

## RETO 1 (Regresión): Predecir mortalidades semanales

**Archivo:** Mortalidades\_Centro de cultivos.csv

**Objetivo:** Predecir la cantidad de mortalidades semanales a partir de edad, especie, centro, densidad, alimentación, etc.

### Pasos a seguir:

1. Importar bibliotecas necesarias (pandas, numpy, seaborn, matplotlib, sklearn)
2. Cargar el archivo CSV
3. Explorar el dataset (.head(), .info(), .describe())
4. Tratar valores nulos (imputar con mediana)
5. Convertir tipos de datos adecuados
6. Detectar y tratar outliers (IQR)
7. Visualizar relaciones clave: Edad vs Mortalidad, Centro vs Mortalidad
8. Codificar variables categóricas (Centro, Especie, Alimentación) con OneHotEncoder
9. Separar variables predictoras (X) y objetivo (y)
10. Dividir datos en entrenamiento y prueba (80/20)
11. Entrenar modelo de regresión lineal
12. **Aplicar validación cruzada:**
  - o K-Fold Cross-Validation (k=5)
  - o Leave-One-Out (LOO)
13. Evaluar el modelo con:
  - o MSE (error cuadrático medio)
  - o  $R^2$  (coeficiente de determinación)
14. **Visualizar resultados con gráficos:**
  - o Gráfico reales vs predichos
  - o Boxplots de MSE y  $R^2$  por validación
15. **Crear un dashboard en HTML** que incluya:
  - o Tabla de métricas
  - o Gráficos comparativos de validación
  - o Interpretación automática del nivel de ajuste

---

## RETO 2 (Clasificación): ¿Fue una cosecha de alto rendimiento?

**Archivo:** Cosechas\_Cosechas\_2023.csv

**Objetivo:** Clasificar si una cosecha fue de alto o bajo rendimiento.

### Pasos a seguir:

1. Importar bibliotecas
2. Cargar y explorar el CSV
3. Imputar valores faltantes
4. Codificar variables categóricas (Cultivo, Región)
5. Crear variable binaria: rendimiento = alto (peso\_total > Q75), bajo (peso\_total < Q25)
6. Separar X e y
7. Dividir en entrenamiento/prueba
8. Entrenar modelo de clasificación (LogisticRegression o Árbol de decisión)
9. **Aplicar validación cruzada:**
  - o Stratified K-Fold (preserva proporciones de clase)
  - o Repeated K-Fold (k=5, rep=3)
10. Calcular métricas por validación:
  - o Accuracy
  - o Precision

- o Recall
  - o F1-Score
  - 11. **Crear dashboard en HTML** con:
    - o Tabla de métricas
    - o Gráfica de barras comparativa
    - o Interpretación automática de desempeño
    - o Matriz de confusión visual
- 

### **RETO 3 (Clasificación): ¿Hubo evento crítico de mortalidad?**

**Archivo:** Mortalidades\_Centro de cultivos.csv

**Objetivo:** Detectar si hubo un evento crítico (mortalidad > percentil 90)

#### **Pasos a seguir:**

1. Cargar y limpiar los datos
  2. Crear columna evento\_critico = 1 si mortalidad > P90
  3. Codificar variables categóricas
  4. Definir X e y
  5. Dividir en entrenamiento/prueba
  6. Entrenar modelo de clasificación
  7. **Aplicar validación cruzada:**
    - o Stratified K-Fold
    - o Leave-One-Out (si el dataset es pequeño)
  8. Calcular métricas de evaluación:
    - o F1-score, Accuracy, Matriz de confusión
  9. Visualizar resultados
  10. **Generar dashboard HTML** con:
    - o Análisis comparativo de validación
    - o Tabla de resultados
    - o Interpretación automática por modelo
- 

### **RETO 4 (Regresión): ¿Cuánto tarda una cosecha en completarse?**

**Archivo:** Cosechas\_Cosechas\_2023.csv

**Objetivo:** Predecir la semana en que se cosecha a partir de variables como cajas, peso total, tipo de cultivo y región.

#### **Pasos a seguir:**

1. Importar bibliotecas y cargar datos
2. Visualizar y preparar los datos
3. Codificar variables categóricas
4. Definir X e y (semana de cosecha)
5. Separar entrenamiento y prueba
6. Entrenar modelo de regresión lineal
7. **Validar el modelo usando:**
  - o K-Fold CV
  - o Repeated K-Fold
8. Calcular y graficar métricas:
  - o MSE
  - o  $R^2$
9. **Crear dashboard HTML** que incluya:

- o Tabla de resultados por validación
  - o Gráfico semana real vs predicha
  - o Boxplot de MSE y  $R^2$
  - o Conclusión sobre el nivel de ajuste
- 



### **Nota final para todos los retos:**



Se espera que cada reto termine con la **generación de un archivo HTML interactivo**, como se mostró en los ejemplos anteriores, que incluya:

- Tabla de resultados
- Gráficas con matplotlib o seaborn
- Interpretación automática del rendimiento
- Estilo visual con Bootstrap