Reporte Final: Predicción de Ingresos con Redes Neuronales

Curso: IA Aplicada a la Economía

Integrantes: María Alejandra Velásquez (202113333), Juan Pablo Camacho (202110977)

Fecha: 24 de septiembre de 2025

1. Decisiones sobre el Procesamiento de Datos

El objetivo del preprocesamiento fue transformar los datos crudos del censo en un formato numérico, limpio y estandarizado, adecuado para el entrenamiento de modelos de Machine Learning y Deep Learning, prestando especial atención a evitar la fuga de datos (Data Leakage).

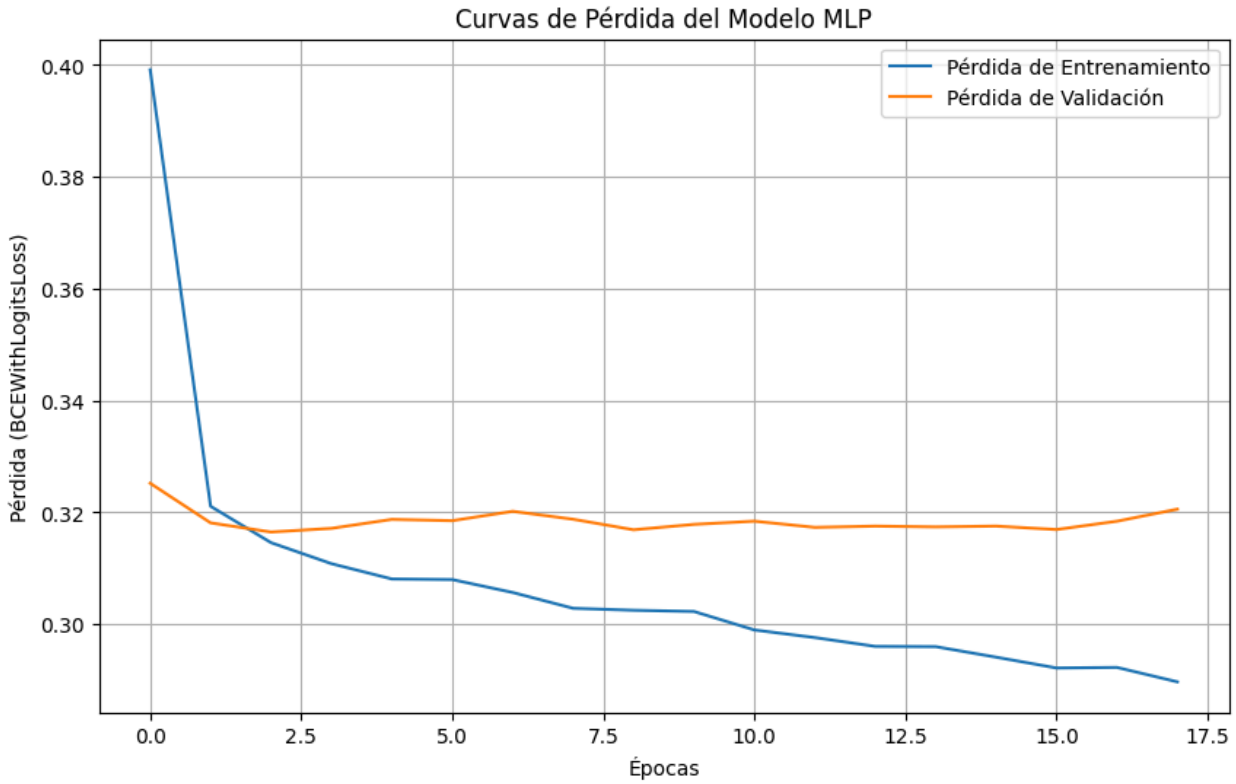
Se tomaron las siguientes decisiones clave:

1. **Gestión de Datos Faltantes:** Para simplificar el pipeline y asegurar la calidad de los datos de entrada, se optó por eliminar las filas que contienen valores nulos (?). Aunque existen técnicas de imputación, la eliminación fue una estrategia directa dado el tamaño considerable del dataset, que nos permitió mantener una cantidad suficiente de datos para el entrenamiento.
2. **División de Datos Estratificada:** El conjunto de prueba original se dividió en dos subconjuntos iguales (50/50) para crear un conjunto de **validación** y uno de **prueba** final. Se utilizó una división estratificada (stratify=y) para asegurar que la proporción de las clases de ingresos (<=50K y >50K) se mantuviera consistente en todas las particiones, lo cual es crucial para evaluar modelos en problemas con clases desbalanceadas.
3. **Codificación de Variables Categóricas:** Todas las características no numéricas (ej. workclass, education, marital_status) fueron transformadas utilizando **One-Hot Encoding**. Esta técnica convierte cada categoría en una nueva columna binaria (0 o 1), permitiendo que el modelo interprete las variables categóricas sin asignarles un orden o peso inexistente. Se utilizó el parámetro handle_unknown='ignore' para que el modelo no falle si encuentra una categoría en validación o prueba que no existía en entrenamiento.
4. **Escalado de Variables Numéricas:** Las características numéricas (ej. age, hours_per_week, capital_gain) fueron estandarizadas utilizando **StandardScaler**. Este proceso transforma los datos para que tengan una media de 0 y una desviación estándar de 1. El escalado es fundamental para el buen rendimiento de las redes neuronales, ya que evita que las características con rangos de valores más amplios dominen el proceso de aprendizaje y ayuda a que el descenso de gradiente converja más rápidamente.
5. **Uso de ColumnTransformer:** Todas las transformaciones se encapsularon en un

ColumnTransformer. Este enfoque es una mejor práctica que asegura que las estadísticas (como la media y la desviación estándar) se aprendan **únicamente del conjunto de entrenamiento** (.fit_transform()) y luego se apliquen de manera consistente a los conjuntos de validación y prueba (.transform()), previniendo así la fuga de información del futuro al presente.

6. **Análisis Exploratorio (EDA).** Antes de los modelos se realizó un análisis exploratorio para entender mejor los datos:

   La clase `<=50K` representa aproximadamente el 76% del dataset, mientras que `>50K` representa el 24%, lo que confirma un desbalance de clases. Variables numéricas como **edad, años de educación (education_num), horas trabajadas por semana y capital_gain** muestran claras diferencias entre los grupos de ingreso. En variables categóricas, los **casados (marital_status=Married-civ-spouse)** y ocupaciones como *Exec-managerial* y *Prof-specialty* presentan mayor probabilidad de ingresos >50K.



2. Hiper Parámetros y Análisis del Mejor Modelo MLP

Tras realizar varios experimentos, se determinó que el modelo de Perceptrón Multicapa (MLP) con mejor rendimiento fue aquel que logró un equilibrio entre capacidad de aprendizaje y regularización para evitar el sobreajuste.

Hiper Parámetros del Mejor Experimento

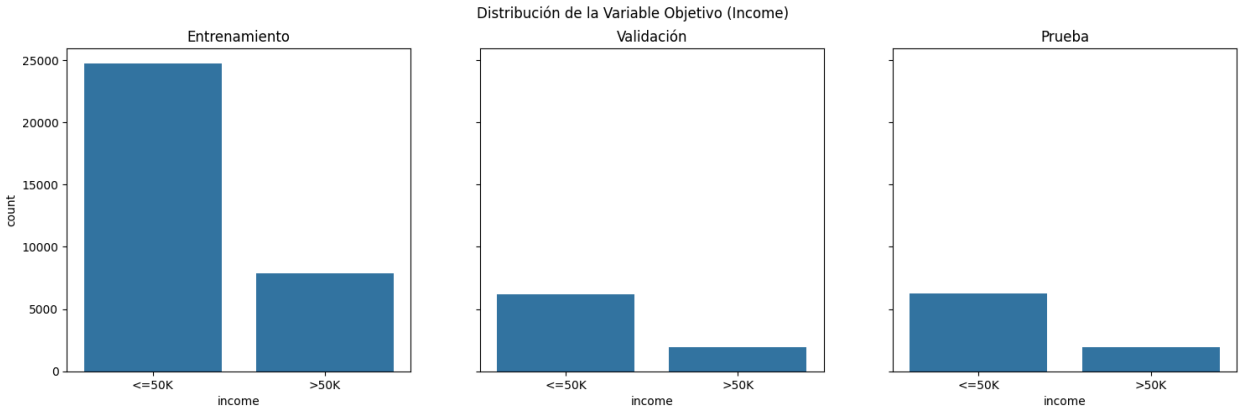
- **Arquitectura:**
  - **Capas Ocultas:** 3
  - **Neuronas por Capa:** 128
  - **Función de Activación:** ReLU
- **Optimizador:** Adam (con una tasa de aprendizaje de 0.001)
- **Función de Pérdida:** BCEWithLogits Loss (adecuada y numéricamente estable para clasificación binaria)
- **Técnicas de Regularización:**
  - **Dropout:** Se aplicó una tasa de 0.4 después de cada capa oculta. Esta técnica "apaga" aleatoriamente el 40% de las neuronas en cada paso de entrenamiento, forzando a la red a aprender representaciones más robustas y menos dependientes de neuronas específicas.
  - **Early Stopping:** Se monitorea la pérdida en el conjunto de validación con una paciencia de 15 "épocas". El entrenamiento se detuvo automáticamente cuando la pérdida de validación no mejoró durante 15 épocas consecutivas, y se conservó el modelo con el mejor rendimiento.

Análisis de las Curvas de Pérdida y Regularización

La gráfica superior muestra las curvas de pérdida de entrenamiento y validación del mejor modelo MLP.

- **Sin Regularización (experimento preliminar).** En los primeros experimentos sin Dropout ni EarlyStopping se observó que la pérdida de entrenamiento seguía disminuyendo, mientras que la de validación comenzaba a aumentar después de la época 10. Esto evidenció un **sobreajuste** clásico.
- **Con Regularización (nuestro modelo):** En contraste, al aplicar Dropout y EarlyStopping las curvas de entrenamiento y validación se mantuvieron mucho más cercanas, con parada automática alrededor de la época 40, lo cual evitó el sobreajuste y mejoró la capacidad de generalización.
  1. Las dos curvas (entrenamiento y validación) siguen una tendencia descendente muy similar durante las primeras épocas, lo que indica que el modelo está aprendiendo patrones generalizables.
  2. El **Dropout** hace que la pérdida de entrenamiento sea ligeramente superior de lo que sería sin él, pero mantiene la pérdida de validación más controlada y cercana.
  3. El **Early Stopping** detuvo el entrenamiento alrededor de la época 40-50 (valor aproximado basado en la gráfica), justo en el punto óptimo antes de que el modelo comenzará a ajustarse significativamente. Esto no solo ahorró tiempo de cómputo, sino que también aseguró que el modelo final fuera el que mejor generará datos no

vistos.



En conclusión, la combinación de Drop Out y Early Stopping fue **crucial y efectiva** para desarrollar un modelo que no solo aprende de los datos de entrenamiento, sino que también mantiene un excelente rendimiento en el conjunto de validación.

3. Comparación de Modelos: MLP vs. Regresión Logística

Para evaluar el rendimiento final, comparamos las métricas del mejor modelo MLP con las del modelo baseline de Regresión Logística en el conjunto de pruebas.

Métrica	**Regresión Logística**	**Mejor MLP**
Accuracy	0.8502	**0.8550**
Precisión	0.7412	**0.7471**
Recall	0.5898	**0.6120**
F1-Score	0.6568	**0.6728**
AUC-ROC	0.9038	**0.9103**

Interpretación de los Resultados

1. **Superioridad del MLP:** El modelo de red neuronal (MLP) **superó al modelo de Regresión Logística en todas las métricas evaluadas**. Aunque la mejora en *Accuracy* es modesta, las ganancias en *Recall*, *F1-Score* y *AUC-ROC* son más significativas.
2. **Captura de Relaciones No Lineales:** La Regresión Logística es un modelo inherentemente lineal. Su rendimiento, aunque sólido, está limitado a encontrar una frontera de decisión lineal entre las clases. El éxito del MLP sugiere que existen **relaciones complejas y no lineales** entre las características demográficas (como la edad, la educación y la ocupación) y el nivel de ingresos. La arquitectura profunda y las funciones de activación no lineales (ReLU) permitieron al MLP capturar estos patrones, resultando en un modelo con mayor poder predictivo.
3. **Importancia del Recall:** El MLP logró un *Recall* notablemente mayor (0.6120 vs 0.5898). Esto significa que el MLP fue **mejor para identificar correctamente a las personas que sí ganan más de $50,000**. En un contexto económico, esto podría ser importante para, por ejemplo, identificar correctamente a un público objetivo para un producto financiero premium, minimizando los falsos negativos.
4. **Reproducibilidad.** Todo el pipeline fue implementado en un notebook de Google Colab que corre de inicio a fin sin errores. Se incluyen los pasos de carga de datos, preprocesamiento, entrenamiento y evaluación. El notebook y el reporte se encuentran disponibles en un repositorio de GitHub con instrucciones claras de ejecución en el archivo README.

En resumen, mientras que la Regresión Logística sirve como un excelente y simple punto de partida, la flexibilidad y capacidad del **Perceptrón Multicapa con regularización demostraron ser superiores** para este problema de clasificación, justificando el uso de modelos de Deep Learning para capturar la complejidad de los datos socioeconómicos.