

Informe Práctica 3: Algoritmos supervisados y métodos Ensemble

Mario Polaino, Antonio Moya, Leon Skalczynski, Fernando García

`mariopolaino@uma.es, amoyam05@uma.es, leonskal@uma.es, nano_2005@uma.es`
3º Software. Universidad de Málaga.

1 Introducción

En esta práctica trabajamos con distintos algoritmos de clasificación supervisada y con la construcción posterior de métodos Ensemble para mejorar su rendimiento. Los conjuntos de datos utilizados provienen de la práctica anterior, en la cual se realizaron todas las tareas de preprocesado: estandarización, normalización, reducción mediante PCA y, de forma especialmente relevante, la generación de los cinco pliegues correspondientes a la validación cruzada.

De este modo, en lugar de construir los pliegues en esta práctica, hacemos uso directo de dichos conjuntos de entrenamiento y prueba ya preparados. A partir de ellos, entrenamos distintos clasificadores para cada combinación de dataset y fold, evaluamos su comportamiento individual y, finalmente, estudiamos cómo la combinación de modelos mediante técnicas Ensemble (votación, media y mediana) puede mejorar la estabilidad y el rendimiento global del sistema.

2 Entrenamiento de modelos (`train.ipynb`)

El archivo `train.ipynb` contiene el proceso de entrenamiento de los clasificadores utilizados en la práctica. Para facilitar su comprensión, organizamos su funcionamiento en varias etapas, siguiendo la estructura lógica del notebook.

2.1 Definición de modelos supervisados

En una primera celda se definen los cuatro métodos de clasificación que se emplearán a lo largo de la práctica. Cada uno se inicializa con una configuración concreta:

- **KNN**: implementado mediante `KNeighborsClassifier`, usando $k = 3$ vecinos.
- **SVM**: modelo `SVC` con la opción `probability=True` activada.
- **Naive Bayes**: usando `GaussianNB`.
- **Random Forest**: compuesto por 100 árboles y semilla fija para garantizar reproducibilidad.

2.2 Estructura de carpetas y datasets

Se define un diccionario que relaciona cada variante del dataset con su ubicación física y sufijo de archivo. Esto permite reutilizar el mismo bloque de código para cargar cualquiera de las nueve versiones del conjunto de datos generadas previamente.

Cada combinación (`dataset, fold`) corresponde a un archivo CSV de entrenamiento de la forma:

```
training{fold}{suffix}.csv
```

2.3 Función de entrenamiento por modelo y fold

El núcleo del archivo es la función `train_and_save_model()`, responsable de:

1. Localizar el archivo de entrenamiento correspondiente.
2. Cargar los datos separando características y etiquetas.
3. Seleccionar el modelo adecuado según el método indicado.
4. Entrenar mediante `fit`.
5. Guardar el modelo entrenado en formato `.pk1`, identificado por fold, dataset y método.

Finalmente, el notebook ejecuta esta función para las cinco iteraciones de validación cruzada existentes y para cada una de las variantes del dataset, generando así todos los modelos necesarios para la fase de evaluación.

3 Evaluación de los modelos base (eval.ipynb)

El archivo `eval.ipynb` se encarga de evaluar cada modelo individual entrenado previamente.

3.1 Carga de modelos y datos de test

Para cada fold y cada dataset, se carga el archivo `test` correspondiente:

```
test{fold}{suffix}.csv
```

A continuación, se carga el modelo asociado mediante `joblib.load`, lo que permite evaluar cada combinación de forma independiente.

3.2 Generación de predicciones y probabilidades

Cada clasificador predice:

- La clase esperada mediante `predict`.
- Las probabilidades por clase mediante `predict_proba`, cuando el modelo lo permite.

En caso de que un modelo no implemente `predict_proba`, se genera una matriz de probabilidad manual construida a partir de la clase predicha.

3.3 Cálculo de métricas mediante un enfoque One-vs-All

El notebook implementa el cálculo manual de:

- Matriz de confusión
- TP, FP, FN y TN por clase
- Sensibilidad, Especificidad, FPR y FNR
- Accuracy, Precision, Recall y F1-Score
- AUC multiclase mediante `roc_auc_score`

Para cada combinación evaluada se generan valores globales promediando las métricas de cada clase.

3.4 Almacenamiento de resultados y predicciones

Se guardan dos tipos de archivos:

- Predicciones individuales en la carpeta `Predicciones/`.
- Métricas globales añadidas al fichero: `Resultados/all_metrics_raw.csv`

Este fichero es utilizado posteriormente para generar resultados agregados y ensembles.

4 Métodos Ensemble y análisis final (`results.ipynb`)

El archivo `results.ipynb` constituye la fase final del proyecto, donde se combinan los resultados individuales de los clasificadores y se generan análisis comparativos.

4.1 Carga de resultados base

Se inicia cargando los datos de `all_metrics_raw.csv`, así como todas las predicciones individuales almacenadas en la carpeta `Predicciones/`.

4.2 Construcción de métodos Ensemble

Se implementan tres métodos de combinación:

- **Votación mayoritaria:** selecciona la clase más votada entre los modelos base.
- **Media:** media de las probabilidades por clase.
- **Mediana:** mediana de las probabilidades, reduciendo la influencia de valores extremos.

Cada uno de estos métodos combina los resultados de las cuatro métricas anteriormente mencionadas.

4.3 Cálculo de métricas para los Ensemble

Se reutiliza la función de métricas definida en `eval.ipynb`, permitiendo comparar directamente los modelos base con los modelos Ensemble.

4.4 Cálculo de estadísticas agregadas

Los resultados se agrupan por (`Método`, `Dataset`) y se calcula la media y la desviación típica de cada métrica, exportándose todo a ficheros CSV para su análisis posterior.

4.5 Generación de gráficas de análisis

Finalmente, se generan distintas visualizaciones que permiten interpretar mejor las diferencias entre los modelos. A continuación se muestran las tres principales:

FPR vs FNR:

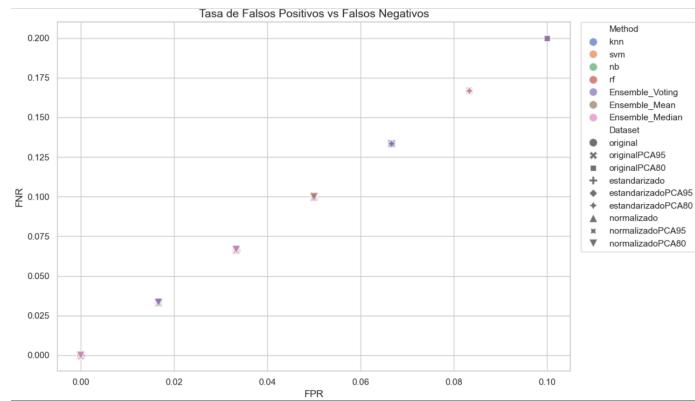


Fig. 1. Relación entre FPR y FNR para los distintos clasificadores y datasets.

Precisión vs Recall:

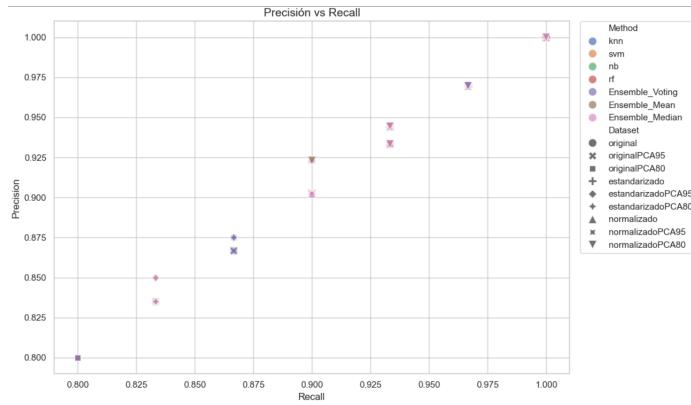


Fig. 2. Relación entre Precisión y Recall para los clasificadores evaluados.

Accuracy vs F1-Score:

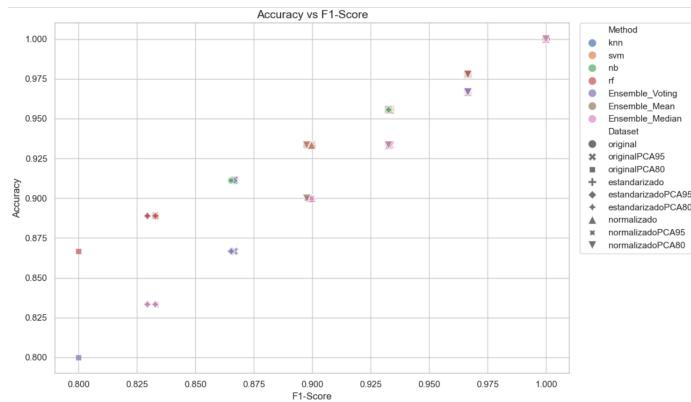


Fig. 3. Comparativa entre Accuracy y F1-Score.

Estas figuras se guardan en la carpeta **Figuras/** y se encuentran listas para añadirse al informe.

5 Resultados y análisis

En general, los distintos clasificadores obtienen valores muy altos en todas las métricas. SVM y Random Forest son los métodos que muestran un comportamiento más sólido y estable, manteniendo Accuracy y F1-Score muy próximas al valor óptimo. K-NN también funciona bien, aunque su rendimiento se resiente

ligeramente cuando la reducción de dimensionalidad es más agresiva. Naive Bayes es el modelo que más varía entre datasets, especialmente en las versiones con PCA al 80%, donde se aprecia una pequeña pérdida de calidad. El preprocessado influye de forma clara: trabajar con los datos normalizados o estandarizados tiende a ofrecer resultados más constantes y con tasas de error muy bajas. La reducción mediante PCA funciona bien cuando se conserva el 95% de la varianza, pero al reducir al 80% sí se pierde algo de información útil para algunos modelos. Por último, los métodos Ensemble ayudan a suavizar estas diferencias y proporcionan predicciones más regulares, aportando pequeñas mejoras cuando algún clasificador individual muestra más variabilidad. En conjunto, el rendimiento global es excelente en prácticamente todos los casos.

6 Conclusiones

A lo largo de esta práctica hemos comparado diferentes clasificadores supervisados y estudiado la mejora que supone combinarlos mediante métodos Ensemble. El uso de los pliegues generados en la práctica anterior ha permitido evaluar los modelos de manera más estable y realista.

Los métodos Ensemble han demostrado ser una herramienta eficaz para aumentar la robustez y mejorar el rendimiento global de los clasificadores, especialmente en métricas como el F1-Score.

7 Uso de inteligencia artificial

Nos hemos apoyado en esta herramienta para aclarar conceptos teóricos, así como para la ayuda y corrección de errores y repaso del código.



Fig. 4. Logo de la Universidad de Málaga