DEPARTAMENTO
DE COMPUTACION
Facultad de Ciencias Exactas y Naturales - UBA

Trabajo Práctico 1

2c 2024 Laboratorio de Datos

Integrante	LU	Correo electrónico
Castro, Lucia	278/21	licastro@dc.uba.ar
Padulo R., Javier	361/05	rjpadulo@gmail.com
Flores, Leandro	277/16	leannicolasflores@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

Índice

1. Introducción

El presente trabajo práctico tiene como objetivo aplicar los conocimientos adquiridos sobre clasificación y selección de modelos, así como la validación cruzada, utilizando el conjunto de datos MNIST-C en su versión "Motion Blur". Este dataset consiste en imágenes de dígitos manuscritos (del 0 al 9) con un nivel de corrupción por desenfoque de movimiento. A través de este trabajo, se busca desarrollar un modelo de clasificación capaz de predecir el dígito correspondiente a una imagen, utilizando herramientas y técnicas como la clasificación multiclase y los árboles de decisión, con especial enfoque en la optimización del modelo mediante la validación cruzada.

El dataset MNIST-C se presenta como una variación del dataset MNIST tradicional, con imágenes de 28x28 píxeles en escala de grises. En este caso, la principal diferencia es la introducción de distorsiones que simulan un desenfoque de movimiento, lo que hace que la tarea de clasificación sea más desafiante. El análisis de los datos, así como la aplicación de modelos de clasificación, se centrará en explorar qué atributos (píxeles) son más relevantes para la predicción, qué relaciones existen entre las diferentes clases (dígitos) y qué desempeño pueden alcanzar los modelos ajustados para diferentes profundidades de árbol.

2. Analisis exploratorio

El dataset MNIST-C está compuesto por dos archivos principales. El archivo mnistc_images.npy contiene un total de 10.000 imágenes, cada una representada como una matriz de 28x28 píxeles, lo que da un total de 784 atributos por imagen. Cada atributo corresponde a la intensidad de un píxel en escala de grises, con valores que oscilan entre 0 y 255, donde 0 representa la ausencia de intensidad (negro) y 255 la máxima intensidad (blanco). Estas intensidades de píxeles son las que definen la forma y la estructura del dígito escrito a mano. El otro archivo, mnistc_labels.npy, contiene las etiquetas correspondientes a cada imagen. Estas etiquetas son números enteros que indican el dígito representado en la imagen, y pueden ser cualquier valor entre 0 y 9, representando los 10 posibles dígitos escritos a mano.

En cuanto a la distribución de las clases, el conjunto de datos incluye 10 clases diferentes, correspondientes a los dígitos 0, 1, 2, 3, 4, 5, 6, 7, 8 y 9 tal como se puede ver en la Figura 1. La distribución de estas clases es relativamente equilibrada, con las siguientes cantidades por dígito: 980 imágenes del dígito 0, 1135 del dígito 1, 1032 del dígito 2, 1010 del dígito 3, 982 del dígito 4, 892 del dígito 5, 958 del dígito 6, 1028 del dígito 7, 974 del dígito 8 y 1009 del dígito 9.

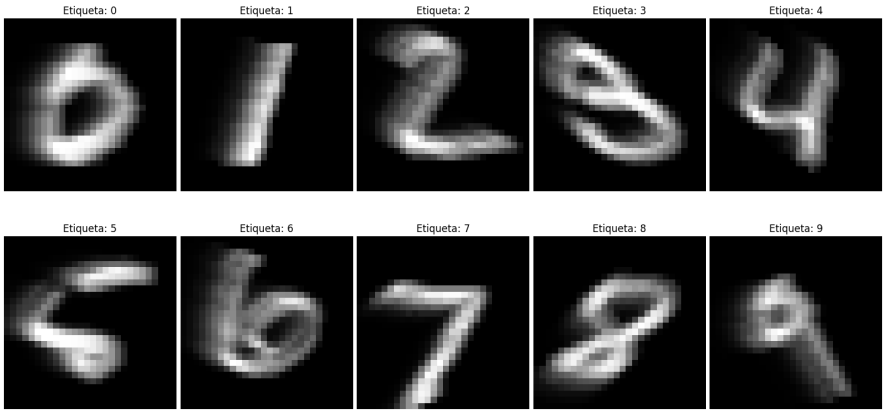


Figura 1: Ejemplos representativos de imágenes de las distintas clases en el conjunto de datos MNIST-C, con sus etiquetas correspondientes al dígito.

Para explorar las semejanzas dentro de las imágenes de una misma clase, es posible calcular la imagen promedio de cada dígito. Este procedimiento consiste en sumar los valores de intensidad de cada píxel para todas las imágenes de una misma clase y luego dividir el resultado por la cantidad de imágenes de esa clase. Así, se obtiene una representación promedio que resalta las características comunes a las imágenes de cada dígito. En la Figura 2 se presentan los resultados de este análisis, donde las imágenes promedio para cada dígito son claramente identificables a simple vista. Esto indica que existe una notable coherencia dentro de las clases, ya que los patrones característicos de cada dígito permanecen bien definidos incluso después de promediar las imágenes. Este resultado sugiere una gran similitud entre las imágenes de una misma clase, lo que podría facilitar su diferenciación en tareas de clasificación.

Para complementar el análisis de semejanzas dentro de las imágenes de una misma clase, se realizó una visualización de las dispersiones de intensidad en los píxeles de cada clase, representada mediante la desviación estándar para cada dígito. En la Figura 3 se puede observar que las mayores dispersiones se encuentran en las áreas donde están trazados los contornos de las

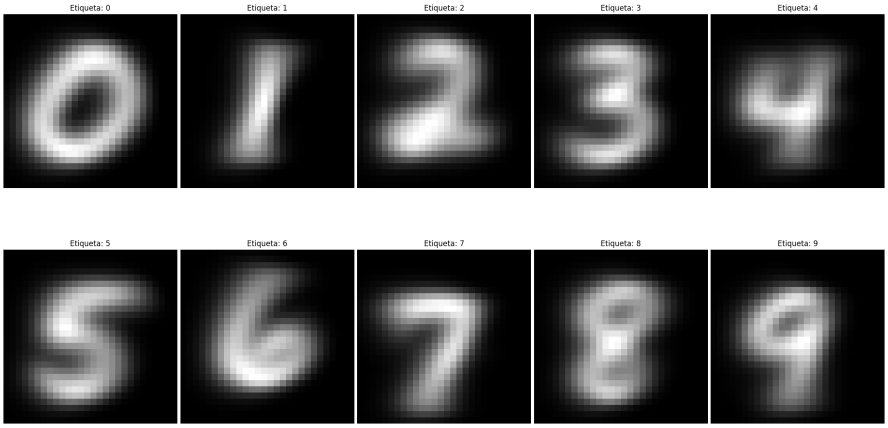


Figura 2: Imágenes promedio para cada dígito basado en la intensidad de píxeles.

figuras, indicando que los trazos de los dígitos generan un fuerte contraste con el fondo. Esto sugiere que, aunque las imágenes dentro de una misma clase no son idénticas en cuanto a la distribución de intensidad de píxeles, respetan de manera consistente la posición y forma general del dígito en el marco. Además, el cálculo de los promedios de las desviaciones estándar para cada clase permitió evaluar el grado de dispersión general dentro de cada grupo. Los resultados obtenidos en orden por valor de dispersión son los siguientes:

Dígito	STD
Dígito 1	16.81
Dígito 9	26.37
Dígito 7	27.52
Dígito 4	27.65
Dígito 6	30.04
Dígito 8	30.86
Dígito 3	32.86
Dígito 0	33.63
Dígito 5	34.01
Dígito 2	35.64

La mediana de los promedios de las desviaciones estándar es 30.45, lo que ayuda a identificar qué clases son más homogéneas. Los dígitos con menor dispersión, como el 1, 9 y 7, presentan una menor variabilidad entre las imágenes de su clase. Esto podría reflejar que los trazos de estas clases son más consistentes. En contraste, los dígitos con mayor dispersión, como el 0, 5 y 2, presentan trazos menos uniformes o variaciones más significativas en la distribución de intensidad de los píxeles.

Este análisis sugiere que, aunque los contornos y posiciones son consistentes dentro de cada clase, las intensidades de los píxeles pueden variar considerablemente. Esto podría tener implicaciones en tareas de clasificación, ya que las clases con menor dispersión probablemente sean más fáciles de identificar para un modelo.

Para identificar los atributos más relevantes en la diferenciación entre los dígitos, se analizó la dispersión de los valores promedio de intensidad de cada píxel, calculada a partir de los promedios de las 10 clases. Este enfoque permite determinar qué píxeles presentan mayor variabilidad entre las clases, lo que los convierte en indicadores clave para distinguir los dígitos. Por el contrario, los píxeles con baja dispersión son menos útiles, ya que su valor es similar independientemente de la clase y, por lo tanto, no contribuyen significativamente a la diferenciación. En la Figura 4, se observa la distribución de la dispersión de los promedios entre las clases. Como era esperable, los bordes de las imágenes muestran valores de dispersión muy bajos, ya que estas áreas no suelen contener trazos de los dígitos. En cambio, la región central, donde generalmente se ubican los dígitos, presenta una zona de alta dispersión en forma ovalada, que corresponde a las áreas clave donde se concentran las diferencias entre las clases. Además, alrededor de esta zona central se encuentra una región de dispersión media, que representa las transiciones entre los trazos del dígito y el fondo. Curiosamente, dentro de la zona de alta dispersión se observan dos pequeñas áreas con valores de dispersión menores. Estas áreas podrían estar relacionadas con características específicas o patrones recurrentes en los dígitos que, por su consistencia entre clases, muestran menor variabilidad.

Para cuantificar cuántos píxeles son relevantes, se estableció un umbral de dispersión. Por ejemplo, al considerar un umbral de 20 (es decir, píxeles con una dispersión mayor a este valor), se identificaron 308 píxeles que cumplen con este criterio. Esto equivale aproximadamente al 40 % del total de atributos (proporción: 0.3929). Estos píxeles son los más relevantes para diferenciar entre las clases de dígitos, mientras que el resto aporta menos información o es redundante en el contexto de clasificación.

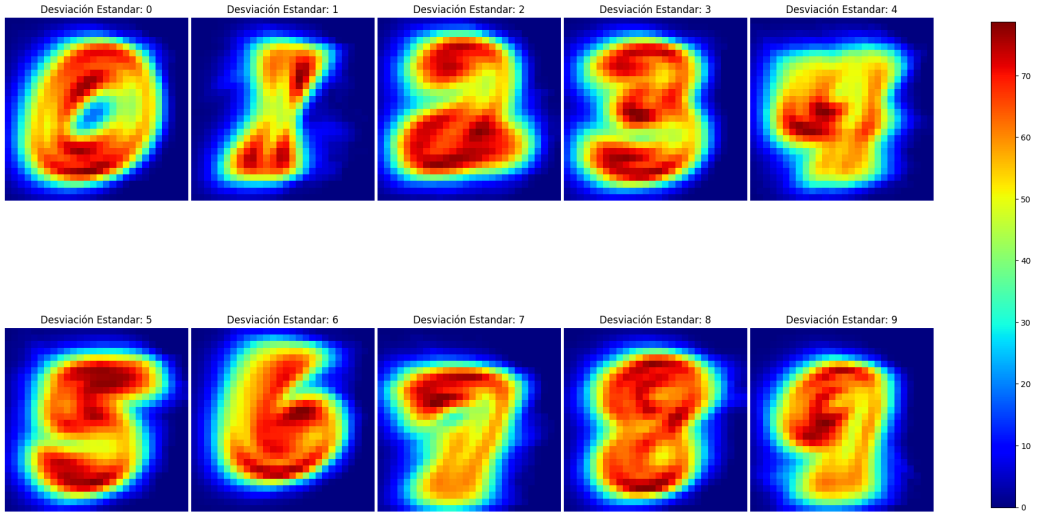


Figura 3: Dispersión de intensidades de píxeles en los dígitos basada en la desviación estándar.

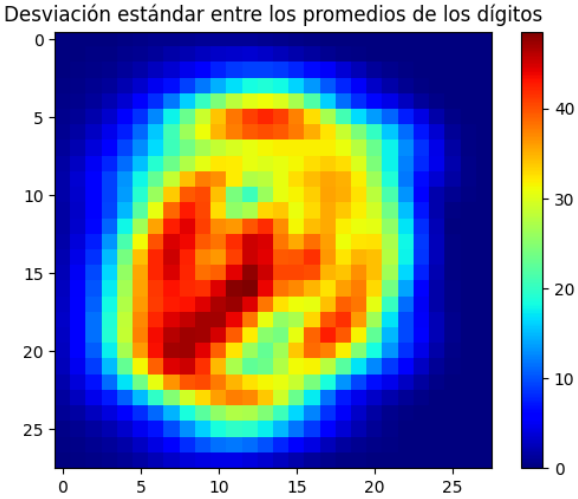


Figura 4: Mapa de dispersión promedio de píxeles entre clases.

Para analizar qué dígitos son más similares o diferentes entre sí, calculamos la distancia Euclidiana entre las imágenes promedio de cada clase. Este método nos permite cuantificar el grado de diferencia global entre los patrones promedio de los píxeles de cada dígito. Una menor distancia indica que las formas promedio de los dígitos son más similares, mientras que una mayor distancia refleja diferencias más significativas.

La matriz de distancias Euclidianas obtenida (ver Tabla 5) muestra, por ejemplo, que los dígitos 3 y 5 tienen una distancia muy baja (590), lo que sugiere similitudes significativas entre ellos, probablemente en trazos curvos. En contraste, los dígitos 0 y 1 tienen una distancia mayor (1467), reflejando sus formas claramente distintas. Este análisis ofrece una visión global que complementa el estudio pixel a pixel, ayudando a identificar qué pares de dígitos podrían ser más difíciles de diferenciar.

3. Clasificación multiclase

I. Conjunto de datos

Trabajamos con un subconjunto de datos que incluye únicamente las imágenes correspondientes a los dígitos $1, 2, 3, 7, 8$. Para seleccionar estas imágenes, se filtraron las etiquetas del conjunto completo, manteniendo únicamente aquellas que correspondieran a los dígitos mencionados.

Para garantizar la reproducibilidad en la separación de los datos, utilizamos una semilla fija en el proceso de partición. Esto asegura que, al ejecutar nuevamente el experimento, los datos se dividan de la misma forma entre los conjuntos de desarrollo y

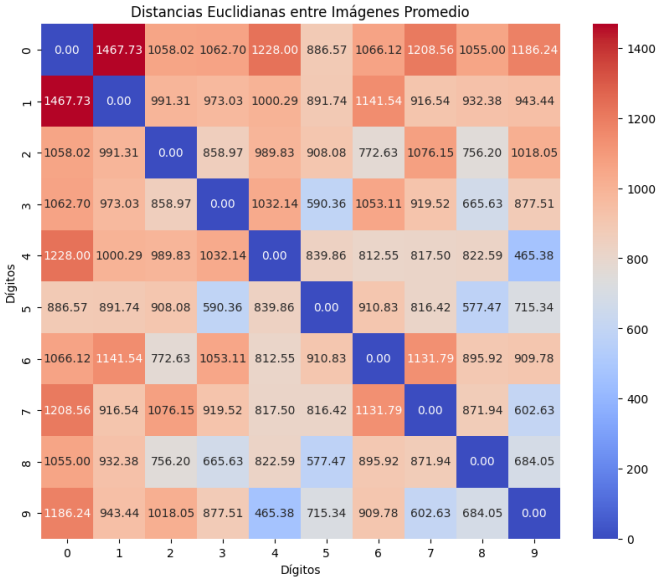


Figura 5: Matriz de distancias euclidianas entre los promedios de los dígitos.

prueba. La partición se realizó de manera aleatoria, asignando un 70 % de los datos al conjunto de desarrollo y el restante 30 % al conjunto de test.

El conjunto de desarrollo fue utilizado para entrenar y ajustar los modelos de árboles de decisión, mientras que el conjunto de test (o hold-out) se reservó exclusivamente para evaluar el desempeño final del modelo en datos no vistos previamente.

II. Ajuste del modelo de árbol de decisión

Se ajustó un modelo de árbol de decisión utilizando el criterio de *Entropia* y se exploraron distintas profundidades, desde 1 hasta 28, evaluando su desempeño mediante validación cruzada. En la Figura 6, se muestra el comportamiento de la exactitud promedio y el tiempo promedio de ejecución en función de la profundidad del árbol.

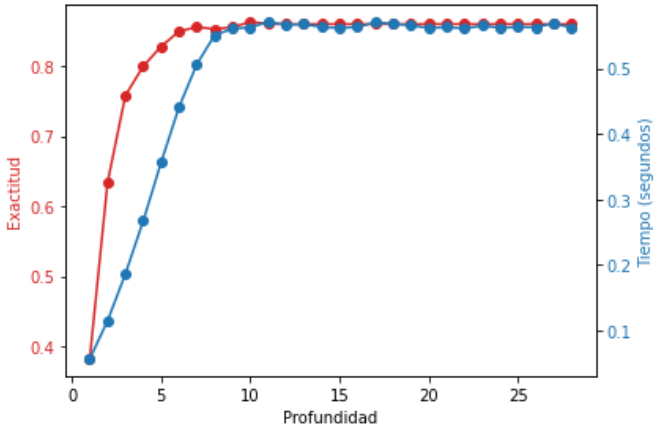
El análisis de las curvas revela que a partir de una profundidad de 6, las variaciones en la exactitud se estabilizan considerablemente, alcanzando un rango aproximado entre 0,85 y 0,87. De manera similar, los tiempos de ejecución convergen alrededor de 0,56 segundos, mostrando un incremento menos pronunciado a profundidades mayores.

El valor máximo de exactitud promedio, 0,8626, se alcanzó a una profundidad de 10. Sin embargo, cabe considerar que en profundidad 7 también se observa un máximo local con un menor tiempo promedio de ejecución. Para decidir entre estas profundidades, se evaluaron las diferencias en exactitud y tiempo promedio. Como se observa en la Tabla 1, la profundidad 10 logra una mejora del 0.72 % en exactitud en comparación con la profundidad 7, pero a costa de un aumento del 11.12 % en tiempo de ejecución. Esto sugiere que la profundidad 7 podría ser una opción más eficiente, dependiendo de las prioridades del problema (por ejemplo, velocidad vs. precisión).

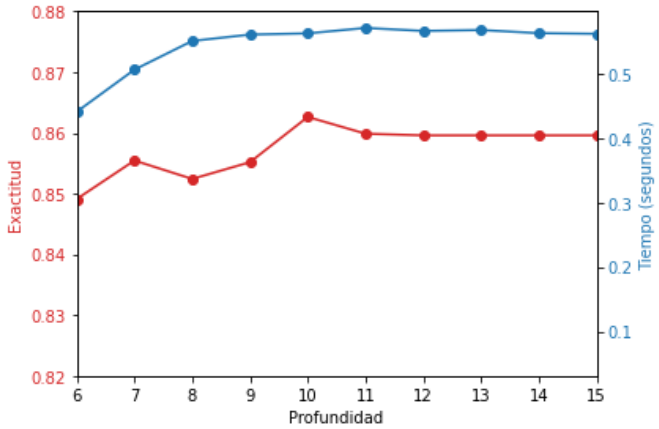
Profundidad	**Exactitud promedio**	**Tiempo promedio (s)**	**Incremento respecto a Prof. 7**
7	0.8554	0.5062	-
10	0.8626	0.5625	+0.72 % en exactitud, +11.12 % en tiempo

Cuadro 1: Comparación de desempeño entre las profundidades 7 y 10.

Este análisis sugiere que la profundidad de 10, que es la de mayor exactitud, presentan una ganancia marginal en exactitud a expensas de un aumento en el tiempo de ejecución comparada con una profundidad de 7. Por lo tanto, dependiendo de las restricciones computacionales y la importancia relativa de la exactitud frente al tiempo, podría ser razonable considerar una profundidad de 7 como un punto de equilibrio entre desempeño y eficiencia computacional.



(a) Exactitud y tiempo en función de la profundidad (rango completo).



(b) Exactitud y tiempo ampliado entre las profundidades entre 6 y 15.

Figura 6: Comparación de la exactitud promedio de un árbol de decisión con diferentes profundidades (rojo). También se muestran tiempos de ejecución para las distintas profundidades (azul). Se tiene una vista para el rango completo de profundidad (6a) y una ampliada para en un rango de profundidad (6b)

III. Comparativa de diferentes K foldings

Para evaluar el desempeño de un modelo de árbol de decisión con el criterio de gini y entropy, se realizó un experimento variando el número de pliegues en la validación cruzada (K-fold) sin modificar la profundidad del árbol, que fue fijada en 10 debido a que se había determinado previamente como la profundidad que maximiza la exactitud. El propósito de este análisis fue observar cómo el número de pliegues influye en la exactitud de los modelos y cuál de los dos criterios ofrece el mejor desempeño bajo diferentes configuraciones. Los experimentos se llevaron a cabo utilizando tres configuraciones de K-fold: 3, 5 y 10 pliegues. Para cada configuración, se entrenó el modelo con ambos criterios (entropy y gini) y se calculó la exactitud promedio de los resultados obtenidos en las distintas particiones de validación cruzada.

Con el criterio de entropy, se observó que el modelo alcanzó una exactitud promedio más alta cuando se usaron 5 pliegues, logrando una exactitud de 0.8626 a una profundidad de 10. La exactitud promedio fue bastante estable a lo largo de los diferentes números de pliegues, con ligeras variaciones entre las configuraciones de 3, 5 y 10 pliegues, pero sin diferencias significativas que cambiaran la conclusión principal de que la profundidad 10 es la más efectiva para maximizar el desempeño (fig. 7a).

Por otro lado, con el criterio de gini, el mejor desempeño también se alcanzó con una profundidad de 10, aunque la configuración de 10 pliegues dio como resultado una exactitud promedio de 0.8618, ligeramente inferior a la de entropy. En este caso, las configuraciones con 3 y 5 pliegues mostraron desempeños similares, pero también se pudo observar que la mayor cantidad de pliegues (10) resultó en una leve mejora de la exactitud, lo que sugiere que utilizar un mayor número de pliegues contribuye a un mejor desempeño en términos de estabilidad del modelo (fig. 7b).

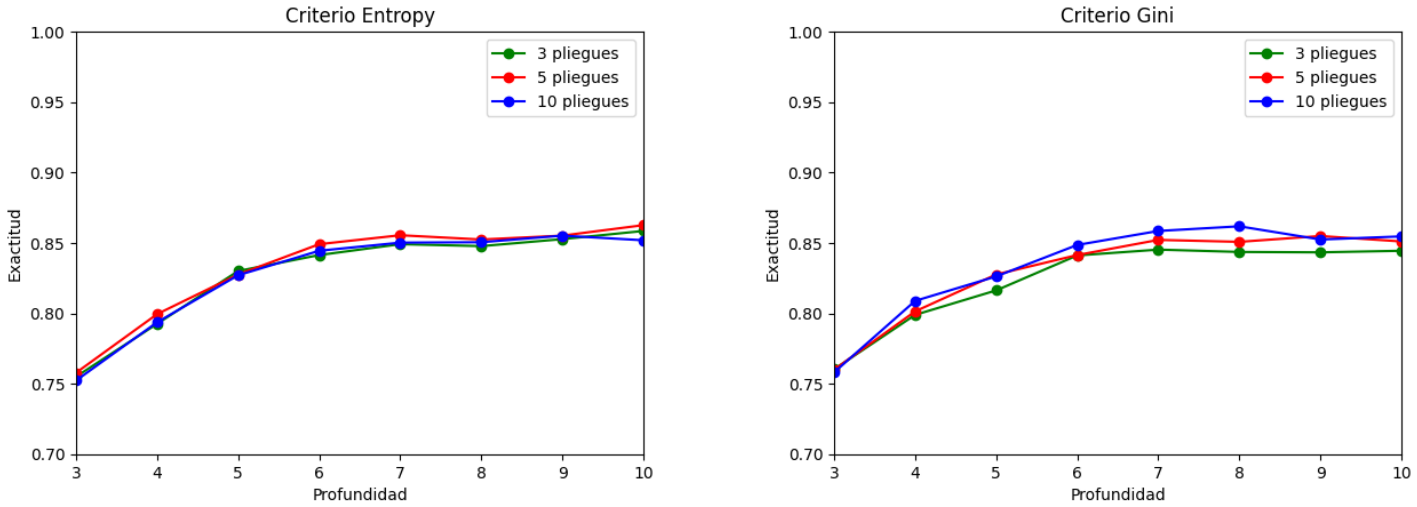
Al comparar ambos criterios, el modelo basado en el criterio de entropy logró una exactitud promedio superior a 0.86 en todos los casos evaluados, siendo ligeramente más efectivo que el modelo con gini. Específicamente, la configuración con 5 pliegues y el criterio entropy alcanzó la mejor exactitud (0.8626), mientras que el modelo con gini obtuvo una exactitud de 0.8618 con 10 pliegues.

Este análisis confirma que, aunque el número de pliegues puede tener un impacto marginal en el desempeño, la principal diferencia radica en el criterio de división utilizado. El criterio de entropy resulta ser ligeramente más efectivo en términos de exactitud para este conjunto de datos, y el número de pliegues influye en la estabilidad y precisión del modelo, siendo 5 pliegues la mejor opción.

IV. Test de predicción sobre los datos *HoldOut*

Después de seleccionar el mejor modelo basado en el análisis previo, se entrenó un Árbol de Decisión con los siguientes hiperparámetros: criterio Gini y profundidad máxima 7. El modelo se entrenó utilizando el conjunto de desarrollo completo y luego se evaluó en el conjunto de prueba (hold-out). Los resultados de esta evaluación incluyen la exactitud global, la matriz de confusión y las métricas de clasificación detalladas.

Se armó una matriz de confusión para mostrar el desempeño del modelo en términos de la cantidad de predicciones correctas



(a) Criterio Entropy.

(b) Criterio Gini.

Figura 7: Gráfico de la exactitud en función de la profundidad para 3, 5 y 10 pliegues, visto con criterios Entropy y Gini.

e incorrectas para cada clase. Los valores diagonales representan las predicciones correctas, mientras que los valores fuera de la diagonal indican errores. Se observa un buen equilibrio en las predicciones, aunque algunas clases presentan confusiones más marcadas (por ejemplo, los dígitos 8 y 2).

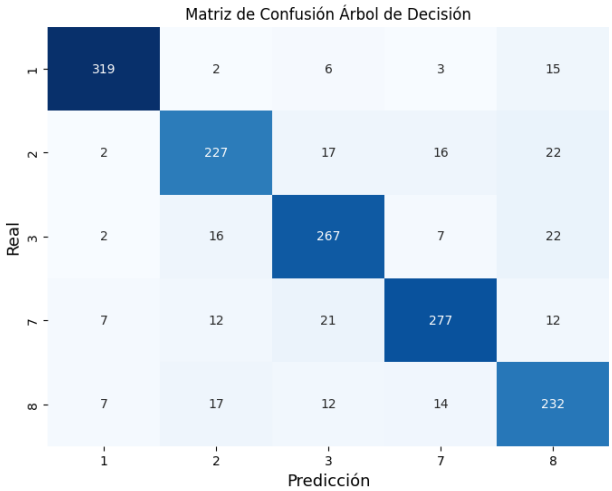


Figura 8: Matriz de confusion para la clasificacion mediante arbol de Gini, con profundidad 7.

Al comparar la matriz de confusión (fig. 8) con la matriz de distancias entre los promedios de los dígitos (En la tabla 2 se puede observar las distancias solo para los digitos que estamos trabajando), se pueden observar algunas coincidencias interesantes que ayudan a entender el desempeño del modelo. En primer lugar, entre los dígitos 3 y 8 se observa una menor distancia en la matriz de distancias, lo que indica que estos dos dígitos son más similares en términos de sus características visuales. Esta similitud parece reflejarse en la matriz de confusión, donde el modelo presenta errores frecuentes al intentar diferenciarlos, especialmente en la predicción del dígito 3 como 8 y viceversa. Un patrón similar ocurre entre 2 y 8. Por otro lado, la distancia considerable entre los dígitos 2 y 7 en la matriz de distancias sugiere que deberían ser fácilmente diferenciables, dado que sus características visuales son bastante distintas. Sin embargo, en la matriz de confusión, se observa que el modelo todavía comete varios errores al predecir 2 como 7 y viceversa. Esto podría deberse a otros factores, como las variaciones dentro de las imágenes de cada dígito que dificultan la clasificación. Esta hipótesis podría tener sentido siendo que 2 es el dígito con mayor despersion. Finalmente, el dígito 1, que tiene una distancia considerable con los demás, es clasificado con mayor precisión, lo que concuerda con su bajo número de errores en la matriz de confusión.

Por otro lado, el reporte de clasificación (tabla 3) proporciona métricas detalladas como precisión, recall y F1-score para cada clase. En particular, el dígito 1 obtuvo los valores más altos en todas las métricas, con una precisión del 95 % y un F1-score de 0.94, evidenciando una excelente capacidad del modelo para identificar esta clase. Lo cual tambien es consistente con el analisis

	1	2	3	7	8
1	0	991	973	916	932
2	991	0	858	1076	756
3	973	858	0	919	665
7	916	1076	919	0	871
8	932	756	665	871	0

Cuadro 2: Matriz de distancias euclidianas entre los promedios de los dígitos 1, 2, 3, 7 y 8.

que hicimos con la matriz de confusion y la matriz de distancias. El dígito 8, aunque con métricas algo inferiores, alcanzó un F1-score de 0.79, lo que indica un desempeño aceptable, aunque con mayor propensión a errores. El promedio macro y ponderado de estas métricas fue de 0.85, consolidando al modelo como una herramienta eficaz para este problema de clasificación.

Cuadro 3: Reporte de clasificación del árbol de decisión con criterio Gini, profundidad 7

Clase	Precisión	Recall	F1-Score	Support
1	0.95	0.92	0.94	345
2	0.83	0.80	0.81	284
3	0.83	0.85	0.84	314
7	0.87	0.84	0.86	329
8	0.77	0.82	0.79	282
Exactitud			0.85	1554
Promedio macro	0.85	0.85	0.85	1554
Promedio ponderado	0.85	0.85	0.85	1554

En síntesis, el modelo entrenado con el criterio gini y una profundidad máxima de 7 demostró ser efectivo, alcanzando un desempeño consistente y satisfactorio en el conjunto de prueba. La evaluación realizada confirma su capacidad para abordar el problema de clasificación planteado y proporciona una base sólida para futuras mejoras.

V. Test de predicción usando un modelo KNN

El modelo KNN se evaluó con diferentes valores de k, en un rango de 1 a 20. Para cada valor de k, se realizó un entrenamiento con los datos de desarrollo y se obtuvo la exactitud en el conjunto de prueba (hold-out) y en el conjunto de desarrollo.

Los resultados que se pueden observar en la fig.9 muestran que la exactitud promedio en el conjunto de entrenamiento es muy alta, alcanzando el 100 % cuando se utiliza un solo vecino (k=1), pero disminuyendo ligeramente a medida que se aumenta el número de vecinos. Este comportamiento es esperado, ya que un mayor número de vecinos introduce una mayor generalización, lo que reduce el sobreajuste. En el conjunto de prueba, el modelo presenta un rendimiento más estable, con la mayor exactitud (95.56 %) alcanzada cuando se usa k=3. Posteriormente, la exactitud comienza a disminuir a medida que se incrementa k, aunque el desempeño sigue siendo bastante bueno, con exactitudes superiores al 92 % para la mayoría de los valores de k.

La matriz de confusión (10) muestra que el modelo presenta pocos errores en general, con valores diagonales muy altos que indican una correcta clasificación de las clases. Además, la matriz de confusión revela que los errores más frecuentes ocurrieron entre las clases 2 y 8, y 3 y 8, lo que está alineado con las distancias más pequeñas observadas en el análisis exploratorio.

El reporte de clasificación (tabla 4)confirma los buenos resultados, con un F1-score promedio ponderado de 0.96, y destaca una excelente capacidad de clasificación para todos los dígitos, especialmente para el dígito 1, que muestra una alta precisión y recall.

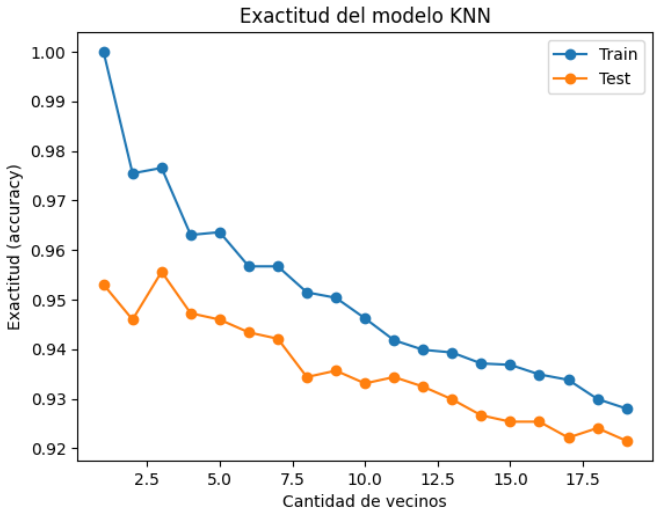


Figura 9: Exactitud del modelo KNN en función de la cantidad de vecinos.

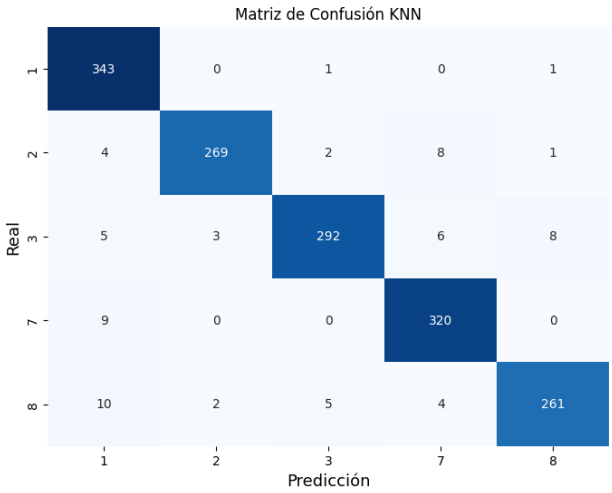


Figura 10: Matriz de confusion para la clasificacion mediante modelo de knn.

Cuadro 4: Reporte de clasificación de modelo KNN

Clase	Precisión	Recall	F1-Score	Support
1	0.92	0.99	0.96	345
2	0.98	0.95	0.96	284
3	0.97	0.95	0.95	314
7	0.95	0.97	0.96	329
8	0.96	0.93	0.94	282
Exactitud			0.96	1554
Promedio macro	0.96	0.95	0.96	1554
Promedio ponderado	0.96	0.96	0.96	1554

El desempeño del modelo KNN fue notable, con una alta exactitud en el conjunto de prueba (95.56 %) y buenos resultados en el reporte de clasificación, especialmente en la clasificación de dígitos como el 1, 2 y 7. La matriz de confusión mostró pocos errores, y los resultados fueron estables a lo largo de los diferentes valores de k evaluados.

En comparación con el modelo de árbol de decisión con el criterio Gini (profundidad 7), el modelo KNN mostró un desempeño superior. Aunque el árbol de decisión también obtuvo buenos resultados, su exactitud fue ligeramente menor en el conjunto de

prueba. Esto sugiere que el KNN fue más adecuado para esta tarea específica de clasificación de dígitos, ya que su capacidad para manejar la similitud entre las clases le permitió clasificar los dígitos de manera más efectiva. El análisis y los resultados obtenidos refuerzan la elección del KNN como el modelo más eficiente para este conjunto de datos.

4. Conclusiones

En este trabajo práctico se ha aplicado un modelo de clasificación sobre el conjunto de datos MNIST-C en su versión "Motion Blur", con el objetivo de predecir dígitos manuscritos corrompidos por desenfoque de movimiento. A través de un análisis exploratorio, se identificaron patrones clave en la variabilidad de las clases, destacándose que los dígitos como 1, 7 y 9 son más homogéneos, mientras que otros como 0, 5 y 2 presentan mayor dispersión. También se observaron 308 píxeles con alta dispersión entre las clases, lo que podría haber sido útil para reducir la dimensionalidad y mejorar la eficiencia del modelo, aunque no se implementó en este caso.

El análisis incluyó dos modelos de clasificación: el Árbol de Decisión y KNN. Aunque el Árbol de Decisión mostró un buen rendimiento, alcanzando una exactitud promedio de 0.8626 con una profundidad óptima de 10, los resultados del modelo KNN fueron superiores en términos de consistencia. KNN presentó una mayor exactitud en las predicciones.

El modelo de Árbol de Decisión fue efectivo, pero mostró un comportamiento de estabilización en la exactitud a partir de una profundidad de 6, sin mejoras significativas al aumentar la profundidad. Además, KNN presentó mejores resultados en la clasificación de dígitos visualmente similares como el 7 y el 8.

En resumen, el modelo KNN fue el que logró un mejor rendimiento en términos de exactitud y consistencia para la tarea de clasificación en este conjunto de datos, superando al Árbol de Decisión. Ambos enfoques ofrecen buenos resultados, pero KNN parece ser más adecuado para este tipo de tarea de clasificación con imágenes corrompidas por desenfoque de movimiento.