

Reporte de Resultados

1) Decisiones de procesamiento de datos

Antes de entrenar los modelos se tomaron varias decisiones basadas en la estructura del dataset Adult y con el objetivo de evitar fuga de información entre particiones. En cuanto a variables, se emplearon todas las reportadas en *adult.names*, incluyendo numéricas y categóricas, y la variable objetivo *income* se mapeó a 0 para $\leq 50K$ y 1 para $> 50K$. Respecto de los valores faltantes, en *work class*, *occupation* y *native-country* se imputó la categoría *Unknown* para no descartar filas y permitir que el modelo aprenda la señal asociada a la ausencia de información. Para la codificación y el escalamiento, las variables categóricas se transformaron con One-Hot Encoding configurado con *handle_unknown='ignore'* para prevenir errores ante categorías no vistas en validación o prueba, mientras que las numéricas se estandarizaron con *StandardScaler*. Tanto el OHE como el *scaler* se ajustaron exclusivamente sobre el conjunto de entrenamiento para evitar *data leakage*.

El particionado utilizó *adult.data* como entrenamiento y dividió *adult.test* en partes de validación y prueba final con una proporción 50/50 y estratificación por *income* para asegurar estabilidad en las proporciones de clase. En esta entrega no se añadieron características sintéticas; la prioridad fue mantener un flujo reproducible y claro con imputación mínima, codificación OHE para las categóricas y normalización de las numéricas.

2) Hiperparámetros del mejor MLP y comparación sin/con regularización

Se exploraron múltiples configuraciones variando profundidad y ancho de la red, *dropout*, *weight decay* y número de épocas. La selección del mejor modelo se realizó con base en la menor pérdida de validación (*val_loss*) empleando *Early Stopping* con *patience=10*. El mejor MLP seleccionado tuvo capas ocultas de tamaños [256, 128, 64], *dropout* de 0,5, función de pérdida *BCEWithLogitsLoss* con *pos_weight = #neg/#pos* para manejar el desbalance, optimizador Adam con *lr=1e-3* y *weight_decay=1e-4*, *batch size* de 128 y criterio de parada temprana sobre *val_loss*.

Al comparar configuraciones sin regularización frente a la variante regularizada, se observó que, sin regularización (por ejemplo, una red con [64] neuronas, *dropout* 0,0 y *weight_decay* 0,0), las curvas mostraron una caída rápida del *loss* de entrenamiento pero un estancamiento o empeoramiento del *loss* de validación después de varias épocas, evidenciando sobreajuste. En cambio, con regularización (el mejor MLP con *dropout* 0,5 más *weight_decay* $1e-4$ y *Early Stopping*), la pérdida de validación se estabilizó de forma temprana, se evitó el sobreajuste y además se incrementaron el *recall* y el ROC-AUC finales. En síntesis, la combinación de *dropout*, *weight decay* y parada temprana mejora la generalización; si el objetivo es capturar más positivos ($> 50K$), esta configuración es preferible.

3) Comparación del mejor MLP frente a Regresión Logística (línea base) e interpretación

Evaluación del mejor MLP

Split	Accuracy	Precisión	Recall	F1	ROC-AUC
Train	0.8274	0.5946	0.8901	0.7129	0.9277
Val	0.8104	0.5654	0.8539	0.6803	0.9091
Test	0.8111	0.5655	0.8638	0.6835	0.9143

Comparativa final: LogReg vs MLP

Modelo	Accuracy	Precision	Recall	F1	ROC-AUC
LogReg	0.8571	0.7375	0.6136	0.6699	0.9081
MLP	0.8111	0.5655	0.8638	0.6835	0.9143

El mejor MLP evaluado obtuvo, en *train*, *val* y *test*, desempeños consistentes: *accuracy* 0,8274, 0,8104 y 0,8111; *precision* 0,5946, 0,5654 y 0,5655; *recall* 0,8901, 0,8539 y 0,8638; *F1* 0,7129, 0,6803 y 0,6835; y ROC-AUC 0,9277, 0,9091 y 0,9143, respectivamente. En la comparativa final entre modelos, la Regresión Logística alcanzó *accuracy* 0,8571, *precision* 0,7375, *recall* 0,6136, *F1* 0,6699 y ROC-AUC 0,9081; por su parte, el MLP logró *accuracy* 0,8111, *precision* 0,5655, *recall* 0,8638, *F1* 0,6835 y ROC-AUC 0,9143.

La interpretación es la siguiente. El MLP consigue un *recall* superior (0,864) y un ROC-AUC más alto (0,914), lo que indica que detecta más casos de ingresos >50K y separa mejor las clases en términos probabilísticos; es especialmente útil cuando el costo de perder un positivo (falso negativo) es alto. La Regresión Logística presenta mayor *accuracy* (0,857) y mejor *precision* (0,738), por lo que comete menos falsos positivos, algo ventajoso si estos resultan más costosos; sin embargo, deja escapar más positivos reales debido a su menor *recall*. El *F1* del MLP (0,683) supera ligeramente al de la Logística (0,670), coherente con un aumento notable del *recall* incluso sacrificando cierta *precision*. Operativamente, no conviene usar un umbral fijo de 0,5; es preferible ajustar el umbral con la curva *Precision–Recall* y una función de costos que refleje el balance entre falsos negativos y falsos positivos para el problema. El MLP, además, permite desplazar el punto de operación para elevar la *precision* manteniendo un *recall* alto si así se requiere.

Notas finales

El flujo seguido respeta buenas prácticas: OHE y *scaler* ajustados solo en entrenamiento, separación explícita de validación, y tratamiento del desbalance incorporado en la pérdida del MLP. Si se necesita, puede añadirse una sección de elección de umbrales óptimos (seleccionados en validación por *F1* o por costo) y la inclusión de las figuras de ROC y *Precision–Recall* exportadas desde el *notebook*.

