

Caso: Predicción operativa en centros de cultivo de salmón

Objetivo. Entender el problema y los datos (sectores/jaulas, tiempo, tamaño, peso, edad, mortalidad, FCR, temperatura, oxígeno disuelto, salinidad, tratamientos, eventos sanitarios).



Conceptos

1) Datos y particiones -1

- **Conjunto de entrenamiento / validación / prueba (train/val/test)**
- *Qué es:* dividir el dataset en tres partes:
 - **Train:** para ajustar (entrenar) el modelo.
 - **Validación:** para elegir hiperparámetros (no se entrena con esto).
 - **Prueba:** para medir el resultado final “como si fuera el mundo real”.
- *Por qué importa:* evita engañarnos con resultados aprendidos “de memoria”.
- *Ejemplo:* 2022–2024Q3 (train), 2024Q4 (val), 2025H1 (test).

1) Datos y particiones - 2

- **Fuga de datos (data leakage)**

- *Qué es:* usar sin querer información del futuro o del conjunto de validación/prueba al entrenar.
- *Por qué importa:* sube las métricas artificialmente y luego el modelo falla en la realidad.
- *Ejemplo:* calcular el promedio de DO (oxígeno) usando datos de 2025 y aplicarlo a entrenar el modelo de 2023.

1) Datos y particiones - 3

- **Estratificación (stratified split)**

- *Qué es:* dividir manteniendo proporciones de clases (por ejemplo, “Alta Mortalidad” = 10% en cada partición).
- *Por qué importa:* evita que validación o prueba queden “desbalanceadas”.

- **Imputación**

- *Qué es:* rellenar datos faltantes (NaN) con una regla (media, mediana, interpolación en series).
- *Ejemplo:* Si falta DO del día 10, usar promedio entre día 9 y 11.

- **Estandarización / z-score**

- *Qué es:* transformar una variable para que tenga media 0 y desviación estándar 1.
- *Fórmula:* $z = (x - \text{media}) / \text{desviación}$
- *Ejemplo:* Si DO tiene media 7.5 y desv. 1.0, un $DO=6.5 \Rightarrow z = (6.5-7.5)/1.0 = -1.0$.

1) Datos y particiones - 4

- **Normalización Min-Max**

- *Qué es:* reescalar a un rango (típicamente 0 a 1).
- *Fórmula:* $(x - \min) / (\max - \min)$.

- **Outlier (valor atípico)**

- *Qué es:* dato “extremo” respecto a la mayoría.
- *Qué hacer:* revisar si es error; si no, atenuar su efecto (p. ej., winsorización: recortar a un percentil).

2) Series temporales y “features”

- **Ventana (L)**
 - *Qué es:* cuántos días hacia atrás mira el modelo.
 - *Ejemplo:* $L=28 \Rightarrow$ el modelo usa DO, T°, peso, etc. de los últimos 28 días.
- **Horizonte (H)**
 - *Qué es:* cuántos días hacia adelante predice.
 - *Ejemplo:* $H=7 \Rightarrow$ queremos estimar la próxima semana.
- **Rezagos (lags)**
 - *Qué es:* el valor de una variable “corrida en el tiempo”.
 - *Ejemplo:* DO-1 (ayer), DO-7 (hace una semana).
- **Funciones en ventana (rolling)**
 - *Qué es:* estadísticos sobre la ventana: mínimo, máximo, promedio, etc.
 - *Ejemplo:* DO_min_7d = mínimo DO en los últimos 7 días.
- **Variación porcentual ($\Delta\%$)**
 - *Qué es:* cuánto cambia una variable en porcentaje.
 - *Fórmula:* $\Delta\% = (\text{valor_actual} - \text{valor_pasado}) / \text{valor_pasado} \times 100$
 - *Ejemplo:* peso pasa de 1200 g a 1260 g $\Rightarrow \Delta\% = (1260-1200)/1200 \times 100 = 5\%$.

3) Clasificación (sí/no o varias clases) - 1

- **Matriz de confusión**
- Qué es: tabla que resume aciertos y errores:
 - **TP** (verdaderos positivos): predijo “Alta” y era “Alta”.
 - **FP** (falsos positivos): predijo “Alta” pero no era.
 - **FN** (falsos negativos): no predijo “Alta” y sí era.
 - **TN** (verdaderos negativos): no predijo “Alta” y no era.
- **Accuracy (exactitud)**
 - *Fórmula:* $(TP + TN) / \text{total}$.
 - *Cuidado:* engañosa cuando la clase positiva es rara (podemos tener 95% acertando “siempre no”).
- **Precision (precisión)**
 - *Fórmula:* $TP / (TP + FP)$
 - *Idea:* de los que predije como “Alta”, ¿cuántos realmente lo eran?
- **Recall (sensibilidad)**
 - *Fórmula:* $TP / (TP + FN)$
 - *Idea:* de todos los casos “Alta”, ¿cuántos detecté?
 - *Operativamente:* si perder un “Alta” es grave, priorizo **recall**.

3) Clasificación (sí/no o varias clases) - 2

- **F1-score**
 - *Fórmula:* $F1 = 2 \times (\text{precision} \times \text{recall}) / (\text{precision} + \text{recall})$
 - *Idea:* equilibrio entre precisión y recall (útil con desbalance).
- **Macro / Micro promedio**
- **Macro:** promedio simple del F1 por clase (trata todas las clases por igual).
- **Micro:** suma TP/FP/FN globales antes de calcular F1 (pondera por tamaño de clase).
- **Curva ROC y AUC-ROC**
 - *ROC:* gráfico de **TPR (=recall)** vs **FPR (FP/negativos)** al mover el umbral.
 - *AUC-ROC:* área bajo esa curva (0.5=azar, 1.0=perfecto).
 - *Cuidado:* con clases raras, ROC puede ser optimista.
- **Curva Precisión-Recall (PR) y AUC-PR**
 - *PR:* precisión vs recall al mover el umbral.
 - *AUC-PR:* área bajo PR; **mejor** que ROC cuando la clase positiva es rara (como “Alta Mortalidad”).

3) Clasificación (sí/no o varias clases) - 3

- **Umbral operativo**
 - *Qué es:* valor de probabilidad a partir del cual disparamos la alerta.
 - *Ejemplo:* Elegimos umbral para alcanzar **Recall ≥ 0.90** (aceptando más FP).
- **Calibración de probabilidades y Brier score**
 - *Calibración:* que “0.7” signifique ~70% de ocurrencia en la práctica.
- **Brier score:** error cuadrático medio entre probabilidad predicha y realidad (0=perfecto, 1=peor).
- **Pesos de clase (class weights)**
 - *Qué es:* dar más peso a la clase minoritaria al entrenar.
 - *Por qué:* ayuda cuando “Alta” es poco frecuente.
- **Focal Loss**
 - *Qué es:* función de pérdida para clases desbalanceadas que **enfatisa ejemplos difíciles**.
 - *Intuición:* reduce la influencia de los fáciles y aumenta la de los que el modelo se equivoca mucho.
- **Label smoothing**
 - *Qué es:* suavizar las etiquetas “duras” (1 y 0) a valores como 0.9 y 0.1.
 - *Por qué:* ayuda a generalizar y evita sobreconfianza del modelo.

4) Regresión (predicción de valores continuos)

- **MAE (Mean Absolute Error)**
 - *Fórmula:* promedio de $|\text{predicción} - \text{real}|$.
 - *Interpretación:* error medio en **unidades de la variable** (fácil de explicar).
 - *Ejemplo:* errores: 30 g, 50 g, 20 g $\Rightarrow \text{MAE} = (30+50+20)/3 = 33.3$ g.
- **MSE (Mean Squared Error) y RMSE (Root MSE)**
 - *MSE:* promedio de $(\text{error})^2$. Penaliza más los errores grandes.
 - *RMSE:* raíz del MSE (vuelve a las unidades originales).
 - *Ejemplo:* errores 30, 50, 20 $\Rightarrow \text{MSE} = (900+2500+400)/3 = 1266.7$; $\text{RMSE} \approx 35.6$ g.
- **R² (coeficiente de determinación)**
 - *Qué es:* proporción de variación explicada (1=perfecto, 0=igual a la media).

Nota: no siempre es la mejor métrica operativa; MAE suele ser más interpretable.

5) Segmentación

- **IoU (Intersection over Union)**
 - *Fórmula:* $\text{IoU} = (\text{área de intersección}) / (\text{área de unión})$
 - *Uso:* segmentación de imágenes (qué tan bien se superponen las máscaras).
- **Coeficiente Dice**
 - *Fórmula:* $\text{Dice} = 2 \times \text{intersección} / (\text{pred} + \text{real})$
 - *Parecido a IoU:* más sensible a clases pequeñas.

6) Funciones de activación y pérdidas comunes

- **Sigmoid**
 - *Qué hace:* transforma un número en probabilidad 0–1.
 - *Uso:* clasificación binaria.
- **Softmax**
 - *Qué hace:* transforma un vector en probabilidades que suman 1.
 - *Uso:* clasificación multiclase.
- **Cross-Entropy (CE) / Binary Cross-Entropy (BCE)**
 - *Qué es:* pérdida que compara la probabilidad predicha con la etiqueta real.
 - *Uso:* CE para multiclase (softmax), **BCE** para binaria (sigmoid) o multietiqueta.
- **MAE/MSE**
 - Pérdidas típicas en regresión.

7) Regularización y estabilidad del entrenamiento - 1

- **Overfitting / Underfitting**
 - *Overfitting*: el modelo aprende “memoria” del train y falla en val/test.
 - *Underfitting*: el modelo es demasiado simple y no aprende el patrón.
- **Dropout**
 - *Qué es*: apagar aleatoriamente neuronas en entrenamiento (p. ej., 20%).
 - *Por qué*: evita sobreajuste, mejora generalización.
- **Weight decay (L2)**
 - *Qué es*: penalización a pesos grandes durante el entrenamiento.
 - *Por qué*: “empareja” los parámetros y reduce sobreajuste.
- **Early stopping**
 - *Qué es*: detener cuando la métrica de validación deja de mejorar.
 - *Por qué*: previene overfitting.
- **Learning Rate (LR)**
 - *Qué es*: tamaño del paso al actualizar pesos.
 - *Demasiado grande*: inestable; *demasiado pequeña*: aprendizaje lento.

7) Regularización y estabilidad del entrenamiento - 2

- **Scheduler de LR**

- *Qué es:* plan para cambiar la LR a lo largo del entrenamiento (por ejemplo, “cosine”).
- *Warmup:* empezar con LR más baja y subirla gradualmente (evita inestabilidad al inicio).

- **Gradient clipping**

- *Qué es:* poner un límite al tamaño del gradiente.
- *Por qué:* evita “explosión de gradientes” (muy común en secuencias).

- **AMP (Mixed Precision)**

- *Qué es:* entrenar con menor precisión numérica en GPU (por ejemplo, 16 bits).
- *Por qué:* más rápido y ahorra memoria, con mínima pérdida de calidad.

8) Interpretabilidad y calibración

- **Importancia de variables**

- *Qué es:* ranking de qué variables influyen más en la predicción.
- *Por qué:* ayuda a explicar “por qué” el modelo dio una alerta.

- **Calibración (otra vez, operativa)**

- *Idea:* si el modelo dice 0.8 de probabilidad, que de 10 casos 8 ocurran realmente.
- *Herramientas:* curvas de confiabilidad (reliability), Brier score.

9) Métricas por subgrupos (equidad/operación)

- **Evaluación por subgrupos**

- *Qué es:* medir métricas separadas por centro, estación, cohorte, etc.
- *Por qué:* detectar si el desempeño es peor en ciertos contextos (invierno, centros nuevos).

10) Mini-tabla de “qué usar cuándo”

- **Clase rara y grave (Alta Mortalidad):** AUC-PR, Recall alto, F1-macro; umbral operativo para $\text{Recall} \geq 0.90$; *class weights* o **Focal Loss**.
- **Pronóstico de peso (g):** **MAE** (fácil de explicar), y **RMSE** si te importan mucho los errores grandes.
- **Entrenamiento estable:** LR con **warmup** + **scheduler**, **dropout**, **weight decay**, **early stopping**, **gradient clipping**.
- **Series temporales:** definir bien **L** (ventana) y **H** (horizonte); usar **lags** y funciones en ventana (min/mean/max).

Problemática

- La empresa salmonera opera múltiples centros y jaulas de cultivo. Cada jaula aloja peces de una cohorte (grupo de la misma edad aproximada). La operación debe anticipar eventos de alta mortalidad (muchas muertes en un período corto) y pronosticar el peso promedio de los peces para planificar alimentación, cosecha y logística.

Actividad

- **Tipificar tarea(s):** p. ej., predicción de mortalidad semanal y pronóstico de crecimiento/peso; identificar variables clave y restricciones (alerta temprana, latencia diaria, trazabilidad para auditorías).
- **Tarea.** Elegir arquitectura candidata (LSTM/TCN/TFT para series; XGBoost/TabTransformer para tabular; Prophet/ARIMA como baseline).
- **Ejemplo aplicado.**
 - **Problema principal:** clasificación (alta mortalidad próxima semana: sí/no) + regresión (peso medio por jaula). Métricas: AUC-PR y F1 (clase “alta mortalidad” rara), MAE/RMSE para peso. Restricciones: explainability para Sernapesca y operación.

Preguntas operativas :

- ¿Podemos predecir si habrá “alta mortalidad” la próxima semana en cada jaula? (alertas tempranas para intervenir).
- ¿Podemos estimar el peso promedio por jaula a 7 días? (planificar alimento y cosecha).

Ambos problemas requieren modelos predictivos que usen datos históricos de producción y ambientales.

Datos fuente (qué son y de dónde salen)

- Producción por jaula (tabular/serie temporal): fecha, centro, jaula, edad (días desde siembra), peso promedio (g), biomasa (kg en jaula), muertes (n° peces/día), FCR (Feed Conversion Ratio: kilos de alimento / kilos de ganancia de biomasa), tratamientos (p. ej., antiparasitarios), tipo de pellet y ración diaria.
- Ambientales (serie temporal): T° (temperatura del agua), DO (Dissolved Oxygen = oxígeno disuelto, mg/L), salinidad, corrientes, clima.
- Catálogo estático: centro (ubicación, zona), jaula (dimensiones), cohorte (año/mes de siembra), genotipo/stock, proveedor de smolt.
- Registros sanitarios/eventos: tratamientos, brotes (bloom), parásitos, inspecciones.
- Etiquetas (targets):
 - Clasificación: Alta Mortalidad Sí/No en la próxima semana (definir umbral, ej.: >X‰ de stock).
 - Regresión: Peso medio estimado a 7 días (g).

Serie temporal: datos ordenados por tiempo (día/semana).

Tabular: tabla clásica (filas=observaciones, columnas=variables).

Glosario

- **ML** (Machine Learning): modelos que aprenden patrones desde datos.
- **DL** (Deep Learning): subcampo de ML con **redes neuronales** profundas.
- **CNN** (Convolutional Neural Network): red para **imágenes**.
- **RNN/LSTM/GRU**: redes para **secuencias**; **LSTM** (Long Short-Term Memory) maneja dependencias largas.
- **TFT** (Temporal Fusion Transformer): arquitectura moderna para **series temporales multivariantes** con variables estáticas y dinámicas.
- **FCR** (Feed Conversion Ratio): eficiencia de conversión de alimento.
- **DO** (Dissolved Oxygen): oxígeno disuelto (mg/L).
- **Ventana (L)**: cantidad de días hacia atrás que el modelo observa.
- **Horizonte (H)**: cuántos días hacia adelante predice.
- **AUC-PR**: área bajo la curva Precisión–Recall (útil cuando la clase positiva es **rara**).
- **F1**: promedio armónico de precisión y recall.
- **MAE/RMSE**: errores de regresión (desviación promedio / cuadrática).
- **Leakage** (fuga de datos): usar sin querer información del futuro o del conjunto de validación en el entrenamiento.

Secuencia de aprendizajes

1) Análisis y comprensión del caso

- **Objetivo.** Entender el problema y definir formalmente tareas, métricas y restricciones.

Actividades

- **Delimitar:** predicción de **Alta Mortalidad (Sí/No)** a 7 días y **Peso** a 7 días por jaula.
- **Unidad de predicción:** una jaula en una fecha.
- **Métricas:**
 - Clasificación: **F1 macro** y **AUC-PR** (porque “Alta” es rara).
 - Regresión: **MAE** (error medio absoluto en gramos).
- **Restricciones:** modelo interpretable (explicar alertas), **latencia** de scoring diaria (<1 s por jaula), datos con **PII = no aplica** (pero sí confidencialidad).
- **Riesgos:** **drift estacional**, sensores fallando (DO/T°), jaulas nuevas sin historia (frío arranque).
- **Tarea/Entregable.** Ficha de 1 página con: problema, targets, métricas, restricciones, riesgos.
- **Ejemplo aplicado.** Umbral “Alta Mortalidad” = >0,8‰ del stock semanal. Horizonte H=7 días.

2) Preparación de los datos (pipeline sin fuga)

- **Objetivo.** Limpiar, transformar y organizar datos para que el modelo aprenda de manera válida.

Actividades - 1

Imputar faltantes; generar **features**: ventanas de 7/14/28 días, tasas ($\Delta\%$ peso), **FCR**, biomasa/jaula, lags de oxígeno/temperatura; normalizar por centro/temporada; codificar eventos sanitarios.

Actividades - 2

Tarea. Documentar pipeline y justificar: por qué esas ventanas, escalado por centro para evitar fuga entre centros.

Ejemplo aplicado. Construir dataset con índice (centro, jaula, fecha); window_size=28, horizon=7; escalado z-score aprendido **solo** en train por centro.

3) Diseño de la arquitectura de la red

- **Objetivo.** Definir un modelo adecuado para **series temporales multivariantes** y variables estáticas.

Actividades - 1

- **Clasificación y regresión conjuntas** (multitarea): un mismo backbone que alimente dos “cabezas” de salida.
- **Elegir arquitectura:**
 - **LSTM** (sencilla, buena base, interpreta secuencias).
 - **TFT** (Temporal Fusion Transformer): maneja **estáticas + dinámicas**, atenciones y “gating” para relevancia de variables.
- **Componentes clave:**
 - **Embeddings** para categóricas (centro, cohorte).
 - **Dropout** (regularización) para evitar sobreajuste.
 - **Head 1** (clasificación): capa densa + **sigmoid**.
 - **Head 2** (regresión): capa densa lineal (salida continua).
- **Pérdidas:**
 - Clasificación: **BCE** (Binary Cross-Entropy) con **class_weight** (desbalance).
 - Regresión: **MAE** o **MSE** (Mean Squared Error).
 - Pérdida total = suma ponderada (ej.: $0.6 \cdot \text{BCE} + 0.4 \cdot \text{MAE}$).
- **Entregable.** Diagrama de la arquitectura + tabla de hiperparámetros iniciales.

Actividades - 2

- **Ejemplo aplicado.** TFT con hidden=128, dropout=0.2; embeddings: centro(16), cohorte(8); salidas: p(alta_mortalidad) y peso_7d.
- **Dropout:** “apaga” neuronas al azar durante el entrenamiento para mejorar generalización.
Class weight: pesos mayores a la clase minoritaria para balancear el aprendizaje.

4) Implementación en código

- **Objetivo.** Organizar el proyecto y codificar de forma **reproducible**.
- **Entregable.** Script/Notebook ejecutable de punta a punta + config.yaml con hiperparámetros.
- **Tarea.** Notebook con celdas documentadas, configuración YAML (ventana, horizonte, HPs).
- **Ejemplo aplicado.** Script que genera tensores [batch, time, features], dos cabezales: clasificación mortalidad (sigmoid) y regresión peso (lineal).
- **AMP:** usa menor precisión en GPU para acelerar sin perder mucha exactitud.
Scheduler: plan para cambiar la **LR** (Learning Rate) durante el entrenamiento.
Early stopping: detener cuando la métrica en validación deja de mejorar.

5) Entrenamiento del modelo

- **Objetivo.** Ajustar pesos del modelo y **monitorear** el aprendizaje.

Actividades

- **Partición estricta por tiempo y entidad:** que una **jaula** no aparezca mezclada entre train y val en el mismo período.
- **Desbalance** en clasificación (pocos “Alta”): usar **class_weight** o **Focal Loss**.
- **Estabilidad:** **gradient clipping** (evitar explosión de gradientes), revisar tamaños de batch/ventana.
- **Registro:** curvas de **pérdida** y **métricas** (clasificación y regresión) por época; tiempos y VRAM.
- **Entregable.** Gráficos de curvas train/val; mejores checkpoints guardados.
- **Ejemplo aplicado.** clip_norm=1.0; **Focal Loss** $\gamma=2$ como prueba vs BCE estándar.
- **Gradient clipping:** limita el tamaño del gradiente para evitar inestabilidad.

6) Evaluación del modelo

- **Objetivo.** Medir desempeño realista y entender errores.

Actividades

- **Clasificación:** AUC-PR, F1 macro, matriz de confusión; **Recall** alto para “Alta” (alertas).
- **Regresión:** MAE y RMSE; gráfico predicho vs real (dispersión).
- **Subgrupos:** métricas por **centro, estación** (frío/cálido), **cohorte**.
- **Curvas:** Precision–Recall para calibrar **umbral operativo** de alerta.
- **Entregable.** Informe con tablas/gráficas + lista comentada de **casos difíciles** (p. ej., DO errático).
- **Ejemplo aplicado.** Umbral operativo elige **Recall \geq 0.90** en “Alta” (aceptamos más falsos positivos para no perder eventos críticos).
- **Umbral operativo:** valor de probabilidad a partir del cual disparo la alerta.
Calibración: ajustar probabilidades para que reflejen frecuencias observadas.

7) Optimización y *fine-tuning*

- **Objetivo.** Mejorar resultados sin romper costos/latencia.

Actividades

- **HPO** (Hyperparameter Optimization): LR, hidden, dropout, L y H.
- **Regularización**: más/menos **dropout**, **weight decay** (penaliza pesos grandes).
- **Comparativas**: LSTM vs TFT; añadir/remover variables (ablation).
- **Transfer**: entrenar primero solo las “heads” y luego “descongelar” capas del backbone.
- **Entregable**. Tabla “baseline vs variantes” con **métrica** y **costo** (tiempo, memoria).
- **Ejemplo aplicado**. Baseline LSTM $F1=0.57 \rightarrow$ TFT + embeddings estáticos $F1=0.63$ (+0.06) con +12% tiempo.
- **Weight decay**: regularización L2 para evitar sobreajuste.
Ablation: quitar variables para medir su aporte real.

8) Interpretabilidad y ética

- **Objetivo.** Explicar por qué el modelo alerta y asegurar uso responsable.

Actividades

- **Importancia de variables** (según TFT/LSTM con atención o métodos poshoc).
- **Casos explicados:** cuando alerta “Alta”, mostrar que **DO bajo** sostenido y **T° alta** fueron claves.
- **Revisión por subgrupos:** ¿peor en centros nuevos o invierno?
- **Uso responsable:** la alerta **no reemplaza** criterio veterinario; guía inspecciones.
- **Entregable.** Sección de interpretabilidad con ejemplos y **riesgos/mitigaciones**.
- **Ejemplo aplicado.** Top factores en alertas: DO mínimo 7 días, caída abrupta de DO 24 h, T°>14°C, FCR alto.
- **Saliency/atención:** técnicas para ver qué entrada influyó más en la predicción.

9) Presentación

- **Objetivo.** Comunicar resultados y límites a decisores.

Actividades

- **Model Card:** propósito, datos, entrenamiento, métricas globales y por subgrupos, límites, usos (apropiado/inapropiado), próximos pasos.
- **Slides (6–10):** historia clara: problema → datos → modelo → resultados → errores típicos → operación (umbral, latencia) → riesgos → roadmap.
- **Tarea/Entregable.** Model Card + presentación.
- **Ejemplo aplicado.** Recomendaciones: subir frecuencia de DO en centros con baja precisión; validar en dos centros nuevos antes de producción.

10) Operación y monitoreo

- Objetivo. Usar el modelo en la práctica y vigilar su salud.

Actividades

- **Scoring diario** por jaula (batch nocturno).
- **Monitoreo de drift** (cambios en distribución de DO/T°) y en **métricas** (AUC-PR semanal).
- **Alertas** integradas al flujo operativo (amarilla ≥ 0.6 , roja ≥ 0.8 , editable).
- **Entregable.** Plan de monitoreo + tablero simple (métricas y alertas por centro).