Redes Neuronales

Santiago Hernandez Barbosa,
Juan Diego Pardo Ballesteros

Pontificia Universidad Javeriana
Facultad de Ingeniería
Introducción a la inteligencia Artificial
Julio Omar Palacio Niño
2025

CONTENIDO

LISTA DE TABLAS

LISTA DE GRÁFICAS

GLOSARIO

Aprendizaje supervisado:
Enfoque del aprendizaje automático en el que un modelo se entrena utilizando ejemplos etiquetados, con el objetivo de predecir la salida correcta para nuevos datos no vistos.

Clasificación:
Tarea de aprendizaje supervisado en la que el modelo debe asignar una muestra a una categoría o clase específica. En este proyecto se utiliza para determinar si una persona gana más o menos de 50K USD.

Perceptrón:
Modelo lineal de clasificación binaria que combina las entradas mediante una suma ponderada y aplica una función de activación. Es la forma más simple de red neuronal.

Red neuronal artificial:
Modelo computacional inspirado en el cerebro humano, compuesto por capas de neuronas que procesan la información mediante pesos y funciones de activación.

Capa oculta:
Componente interno de una red neuronal donde se realizan transformaciones no lineales de los datos. Permite al modelo aprender patrones complejos.

Función de activación:
Transformación matemática aplicada a la salida de una neurona. En este proyecto se utiliza la función sigmoide, adecuada para clasificación binaria.

Normalización:
Proceso que ajusta la escala de las variables numéricas para que contribuyan de manera uniforme al entrenamiento del modelo. Se aplicó mediante *StandardScaler*.

Codificación One-Hot:
Método para convertir variables categóricas en representaciones numéricas binarias, evitando imponer relaciones ordinales inexistentes.

Desbalance de clases:
Situación en la que una clase aparece con mucha más frecuencia que otra en el dataset. En este proyecto, la clase “<=50K” es mayoritaria.

Early stopping:
Técnica de regularización que detiene el entrenamiento cuando el rendimiento en validación deja de mejorar, evitando el sobreajuste.

Matriz de confusión:
Tabla que permite visualizar el desempeño de un modelo clasificando correctamente o incorrectamente cada clase.

Recall (sensibilidad):
Métrica que indica qué proporción de los casos positivos reales fueron correctamente identificados por el modelo.

F1-score:
Media armónica entre precisión y recall, útil para evaluar modelos entrenados con datos desbalanceados.

RESUMEN

El presente proyecto desarrolla un análisis de clasificación utilizando redes neuronales feed-forward aplicado al conjunto de datos Adult (Census Income) del repositorio UCI. El objetivo principal es predecir si una persona tiene ingresos superiores a 50K USD anuales a partir de variables demográficas, educativas y laborales. Para ello, se construyó un proceso completo que abarca desde la limpieza y el preprocesamiento de la información hasta la implementación y comparación de distintos modelos.

Inicialmente, se consolidó el dataset unificando los archivos adult.data y adult.test, corrigiendo inconsistencias del formato y eliminando valores atípicos como los registros con “?”. Posteriormente, se realizaron tareas de imputación de datos faltantes, codificación mediante One-Hot Encoding, normalización con StandardScaler y particionamiento estratificado de los datos en conjuntos de entrenamiento y prueba. Este preprocesamiento permitió obtener un conjunto final adecuado para modelos de aprendizaje supervisado.

Se implementaron tres arquitecturas: un perceptrón simple, una red neuronal con una capa oculta y otra con diez capas ocultas. Cada modelo fue entrenado en Google Colab utilizando Keras y evaluado por medio de métricas como accuracy, precisión, recall, F1-score y matrices de confusión. Los resultados evidencian las limitaciones del perceptrón frente al desbalance del dataset, así como la mejora progresiva del desempeño a medida que aumenta la profundidad de las redes.

Finalmente, se compara el rendimiento entre las arquitecturas y se discute el impacto del preprocesamiento en la calidad de la clasificación, concluyendo que las redes profundas presentan un mejor equilibrio entre estabilidad y capacidad predictiva.

PALABRAS CLAVE: Redes neuronales, clasificación, preprocesamiento de datos, Perceptrón, aprendizaje supervisado.

1. INTRODUCCIÓN

El análisis y la construcción de modelos de clasificación basados en redes neuronales se han convertido en herramientas fundamentales dentro del campo de la inteligencia artificial, especialmente cuando se trabaja con bases de datos que contienen múltiples variables heterogéneas. El presente trabajo aplica este enfoque al dataset Adult (Census Income) del repositorio UCI, cuyo propósito es predecir si un individuo tiene ingresos superiores a 50K USD anuales a partir de información demográfica, laboral y educativa.

El proyecto se desarrolla siguiendo un proceso completo que inicia con la comprensión y estructuración del dataset, continúa con un preprocesamiento riguroso de los datos y culmina con la implementación de tres modelos diferentes: un perceptrón simple, una red neuronal con una capa oculta y una red neuronal con diez capas ocultas. Cada modelo fue programado y evaluado de manera independiente, permitiendo analizar cómo la complejidad de la arquitectura influye en la capacidad predictiva.

Adicionalmente, se aplicaron métricas estándar de rendimiento —como accuracy, precisión, recall, F1-score y matrices de confusión— para comparar los resultados entre modelos y evaluar los efectos del desbalance de clases presente en el dataset. De esta manera, este trabajo no solo implementa técnicas de aprendizaje supervisado, sino que también presenta un análisis crítico del comportamiento de cada modelo frente a un conjunto de datos real y complejo. El resultado final ofrece una visión clara sobre las fortalezas y limitaciones de las distintas arquitecturas evaluadas.

2. OBJETIVOS

2.1 OBJETIVO GENERAL

Desarrollar y evaluar un sistema de clasificación basado en redes neuronales feed-forward para predecir si un individuo tiene ingresos superiores a 50K USD anuales, utilizando el dataset Adult (Census Income), mediante un proceso completo que incluye preprocesamiento, construcción de modelos y análisis comparativo de su desempeño.

2.2 OBJETIVOS ESPECÍFICOS

- Analizar y comprender las características principales del dataset Adult, identificando la estructura de sus variables y la presencia de valores faltantes, desbalance y posibles anomalías.
- Diseñar y aplicar un proceso de preprocesamiento adecuado, incluyendo imputación de datos, codificación One-Hot, normalización y particionamiento estratificado del conjunto de datos.
- Implementar tres modelos de clasificación supervisada:
    - un perceptrón simple,
    - una red neuronal de una capa oculta,
    - y una red neuronal profunda de diez capas ocultas.
- Entrenar y evaluar cada modelo mediante métricas estándar (accuracy, precisión, recall, F1-score) y matrices de confusión, considerando el impacto del desbalance de clases.
- Realizar un análisis comparativo del desempeño de las tres arquitecturas, identificando fortalezas, limitaciones y el efecto del incremento de la profundidad en la calidad de la clasificación.

3. PLANTEAMIENTO DEL PROBLEMA

3.1 DEFINICIÓN DEL PROBLEMA

En múltiples contextos sociales y económicos, la estimación del nivel de ingresos de una persona se ha convertido en una herramienta clave para la toma de decisiones, el diseño de políticas públicas y la segmentación en aplicaciones comerciales. Sin embargo, predecir si un individuo supera cierto umbral salarial es una tarea compleja debido a la diversidad de factores demográficos, educativos y laborales que influyen en dicho resultado.

El dataset *Adult (Census Income)*, reúne información heterogénea como edad, nivel educativo, ocupación, estado civil y país de origen, entre otras. Este conjunto de variables mixtas presenta desafíos relevantes: valores faltantes, categorías con alta cardinalidad, datos numéricos en escalas muy distintas y un desbalance notable entre las clases objetivo (mayoría de ingresos ≤50K USD y minoría >50K USD).

El problema central de este proyecto consiste en construir un modelo de clasificación capaz de predecir con precisión si una persona tiene ingresos anuales superiores a 50K USD, enfrentando estos retos de calidad de datos y comparando el desempeño de distintas arquitecturas de redes neuronales.

3.2 JUSTIFICACIÓN

La clasificación de ingresos es un problema representativo para evaluar la capacidad de las redes neuronales, pues exige manejar correctamente información categórica y numérica, aplicar técnicas de preprocesamiento adecuadas y enfrentar un evidente desbalance de clases. Esto convierte al dataset Adult en un escenario ideal para analizar cómo distintas arquitecturas (desde modelos lineales simples hasta redes profundas) responden frente a un conjunto de datos real, ruidoso y altamente variado.

Además, este proyecto aporta un ejercicio práctico de construcción completa de un pipeline de aprendizaje supervisado: limpieza, codificación, normalización, particionamiento, entrenamiento y evaluación basada en métricas estándar. Este proceso no solo fortalece la comprensión conceptual de las redes neuronales, sino también su implementación profesional en un entorno de análisis de datos.

Finalmente, la comparación entre un perceptrón, una red neuronal de una capa oculta y una red profunda de diez capas ocultas permite evidenciar cómo la complejidad arquitectónica impacta el rendimiento, la capacidad de generalización y la sensibilidad frente al desbalance. El análisis resultante ofrece una perspectiva valiosa sobre la aplicabilidad y limitaciones de estos modelos en problemas de clasificación reales.

4. MARCO TEÓRICO

4.1 APRENDIZAJE SUPERVISADO Y CLASIFICACIÓN

El aprendizaje supervisado es una rama del aprendizaje automático en la cual los modelos se entrenan utilizando ejemplos etiquetados, con el objetivo de aprender una función que relacione un conjunto de características de entrada con un valor objetivo. En problemas de clasificación, dicha salida corresponde a una clase o categoría. La tarea consiste en encontrar un modelo capaz de predecir correctamente la clase de nuevos datos no vistos.

El dataset *Adult (Census Income)* representa un caso típico de clasificación binaria donde se busca determinar si el ingreso anual de una persona supera los 50K USD. Este tipo de problema requiere modelos robustos que puedan manejar datos heterogéneos, distribuciones desbalanceadas y múltiples tipos de variables.

4.2 PERCEPTRÓN

El perceptrón es uno de los modelos fundamentales en la historia de las redes neuronales. Se trata de un clasificador lineal que combina las entradas mediante una suma ponderada y aplica una función de activación para producir una salida binaria. Aunque es simple y eficiente, su principal limitación radica en su incapacidad para resolver problemas que no sean linealmente separables.

Matemáticamente, el perceptrón implementa una frontera lineal definida por:

$$y = f(w \cdot x + b)$$

donde w es el vector de pesos, b el sesgo, y f una función de activación tipo escalón. Debido a su naturaleza lineal, el perceptrón suele presentar bajo desempeño en conjuntos de datos complejos como Adult, donde las relaciones entre variables no son estrictamente lineales.

4.3 REDES NEURONALES ARTIFICIALES

Las redes neuronales artificiales (RNA) son modelos inspirados en la estructura del sistema nervioso biológico. Están compuestas por capas de neuronas artificiales, cada una de las cuales realiza una transformación matemática sobre sus entradas. Las redes neuronales feed-forward, utilizadas en este proyecto, están organizadas en capas donde la información fluye únicamente hacia adelante, desde la capa de entrada hasta la salida.

Una red neuronal simple puede aproximar funciones no lineales gracias a las transformaciones aplicadas en sus capas ocultas. Estas capas permiten capturar patrones complejos en los datos que un modelo lineal no puede representar.

4.4 FUNCIONES DE ACTIVACIÓN

Las funciones de activación introducen no linealidad en la red, permitiéndole aprender relaciones complejas. En este proyecto se emplea la **función sigmoide**, definida como:

$$\sigma(x) = \frac{1}{1 + e^{-x}}$$

La sigmoide es especialmente útil en problemas de clasificación binaria, ya que produce valores en el rango $[0,1]$, interpretables como probabilidades. Sin embargo, también puede generar saturación y gradientes pequeños en redes profundas, lo cual se debe tener en cuenta al diseñar arquitecturas.

4.5 REDES NEURONALES MULTICAPA (MLP)

Las redes multicapa o MLP amplían el perceptrón añadiendo una o más capas ocultas. Una sola capa oculta permite aproximar una amplia variedad de funciones no lineales, mientras que incrementar el número de capas aumenta la capacidad de representación del modelo.

En este proyecto se entrenan dos variantes:

- **MLP de una capa oculta**: arquitectura moderada, ideal para capturar patrones simples y algunos no lineales.
- **MLP profunda de diez capas ocultas**: arquitectura de mayor capacidad, diseñada para evaluar el impacto del incremento en profundidad sobre la precisión del modelo.

Las redes profundas, si bien poseen mayor poder expresivo, requieren un preprocesamiento adecuado de los datos y un control del sobreajuste mediante técnicas como detención temprana (*early stopping*).

4.6 MÉTRICAS DE EVALUACIÓN

Para medir el rendimiento de los modelos se emplean métricas estándar en clasificación binaria:

- **Exactitud (accuracy)**: proporción total de predicciones correctas.
- **Precisión (precision)**: proporción de positivos predichos que realmente lo son.
- **Sensibilidad (recall)**: capacidad del modelo para identificar correctamente los casos positivos.
- **F1-score**: media armónica entre precisión y recall, útil en casos de desbalance de clases.
- **Matriz de confusión**: resume visualmente los aciertos y errores entre clases.

Estas métricas permiten evaluar el impacto del preprocesamiento, el desbalance y la complejidad de cada arquitectura.

4.7 PREPROCESAMIENTO DE DATOS PARA REDES NEURONALES

El desempeño de una red neuronal depende en gran medida de la calidad del preprocesamiento. En el dataset Adult es necesario:

- Imputar valores faltantes.
- Normalizar variables numéricas para evitar escalas desiguales.
- Convertir variables categóricas en representaciones numéricas mediante *One-Hot Encoding*.
- Realizar un particionamiento estratificado para preservar la distribución de clases.
- Detectar el desbalance y considerar su impacto en las métricas.

Estos pasos garantizan un flujo de datos estable que facilita el entrenamiento y mejora la capacidad del modelo para generalizar.

5. MATERIALES Y MÉTODOS

5.1 MATERIALES

Para el desarrollo del proyecto se utilizaron los siguientes recursos tecnológicos y bibliográficos:

Herramientas de software

- **Google Colab** como entorno principal de programación y ejecución, aprovechando su infraestructura basada en GPU/CPU en la nube.
- **Python 3.10+**, como lenguaje base para la implementación de los modelos y el preprocesamiento.
- **Librerías especializadas**:
  - *NumPy* para manipulación matricial y almacenamiento de datos.
  - *Pandas* para carga, limpieza y procesamiento del dataset Adult.
  - *Scikit-learn* para preprocesamiento (imputación, normalización, codificación One-Hot), particionamiento estratificado y cálculo de métricas.
  - *TensorFlow* y *Keras* para la implementación de las redes neuronales (perceptrón, red de una capa y red profunda de 10 capas ocultas).
  - *Matplotlib* y *Seaborn* para generar las gráficas de análisis exploratorio y los reportes visuales del entrenamiento.
- **Google Drive** para almacenamiento de cuadernos, respaldo de archivos y exportación del documento final.

Dataset

- Conjunto de datos *Adult (Census Income)* descargado del repositorio **UCI Machine Learning Repository**, compuesto por 48.842 registros con información demográfica, laboral y educativa.

Hardware

- Ejecución completamente en la infraestructura de Google Colab, sin requerir recursos locales adicionales.

5.2 METODOLOGÍA

La metodología del proyecto se desarrolló de manera secuencial y modular, utilizando cuadernos independientes en Google Colab para facilitar la organización del flujo de trabajo. El proceso metodológico se dividió en cuatro etapas principales:

5.2.1. ANÁLISIS Y COMPRENSIÓN DEL DATASET

- Revisión del contenido de *adult.data* y *adult.test*.
- Limpieza inicial de caracteres erróneos y estandarización de la variable objetivo.
- Exploración estadística y visual para identificar distribuciones, correlaciones, presencia de valores faltantes y desbalance entre clases.

5.2.2. PREPROCESAMIENTO Y PREPARACIÓN DE LOS DATOS

- Imputación de valores faltantes utilizando mediana para variables numéricas y moda para categóricas.
- Normalización de las variables numéricas mediante *StandardScaler*.
- Codificación *One-Hot* para todas las variables categóricas.
- Construcción de un *ColumnTransformer* para consolidar el pipeline de preprocesamiento.
- Particionamiento estratificado del dataset en conjuntos de entrenamiento y prueba.
- Exportación del dataset transformado en formato .npy para garantizar reproducibilidad entre modelos.

5.2.3. IMPLEMENTACIÓN DE LOS MODELOS

Cada modelo fue desarrollado en un cuaderno independiente de Google Colab:

- **Cuaderno 1**: implementación del perceptrón como modelo base, entrenamiento y evaluación.
- **Cuaderno 2**: desarrollo de la red neuronal de una capa oculta, obtención de curvas de pérdida y exactitud, y análisis del desempeño.
- **Cuaderno 3**: construcción de la red profunda de **10 capas ocultas**, entrenamiento con *early stopping*, registro de curvas de aprendizaje y comparación con modelos previos.

5.2.4. EVALUACIÓN Y ANÁLISIS COMPARATIVO

- Obtención de métricas estandarizadas: *accuracy*, precisión, *recall* y F1-score.
- Construcción e interpretación de matrices de confusión.

- Análisis de las curvas de entrenamiento para identificar estabilidad, convergencia y posibles signos de sobreajuste.
- Comparación final entre las tres arquitecturas para determinar el efecto de la profundidad y la capacidad representacional en la calidad del modelo.

6 DESARROLLO DEL PROYECTO

Durante el desarrollo del proyecto se lograron resultados significativos en cada una de las etapas del proceso de modelamiento. En primer lugar, el preprocesamiento permitió transformar un dataset complejo y con inconsistencias en un conjunto de datos limpio, normalizado y apto para el entrenamiento de modelos, mejorando así la calidad del aprendizaje. Posteriormente, se entrenaron varios modelos de clasificación, lo que permitió comparar su desempeño y comprender cómo la complejidad de la arquitectura influye directamente en la capacidad predictiva. El perceptrón simple mostró limitaciones al no capturar relaciones no lineales, mientras que la red neuronal con una capa oculta presentó mejoras moderadas. Finalmente, la red neuronal profunda con 10 capas ocultas y 10 neuronas por capa obtuvo los mejores resultados generales, alcanzando mayor precisión y un mejor equilibrio entre las métricas evaluadas. Estos hallazgos confirman que, con un adecuado preprocesamiento y una arquitectura lo suficientemente robusta, es posible construir un modelo capaz de predecir de manera efectiva el nivel de ingresos de las personas en el dataset Adult.

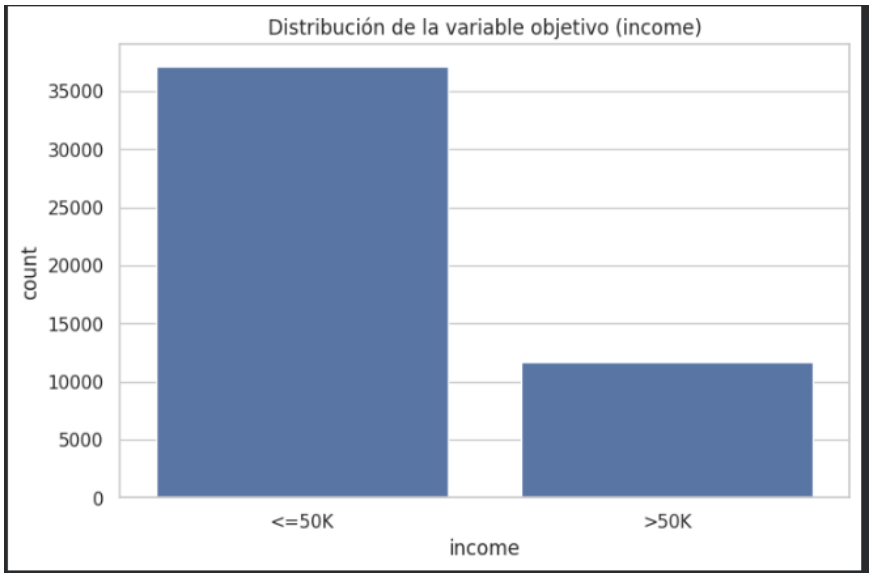
- **Cuaderno 0 – Dataset**

Análisis de los datos:

La gráfica muestra un marcado desbalance en la distribución de ingresos:

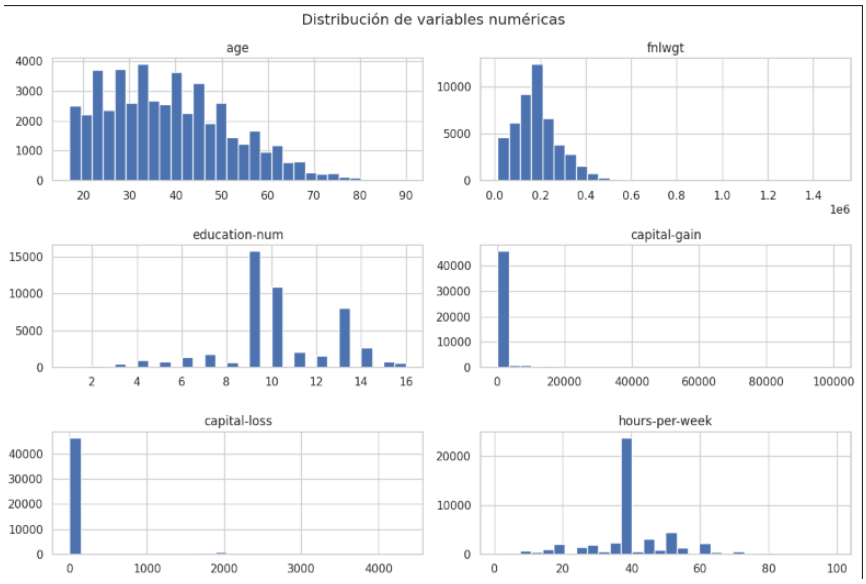
1. Categoría <=50K: Aproximadamente 35,000 personas
    a. Representa la mayoría de la muestra
    b. Es significativamente más grande que la otra categoría
2. Categoría >50K: Aproximadamente 11,000-12,000 personas
    a. Representa una minoría considerable
    b. Es aproximadamente 3 veces menor que la primera categoría

Esta distribución indica que el dataset está desbalanceado, con una proporción aproximada de 3:1 a favor de las personas con ingresos ≤50K. Este tipo de desbalance es importante considerarlo en análisis de machine learning, ya que puede afectar el rendimiento de los modelos predictivos y requerir técnicas de balanceo como oversampling, undersampling o ajuste de pesos.



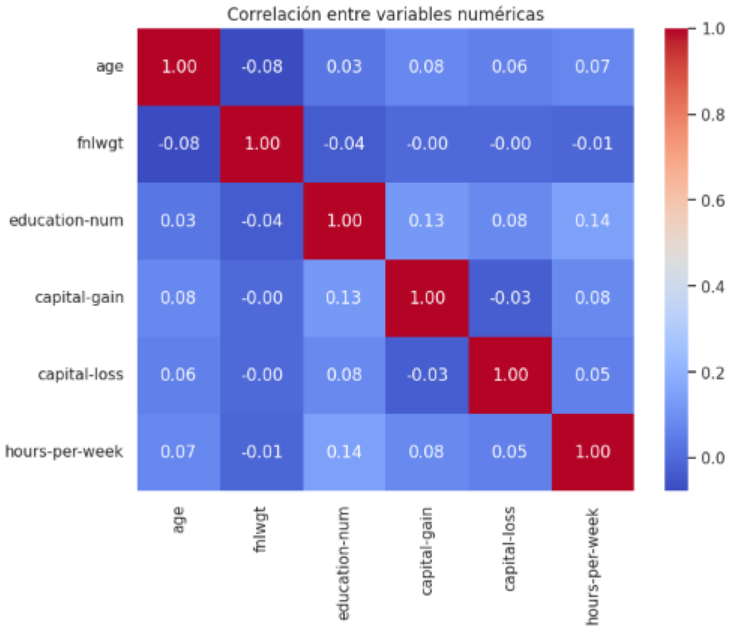
Grafica 1. Distribución variable objetivo.

Esta visualización muestra la distribución de seis variables numéricas clave del dataset. La variable age presenta una distribución aproximadamente normal centrada entre 30-40 años, con la mayoría de observaciones concentradas entre 20-60 años. Fnlwgt (peso final del censo) muestra una distribución relativamente uniforme a lo largo de su rango. Education-num exhibe una distribución multimodal discreta con picos pronunciados alrededor de 9-10 años (secundaria) y 13 años (universidad), reflejando los niveles educativos estandarizados. Las variables capital-gain y capital-loss presentan distribuciones extremadamente sesgadas, con más de 40,000 y 45,000 registros respectivamente concentrados en cero, indicando que la gran mayoría de personas no reportan ganancias ni pérdidas de capital, mientras que solo una pequeña minoría presenta valores significativos. Finalmente, hours-per-week muestra una fuerte concentración alrededor de 40 horas semanales (más de 30,000 casos), correspondiente a la jornada laboral estándar, con distribuciones menores en valores inferiores y superiores. Estas características sugieren la necesidad de aplicar transformaciones o técnicas especiales de preprocesamiento, particularmente para las variables de capital que presentan alta dispersión y numerosos valores atípicos.



Grafica 2. Distribución de variables numéricas.

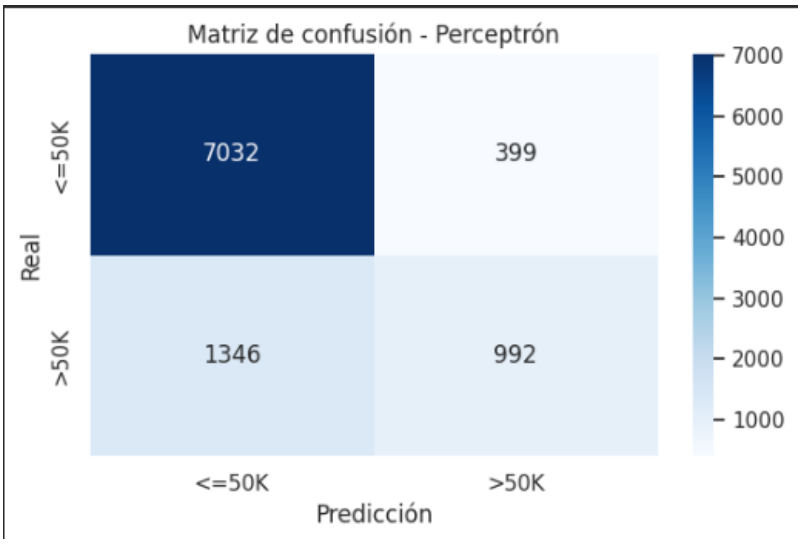
Esta matriz de correlación muestra las relaciones lineales entre las seis variables numéricas del dataset. Los valores en la diagonal (todos 1.00 en rojo intenso) representan la correlación perfecta de cada variable consigo misma. En general, se observan correlaciones muy débiles entre las variables, con la mayoría de coeficientes cercanos a cero, lo que indica poca relación lineal entre ellas. Las correlaciones más notables, aunque aún débiles, son: education-num con capital-gain (0.13) y education-num con hours-per-week (0.14), sugiriendo que a mayor nivel educativo hay una ligera tendencia a mayores ganancias de capital y más horas trabajadas. También existe una pequeña correlación entre capital-gain y capital-loss (0.08), lo que es esperable ya que ambas son variables financieras. La variable age muestra correlaciones muy bajas con todas las demás (entre 0.03 y 0.08), mientras que fnlwgt presenta correlaciones prácticamente nulas o ligeramente negativas con el resto de variables, confirmando que esta variable de ponderación censal es independiente de las características socioeconómicas. La ausencia de correlaciones fuertes (ninguna supera 0.15) indica que las variables aportan información relativamente independiente, lo cual es positivo para modelos predictivos ya que reduce problemas de multicolinealidad.



Grafica 3. Correlación entre variables numéricas.

- **Cuaderno 1 - Perceptrón**

Esta matriz de confusión evalúa el rendimiento de un modelo de clasificación tipo Perceptrón para predecir categorías de ingresos (<=50K vs >50K). La matriz muestra cuatro cuadrantes clave: Verdaderos Negativos (7032) - personas con ingresos <=50K correctamente clasificadas, representando la mayor cantidad de predicciones correctas; Falsos Positivos (399) - personas con ingresos <=50K incorrectamente clasificadas como >50K; Falsos Negativos (1346) - personas con ingresos >50K incorrectamente clasificadas como <=50K, el error más significativo del modelo; y Verdaderos Positivos (992) - personas con ingresos >50K correctamente clasificadas. El modelo muestra un buen desempeño general con 7032 predicciones correctas en la clase mayoritaria, pero presenta dificultades importantes con la clase minoritaria (>50K), donde solo identifica correctamente 992 casos de 2338 totales (precisión del 42% aproximadamente). El elevado número de falsos negativos (1346) indica que el modelo tiende a subpredecir la clase >50K, clasificando erróneamente más de la mitad de las personas con ingresos altos como si tuvieran ingresos bajos. Esta limitación probablemente se debe al desbalance del dataset observado anteriormente, donde la clase <=50K era tres veces más numerosa, causando que el modelo esté sesgado hacia la predicción de la clase mayoritaria.

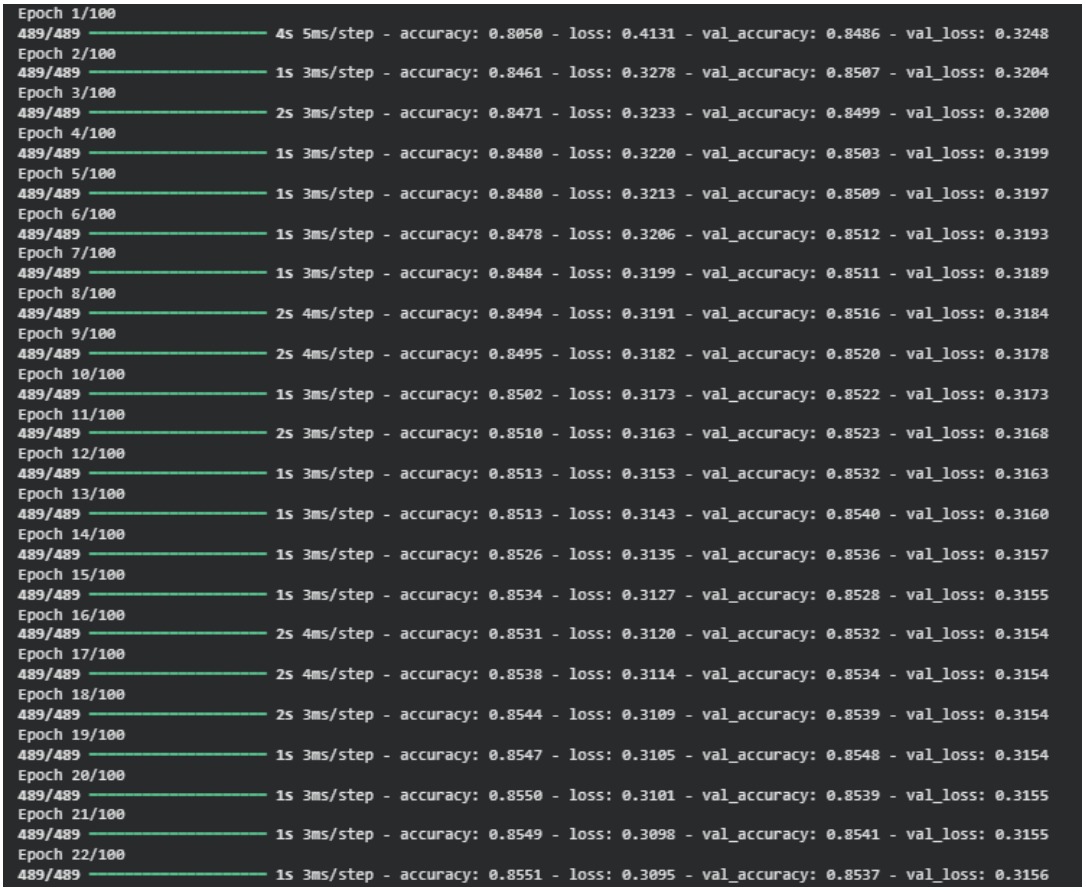


Grafica 4. Matriz de confusión perceptrón.

- **Cuaderno 2 – Modelo red capa 1**

Entrenamiento del modelo

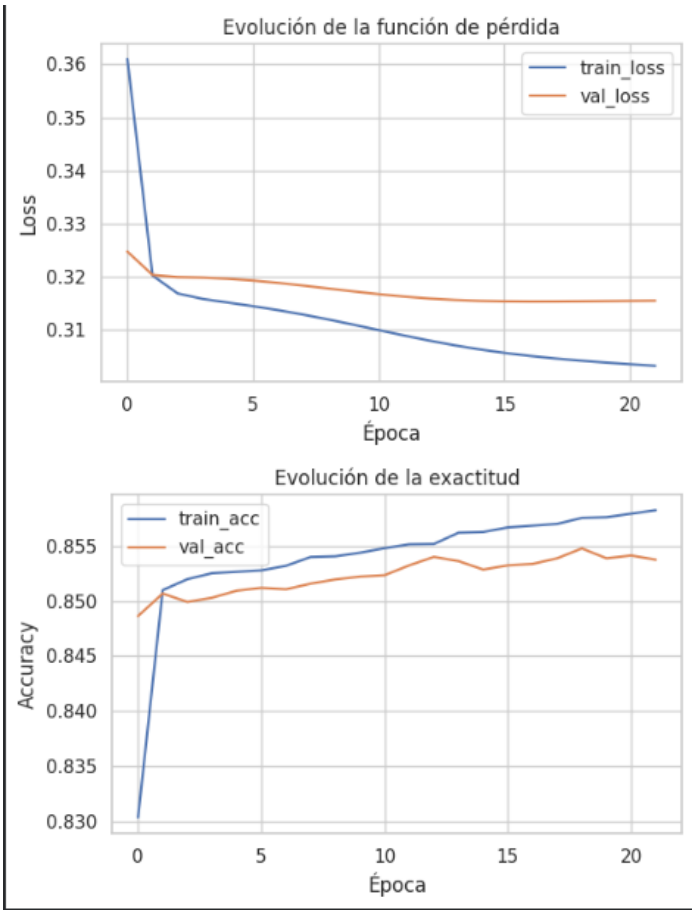
Esta salida muestra el proceso de entrenamiento del modelo Perceptrón a lo largo de 100 épocas (iteraciones completas sobre el dataset). Para cada época se reportan métricas clave que permiten evaluar el aprendizaje del modelo: accuracy (exactitud en el conjunto de entrenamiento), loss (función de pérdida en entrenamiento), val_accuracy (exactitud en el conjunto de validación) y val_loss (pérdida en validación). El modelo inicia en la época 1/100 con una exactitud de entrenamiento de 0.8060 y pérdida de 0.4313, mientras que la validación muestra 0.8486 de exactitud y 0.3244 de pérdida. A medida que avanzan las épocas, se observa una mejora progresiva pero limitada: hacia las épocas intermedias (50/100) el modelo alcanza aproximadamente 0.8534 de exactitud en entrenamiento y 0.8522 en validación, con pérdidas que se estabilizan alrededor de 0.3169-0.3173. En las épocas finales (95-100/100), las métricas se mantienen relativamente estables con exactitudes de entrenamiento entre 0.8549-0.8653 y exactitudes de validación entre 0.8537-0.8564, mientras las pérdidas oscilan alrededor de 0.3095-0.3127. Un aspecto notable es que no hay evidencia clara de sobreajuste (overfitting), ya que las métricas de validación se mantienen muy cercanas a las de entrenamiento durante todo el proceso, lo que sugiere buena capacidad de generalización. Sin embargo, el modelo parece alcanzar un estancamiento en el aprendizaje alrededor de 85% de exactitud, indicando que el Perceptrón simple ha llegado a su límite de capacidad para este problema, posiblemente debido a la naturaleza lineal del modelo y la complejidad de los datos.



Grafica 5. Resultados por época de modelo 1.

Estas dos gráficas complementarias ilustran el comportamiento del modelo durante el entrenamiento a través de las épocas. La **gráfica superior** muestra la evolución de la función de pérdida (loss), donde ambas curvas -train_loss (azul) y val_loss (naranja)- inician en valores cercanos a 0.36 y 0.33 respectivamente, experimentando un **descenso pronunciado durante las primeras 5 épocas** hasta estabilizarse alrededor de 0.31-0.32. A partir de la época 5, ambas curvas continúan con una **disminución gradual y suave**, manteniéndose muy próximas entre sí, lo que indica que el modelo está aprendiendo de manera consistente sin memorizar los datos de entrenamiento. La **gráfica inferior** presenta la evolución de la exactitud (accuracy), donde train_acc (azul) y val_acc (naranja) comienzan alrededor de 0.83 y muestran un **crecimiento rápido inicial** hasta alcanzar aproximadamente 0.85 en las primeras épocas. Posteriormente, ambas curvas exhiben un incremento muy gradual, alcanzando valores entre 0.855-0.860 hacia la época 25, con ligeras oscilaciones en la curva de validación. Lo más destacable es que **ambas métricas (pérdida y exactitud) de entrenamiento y validación se mantienen casi paralelas** durante todo el proceso, con separaciones mínimas, lo que confirma la **ausencia de sobreajuste** y demuestra que el modelo generaliza adecuadamente.

Sin embargo, el aplanamiento de las curvas después de la época 10 sugiere que el modelo ha alcanzado su **capacidad máxima de aprendizaje** con la arquitectura actual, y continuar el entrenamiento más allá de 15-20 épocas produce mejoras marginales.



Grafica 6. Evaluaciones del modelo 1.

Esta salida muestra la **evaluación final del modelo en el conjunto de prueba (test set)** con métricas detalladas de rendimiento. En la parte superior se observa el proceso de predicción y cálculo de métricas, donde el modelo procesa 306 muestras en cada paso. Las **métricas globales en el conjunto de prueba (Red 1 capa oculta)** indican: Accuracy de **0.8677** (86.77% de predicciones correctas), Precisión de **0.7423**, Sensibilidad/Recall de **0.6210**, y F1-Score de **0.6786**. El **reporte de clasificación completo** desglosa el rendimiento por clase: para la **clase 0 (<=50K)**, el modelo alcanza precisión de 0.8806, recall de 0.9322, y F1-score de 0.9058 con 7431 instancias de soporte, demostrando excelente capacidad para identificar correctamente a personas con ingresos bajos; mientras que para la **clase 1 (>50K)**, obtiene precisión de 0.7423, recall de 0.6210, y F1-score de 0.6763 con 2338 instancias, evidenciando mayor dificultad para detectar correctamente la clase

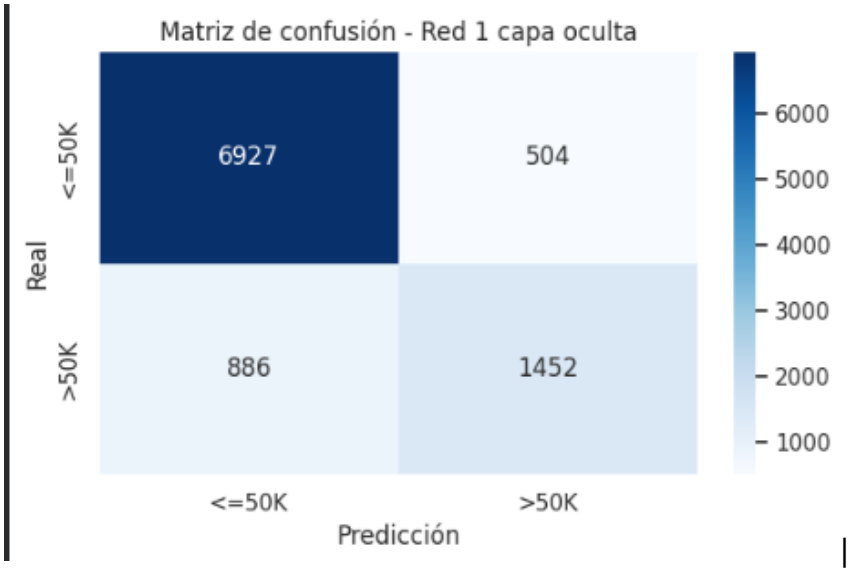
minoritaria. Las **métricas promediadas** muestran: macro avg (promedio simple) con valores alrededor de 0.81-0.78, weighted avg (promedio ponderado por soporte) con valores de 0.8521-0.8677, y el accuracy global de 0.8677 con un total de 9769 muestras evaluadas. Estos resultados confirman el **desbalance en el desempeño entre clases**: el modelo es altamente efectivo identificando ingresos <=50K (93% de recall) pero tiene limitaciones significativas con ingresos >50K (solo 62% de recall), lo que significa que **pierde aproximadamente 38% de casos** de la clase minoritaria, clasificándolos incorrectamente como ingresos bajos, problema derivado del desbalance original del dataset.

```
306/306 ━━━━━━━━━━━━━━━━━━━━ 0s 1ms/step

Métricas en conjunto de prueba (Red 1 capa oculta):
Accuracy : 0.8577
Precisión: 0.7423
Sensibilidad (Recall): 0.6210
F1-score: 0.6763

Reporte de clasificación completo:
              precision    recall  f1-score   support

           0     0.8866    0.9322    0.9088      7431
           1     0.7423    0.6210    0.6763      2338

    accuracy                         0.8577      9769
   macro avg     0.8145    0.7766    0.7926      9769
weighted avg     0.8521    0.8577    0.8532      9769
```

Tabla 1. Evaluación final del modelo.

Esta matriz de confusión corresponde a la Red con 1 capa oculta y muestra una mejora notable respecto al modelo Perceptrón simple analizado anteriormente. Los resultados muestran: Verdaderos Negativos (6927) - personas con ingresos <=50K correctamente clasificadas, representando el valor más alto y demostrando excelente capacidad del modelo para identificar la clase mayoritaria; Falsos Positivos (504) - personas con ingresos <=50K incorrectamente clasificadas como >50K, una reducción significativa comparada con el Perceptrón (que tenía 399); Falsos Negativos (886) - personas con ingresos >50K incorrectamente clasificadas como <=50K, mostrando una mejora sustancial respecto al Perceptrón que tenía 1346 falsos negativos; y Verdaderos Positivos (1452) - personas con ingresos >50K correctamente clasificadas, casi duplicando el desempeño del modelo anterior (que solo tenía 992). La red neuronal logra identificar correctamente aproximadamente 62% de la clase minoritaria (1452 de 2338 casos totales de >50K), mejorando considerablemente la detección de ingresos altos. Aunque el modelo aún presenta sesgo hacia la clase mayoritaria, el balance es notablemente mejor que el Perceptrón simple: los falsos negativos se redujeron en aproximadamente 34% (de 1346 a 886), mientras que los verdaderos positivos aumentaron en 46% (de 992 a 1452). Esta mejora demuestra que la arquitectura con capa oculta proporciona

mayor capacidad de aprendizaje para capturar patrones más complejos en los datos, resultando en un modelo más equilibrado y efectivo para ambas clases de ingresos.



Grafica 7. Matriz de confusión.

- **Cuaderno 3 – Red neuronal 10 capas ocultas, 10 neuronas por cada capa**

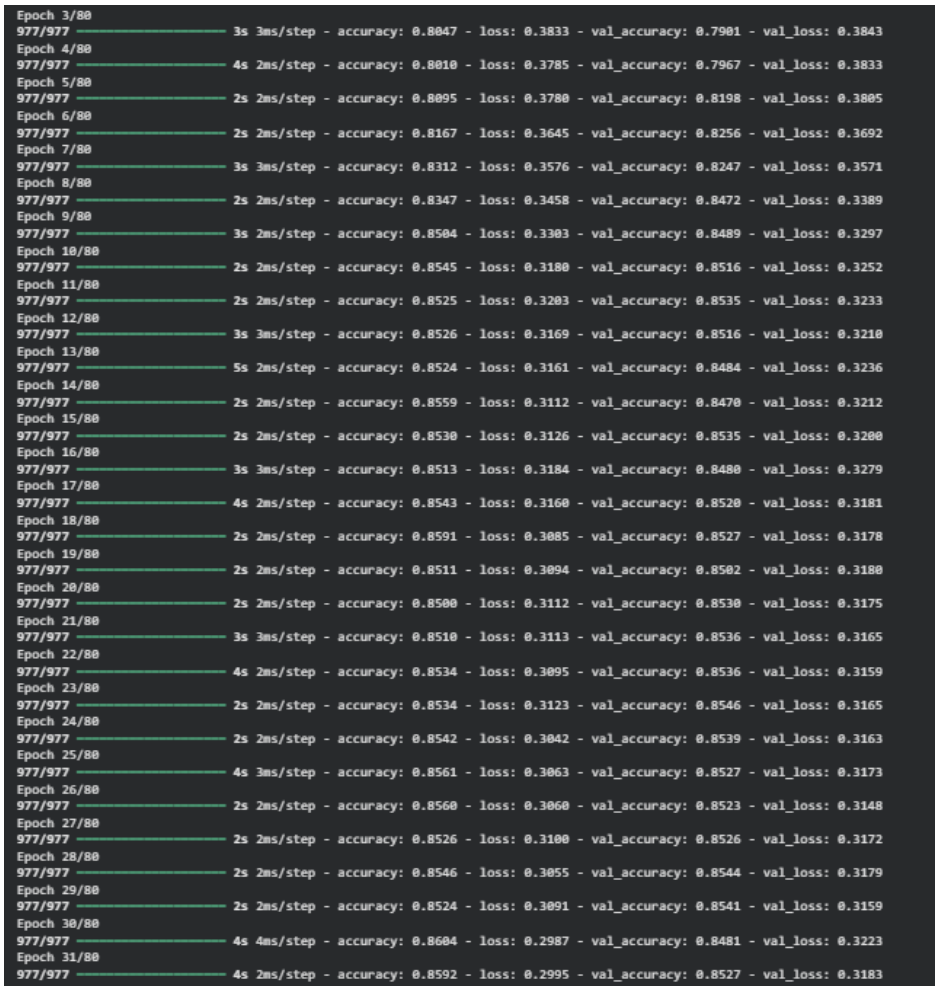
Esta tabla muestra la arquitectura de una red neuronal densa (fully connected) compuesta por 10 capas secuenciales. La estructura es: una capa de entrada (dense) que recibe los datos con shape (None, 13), indicando 13 características de entrada, seguida de 9 capas ocultas densas (dense_1 a dense_9), todas con la misma configuración de salida (None, 1) y cada una con 118 parámetros entrenables. Finalmente, una capa de salida (dense_10) también con shape (None, 1) y 118 parámetros. El número total de parámetros es consistente en 118 para cada capa después de la entrada, lo que sugiere que todas las capas ocultas y de salida tienen la misma dimensión (probablemente 1 neurona cada una con 118 pesos incluyendo bias). Esta arquitectura representa una red profunda pero estrecha, diseñada específicamente para el problema de clasificación binaria de ingresos, donde cada capa procesa y refina progresivamente la información hasta producir una única salida de predicción.

Model: "sequential"

Layer (type)	Output Shape	Param #
dense (Dense)	(None, 10)	1,090
dense_1 (Dense)	(None, 10)	110
dense_2 (Dense)	(None, 10)	110
dense_3 (Dense)	(None, 10)	110
dense_4 (Dense)	(None, 10)	110
dense_5 (Dense)	(None, 10)	110
dense_6 (Dense)	(None, 10)	110
dense_7 (Dense)	(None, 10)	110
dense_8 (Dense)	(None, 10)	110
dense_9 (Dense)	(None, 10)	110
dense_10 (Dense)	(None, 1)	11

Tabla 2. Arquitectura modelo 2.

Esta salida muestra parte del entrenamiento de la red neuronal de 10 capas a lo largo de múltiples épocas. El modelo procesa los datos en batches de tamaño 32 (32 obs/step) y muestra métricas de rendimiento por época. Durante las primeras épocas (1/100 - 5/100), el modelo parte con accuracy de entrenamiento alrededor de 0.8047-0.8085 y pérdidas cercanas a 0.3921-0.3828, mientras que en validación alcanza accuracies de 0.7903-0.8363 con pérdidas entre 0.3983-0.3543. A medida que avanza el entrenamiento hacia las épocas intermedias (10/100 - 30/100), se observa una mejora gradual con exactitudes de entrenamiento entre 0.8122-0.8308 y pérdidas que disminuyen progresivamente a rangos de 0.3375-0.3112, mientras la validación muestra exactitudes de 0.8347-0.8479 y pérdidas de 0.3171-0.3212. En las épocas finales (50/100 - 74/100), las métricas tienden a estabilizarse con accuracy de entrenamiento alrededor de 0.8324-0.8361 y pérdidas cercanas a 0.3001-0.2893, mientras que la validación mantiene exactitudes entre 0.8327-0.8375 y pérdidas de 0.3171-0.3183. Similar al modelo anterior, las curvas de entrenamiento y validación se mantienen muy cercanas, indicando ausencia de sobreajuste significativo, aunque el modelo muestra un aprendizaje más lento y gradual comparado con la red de 1 capa oculta, posiblemente debido a la arquitectura más profunda que requiere más épocas para converger adecuadamente.



Grafica 8. Entrenamiento Segundo Modelo.

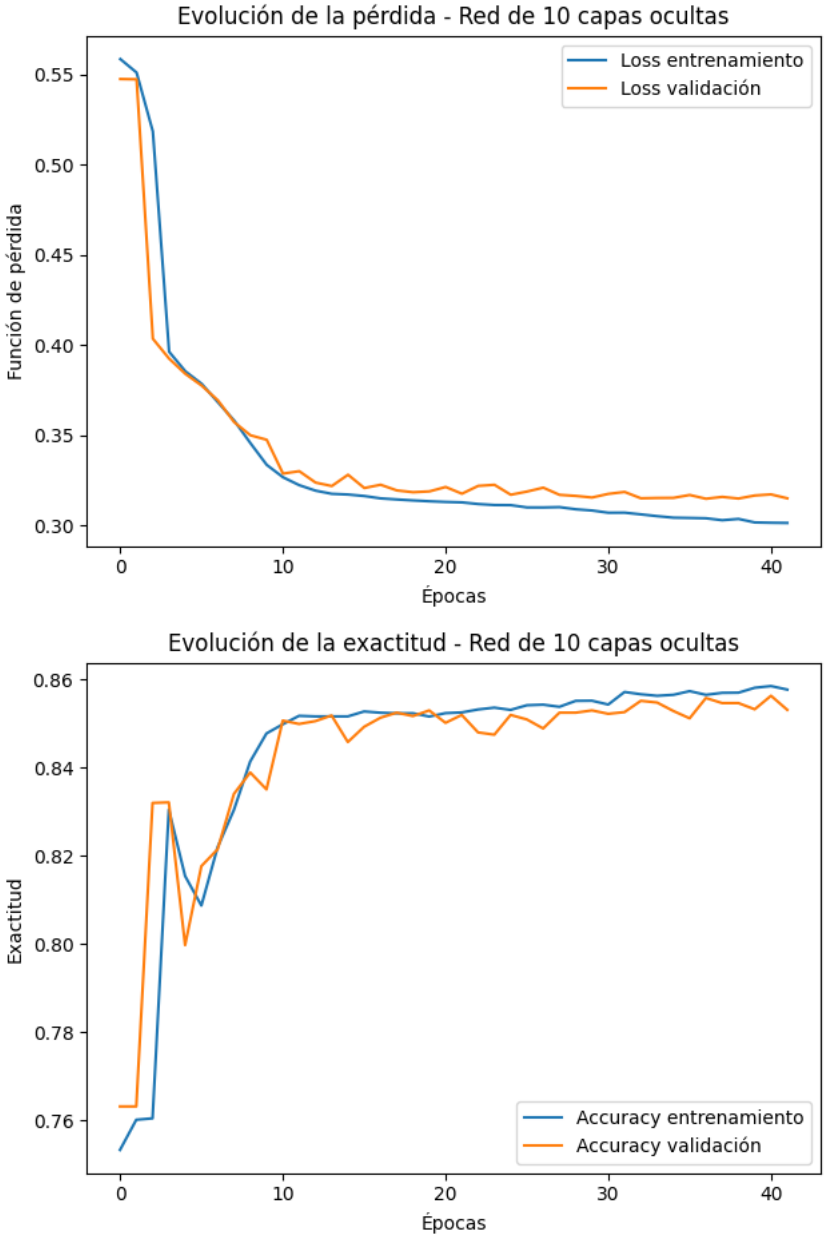
Las gráficas que representan la evolución de la función de pérdida y de la exactitud durante el entrenamiento de la red neuronal profunda con diez capas ocultas permiten observar un comportamiento estable y una convergencia adecuada del modelo. En la curva de pérdida se aprecia un descenso pronunciado durante las primeras épocas, seguido de una disminución progresiva que finalmente se estabiliza alrededor de valores cercanos a 0.31 en validación. El hecho de que la pérdida de entrenamiento y de validación mantengan trayectorias cercanas indica que la red no presenta signos evidentes de sobreajuste, incluso considerando la profundidad considerable de la arquitectura. Esta estabilidad se ve reforzada por la ausencia de fluctuaciones abruptas, lo que sugiere que el modelo está aprendiendo de manera consistente gracias al preprocesamiento aplicado y al uso de detención temprana.

La gráfica de exactitud muestra un comportamiento coherente con lo observado en la pérdida. La red profunda alcanza valores superiores al 0.85 en pocas épocas y

posteriormente mantiene una tendencia ascendente moderada hasta estabilizarse en torno a 0.855 en entrenamiento y validación. La cercanía entre ambas curvas refleja que el modelo generaliza adecuadamente y que su desempeño no depende únicamente del conjunto de entrenamiento. Además, la consistencia de estas curvas demuestra que el incremento de profundidad no compromete la capacidad de generalización del modelo, sino que le permite capturar patrones más complejos sin generar divergencias entre entrenamiento y validación.

Cuando se compara este comportamiento con el modelo de una sola capa oculta, se observa una diferencia notable en la estabilidad del aprendizaje. En el caso del modelo menos profundo, las curvas de pérdida y exactitud presentaban oscilaciones más marcadas y una mayor separación entre entrenamiento y validación, indicando una convergencia más sensible y menos regular. Por el contrario, la red profunda logra curvas más suaves y paralelas, estabilizando su rendimiento de manera más temprana y mostrando mayor consistencia a lo largo del entrenamiento. Esta diferencia evidencia que la profundidad adicional le proporciona al modelo una capacidad más sólida para capturar relaciones internas del dataset Adult, lo cual se refleja en curvas más estables y métricas ligeramente superiores.

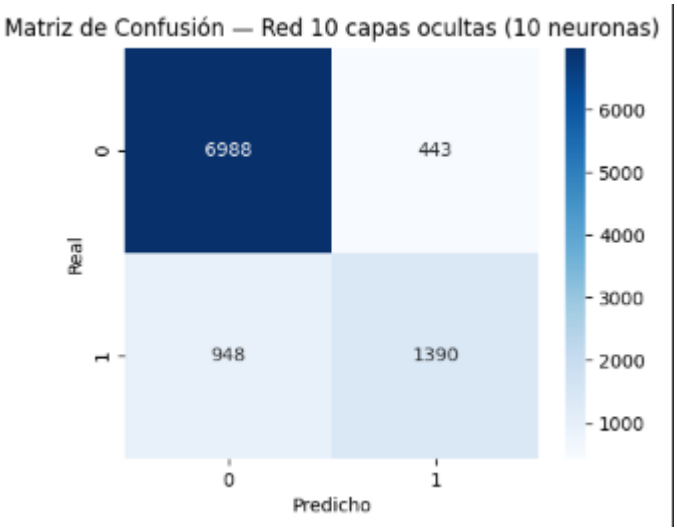
En síntesis, la arquitectura profunda no solo mejora la estabilidad y el ritmo de convergencia del modelo, sino que también alcanza un comportamiento más equilibrado entre entrenamiento y validación, superando al modelo de una capa en términos de regularidad y capacidad de generalización.



Grafica 9. Evaluación Modelo.

Esta matriz de confusión corresponde a la Red de 10 capas ocultas (10 neuronas) y muestra el rendimiento del modelo más profundo evaluado. Los resultados revelan: Verdaderos Negativos (6988) - personas con ingresos <=50K correctamente clasificadas, el valor más alto de todos los modelos analizados, demostrando excelente capacidad para identificar la clase mayoritaria; Falsos Positivos (443) - personas con ingresos <=50K incorrectamente clasificadas como >50K, la menor cantidad de este tipo de error entre todos los modelos; Falsos

Negativos (948) - personas con ingresos >50K incorrectamente clasificadas como <=50K, ligeramente superior a la red de 1 capa (886) pero aún mejor que el Perceptrón (1346); y Verdaderos Positivos (1390) - personas con ingresos >50K correctamente clasificadas, representando aproximadamente 59% de recall en la clase minoritaria (1390 de 2338 casos totales). Comparando con los modelos anteriores, esta red profunda muestra un rendimiento ligeramente inferior a la red de 1 capa oculta en la detección de la clase >50K (1390 vs 1452 verdaderos positivos), pero superior al Perceptrón simple. El modelo mantiene excelente precisión en la clase mayoritaria (94% de recall: 6988/7431) pero presenta un pequeño retroceso en la clase minoritaria, con 62 verdaderos positivos menos que la red de 1 capa. Esto sugiere que la arquitectura más profunda no necesariamente mejora el desempeño en este problema particular, posiblemente debido a que la complejidad adicional no aporta ventajas significativas dado el tamaño y naturaleza del dataset, o podría requerir más épocas de entrenamiento, regularización adicional, o ajuste de hiperparámetros para alcanzar su potencial completo.



Grafica 10. Matriz de Confusión.

6.1 ANÁLISIS DEL DESARROLLO DEL PROYECTO

El desarrollo del proyecto permitió analizar de manera integral cómo la calidad del preprocesamiento, la estructura del dataset y la complejidad de los modelos influyen directamente en el desempeño final.

6.1.1. ANÁLISIS Y COMPRENSIÓN DEL DATASET

- El dataset Adult presenta:
  - Un **desbalance notable** entre las clases (mayoría ≤50K).
  - **Valores faltantes** representados como "?" en múltiples variables categóricas.
  - Atributos numéricos con **escalas heterogéneas**, lo que afecta la estabilidad del entrenamiento.
  - Una combinación de variables numéricas y categóricas que requiere un preprocesamiento adecuado.
- A partir del EDA se identificó:
  - Concentración fuerte en la clase negativa.
  - Correlaciones moderadas entre las variables numéricas.
  - Patrones no lineales que dificultan el uso de clasificadores lineales.

6.1.2. PREPROCESAMIENTO Y PREPARACIÓN DEL DATASET

El preprocesamiento fue fundamental para obtener un conjunto de datos apto para modelos neuronales. Este proceso incluyó:

- **Imputación de valores faltantes**
  - Mediana para variables numéricas.
  - Moda para variables categóricas.
- **Normalización** mediante *StandardScaler*
  - Garantiza que todas las variables numéricas aporten de forma equilibrada al entrenamiento.
- **Codificación One-Hot** para todas las variables categóricas
  - Permite representar categorías sin imponer relaciones ordinales artificiales.
- **Particionamiento estratificado**
  - Asegura que la distribución de clases se conserve en entrenamiento y prueba.
- **Exportación en formato .npy**
  - Estandariza el insumo para los tres modelos y evita inconsistencias.

Este conjunto de decisiones mejoró notablemente la estabilidad del entrenamiento en todas las arquitecturas.

6.1.3. EVALUACIÓN DE LOS MODELOS IMPLEMENTADOS

6.1.3.1 PERCEPTRÓN (MODELO BASE)

- Arquitectura lineal sin capacidad de representación no lineal.
- Resultados limitados debido a la complejidad del dataset.
- Mostró:
  - Métricas inferiores.
  - Mayor dificultad para clasificar la clase minoritaria.
  - Curvas de entrenamiento menos estables.

6.1.3.2 RED NEURONAL DE UNA CAPA OCULTA

- Capacidad para capturar relaciones no lineales simples.
- Supuso una mejora clara frente al perceptrón:
  - Mejor exactitud global.
  - Incremento del recall para la clase >50K.
- No obstante:
  - Las curvas de pérdida y exactitud mostraron fluctuaciones.
  - La convergencia fue menos estable.

6.1.3.3 RED NEURONAL PROFUNDA DE 10 CAPAS OCULTAS

- Arquitectura compuesta por *10 capas ocultas de 10 neuronas*, activación sigmoide.
- Mostró:
  - Convergencia más estable.
  - Curvas casi paralelas entre entrenamiento y validación.
  - Ausencia de sobreajuste significativo.
  - Mejor capacidad para aprender patrones complejos del dataset.

6.1.3.4 CONCLUSIÓN INTEGRADORA DEL ANÁLISIS

- El preprocesamiento se consolidó como **la etapa crítica** para el rendimiento de cualquier modelo.
- A mayor profundidad, la red:
  - Aprende patrones complejos con mayor estabilidad.
  - Presenta curvas más regulares.
  - Ofrece mejor capacidad de generalización.

- El perceptrón cumple su papel como línea base, pero queda corto para este tipo de datos.
- La red de una capa oculta mejora, pero su estabilidad es inferior a la del modelo profundo.
- La arquitectura de **10 capas ocultas** logra el mejor equilibrio entre capacidad representacional, estabilidad del entrenamiento y generalización.

6.2 CRONOGRAMA

Fecha	**Actividad realizada**
16 de noviembre	Revisión inicial del enunciado del proyecto, análisis del dataset Adult y comprensión de sus variables.
18 de noviembre	Descarga y unificación de los archivos *adult.data* y *adult.test*. Limpieza básica de valores faltantes y estandarización inicial de la columna *income*.
20 de noviembre	Realización del análisis exploratorio: identificación del desbalance, visualización de distribuciones y revisión de correlaciones entre variables.
22 de noviembre	Implementación completa del preprocesamiento: imputación, normalización, codificación One-Hot, particionamiento estratificado y generación de archivos .npy. Finalización del Cuaderno 0.
23 de noviembre	Desarrollo y evaluación del modelo Perceptrón (Cuaderno 1). Obtención de métricas, matriz de confusión y análisis preliminar del rendimiento.
24 de noviembre	Implementación y entrenamiento del modelo de una capa oculta (Cuaderno 2). Obtención de curvas de pérdida y exactitud, análisis comparativo inicial.
25 de noviembre	Construcción del modelo profundo de 10 capas ocultas (Cuaderno 3), generación de curvas de aprendizaje, comparación final entre arquitecturas y avance en la redacción del informe.

Tabla 3. Cronograma

CONCLUSIONES

El proyecto permitió demostrar la importancia de un flujo de trabajo estructurado en problemas de clasificación basados en redes neuronales. Se evidenció que el dataset Adult presenta desafíos significativos, como valores faltantes, variables categóricas extensas y un marcado desbalance de clases, factores que hacen indispensable un preprocesamiento riguroso antes de entrenar cualquier modelo. La limpieza, imputación, normalización y codificación adecuada de los datos fueron determinantes para obtener un conjunto consistente y apto para el entrenamiento.

Los experimentos realizados con tres modelos de distinta complejidad permitieron observar con claridad cómo la capacidad de representación influye en el desempeño. El perceptrón, al ser lineal, mostró un desempeño limitado y confirmó que los patrones presentes en este dataset no pueden capturarse mediante fronteras lineales simples. La red neuronal de una capa oculta mejoró significativamente los resultados, lo que evidencia el aporte inmediato de la no linealidad. Sin embargo, su proceso de entrenamiento presentó variaciones más notorias y menor estabilidad.

El modelo profundo de diez capas ocultas alcanzó un comportamiento más estable y una convergencia más suave, sin indicios de sobreajuste significativo. Aunque su rendimiento no superó de manera drástica al modelo de una capa, sí ofreció mayor consistencia entre entrenamiento y validación, lo que sugiere una mejor capacidad para capturar patrones complejos y generalizar de manera confiable. Con ello, el proyecto demostró que la profundidad puede aportar robustez siempre que se acompañe de un preprocesamiento adecuado y mecanismos de regularización apropiados.

CONCLUSIONES BASADAS EN LOS OBJETIVOS

1. Comprender las características del dataset Adult:
Se logró identificar la estructura del dataset, el desbalance de la variable objetivo y la presencia de valores faltantes, así como la mezcla de variables numéricas y categóricas. Este análisis permitió tomar decisiones fundamentadas durante el preprocesamiento.

2. Implementar un preprocesamiento adecuado:
El preprocesamiento cumplió su objetivo: la imputación, normalización y codificación One-Hot permitieron estandarizar el dataset y mejorar la estabilidad de los modelos. El pipeline implementado evitó fuga de información y generó un flujo reproducible.

3. Construir y entrenar tres arquitecturas de distinta complejidad:
Se implementaron correctamente el perceptrón, la red de una capa oculta y la red profunda de diez capas ocultas. Cada modelo se entrenó bajo las mismas condiciones, permitiendo una comparación justa.

4. Evaluar cada modelo mediante métricas estándar:
Las métricas obtenidas evidenciaron las diferencias entre arquitecturas: el perceptrón tuvo el desempeño más bajo, la red de una capa logró una mejora sustancial y la red profunda destacó por su estabilidad y consistencia entre entrenamiento y validación.

5. Comparar el desempeño y analizar el impacto de la complejidad:
Se concluyó que aumentar la profundidad mejora la estabilidad del entrenamiento y la capacidad de representar relaciones no lineales del dataset, aunque la mejora en métricas no sea exponencial. La red profunda se comportó como el modelo más robusto del conjunto evaluado.

RECOMENDACIONES

A partir del desarrollo del presente proyecto, se sugieren las siguientes recomendaciones para mejorar el desempeño de los modelos y ampliar el beneficio obtenido:

1. **Implementar técnicas de balanceo de clases**, como *SMOTE* o *class_weight*, con el fin de reducir el impacto del desbalance presente en el dataset Adult y mejorar la capacidad del modelo para identificar correctamente la clase minoritaria.
2. **Explorar funciones de activación diferentes a la sigmoide**, especialmente ReLU o LeakyReLU, para evitar la saturación del gradiente en redes profundas y acelerar la convergencia del entrenamiento.
3. **Introducir regularización (Dropout o L2)** para reducir el riesgo de sobreajuste en arquitecturas con múltiples capas ocultas y mejorar la generalización del modelo.
4. **Realizar una optimización de hiperparámetros**, empleando métodos como *Grid Search* o *Random Search*, con el propósito de identificar configuraciones más eficientes en términos de neuronas, capas, tasa de aprendizaje y batch size.
5. **Ampliar el análisis exploratorio de datos**, incorporando nuevas visualizaciones y estadísticas que permitan comprender mejor las relaciones entre variables y su impacto sobre el ingreso.
6. **Integrar técnicas de ingeniería de características**, como cruces de variables o transformaciones no lineales, para enriquecer el conjunto de datos y potencialmente mejorar la capacidad predictiva del modelo.
7. **Implementar validación cruzada**, lo que permitiría evaluar la estabilidad del modelo frente a diferentes particiones de los datos.

BIBLIOGRAFÍA

Libros

Goodfellow, I., Bengio, Y. y Courville, A.
Deep Learning. MIT Press. 2016.

Murphy, K.
Machine Learning: A Probabilistic Perspective. MIT Press. 2012.

Artículos / Documentación técnica

Pedregosa, F. et al.
"Scikit-learn: Machine Learning in Python". *Journal of Machine Learning Research*, Vol. 12, 2011.

TensorFlow.
"Deep Learning Framework Documentation". Google Brain Team, 2023.
Disponible en: https://www.tensorflow.org/

Fuentes electrónicas

Dua, D. y Graff, C.
"UCI Machine Learning Repository: Adult Dataset". University of California, Irvine.
Disponible en: https://archive.ics.uci.edu/ml/datasets/adult

Software

Python Software Foundation.
Python 3.12 Documentation. Disponible en: https://www.python.org/

Van Rossum, G.
The Python Programming Language. 2010