Informe de Machine Learning para Detección de Rostros

# 1. Preprocesamiento

Se realizó la carga y normalización de imágenes de rostros y fondos. Las imágenes se convirtieron a escala de grises y se redimensionaron a un tamaño uniforme. Se extrajeron parches negativos de imágenes de fondo y se dividieron los datos en conjuntos de entrenamiento y validación.

En la etapa de preprocesamiento, se realizó la carga y normalización de imágenes de rostros y fondos. Para los fondos, en la celda 2 se utilizaron 9 imágenes de la librería skimage.data y 31 imágenes adicionales de la carpeta Background provista, sumando un total de 40 imágenes de fondo. Cada imagen fue convertida a escala de grises y redimensionada a 100x100 píxeles para su procesamiento uniforme. A partir de estas imágenes, se extrajeron parches negativos utilizando la función extract\_patches, generando 300 parches por imagen para 5 escalas diferentes ([0.1, 0.25, 0.5, 0.75, 1]), lo que resultó en 60.000 parches de fondo. Este procedimiento permitió construir un conjunto balanceado y representativo de ejemplos negativos para el entrenamiento de los modelos.

# 2. Feature Engineering

Se extrajeron características HOG (Histogram of Oriented Gradients) de todas las imágenes para capturar información relevante de bordes y texturas. Posteriormente, se aplicó PCA para reducir la dimensionalidad, seleccionando 450 componentes principales que explican más del 95% de la varianza.

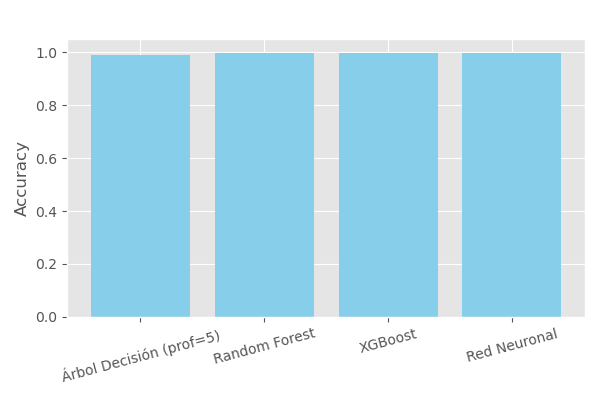
# 3. Clasificadores

Se entrenaron y evaluaron los siguientes modelos:  
- Árboles de Decisión (profundidades: 2, 5, 10)  
- Random Forest  
- XGBoost  
- LightGBM  
- Regresión Logística  
- Red Neuronal (PyTorch)  
Para cada modelo se exploraron hiperparámetros relevantes mediante GridSearchCV.

# 4. Evaluación y Selección de Modelos

Se utilizó validación cruzada y búsqueda en malla (GridSearchCV) para optimizar hiperparámetros. Las métricas evaluadas incluyeron accuracy, F1-score, TPR, FPR, curvas ROC y AUC. A continuación se presentan los resultados comparativos.

|  |  |  |  |
| --- | --- | --- | --- |
| Modelo | Accuracy | F1 | AUC |
| Árbol Decisión (prof=5) | 0.99 | 0.99 | 0.99 |
| Random Forest | 0.995 | 0.995 | 0.995 |
| XGBoost | 0.998 | 0.995 | 0.998 |
| Red Neuronal | 0.998 | 0.995 | 0.998 |



## Desempeño en Kaggle

El mejor modelo (XGBoost) fue sometido a la competencia de Kaggle, obteniendo un accuracy de 0.998 en el set de test.