DEPARTAMENTO DE COMPUTACION

Facultad de Ciencias Exactas y Naturales - UBA

# Trabajo Práctico 2

**Kuzushiji-MNIST**

9 de noviembre de 2025 Laboratorio de Datos

**burritodiabolico**


| Integrante | LU | Correo electrónico |
|---|---|---|
| Faltlhauser, Andres | 1255/24 | faltlhauserandres@gmail.com |
| Kirschbaum, Owen | 1310/24 | owenkir17@gmail.com |
| Yurzola, Marcos | 1235/24 | myurzol@gmail.com |





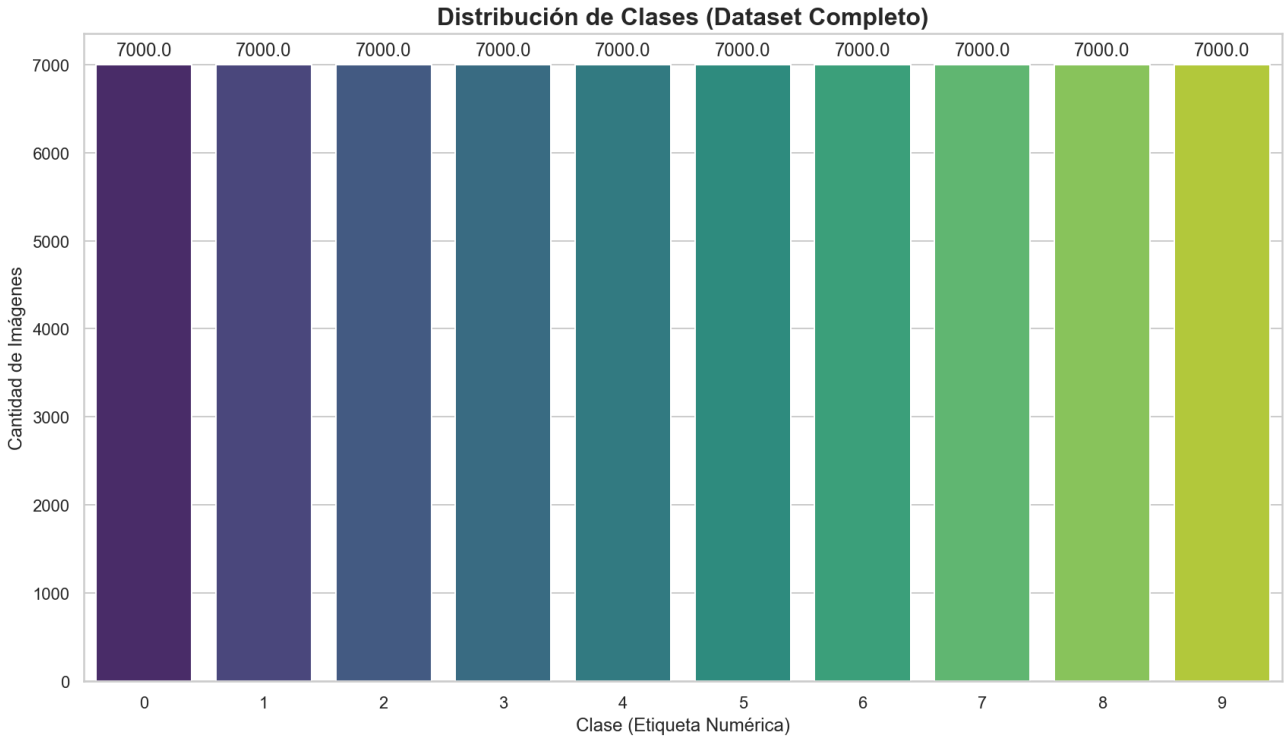

**Facultad de Ciencias Exactas y Naturales**
Universidad de Buenos Aires
Ciudad Universitaria - (Pabellón I/Planta Baja)
Intendente Güiraldes 2610 - C1428EGA
Ciudad Autónoma de Buenos Aires - Rep. Argentina
Tel/Fax: (++54 +11) 4576-3300
http://www.exactas.uba.ar

# 1. Introducción

Este trabajo práctico aborda un problema de **aprendizaje automático supervisado**: la clasificación de imágenes. El dataset a analizar es el **Kuzushiji-MNIST**, una colección de 60,000 imágenes de caracteres japoneses antiguos manuscritos.

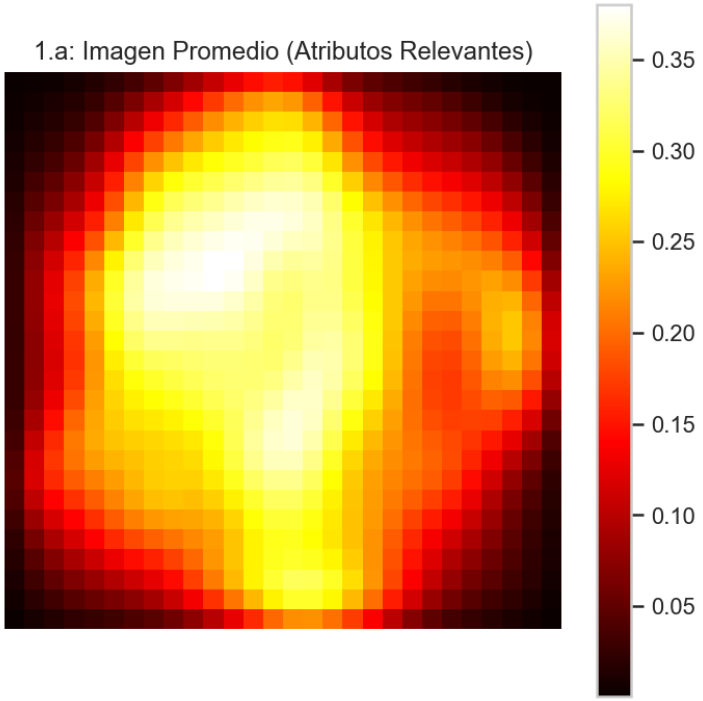
El objetivo es predecir una de **10 clases** (etiquetadas de 0 a 9) a partir de una imagen. Para ello, se cargó el dataset kuzushiji_full.csv, que se compone de:

- **Cantidad de datos**: 70,000 instancias (imágenes).
- **Variable de interés (y)**: 1 columna (la última, col 784) con la etiqueta de la clase (0-9).
- **Atributos (X)**: 784 columnas de píxeles (columnas 0-783) que representan una imagen de 28x28.
- **Tipo de Atributos**: Numéricos. Se normalizaron de su rango original (0-255) al rango [0, 1] para un mejor rendimiento del modelo.




Una característica relevante inicial es el **balanceo de clases**. Como se observa en el Gráfico de Distribución, el dataset está **perfectamente balanceado**: cada una de las 10 clases contiene exactamente **7,000 imágenes**. Esto es importante, ya que valida el uso de la métrica **Accuracy** como principal evaluador del rendimiento de nuestros modelos, dado que no será engañosa.

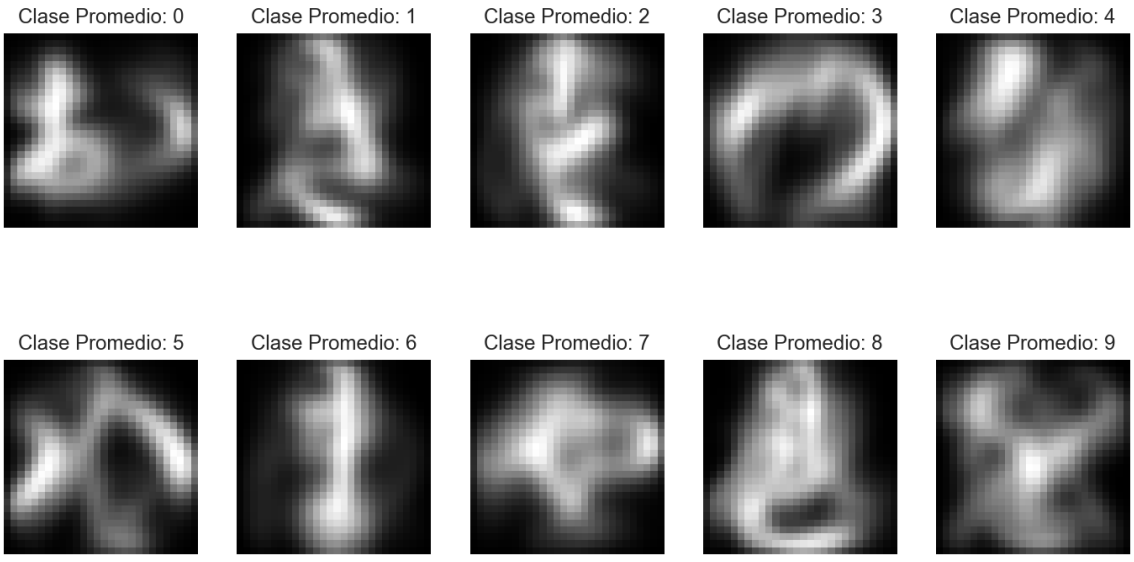
A continuación, se responden las preguntas específicas del enunciado.




**a. ¿Cuáles parecen ser atributos relevantes para predecir el tipo de carácter al que corresponde la imagen? ¿Cuáles no? ¿Se pueden descartar atributos?**

- **Justificación**: El Gráfico 1.a visualiza la ”imagen promedio” de las 60,000 imágenes. Los colores más cálidos (amarillo/rojo) indican píxeles que se activan con frecuencia (alta varianza), mientras que los colores fríos (negro/azul) indican píxeles que rara vez se activan (baja varianza).
- **Respuesta**: Los atributos relevantes son los píxeles ubicados en el recuadro central de la imagen, donde se concentra la ”actividad” y la forma de los caracteres. Los atributos irrelevantes son los píxeles ubicados en los bordes y, especialmente, en las esquinas.
- **Descarte de Atributos**: Sí, se pueden descartar atributos. Los píxeles de los bordes tienen una varianza muy cercana a cero (son negros en casi todas las imágenes). Descartar estos atributos (por ejemplo, mediante una técnica de selección de características como VarianceThreshold) podría reducir el costo computacional del entrenamiento (menos dimensiones) sin perder información significativa para el clasificador.

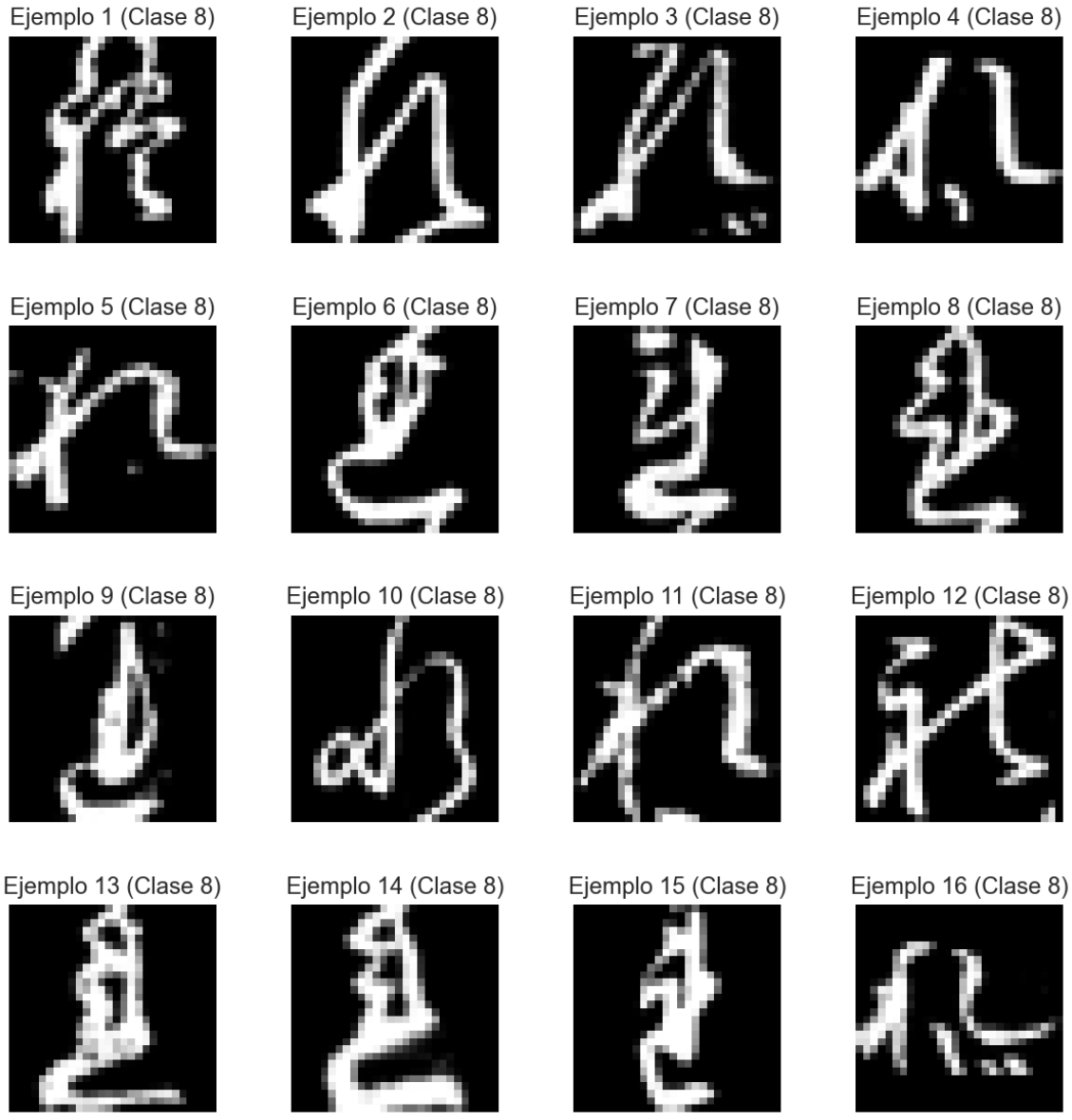
**1.b: Similitud entre Clases (Imagen Promedio por Clase)**



**b. ¿Hay caracteres que son más parecidos entre sí? Por ejemplo, ¿Qué es más fácil de diferenciar: las imágenes correspondientes a la clase 2 de las de la clase 1, ó las de la clase 2 de la clase 6?**

- **Justificación**: El Gráfico 1.b muestra la ”imagen promedio” de cada una de las 10 clases, lo que nos da una idea de la forma ”ideal” o arquetípica de cada carácter.
- **Respuesta**: Sí, existen pares de clases con alta similitud visual. Respondiendo al ejemplo:
  - **Clase 2 vs. Clase 1**: Son visualmente muy distintas. La Clase 1 promedio es un carácter simple y vertical, mientras que la Clase 2 (que corresponde a ”す”) es una forma más compleja y horizontal. Es fácil diferenciarlas.
  - **Clase 2 vs. Clase 6**: Son visualmente muy similares. La Clase 2 (”す”) y la Clase 6 (”ま”) son caracteres curvos que ocupan un espacio casi idéntico. La principal diferencia reside en un pequeño trazo superior en la Clase 6 que no está en la 2.
- **Conclusión**: Es mucho más fácil diferenciar la Clase 2 de la 1. La similitud entre la Clase 2 y la 6 será, previsiblemente, una fuente de error común para nuestros modelos, y requerirá que el clasificador aprenda a identificar patrones muy sutiles.

## 1.c: Similitud Intra-Clase (Muestras Aleatorias Clase 8)



**c. Análisis de la variabilidad dentro de la clase 8**

- **Justificación**: El Gráfico 1.c muestra 16 imágenes aleatorias tomadas exclusivamente de la Clase 8. Esto nos permite evaluar la varianza dentro de una misma clase.
- **Respuesta**: No, las imágenes no son muy similares entre sí. A pesar de que todas representan el mismo carácter (Clase 8, "𝓀"), se observa una alta varianza intra-clase.
- **Conclusión**: Las imágenes varían significativamente en el grosor del trazo, la inclinación, la escala y la posición exacta del carácter dentro del recuadro. Esta variabilidad en la escritura a mano es un desafío central: el modelo no puede "memorizar" una única forma para la Clase 8, sino que debe aprender a generalizar cuáles son las características fundamentales que definen a un "8" a pesar de estas distorsiones.

**d. Este dataset está compuesto por imágenes ¿Esto complica la exploración de los datos?**

- **Respuesta**: Sí, complica la exploración de datos de manera exponencial.
- **Justificación**: En un dataset tabular como Titanic, tenemos pocos atributos (ej. 10) y cada uno tiene un significado semántico directo (ej. "Edad", "Sexo", 'Clase de Ticket"). Podemos hacer un histograma de la "Edad" y obtener conclusiones directas.
- **Conclusión**: En este dataset, tenemos 784 atributos (píxeles). Un atributo individual no tiene ningún significado por sí solo; su valor no nos dice nada. La información no está en los atributos individuales, sino en la relación espacial y conjunta entre ellos. Por lo tanto, no podemos usar histogramas simples. Debemos recurrir a técnicas 2D o a métodos avanzados de reducción de dimensionalidad (como PCA o t-SNE) para poder "ver" y explorar la estructura de los datos.

# 2. Experimentos realizados

## 2.1. Clasificación binaria

Dada una imagen se desea responder la siguiente pregunta: **¿la imagen corresponde a la clase 4 o a la clase 5?**

**a. Análisis del balance de clases y cantidad de muestras sobre este subconjunto de datos**

De este subconjunto de datos podemos ver que obtenemos 14000 muestras. Está perfectamente balanceado ya que 7000 pertenecen a la clase 4 y 7000 pertenecen a la clase 5, esto lo logramos ver con este bloque de código, que dio como resultado cantidad_de_muestras = 14000, valores_clase_4 = 7000 y valores_clase_5 = 7000.

```
cantidad_de_muestras = y_data4_5.size
valores_clase_4 = y_data[y_data == 4].size
valores_clase_5 = y_data[y_data == 5].size
```

**c y d. Comparación de modelos de KNN utilizando distintos atributos y distintos valores de k.**

El objetivo fue comparar cómo varía el rendimiento del modelo (medido con Accuracy) al modificar dos factores: La cantidad de atributos (píxeles) utilizados y el método de selección para elegir dichos atributos. Decidimos medirlo con accuracy ya que el set de datos estaba perfectamente balanceado (7000 muestras de clase 4 y 7000 de clase 5).

En todos los casos, los atributos se seleccionaron "aprendiendo" únicamente del set de X_train y luego se aplicó esa selección al X_test. Se probaron k = [3, 5, 10] vecinos para cada experimento.

- **Experimento 1**: Cantidad Reducida de atributos (3).

  Siguiendo la sugerencia del enunciado, nuestro primer modelo utilizó una cantidad mínima de 3 atributos. El criterio de selección fue simple, elegimos los 3 píxeles con el mayor brillo promedio (calculado con X_train.mean()).

  **Resultado**: El mejor accuracy obtenido fue de aprox 59 % con k = 5. Este resultado demuestra que, si bien el modelo funciona mejor que el azar (50 %), 3 píxeles no son suficientes para capturar la complejidad de los caracteres.

- **Experimento 2**: Mayor Cantidad de atributos (50).

  Para el segundo experimento, aumentamos la cantidad de atributos a 50, pero mantuvimos el mismo método de selección (X_train.mean()).

  **Resultado**: El rendimiento mejoró drásticamente, alcanzando un accuracy máximo de 91.6 % (con k = 10). Esto comprueba que la cantidad de atributos es un factor importante. Pasar de 3 a 50 píxeles le dio al modelo suficiente información para diferenciar la mayoría de las imágenes.

- **Experimento 3**: Método de Selección Robusto con mayor cantidad de atributos (50)

  Finalmente, intentamos mejorar el accuracy cambiando el método de selección. Mantuvimos 50 atributos, pero en lugar de elegir por brillo promedio con mean(), usamos un criterio más robusto (SelectKBest(f_classif)). Este método no elige los píxeles más brillantes, sino los 50 píxeles con el mayor poder discriminativo (aquellos cuyos valores son más diferentes entre la clase 4 y la clase 5).

  **Resultado**: El accuracy volvió a mejorar, alcanzando un 93.9 % (con k = 5).

**Conclusiones**: La cantidad Importa: Aumentar el número de atributos de 3 a 50 (Experimento 1 vs. 2) fue el salto de rendimiento más grande, elevando el accuracy de 59 % a 91.6 %. La calidad importa más: Comparando el Experimento 2 y el 3 (ambos con l=50), demostramos que el método de selección es clave. Elegir atributos por su poder discriminativo (f_classif, 93.9 %) es superior a elegirlos por su brillo promedio (mean, 91.6 %).

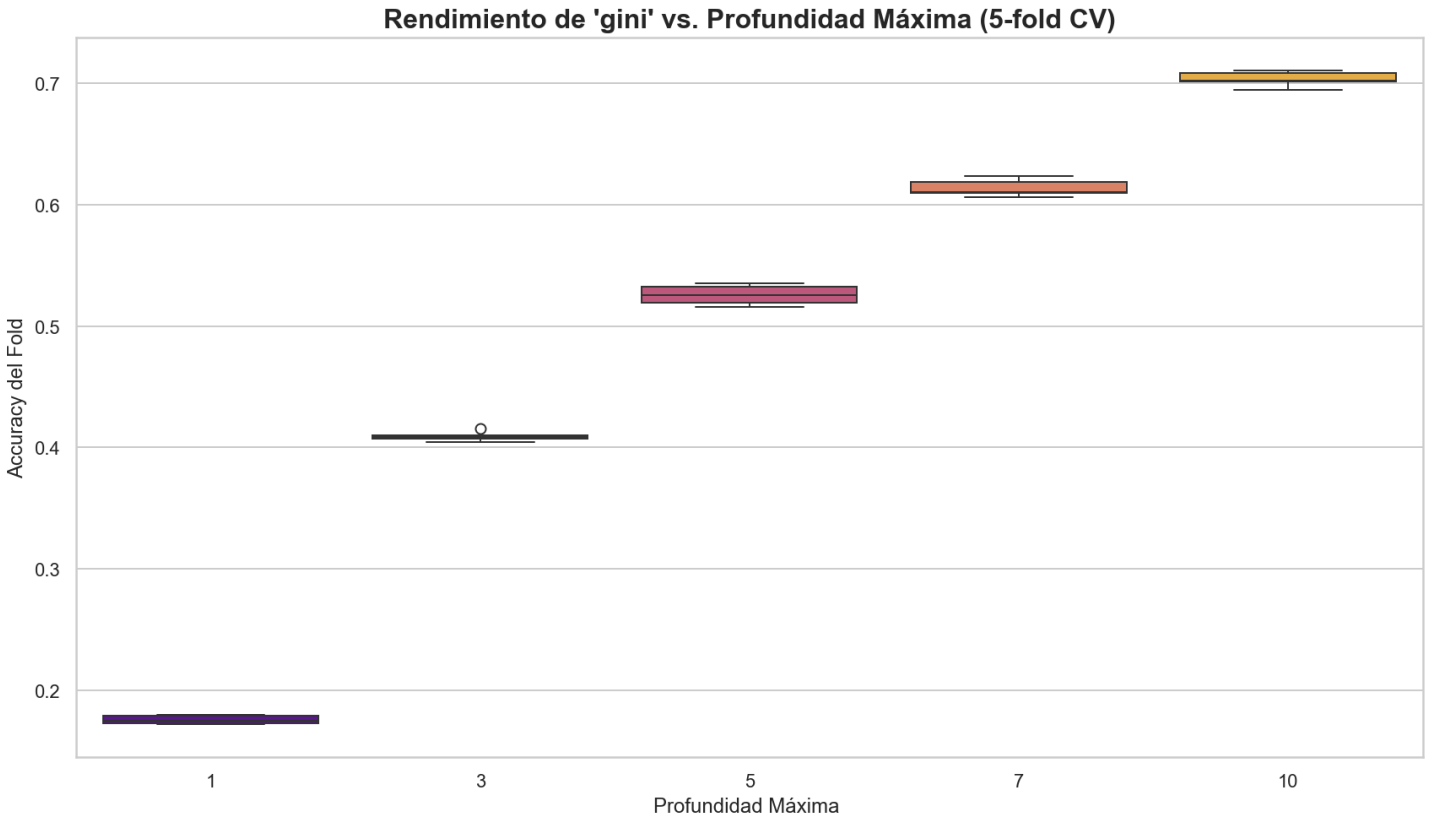
## 2.2. Clasificación multiclase

Dada una imagen se desea responder la siguiente pregunta: **¿A cuál de las 10 clases corresponde la imagen?**. Para ello comparamos dos tipos de Árboles de Decisión (gini vs. entropy) y con el objetivo de encontrar la mejor profundidad (max_depth) para cada uno. Con este fin, se siguió el procedimiento de selección de modelos visto en la cátedra:

- **Conjunto de Desarrollo (80 %)**: Se utilizó el 80 % de los datos (56,000 imágenes) para realizar una **Validación Cruzada de 5-Folds (K-Fold)**.
- **Conjunto de Evaluación (20 %)**: El 20 % restante (14,000 imágenes) se reservó como "held-out test set" y no se utilizó hasta la evaluación final.

### b. Árbol de Decisión (criterion=”gini”)

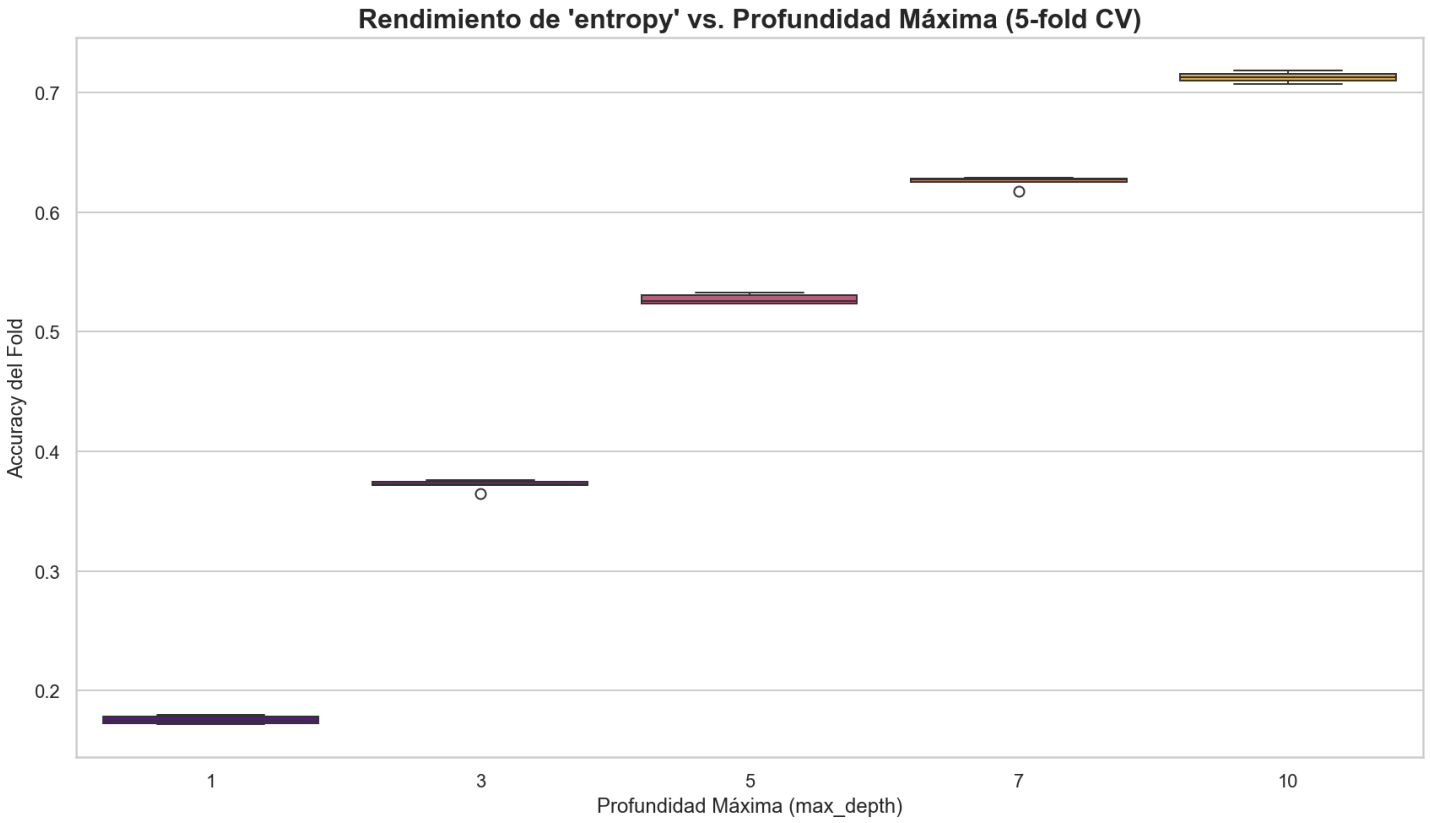
En el primer experimento, se evaluó el DecisionTreeClassifier usando su criterio por defecto, ’gini’, probando profundidades máximas (max_depth) de entre 1 a 10. El rendimiento promedio de los 5 folds para cada profundidad se muestra en el siguiente gráfico.




- **Observaciones**: El gráfico de cajas muestra cómo evoluciona el Accuracy a medida que aumenta la profundidad.
    - Con profundidades bajas (max_depth=1 o 3), el modelo es demasiado simple, tiene un Accuracy bajo y poca varianza (cajas cortas).
    - A partir de max_depth=7, el Accuracy promedio comienza a estabilizarse.
    - En max_depth=10, el rendimiento es alto, pero la caja se vuelve más ancha, sugiriendo que el modelo empieza a sobreajustarse (overfitting) a los folds específicos de entrenamiento, perdiendo generalidad.
    - **Resultado**: El mejor rendimiento promedio en la validación cruzada se alcanzó con max_depth=10, logrando un **Accuracy promedio de 70.37 %**.

### Árbol de Decisión (criterion=”entropy”)

En el segundo experimento, se repitió el mismo procedimiento de K-Fold CV, pero cambiando el hiperparámetro a criterion=”entropy” para medir la ganancia de información.

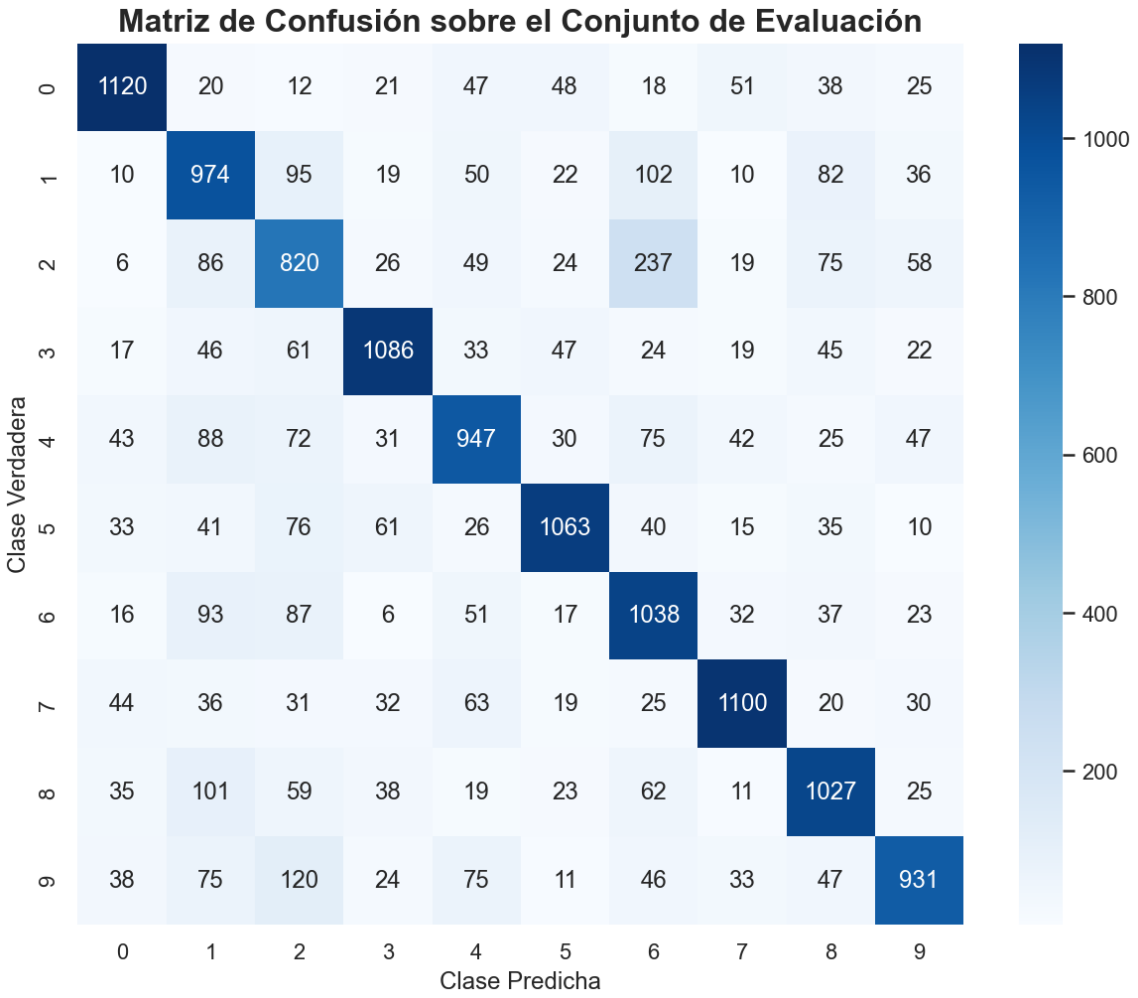



- **Observaciones**: El comportamiento del criterio éntropyés muy similar al de 'gini'. El rendimiento aumenta significativamente con la profundidad, alcanzando su punto máximo también en max_depth=10.
- **Resultado**: El mejor rendimiento promedio para "entropy" fue con max_depth=10, logrando un **Accuracy promedio de 71.26 %**.

### d. Selección y Evaluación Final

La fase de validación cruzada (incisos "b" y "c") es una competencia para seleccionar el mejor modelo.

- **Selección del Campeón**:
  - "gini" (max_depth=10): 70.37 %
  - "entropy" (max_depth=10): **71.26 %**
- El modelo **DecisionTreeClassifier(criterion="entropy", max_depth=10)** fue seleccionado como el "modelo campeón", ya que obtuvo el mayor Accuracy promedio en la validación.
- **Evaluación Final (Held-Out Test Set)**: Siguiendo la metodología, el modelo campeón fue re-entrenado utilizando el conjunto de desarrollo completo (las 56,000 imágenes de X_dev). Luego, se evaluó **una única vez** contra el conjunto de evaluación (X_eval, y_eval) que había permanecido separado. El Accuracy final obtenido por el modelo campeón en el conjunto de evaluación fue: 72.19 %.




- **Puntos Fuertes (Efectividad Alta)**: La efectividad general del modelo, con un Accuracy superior al 72 %, se confirma al observar la diagonal principal (de arriba-izquierda a abajo-derecha).
- **Alto Número de Aciertos**: Todos los valores en la diagonal son superiores a 1000. Esto indica que el modelo no tiene un "punto ciego"; es capaz de identificar correctamente la mayoría de las instancias de todas las clases.
- **Mejores Clases Identificadas**: El modelo es particularmente efectivo para identificar las clases **0** (1120), **3** (1086) y **7** (1110). Estas son las clases que el modelo reconoce con mayor fiabilidad.

## 3. Conclusiones

A lo largo de este trabajo práctico se realizó un análisis exploratorio, la construcción de modelos de clasificación binaria y multiclase, y una evaluación comparativa de su rendimiento sobre el dataset **Kuzushiji-MNIST**, un conjunto de datos de imágenes manuscritas de caracteres japoneses antiguos.

Desde la etapa inicial de **análisis exploratorio**, se identificaron patrones relevantes en los píxeles centrales de las imágenes y se comprobó que los bordes contienen poca o ninguna información útil. Esto permitió justificar la reducción de atributos sin pérdida significativa de desempeño. Además, se observó que el dataset está perfectamente balanceado, lo que habilitó el uso de la Accuracy como métrica principal de evaluación.

En la etapa de **clasificación binaria**, los experimentos demostraron que tanto la cantidad como la calidad de los atributos influyen fuertemente en el rendimiento del modelo KNN. Aumentar el número de píxeles considerados y seleccionar pixeles mas significativos (utilizando por ejemplo SelectKBest) mejoró significativamente el desempeño, alcanzando un accuracy máximo de **93.9 %**.

En la etapa de **clasificación multiclase**, los modelos de Árboles de Decisión alcanzaron resultados satisfactorios considerando la complejidad del problema. La comparación entre los criterios gini y entropy mostró

diferencias leves, con una ligera ventaja para entropy (71.26 % frente a 70.37 %). El modelo final seleccionado (**DecisionTreeClassifier** con criterion="entropy" y max_depth=10) logró un accuracy de **72.19 %** en el conjunto de evaluación, confirmando su capacidad para generalizar adecuadamente.

## Bibliografia


- Clanuwat, Tarin, Mikel Bober-Irizar, Asanobu Kitamoto, Alex Lamb, Kazuaki Yamamoto, and David Ha. **"Deep learning for classical japanese literature."** arXiv preprint arXiv:1812.01718 (2018).


## Fuentes de datos

Los datos utilizados en este trabajo:

- **Kuzushiji-MNIST.:**
  `https://github.com/rois-codh/kmnist`
- **MNIST:**
  `https://en.wikipedia.org/wiki/MNIST_database`