio y depende del muestreo; no se extraen conclusiones predictivas fuertes sin validación de estabilidad.

### 4.2 Contraste de hipótesis

Contraste entre grupos de `income` sobre `hours\_per\_week` usando \*\*Mann–Whitney U\*\* (prueba no paramétrica, no requiere normalidad).

Medias: <=50K = \*\*38.84\*\*, >50K = \*\*45.45\*\*. Medianas: <=50K = \*\*40.00\*\*, >50K = \*\*40.00\*\*.

p-value = \*\*< 1e-300\*\*.

Interpretación: evidencia estadística fuerte de diferencias entre grupos; esto indica asociación, no causalidad.

## 5. Representación de resultados

Figuras generadas en `reports/figures/`: `roc\_curve.png` y `confusion\_matrix.png`. Tablas en `reports/tables/`.

## 5. Representación de resultados

### 5.1 Vista previa del dataset limpio

Primeras 5 filas (columnas seleccionadas):

Muestra estratificada (3 filas de `<=50K` y 2 filas de `>50K`):

```
| age | workclass | education | hours_per_week | capital_gain | capital_loss | income |
|---:|:---:|:---:|:---:|:---:|:---:|:---:|
| 28 | Private | Some-college | 25 | 0 | 0 | <=50K |
| 44 | Private | Some-college | 38 | 0 | 0 | <=50K |
| 24 | Unknown | Some-college | 30 | 0 | 0 | <=50K |
| 27 | Self-emp-not-inc | HS-grad | 60 | 0 | 0 | >50K |
| 43 | Self-emp-inc | Assoc-voc | 50 | 0 | 0 | >50K |
```

### 5.2 Métricas del modelo supervisado

Tabla resumida (precision/recall/F1/support):

```
| precision | recall | f1-score | support | |
|---|---|---|---|---|
| 0 | 0.881401 | 0.93207 | 0.906028 | 9289 |
| 1 | 0.735762 | 0.6013 | 0.66177 | 2922 |
| macro avg | 0.808581 | 0.766685 | 0.783899 | 12211 |
| weighted avg | 0.846551 | 0.852919 | 0.847579 | 12211 |
```

### 5.3 Gráficos generados

Se incluyen las figuras principales del análisis:

\*\*ROC Curve\*\*

![ROC Curve](figures/roc\_curve.png)

\*\*Matriz de confusión\*\*

![Confusion Matrix](figures/confusion\_matrix.png)

## 6. Conclusiones

El dataset permite construir un clasificador con buen desempeño (AUC alto) frente al baseline, aunque la recuperación de la clase `>50K` es moderada por el desbalance. El contraste sugiere diferencias consistentes en horas trabajadas entre grupos. El análisis no supervisado se interpreta como exploratorio.

## 7. Código

El código fuente se encuentra en `src/`. Para ejecutar el pipeline: `python -m src.run\_all`.

## 8. Vídeo

Enlace al vídeo (Google Drive UOC): PENDIENTE

## Tabla de contribuciones

Contribuciones	Firma
Investigación previa	AA, BB
Redacción de las respuestas	AA, BB
Desarrollo del código	AA, BB
Participación en el vídeo	AA, BB