LABORATORIO
DE
DATOS

Trabajo Práctico #2

16/06/2025

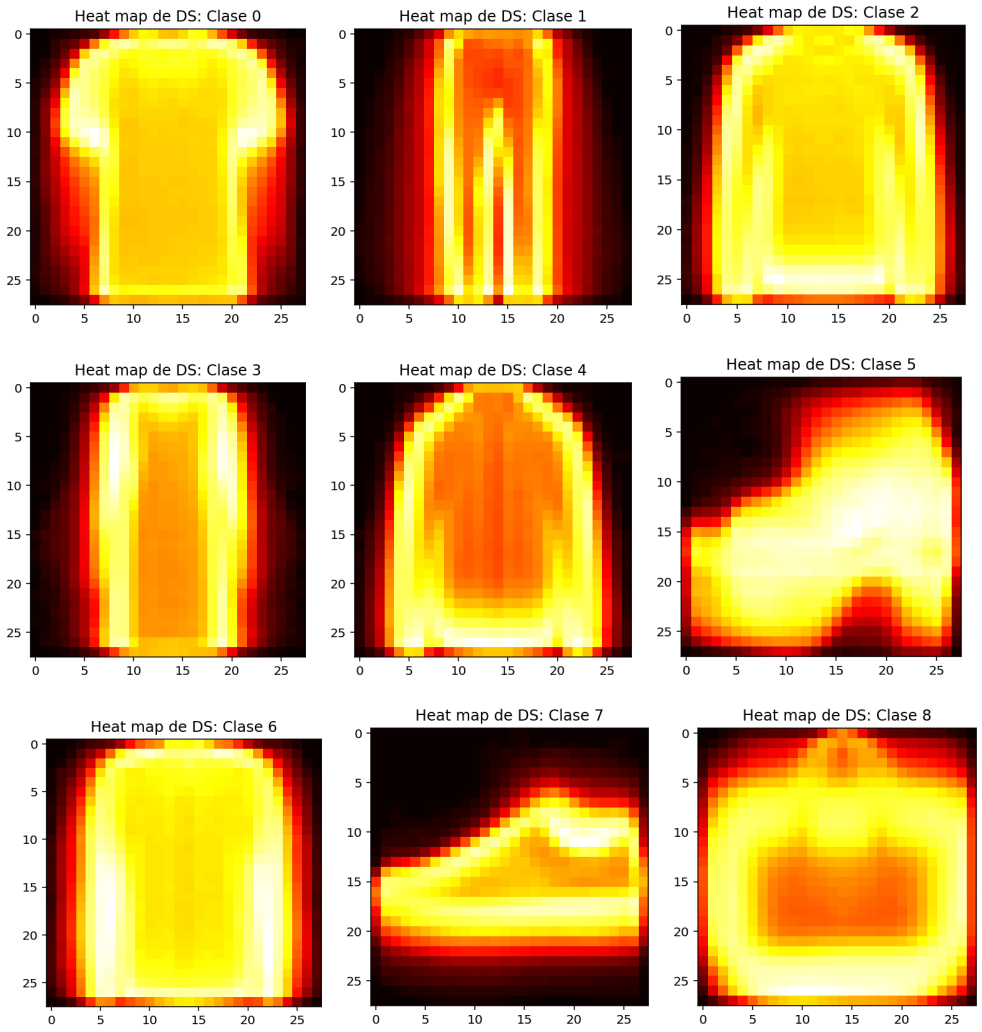
Hernán Blumenthal
Walid Sanchez Richani
Pedro Soldatich

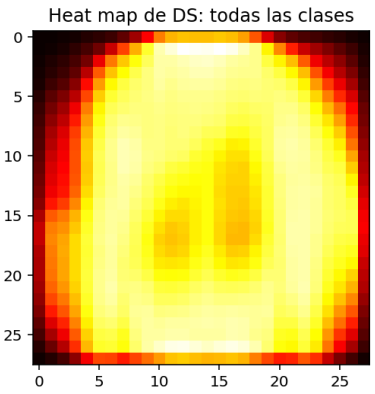
Introducción:

Se recibió una base de datos de 70.000 imágenes 28 x 28 píxeles en escala de grises con rango 0-255. Estas imágenes vinieron etiquetadas en 10 clases diferentes. Esto es equivalente a 784 atributos con variables continuas que debieron ser analizadas exploratoriamente en este trabajo para luego optimizar algoritmos de clasificación por clase.

Análisis Exploratorio

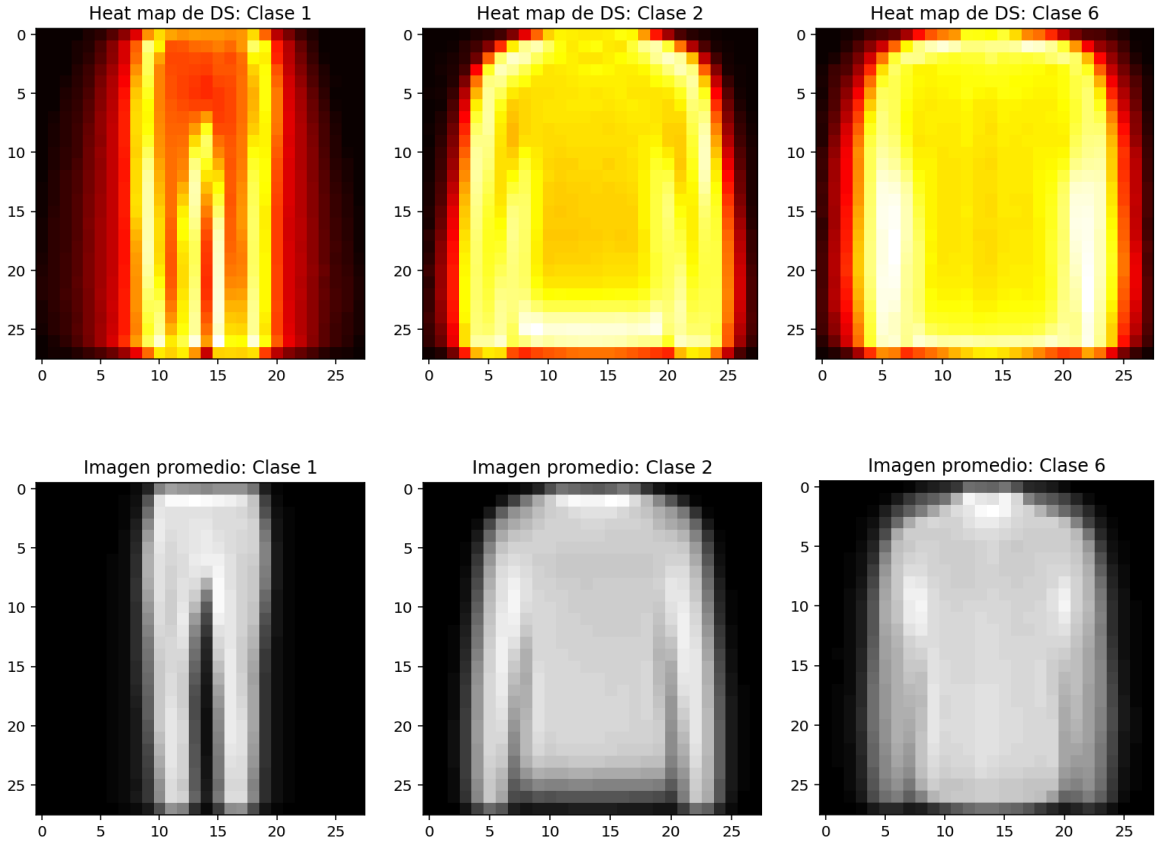
(a) Para discernir entre cuales son los píxeles más y menos relevantes para predecir el tipo de prenda (clase), se graficaron heat maps de los desvíos estándar (DS) intra- e inter-clase. En los mapas de DS intra-clase se pueden visualizar las regiones de menor y mayor dispersión dentro de cada clase. Los atributos que no varían tanto, es decir, los que se 'conservan' (menor DS), o se mantienen relativamente constantes (no nulos), son los más relevantes para el ejercicio de clasificación. Se puede pensar cómo identificar el 'factor común' de la clase. En cambio, los atributos que tienen alta varianza intra-clase no son muy útiles para tal fin.





Luego, en el mapa de DS inter-clase esta relación se ‘invierte’: se buscan las regiones de mayor variabilidad, las cuales pueden ayudar a encontrar diferencias entre las clases y permitir una clasificación optimizada. Y las regiones de menor variabilidad podrían ser descartadas. (se podrían elegir umbrales…)

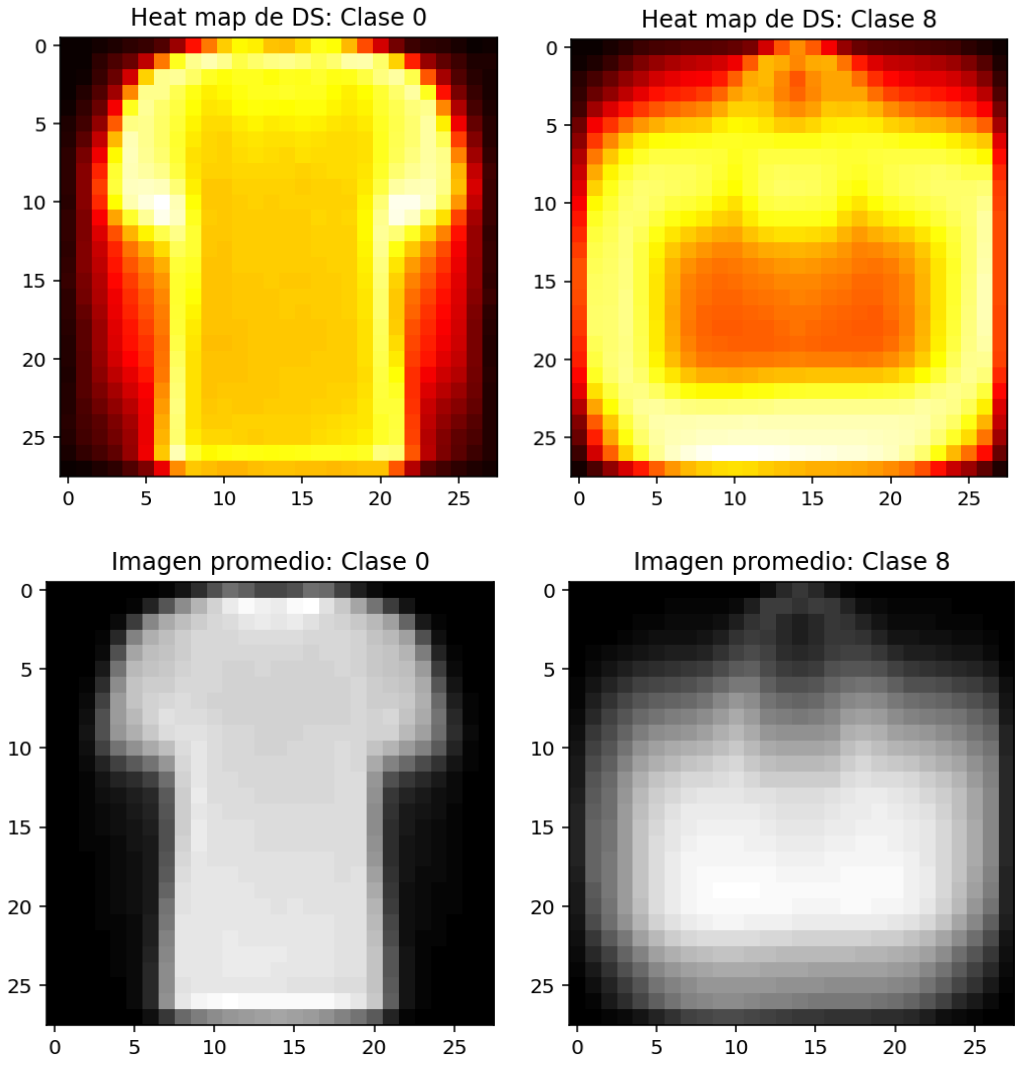
(b)



Como queda en evidencia en esta serié de imágenes promedio y heat maps de DS, efectivamente hay clases que son fáciles de diferenciar, como la 1 y la 2, y otras más parecidas entre sí, como la 2 y la 6, lo cual dificulta su clasificación. Las imágenes promedio 1 y 2 son tan distintas que unos pocos atributos nos servirían para clasificarlas con alta exactitud en sus respectivas clases. En cambio, los resultados obtenidos en las imágenes promedio para las clases 2 y 6, nos sugiere que vamos a necesitar una mayor cantidad de información para clasificarlas con alta exactitud, ya que se nota una alta superposición de medias. Como ejercicio, se podrían restar las

medias de todos los pixeles para encontrar las mayores diferencias y concentrarse en esos atributos para la distinción entre clases. La visualización de los DS también muestran superposición, pero en una menor medida. Parece haber una mayor proporción de atributos con alta variabilidad entre las prendas de clase 6.

(c)



Para decidir si las imágenes de la clase 8 son similares entre si graficamos una imagen promedio y un heat map de desvíos estándar para cada atributo. La imagen promedio ya muestra una silueta poco definida, y el heat map revela una gran proporción de atributos con alta dispersión. Esto nos sugiere que las imágenes no son muy parecidas entre sí dentro de la misma clase. En cambio, la clase 0 muestra mayor similitud intra-clase, ya que la imagen promedio da una silueta mucho más definida, acorde con el tipo de prenda, y el heat map muestra menor dispersión.

Clasificación binaria

(a) Una vez creado el data frame con los subconjuntos de las clases 0 y 8 se confirmó que estaban balanceadas. El total de muestras fue de 14000 (7000 de cada clase).

(b) Luego se procedió a dividir el total en data de entrenamiento y data de testeo en un split 80:20. Esto se hizo de manera pseudoaleatoria, utilizando el parámetro "random_state" de la función "train_test_split", el cual permite fijar el split para todo el experimento. Esto estandariza la división de la muestra para que los modelos sean más comparables.

(c) La selección de los atributos (pixels) con los que se entrenaron los modelos se basó en el análisis exploratorio. Se calcularon las diferencias de las medias de clase de todos los atributos, a mayor diferencia de medias mayor valor predictivo se le atribuyó al pixel. La diferencia de medias osciló entre 0 (baja calidad predictiva) y 150 (alta calidad predictiva). También se revisó que los píxeles seleccionados no tengan desvíos muy grandes, lo cual podría afectar la calidad predictiva. A partir de esta información se hicieron 7 modelos distintos:

Subconjunto 1: 3 atributos de alta calidad (554_526_538)
Subconjunto 2: 3 atributos de baja calidad (0_1_2)
Subconjunto 3: 3 atributos de calidad media (48_590_394)
Subconjunto 4: 6 atributos de alta calidad (554_526_538_510_498_582)
Subconjunto 5: 6 atributos de baja calidad (0_1_2_3_4_5)
Subconjunto 6: 6 atributos de calidad media (48_590_394_306_343_180)
Subconjunto 7: 1 atributo de alta calidad (554)

Para comparar el rendimiento de los modelos se utilizó la métrica de exactitud. Los resultados fueron los siguientes:

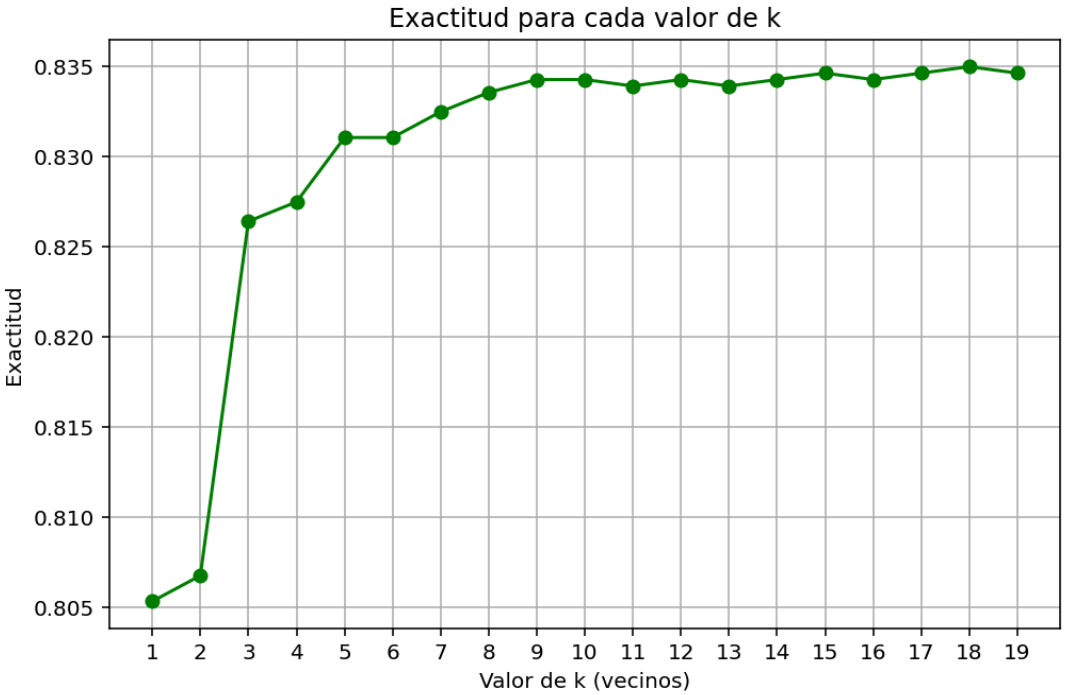
Exactitud del subconjunto 1 (k=3): 91%
Exactitud del subconjunto 2 (k=3): 50%
Exactitud del subconjunto 3 (k=3): 83%
Exactitud del subconjunto 4 (k=3): 92%
Exactitud del subconjunto 5 (k=3): 63%
Exactitud del subconjunto 6 (k=3): 91%
Exactitud del subconjunto 7 (k=3): 90%

En la comparación de los tres primeros subconjuntos, se mantuvieron la cantidad de atributos constantes (3) y se aumentó la calidad de los píxeles. En este caso se notó un drástico incremento en la exactitud de 50% (como tirar una moneda) en el caso de los atributos de baja calidad, hasta una exactitud de 91% para el caso de los de alta calidad. La calidad tuvo un alto impacto en esta comparación. Cuando se repitió el experimento dejando la cantidad de atributos constantes en seis, el incremento de la exactitud no fue tan grande, de hecho, hubo una diferencia muy pequeña entre el subconjunto 4 (calidad media) y 6 (alta calidad). Esto nos llevó a interpretar que a mayor cantidad de atributos en el modelo, menor peso tiene su calidad, lo cual es esperable.

En la tercera comparación, se mantuvo la calidad de los atributos constantes y se aumentó la cantidad de atributos. Comparando los subconjuntos, 1 vs. 4, 2 vs. 5 y 3

vs. 6, podemos notar que el incremento en exactitud por agregado de atributos en modelos con píxeles de alta calidad, no es muy grande, en cambio para modelos con píxeles de calidad media y baja es un salto de aproximadamente 10%. Esto nos llevó a pensar que la cantidad de atributos tiene un alto impacto en el rendimiento de los modelos cuando estos son de calidad media y baja. Por lo tanto, dado un número suficiente de atributos, es posible armar un modelo con buen rendimiento, aunque estos no sean de la mejor calidad. Esto queda reflejado al ver que el subconjunto 4 (calidad media) alcanza prácticamente la misma exactitud (91%) que el 6 (alta calidad).

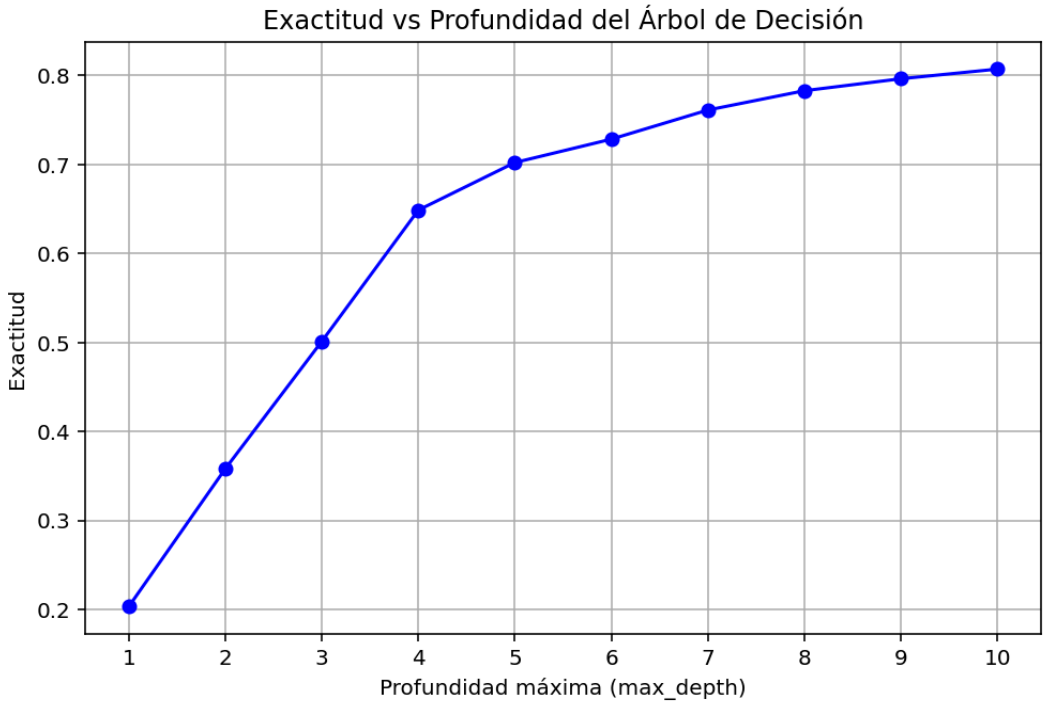
También se varió la cantidad de k-neighbours para evaluar su impacto en la exactitud. Se eligió el subconjunto de 3 atributos de calidad media, con valores de k 1 - 19. El cambio total en la exactitud fue del 3% de k = 1 a 19, con el salto más abrupto de k = 2 a k =3, donde el modelo ya empieza a tener en cuenta más vecinos y se vuelve más confiable y estable.



Finalmente, se nos ocurrió crear un modelo con el atributo de mejor calidad, para conocer su rendimiento. Para nuestra sorpresa el pixel 554 resultó tener una exactitud de casi el 90%. Esto refuerza nuestra interpretación previa, a menor cantidad de atributos en el modelo, mayor peso tiene su calidad.

Clasificación multiclase

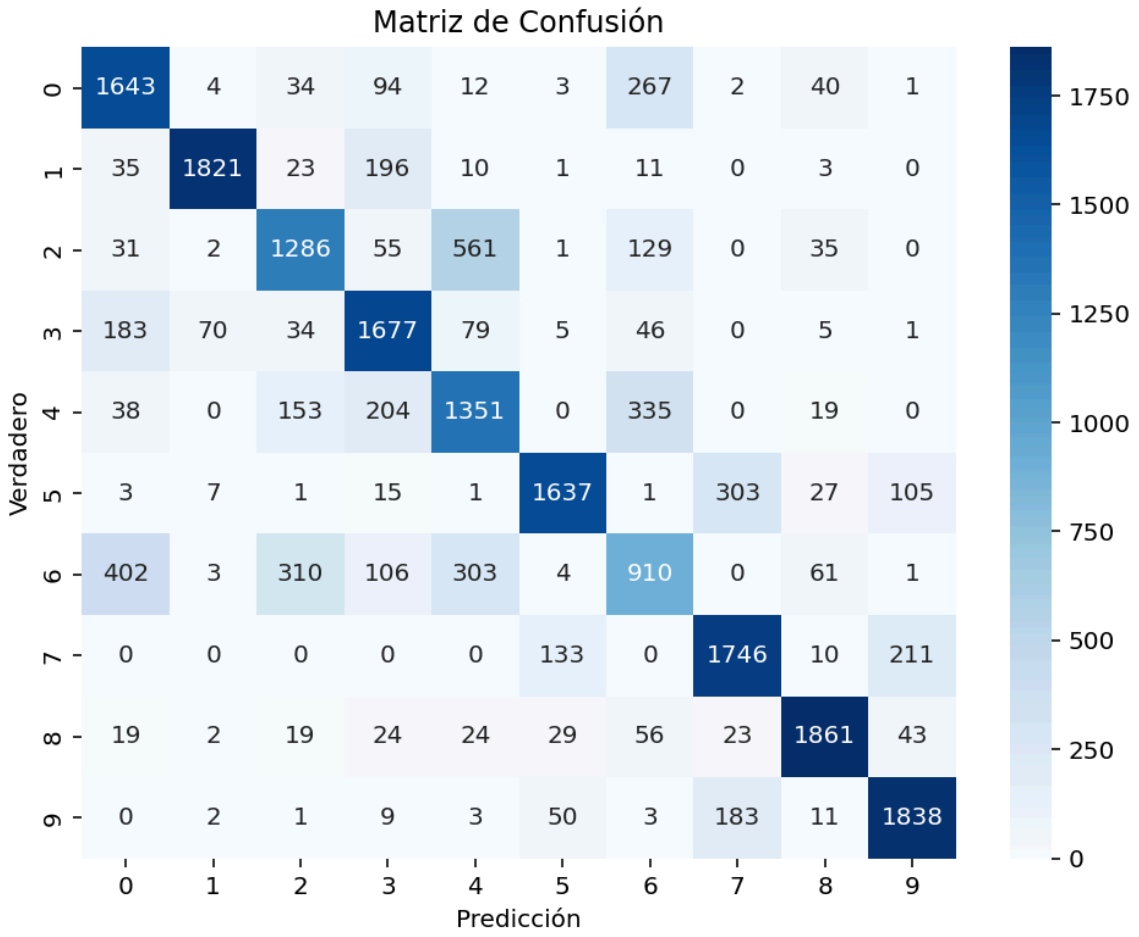
(a) Para esta sección del trabajo se separó el 70% como set de desarrollo (dev) y 30% (held-out) se dejó para validar el modelo.
(b) Se ajustó el modelo de árbol de decisión con profundidades de 1 a 10, y se obtuvieron los siguientes resultados de exactitud:



Se puede ver un incremento de la exactitud en función de la profundidad máxima, pero este efecto se ameseta a medida que nos acercamos a max_depth = 10. Es decir, la profundidad máxima parece ser un hiperparamentro clave para la optimización de los modelos, pero pasado cierto umbral deja de ser computacionalmente costo-efectivo.

(c) Luego se realizaron una serie de experimentos, donde se variaron diferentes hiperparametros *(criterion, max_depth, min_samples_leaf)* y se les evaluó la performance con el método de validación cruzada con k-folding. Dado el alto costo computacional del max_depth se eligieron tres valores para optimizar [2, 4, 6], aun sabiendo que mayores profundidades hubieran dado seguramente una mayor exactitud en esta instancia (a expensas de un potencial sobre-ajuste). La configuración de los hiperparametros del mejor modelo fue → *criterion: entropy, max_depth: 6, min_samples_leaf: 3.* El cual dio un performance medio de 73.7%.

(d) Finalmente, se entrenó el modelo de mejor rendimiento con todo el conjunto de desarrollo (dev) y se utilizó para predecir las clases del conjunto held-out. La exactitud resultó del 75.0%, lo cual fue alentador ya que sugiere que el modelo elegido no estaba sobre-ajustado, en otras palabras, generaliza bien. Para evaluar el modelo también se armó una matriz de confusión de 10 x 10, donde se pudo visualizar cuales clases fueron más difíciles de clasificar.



Acorde con el análisis exploratorio, las clases con las imágenes promedio más similares, osea las prendas más parecidas entre sí [0, 2, 4, 6], fueron las más difíciles de predecir para el modelo, y es donde se produjo mayor cantidad de errores. Esto se ve reflejado por la baja precisión: 69.8% (clase 0), 69.1% (clase 2), 57.6% (clase 4), y 51.8% (clase 6). Los valores advierten una gran cantidad de falsos positivos, que se traduce en poca especificidad para esas clases. Los valores de recall también fueron los más bajos del modelo: 78.2% (clase 0), 61.2% (clase 2), 64.3% (clase 4) y 43.3% (clase 6). Estos resultados reflejan una elevada cantidad de falsos negativos, o dicho de otro modo, poca sensibilidad para las respectivas clases (la clase 6 es la más problemática). En cambio, las clases [1, 3, 5, 7, 8, 9] tuvieron mucho mejores valores de precisión, pocos falsos positivos, entendido como buena especificidad (la mejor especificidad se dio en la clase 1 con 95.3%). Los recalls también fueron buenos para estas clases, con pocos falsos negativos, es decir, buena sensibilidad (hasta un 87.5% para la clase 9).

Precision = [69.8%, 95.3%, 69.1%, 70.5%, 57.6%, 87.9%, 51.8%, 77.4%, 89.8%, 83.5%]

Recall = [78.2%, 86.7%, 61.2%, 79.9%, 64.3%, 78.0%,
43.3%, 83.1%, 88.6%, 87.5%]

En conclusión, la matriz de confusión y las métricas de especificidad y sensibilidad nos permiten tener una idea más detallada sobre cómo está rindiendo el modelo para cada clase en particular. Nos permitió notar una clara diferencia entre las clases [0, 2, 4, 6] vs. [1, 3, 5, 7, 8, 9] (la clase 3 tal vez merecería un grupo aparte), que fue consistente con el análisis exploratorio previo.