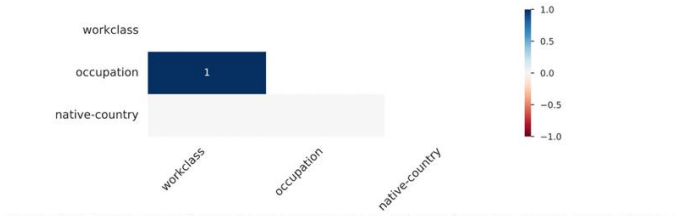
Integrantes: Gabriela Zamora, Gilberto Galeana y Sofía Anzola Ortegón

Reporte Parcial 2

En este proyecto se aborda el problema de predicción de ingresos utilizando el dataset Adult Census Income del repositorio UCI (1994). El objetivo consiste en predecir si una persona gana más de 50.000 USD anuales a partir de variables demográficas y socioeconómicas. Para ello, se implementó un flujo completo de *machine learning* en Google Colab: recolección y procesamiento de datos, análisis exploratorio (EDA), modelado baseline con regresión logística y un modelo de redes neuronales (MLP) en PyTorch. Además, se aplicaron técnicas de regularización (Dropout y EarlyStopping) y se evaluó el desempeño con métricas estándar de clasificación binaria en conjuntos de entrenamiento, validación y prueba. Este trabajo permite comparar un modelo lineal tradicional con una red neuronal más compleja, analizando sus fortalezas y limitaciones en términos de capacidad predictiva y riesgo de sobreajuste.

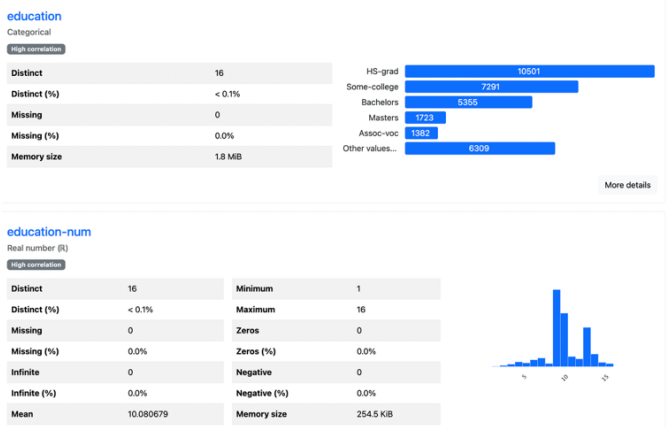
1. Procesamiento de datos.

Primero, la correlación de nulidad reveló que *workclass* y *occupation* tienden a ausentarse conjuntamente. Esto justifica evitar el borrado de filas que sesgaría la muestra y, en su lugar, imputar categóricas con el valor explícito ***"Unknown"***, manteniendo consistencia entre ambas columnas.



The correlation heatmap measures nullity correlation: how strongly the presence or absence of one variable affects the presence of another.

Segundo, comparamos *education* y *education-num* y hallamos correspondencia 1↔1 entre niveles Por ende, conservar ambas infla la dimensionalidad tras el one-hot sin añadir información; por eso eliminamos ***education*** y mantenemos *education-num*, que preserva el orden educativo y simplifica el modelo.



También eliminamos ***fnlwgt*** porque es un peso muestral del esquema de la encuesta, no una característica estable del individuo. Incluirlo puede introducir efectos de diseño que el modelo interpretaría como señal predictiva, degradando la validez externa. Su descarte reduce ruido y evita interpretaciones falsas.

En cuanto al procesamiento, usamos todas las características restantes sin crear nuevas de forma manual, priorizando trazabilidad. Aplicamos un ColumnTransformer: *StandardScaler* en numéricas para centrar y escalar, favoreciendo optimización y comparabilidad y **One-Hot Encoding** en categóricas con *handle_unknown="ignore"* que garantiza que validación/prueba acepten categorías no vistas sin fallar. El ajuste del pipeline se realizó solo en entrenamiento y luego se transformaron los splits, evitando fuga de información.

Dado el desbalance del objetivo *(<=50K* mayoritario), incorporamos pesos de clase durante el entrenamiento. Esta elección modifica la función de pérdida sin alterar la distribución de datos, manteniendo simple el pipeline y la interpretación de métricas; evitamos undersampling para no duplicar observaciones ni introducir varianza adicional.

Finalmente, no aplicamos transformaciones log ni winsorización en *capital-gain/loss* ni reducciones de dimensionalidad (PCA). El criterio fue parcimonia: mientras las métricas de validación se mantuvieron estables con OHE + escalado + pesos de clase, preferimos un flujo más interpretable y reproducible. Como trabajo futuro, podríamos reagrupar países raros en *native-country* y evaluar transformaciones robustas si el modelo mostrara sensibilidad a colas extremas.

2. Hiperparámetros MLP Regularizado y No Regularizado.

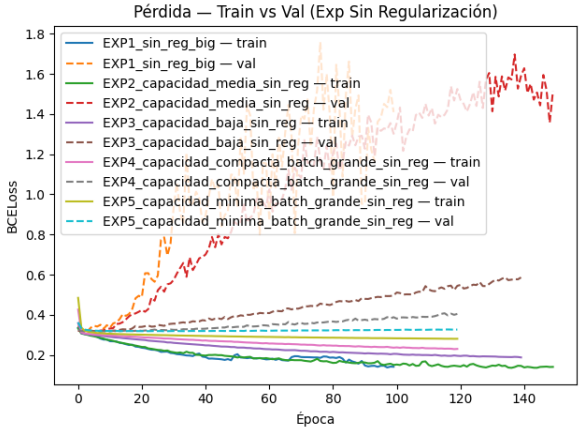
Sin regularización:

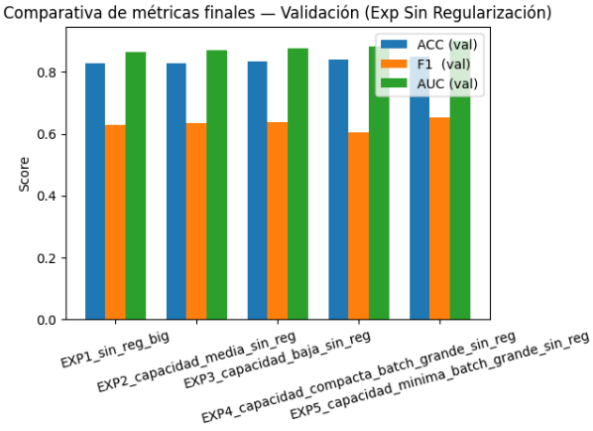
El experimento entrena un MLP sin regularización explícita con BCELoss y AdamW. Los datos llegan por *DataLoader* con *shuffle* en entrenamiento y lectura secuencial en validación; los lotes se mueven a CPU/GPU con *non_blocking=True* para solapar copia y cómputo.

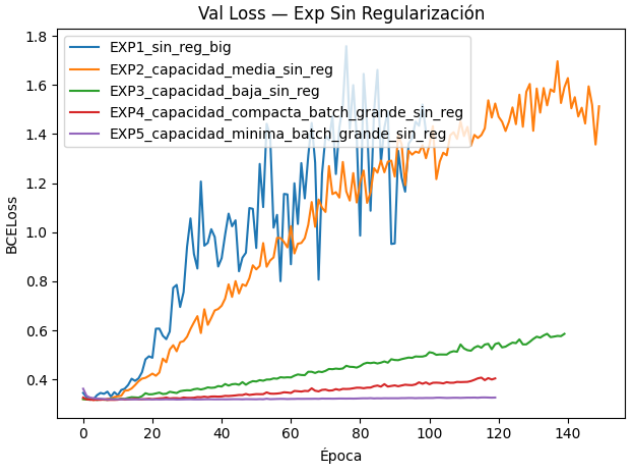
Cada época tiene dos fases. Entrenamiento (*model.train()*): para cada *batch* se hace *forward*, se calcula la pérdida binaria, se limpian gradientes (*optimizer.zero_grad()*), se hace backprop (*loss.backward()*), y se actualizan pesos (*optimizer.step()*). La pérdida de la época se acumula ponderando por tamaño de batch y luego se normaliza por el número de ejemplos del *dataset*.

Validación (*model.eval()*): se desactiva el gradiente con *torch.no_grad()* para ahorrar memoria/tiempo, se hace solo *forward* y acumulación de *val_loss* con la misma normalización por tamaño del conjunto. Al final de cada época se registran *train_losses* y *val_losses*, y se imprime un resumen para monitorear convergencia y brecha train–val sin afectar el estado del optimizador ni del modelo.

Para el caso de los 5 experimentos de MLP sin regularización se obtuvieron los siguientes resultados:







De acuerdo con las gráficas de *Val Loss* y *Train vs Val Loss*, se observa que los modelos con mayor capacidad (EXP1 y EXP2) presentan un fuerte sobreajuste, evidenciado por la divergencia creciente entre la pérdida de entrenamiento y validación.
El EXP3 muestra una tendencia opuesta: el aprendizaje es insuficiente (underfitting), ya que tanto la pérdida de entrenamiento como la de validación se mantienen elevadas.
El modelo EXP4 se comporta de manera más estable, pero es el EXP5_capacidad_mínima_batch_grande_sin_reg el que logra el mejor balance: mantiene Val Loss bajo y estable, con curvas de entrenamiento y validación cercanas, lo que indica una buena capacidad de generalización.

En cuanto a los hiperparámetros del mejor modelo (EXP5):

- **Capas ocultas**: 1
- **Neuronas por capa:** 96
- **Dropout:** no aplicado
- **Weight Decay:** no aplicado
- **Learning rate:** 0.0005
- **Batch size:** 256

- **Épocas:** 120
- **Optimizador:** AdamW (sin weight decay)
- **Función de pérdida:** Binary Cross Entropy (BCELoss)

Ahora bien, en cuanto a las métricas obtenidas del mejor MLP Sin Regularización:

Split	Accuracy	Precision	Recall	F1	AUC	Loss
Train	0.8702	0.7749	0.6494	0.7066	0.9305	0.2788
Val	0.8509	0.7239	0.5959	0.6537	0.9019	0.3264
Test	0.8619	0.7465	0.6292	0.6828	0.9126	–

El análisis de las métricas confirma que EXP5 es el mejor modelo sin regularización:

- Presenta el F1 más alto en validación (0.6537), que es crítico en un problema desbalanceado como *Adult Income*, ya que combina precisión y recall.
- Los resultados en test son consistentes (F1=0.6828, AUC=0.9126), lo que indica que el modelo generaliza correctamente y no depende únicamente del set de validación.
- El diagnóstico del *gap* entre train y val losses sugiere un ajuste razonable, sin caer en underfitting ni en sobreajuste severo.

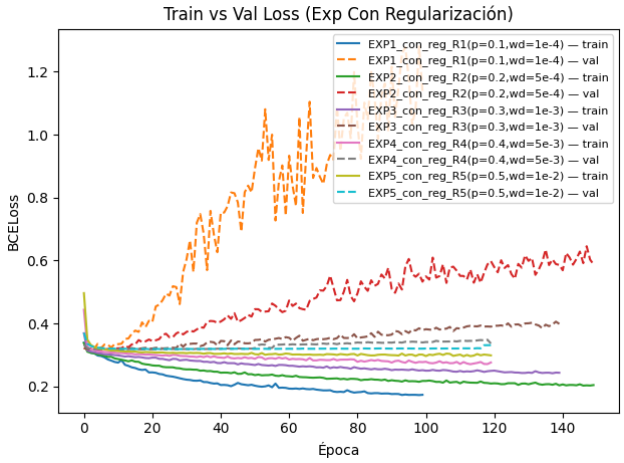
En conclusión, el **EXP5** logra el mejor equilibrio entre capacidad de aprendizaje y generalización, siendo el modelo recomendado dentro del grupo sin regularización.

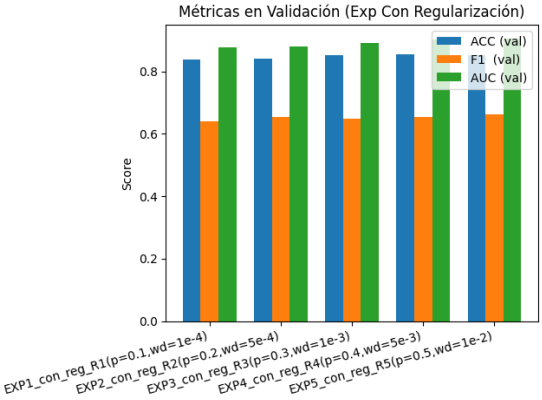
Con regularización:

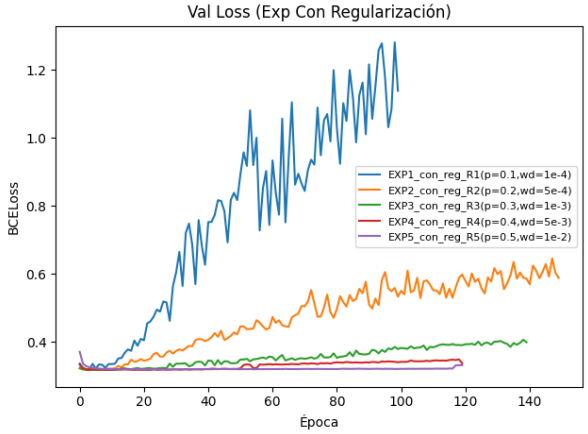
Se entrena un MLP de clasificación con una capa oculta (96 neuronas, **ReLU**) y salida sigmoidal. La regularización es doble: **Dropout p=0.5** en la capa oculta para apagar unidades al azar y **weight decay = 1e-2** vía **AdamW** para penalizar pesos grandes. Esta combinación controla el sobreajuste tanto a nivel de activaciones como de parámetros, que es lo más decisivo del diseño. La pérdida es **BCELoss**, con **learning rate** = 5e-4, **batch size** = 256 y 120 épocas con semilla fija para reproducibilidad.

Los tensores se mueven a **CPU/GPU**; en *train* se hace el ciclo estándar (cero gradiente → *forward* → **BCELoss** → *backprop* → *optimizer.step()*), y en *eval* se desactiva el gradiente para validación por época. Los datos llegan vía DataLoaders aleatorizado solo en entrenamiento. El modelo se evalua posteriormente con métricas fuera del loop (p. ej., **AUC y F1**), sin alterar el procedimiento de entrenamiento.

Por otro lado, en el caso de los 5 experimentos de MLP con regularización se obtuvieron los siguientes resultados:







A partir de las gráficas de Val Loss y Train vs Val Loss, se observa que algunos modelos con menor dropout y weight decay (EXP1 y EXP2) presentan inestabilidad en validación: sus curvas de pérdida aumentan o fluctúan con fuerza, lo cual es indicio de sobreajuste o entrenamiento poco robusto. En contraste, EXP3 y EXP4 logran un comportamiento más estable, con pérdidas de validación más controladas. Finalmente, el modelo EXP5_con_reg_R5(p=0.5, wd=1e-2) es el que ofrece el mejor balance: mantiene una pérdida de validación baja y estable, sin divergencias marcadas entre entrenamiento y validación, y además consigue el mejor F1 en validación, métrica clave en un problema de clasificación binaria desbalanceada.

En cuanto a los hiperparámetros del mejor modelo (EXP5 con Reg):

- **Capas ocultas:** 1
- **Neuronas por capa:** 96
- **Dropout:** 0.5
- **Weight Decay:** 1e-2
- **Learning rate:** 0.0005
- **Batch size:** 256
- **Épocas:** 120
- **Optimizador:** AdamW (con weight decay = 1e-2)
- **Función de pérdida:** Binary Cross Entropy (BCELoss)

A partir de esto, se encontraron las siguientes métricas para el experimento 5:

Split	Accuracy	Precision	Recall	F1	AUC	Loss
Train	0.8675	0.7635	0.6518	0.7033	0.9256	0.2908
Val	0.8528	0.7247	0.6079	0.6612	0.9047	0.3309
Test	0.8646	0.7489	0.6422	0.6915	0.9162	–

El análisis de las métricas confirma que EXP5_con_reg_R5(p=0.5, wd=1e-2) es el mejor modelo con regularización:

- Presenta el F1 más alto en validación (0.6612), lo cual es clave en un problema desbalanceado como Adult Income, ya que refleja un buen balance entre precisión y recall.
- Los resultados en test son consistentes (F1=0.6915, AUC=0.9162), lo que demuestra que el modelo generaliza correctamente y no depende únicamente del set de validación.
- El diagnóstico del gap entre train y val losses sugiere un ajuste razonable, sin señales marcadas de underfitting o sobreajuste.

En conclusión, el **EXP5 con regularización** (Dropout 0.5, Weight Decay 1e-2) logra el mejor equilibrio entre capacidad de aprendizaje y generalización, siendo el modelo recomendado dentro del grupo con regularización.

Con vs Sin Regularización:

Modelo	Accuracy (Val)	F1 (Val)	AUC (Val)	Accuracy (Test)	F1 (Test)	AUC (Test)	Diagnóstico
EXP5 — Sin regularización	0.8509	0.6537	0.9019	0.8619	0.6828	0.9126	Ajuste razonable, gap controlado
EXP5 — Con regularización	0.8528	0.6612	0.9047	0.8646	0.6915	0.9162	Ajuste razonable, gap controlado

- **F1 en validación:** El modelo con regularización (0.6612) supera al sin regularización (0.6537), confirmando que la regularización mejoró el balance entre precisión y recall.
- **Test set:** De igual forma, el modelo con regularización obtiene mejor F1 (0.6915 vs. 0.6828) y AUC más alto (0.9162 vs. 0.9126), lo que indica una mejor capacidad de generalización.
- **Estabilidad**: Ambos muestran un gap controlado entre entrenamiento y validación, sin señales de overfitting fuerte. Sin embargo, la regularización le dio al EXP5_reg un margen extra de robustez.

En conclusión, el **EXP5 con regularización** (Dropout 0.5, Weight Decay 1e-2) es el mejor modelo global, ya que supera consistentemente al EXP5 sin regularización en F1 y AUC tanto en validación como en test, confirmando que la regularización mejoró la capacidad de generalización del modelo.

3. **Compare el mejor MLP y la regresión lineal a partir de sus métricas. Interprete los resultados.**

Teniendo en cuenta el mejor MLP (EXP 5 con regularización) y la regresión lineal que se planteó en un principio (Baseline). Podemos analizar y observar ciertas diferencias y similitudes en sus métricas. Se realizó una tabla donde se pueden ver todos los resultados para así facilitar la interpretación de los mismos:

Modelo / Split	Accuracy	Precision	Recall	F1	AUC
MLP(EXP5_con_reg_R5) – Train	0.8675	0.7635	0.6518	0.7033	0.9256
MLP(EXP5_con_reg_R5) – Val	0.8528	0.7247	0.6079	0.6612	0.9047
MLP(EXP5_con_reg_R5) – Test	0.8646	0.7489	0.6422	0.6915	0.9162
Reg. Logística – Train	0.8116	0.5738	0.8464	0.6840	0.9082
Reg. Logística – Val	0.8004	0.5512	0.8336	0.6636	0.8988
Reg. Logística – Test	0.8154	0.5741	0.8461	0.6840	0.9108

Antes de realizar el análisis de las métricas es importante reconocer que el dataset Adult Income está desbalanceado (aprox. 75% clase 0 y 25% clase 1, lo que hace que el accuracy sea engañoso: un modelo que siempre predijera la clase mayoritaria tendría ≈75% de accuracy, sin ser útil. Por eso, debemos enfocarnos en F1, precision, recall y AUC.

Comparación considerando el desbalance:

1. **Regresión Logística (baseline)**

- **Recall muy alto (~0.84):** logra identificar la mayoría de los casos positivos (>50K).
- **Precisión baja (~0.57):** de los positivos que predice, muchos son falsos positivos.
- **F1 ≈ 0.66–0.68:** balance aceptable entre precision y recall, pero inclinado hacia recall.

- **AUC ≈ 0.91:** excelente capacidad de discriminación global.

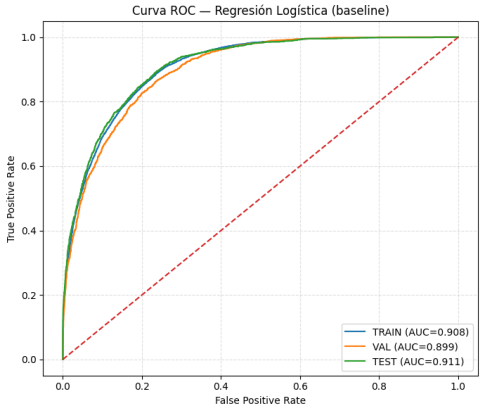
En un dataset desbalanceado, este comportamiento indica que la regresión logística es sensible (recupera muchos positivos), pero arriesga al equivocarse con bastantes falsos positivos.

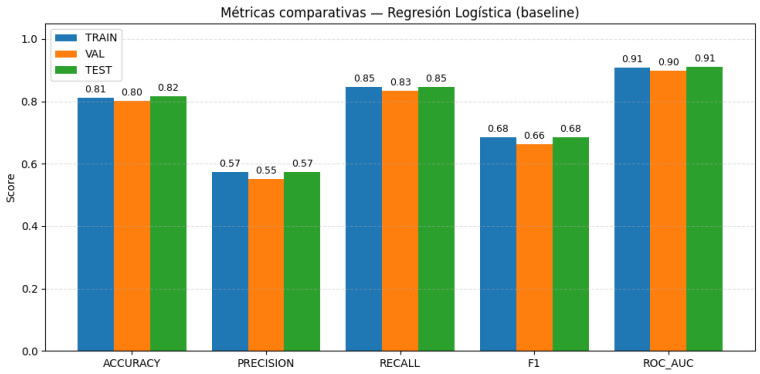
2. **MLP con regularización (EXP5 R5)**

- **Precision alta (~0.74):** cuando predice un positivo, suele acertar más que la regresión logística.
- **Recall moderado (~0.64):** captura menos positivos que la regresión logística.
- **F1 ≈ 0.69:** ligeramente superior al baseline, mostrando un mejor equilibrio entre precision y recall.
- **AUC ≈ 0.916:** un poco mayor que la regresión logística, confirmando mejor capacidad discriminativa.

Este modelo es más conservador: predice menos positivos, pero lo hace con mayor certeza. Esto es deseable si el costo de los falsos positivos es alto.

<u>Interpretación de las gráficas de la regresión lineal:</u>





- **Curvas ROC:** Ambos modelos tienen curvas muy similares, con AUC alto (>0.90), lo que muestra que ambos separan bien las clases. El MLP se ubica apenas por encima, confirmando su ligera superioridad.
- **Loss Train vs Val (MLP):** el EXP5 regulado muestra curvas estables y sin sobreajuste, a diferencia de los MLP más grandes sin regularización.
- **Gráficas de métricas:** confirman el trade-off: la regresión logística se inclina hacia recall, el MLP hacia precision. El F1 y el AUC, que son métricas robustas frente al desbalance, favorecen al MLP.

En conclusión, Dado el desbalance del dataset, accuracy no es un criterio válido para comparar. La Regresión Logística es útil si lo más importante es detectar la mayor cantidad posible de casos positivos, aunque arriesgue falsos positivos. El MLP con regularización (EXP5 R5) logra un mejor F1 y AUC, lo que significa que mantiene un balance más sólido entre precision y recall, y además generaliza mejor. En términos prácticos, el MLP regulado es el modelo recomendado, ya que combina precisión alta con recall aceptable, ofreciendo mejor discriminación global en un contexto desbalanceado.

Bibliografía:

UC Irvine ML Repository. (1996). *Adult.* Recuperado de: https://archive.ics.uci.edu/dataset/2/adult