

# Pipeline de Machine Learning con Redes Neuronales (RNA)

## Clasificación - Dataset Titanic

Desarrollo y evaluación un modelo de aprendizaje automático basado en redes neuronales para predecir la supervivencia de pasajeros del Titanic.

### Fase 1: Preprocesamiento (Preparando el terreno)

- Limpieza: Imputamocion de valores faltantes (usando la media/moda) para no perder información.
- Codificación: datos de texto (como "Sexo" o "Embarque") a números, ya que las RNA solo procesan operaciones matemáticas.
- Balanceo (SMOTE): equilibrio de las clases para que el modelo no se sesgue hacia la mayoría (los que no sobrevivieron).
- Escalado: Ajuste de valores a una escala común (StandardScaler) para facilitar el aprendizaje de la red.

### Fase 2: Procesamiento (El Entrenamiento)

- Modelo Moderno: Optimizador Adam con activación ReLU (Rápido y eficiente).
- Modelo Clásico: Optimizador SGD con activación Tangente Hiperbólica.
- Modelo Estadístico: Optimizador LBFGS con activación Logística.

### Fase 3: Posprocesamiento (La Evaluación)

- Evaluación de la precisión y el F1-Score en datos que la red nunca vio (Test set).
- Verificación que la red no haya "memorizado" los datos (Overfitting) comparando el rendimiento de entrenamiento vs. validación.

## Pipeline Automatizado (Workflow)

Ingesta: Carga de datos crudos desde repositorio (GitHub).

Transformación: Limpieza + División 60/20/20 (Train/Val/Test).

Entrenamiento: Generación de 3 modelos MLP concurrentes.

Validación: Cross-Validation para asegurar robustez.

Despliegue: API Flask lista para recibir consultas.

## Comparativa de rendimiento para seleccionar el modelo final.

Modelo	Tiempo <small>(segundos)</small>	Acc Train <small>% aciertos entrenamiento</small>	Acc Val <small>% aciertos validación</small>	Acc Test <small>% aciertos test</small>	Precision <small>TP / (TP + FP)</small>	Recall <small>TP / (TP + FN)</small>	F1-Score <small>balance precision/recall</small>	Overfit <small>diferencia train-val</small>
Modelo 1 - Adam/ReLU <div>MEJOR</div>	0.061s	0.8237	0.7978	0.7374	0.6571	0.6667	<b>0.6619</b>	0.0259
Modelo 2 - SGD/Tanh	0.5314s	0.7994	0.7809	0.7598	0.6857	0.6957	<b>0.6906</b>	0.0185
Modelo 3 - LBFGS/Logistic	0.7906s	0.9635	0.7584	0.7374	0.6618	0.6522	<b>0.6569</b>	0.2051

### Matrices de Confusion (Test)

Desglosan aciertos y errores binarios: TN/FP/FN/TP para cada modelo.

Modelo 1 - Adam/ReLU			Modelo 2 - SGD/Tanh			Modelo 3 - LBFGS/Logistic		
	Pred: No sobrevivió	Pred: Sobrevivió		Pred: No sobrevivió	Pred: Sobrevivió		Pred: No sobrevivió	Pred: Sobrevivió
Real: No sobrevivió	TN 86	FP 24	Real: No sobrevivió	TN 88	FP 22	Real: No sobrevivió	TN 87	FP 23
Real: Sobrevivió	FN 23	TP 46	Real: Sobrevivió	FN 21	TP 48	Real: Sobrevivió	FN 24	TP 45
Filas: Real (lo que pasó). Columnas: Predicción (lo que el modelo dijo). TN/TP: Aciertos. FP/FN: Errores.			Filas: Real (lo que pasó). Columnas: Predicción (lo que el modelo dijo). TN/TP: Aciertos. FP/FN: Errores.			Filas: Real (lo que pasó). Columnas: Predicción (lo que el modelo dijo). TN/TP: Aciertos. FP/FN: Errores.		