Digit-Recognition

Pipeline completo de Machine Learning

Joaquín Cotrina Santos, Juan Miguel Fernández Tejada,
Francisco Javier Molina Cuenca, Alberto Ramírez Collado

Universidad

Diciembre de 2025

Índice

1. Estructura del Proyecto
2. Script Maestro
3. Preparación y Análisis de Datos
4. Ingeniería de Características
5. Entrenamiento de Modelos
6. Evaluación y Modelo Final
7. Funciones Auxiliares
8. Conclusiones

Estructura General

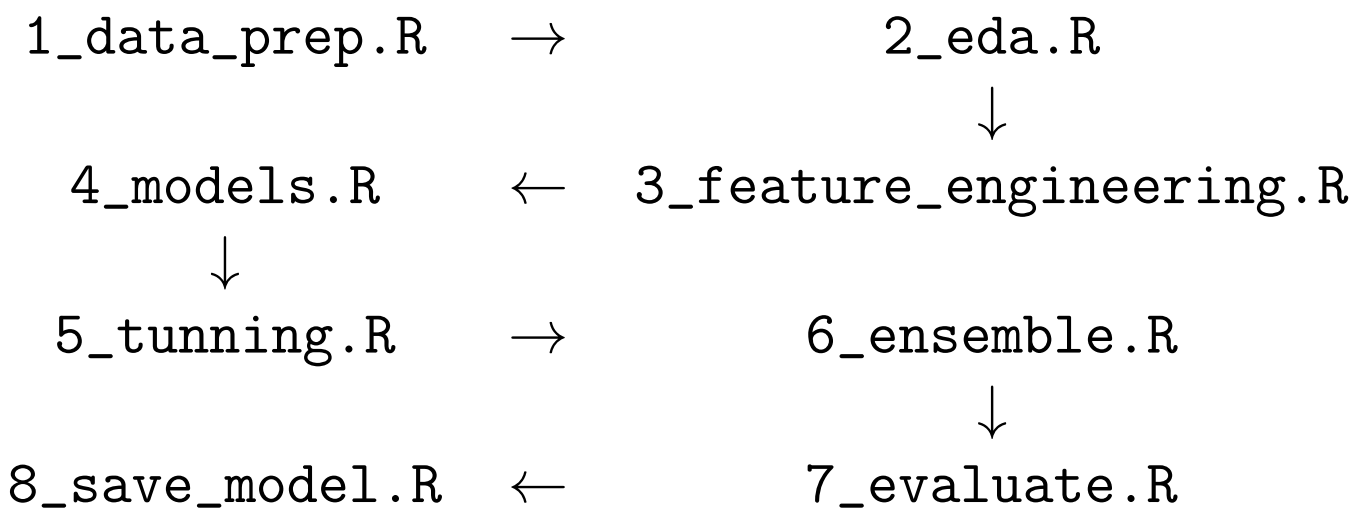
- Proyecto organizado como un **pipeline completo de Machine Learning**. Dicha organización de carpetas es:
  - **data**/: datos originales (`raw`) y datos procesados (`processed`).
  - **models**/: modelos base, modelos ajustados, meta-modelo y modelo final.
  - **results**/: resultados gráficos y métricas de evaluación.
  - **scripts**/: todos los scripts del pipeline junto con `utils.R` y `run_all.R`.
- Flujo automatizado: desde los datos en bruto hasta el modelo final.
- Desglose en scripts independientes y reproducibles.
- Permite ejecutar el proyecto de forma secuencial o por bloques gracias al enfoque modular.

run_all.R

- Script orquestador que controla toda la ejecución del proyecto.
- Verifica e instala las librerías necesarias automáticamente.
- Crea las carpetas del proyecto si no existen.
- Ejecuta los scripts en el orden correcto con control de errores.

Ejecución Secuencial

Orden de ejecución:



- Cada script se ejecuta con `tryCatch`.
- Si un paso falla, el pipeline se detiene automáticamente.
- Se registra el tiempo de ejecución de cada fase.
- Permite localizar fácilmente cuellos de botella.

Archivos

1_data_prep.R

- Carga del fichero original `train.csv`.
- Conversión de las etiquetas numéricas en `L0-L9` para evitar errores en `caret`.
- Normalización de todos los píxeles al rango $[0, 1]$.
- Reducción controlada a 20.000 observaciones por limitaciones de memoria.
- División estratificada: 80 % entrenamiento / 20 % validación.

2_eda.R

- Análisis de la distribución real de las clases.
- Visualización directa de ejemplos representativos de cada dígito.
- Estudio de la densidad de píxeles activos por clase (sparsity).
- Generación del histograma global de activación de píxeles.

Archivos

3_feature_engineering.R

- Eliminación de píxeles irrelevantes mediante selección por varianza.
- Reducción de cientos de variables a un espacio compacto mediante PCA.
- Conservación del 95 % de la varianza original.
- Obtención de los datasets finales `train_pca` y `valid_pca`.

4_models.R

- Entrenamiento de los **modelos base (baseline)** sobre datos PCA.
- Árbol de decisión como clasificador interpretable.
- Random Forest como modelo basado en ensamblado de árboles.
- SVM lineal como clasificador de margen máximo.
- Perceptrón multicapa (MLP) como modelo neuronal no lineal.
- Evaluación preliminar rápida sobre validación.

Archivos

5_tunning.R

- Ajuste de hiperparámetros con **validación cruzada 3-fold**.
- Optimización de `mtry` en Random Forest.
- Ajuste del coste en la SVM lineal.
- Poda del árbol mediante el parámetro `cp`.
- Ajuste de arquitectura del MLP (`size` y `decay`).
- Uso de una muestra estratificada del 10 % para reducir coste.
- Obtención de **cuatro modelos optimizados**.

6_ensemble.R

- Construcción de un sistema de clasificación por **stacking**.
- Uso de las probabilidades de los **4 modelos base** como meta-variables.
- Random Forest, SVM, Árbol y MLP como fuentes de información.
- Limpieza de valores inválidos mediante `na.omit`.

Archivos

7_evaluate.R

- Generación de predicciones probabilísticas de los 4 modelos base.
- Construcción del conjunto de meta-características de validación.
- Predicción final mediante el meta-modelo.
- Construcción de la matriz de confusión.
- Cálculo e impresión de la **accuracy final del ensemble**.
- Guardado del informe completo en `results/evaluation.txt`.

8_save_model.R

- Restricción obligatoria a las **100 primeras observaciones**.
- Generación de meta-características con los **4 modelos**.
- Reentrenamiento del meta-modelo final reducido.
- Conserva la estructura completa del ensemble.
- Guardado del modelo definitivo en `models/final_model_100.rds`.

utils.R

- `set_seed`: control de reproducibilidad.
- `plot_mnist_row`: visualización directa de dígitos.
- `save_model` y `load_model`: persistencia de modelos.
- `pixels_to_matrix`: conversión eficiente a matrices normalizadas.

Conclusiones

- Implementación completa de un pipeline real de Machine Learning.
- Uso combinado de reducción de dimensiones y técnicas de ensemble.
- Sistema completamente reproducible y automatizado.
- Estructura modular preparada para ampliaciones futuras.

Fin

Gracias por la atención