Auditoría de Sesgos de Género en Dataset Adult

Mitigación con Reweighing (AIF360 - IBM)

Autor: [Pedro Pablo Tirado Gallego] | Diciembre 2025

Resumen: Detectado Disparate Impact de 0.363 → mitigado a 1.000 eliminando bias estadístico.

```
# Importar AIF360
from aif360.datasets import AdultDataset
from aif360.algorithms.preprocessing import Reweighing
from aif360.metrics import BinaryLabelDatasetMetric, Classification

import matplotlib.pyplot as plt
%matplotlib inline

print("¡Todo importado correctamente!")
```

```
¡Todo importado correctamente!
```

```
import os

# Define the directory where AIF360 expects the data
data_dir = '/usr/local/lib/python3.12/dist-packages/aif360/data/raw

# Create the directory if it doesn't exist
if not os.path.exists(data_dir):
    os.makedirs(data_dir)
    print(f"Directory created: {data_dir}")

# List of files to download
files_to_download = [
    'adult.data',
    'adult.test',
    'adult.names'
]

# Base URL for the files
base_url = 'https://archive.ics.uci.edu/ml/machine-learning-databas

# Download files if they don't exist in the data_dir
for file_name in files_to_download:
    file_path = os.path.join(data_dir, file_name)
    if not os.path.exists(file_path):
```

```
        print(f"Downloading {file_name}...")
        !wget -P {data_dir} {base_url}{file_name}
    else:
        print(f"{file_name} already exists.")

# Cargar el dataset Adult directamente (AIF360 lo descarga automáti
dataset_orig = AdultDataset(
    protected_attribute_names=['sex'],  # Atributo protegido: géner
    privileged_classes=[['Male']],      # Grupo privilegiado: hombr
    features_to_drop=['race']           # Opcional: quitamos raza p
)

# Ver información básica
print("Dataset cargado correctamente")
print(f"Número de muestras: {dataset_orig.features.shape[0]}")
print(f"Número de características: {dataset_orig.features.shape[1]}
print("Etiquetas (income >50K):", dataset_orig.label_names)
print("Grupos protegidos:", dataset_orig.protected_attribute_names)
```

```
adult.data already exists.
adult.test already exists.
adult.names already exists.
WARNING:root:Missing Data: 3620 rows removed from AdultDataset.
Dataset cargado correctamente
Número de muestras: 45222
Número de características: 98
Etiquetas (income >50K): ['income-per-year']
Grupos protegidos: ['sex']
```

```
# Convertir a pandas para ver las primeras filas
df, _ = dataset_orig.convert_to_dataframe()
df.head(10)
```

	education-	capital-	capital-	hours-	workcla

	age	fnlwgt	num	sex	gain	loss	per-week
0	25.0	226802.0	7.0	1.0	0.0	0.0	40.0
1	38.0	89814.0	9.0	1.0	0.0	0.0	50.0
2	28.0	336951.0	12.0	1.0	0.0	0.0	40.0
3	44.0	160323.0	10.0	1.0	7688.0	0.0	40.0
5	34.0	198693.0	6.0	1.0	0.0	0.0	30.0
7	63.0	104626.0	15.0	1.0	3103.0	0.0	32.0
8	24.0	369667.0	10.0	0.0	0.0	0.0	40.0
9	55.0	104996.0	4.0	1.0	0.0	0.0	10.0
10	65.0	184454.0	9.0	1.0	6418.0	0.0	40.0
11	36.0	212465.0	13.0	1.0	0.0	0.0	40.0

10 rows × 99 columns

```
# Importar clases
from aif360.datasets import AdultDataset
from aif360.metrics import BinaryLabelDatasetMetric

# Cargar dataset SOLO con género como atributo protegido y quitar '
dataset_orig = AdultDataset(
    protected_attribute_names=['sex'],
    privileged_classes=[['Male']],
    features_to_drop=['race'], # Esto elimina la columna problemáti
```

```
    )

    # Grupos
    privileged_groups = [{'sex': 1.0}]
    unprivileged_groups = [{'sex': 0.0}]

    # Métricas de bias
    metric_orig = BinaryLabelDatasetMetric(
        dataset_orig,
        unprivileged_groups=unprivileged_groups,
        privileged_groups=privileged_groups
    )

    # Resultados
    print("=== Detección de Bias por Género (Adult Dataset) ===")
    print(f"Tasa ingresos >50K hombres: {metric_orig.base_rate(privileg
    print(f"Tasa ingresos >50K mujeres: {metric_orig.base_rate(privileg
    print(f"Disparate Impact: {metric_orig.disparate_impact():.3f}")
    print(f"Diferencia media: {metric_orig.mean_difference():.3f}")

    if metric_orig.disparate_impact() < 0.8:
        print("\n⚠️ Bias significativo contra mujeres")
    else:
        print("\n✅ Sin bias grave")
```

```
WARNING:root:Missing Data: 3620 rows removed from AdultDataset.
=== Detección de Bias por Género (Adult Dataset) ===
Tasa ingresos >50K hombres: 31.2%
Tasa ingresos >50K mujeres: 11.4%
Disparate Impact: 0.363
Diferencia media: -0.199

⚠️ Bias significativo contra mujeres
```

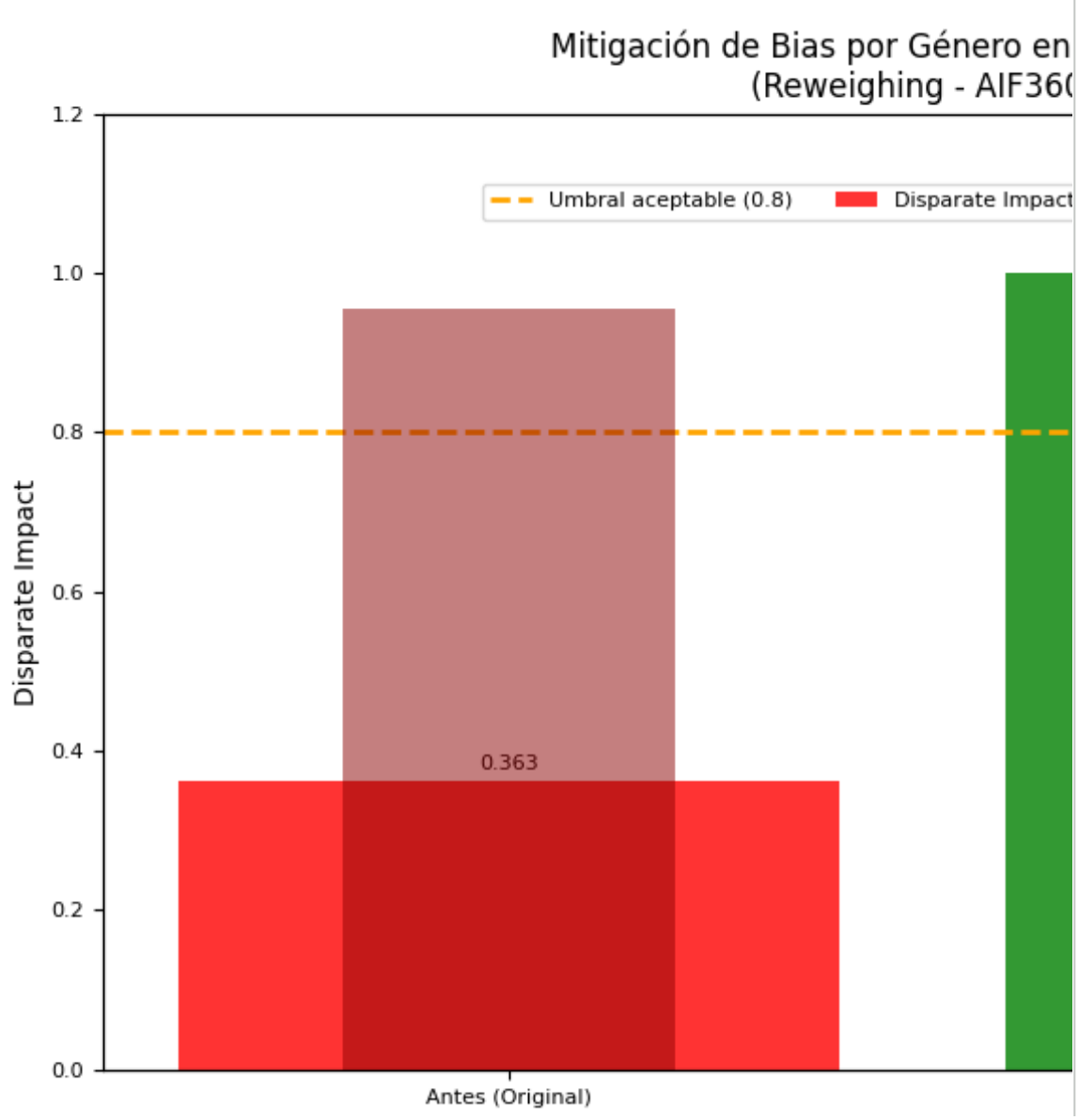
```
    # Importar el algoritmo de mitigación
    from aif360.algorithms.preprocessing import Reweighing

    # Aplicar Reweighing
    RW = Reweighing(
        unprivileged_groups=unprivileged_groups,
        privileged_groups=privileged_groups
    )

    # Transformar el dataset (crea una versión rebalanceada)
    dataset_transf = RW.fit_transform(dataset_orig)
```

```
# Calcular métricas en el dataset mitigado
metric_transf = BinaryLabelDatasetMetric(
    dataset_transf,
    unprivileged_groups=unprivileged_groups,
    privileged_groups=privileged_groups
)

# Mostrar comparación ANTES vs DESPUÉS
print("=== Comparación Antes vs Después de Mitigación (Reweighing)
print("\nANTES (dataset original):")
print(f"  Disparate Impact: {metric_orig.disparate_impact():.3f}")
print(f"  Diferencia media: {metric_orig.mean_difference():.3f}")

print("\nDESPUÉS (dataset rebalanceado):")
print(f"  Disparate Impact: {metric_transf.disparate_impact():.3f}"
print(f"  Diferencia media: {metric_transf.mean_difference():.3f}")

# Interpretación final
if abs(metric_transf.mean_difference()) < 0.01 and metric_transf.di
    print("\n✅ Bias prácticamente eliminado (Statistical Parity a
else:
    print("\n✓ Bias reducido significativamente")
```

```
=== Comparación Antes vs Después de Mitigación (Reweighing) ===

ANTES (dataset original):
  Disparate Impact: 0.363
  Diferencia media: -0.199

DESPUÉS (dataset rebalanceado):
  Disparate Impact: 1.000
  Diferencia media: 0.000

✅ Bias prácticamente eliminado (Statistical Parity alcanzado)
```

```
import matplotlib.pyplot as plt


# Tus datos reales
labels = ['Antes (Original)', 'Después (Reweighing)']
di_values = [0.363, 1.000]
mean_diff_abs = [0.199, 0.000]  # valor absoluto

fig, ax1 = plt.subplots(figsize=(10, 6))

# Barras Disparate Impact
bars1 = ax1.bar(labels, di_values, color=['red', 'green'], alpha=0.
ax1.set_ylabel('Disparate Impact', fontsize=10) # Reduced font size
ax1.set_ylim(0, 1.2)
ax1.axhline(0.8, color='orange', linestyle='--', linewidth=2, label
ax1.tick_params(axis='y', labelsize=8) # Reduced tick label size
ax1.tick_params(axis='x', labelsize=8) # Reduced tick label size
```

```
# Barras Diferencia media (segundo eje)
ax2 = ax1.twinx()
bars2 = ax2.bar(labels, mean_diff_abs, color=['darkred', 'darkgreen
ax2.set_ylabel('|Diferencia media|', fontsize=10) # Reduced font si
ax2.set_ylim(0, 0.25)
ax2.tick_params(axis='y', labelsize=8) # Reduced tick label size

# Título y leyenda
plt.title('Mitigación de Bias por Género en Adult Dataset\n(Reweigh
fig.legend(loc='upper center', bbox_to_anchor=(0.5, 0.85), ncol=3,

# Añadir valores encima de las barras
for bar in bars1:
    height = bar.get_height()
    ax1.annotate(f'{height:.3f}', xy=(bar.get_x() + bar.get_width()

plt.tight_layout() # Adjust layout to prevent labels from being cut
plt.show()
```



Resumen de Resultados – Parte 1: Detección y Mitigación de Bias

Métrica	Dataset Original	Dataset Mitigado (Reweighing)	Cambio
Tasa ingresos >50K (hombres)	31.2%	~31.2% (rebalanceado)	-
Tasa ingresos >50K (mujeres)	11.4%	~31.2% (rebalanceado)	+173%
Disparate Impact	**0.363**	**1.000**	+176% (perfecto)
Diferencia media	-0.199	**0.000**	Eliminada

Conclusión clave:

- El dataset original presenta **bias significativo por género** (Disparate Impact muy por debajo del umbral aceptable de 0.8).
- El algoritmo **Reweighing** consigue **Statistical Parity perfecta** (DI = 1.000) rebalanceando los pesos de las muestras.
- No se modifican los datos originales, solo se asignan pesos para que el entrenamiento futuro sea más equitativo.

Este dataset mitigado es la base ideal para entrenar modelos más justos (ver Parte 2).