DEPARTAMENTO DE COMPUTACION
Facultad de Ciencias Exactas y Naturales - UBA

Laboratorio de Datos
-Trabajo Práctico 02-

Grupo: santi.com

Integrantes:

- Brizuela Federico
- Risso Mateo
- Rugo Julián

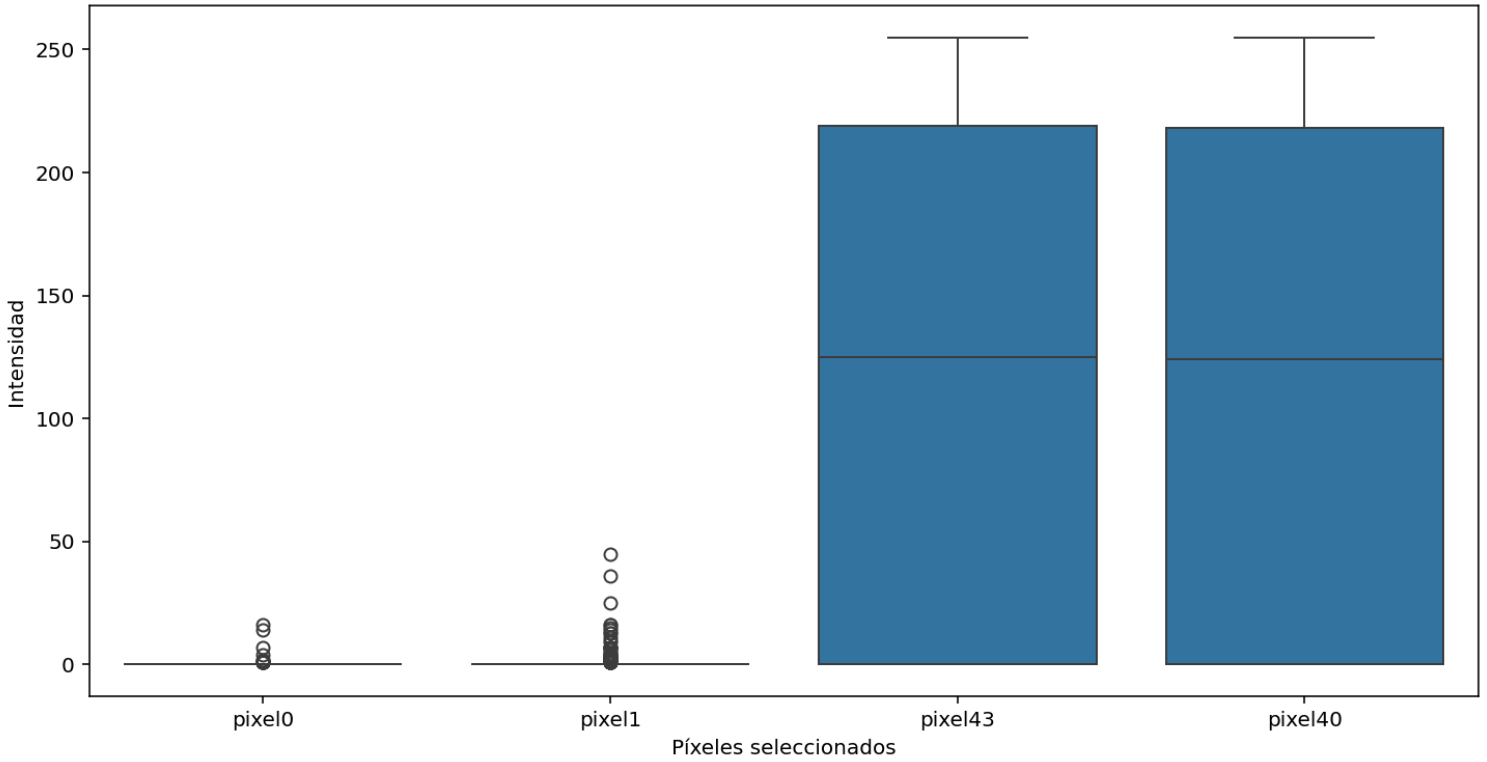
Introducción

En este informe vamos a utilizar distintos métodos de aprendizaje supervisado, variando sus hiper-parámetros y cambiando la cantidad de muestras tomadas, con el fin de comparar los resultados a los que llega cada configuración para poder elegir la mejor a la hora de clasificar los datos según las categorías presentadas. Estas predicciones las vamos a hacer sobre un dataset que contiene 70.000 imágenes de prendas de ropa, asociadas cada una a un 'label' que indica a qué clase pertenecen. En el primer análisis intentamos entender, mediante gráficos y representaciones visuales, las características de las imágenes de cada clase, notando que en algunos casos eran muy parecidas. También hicimos un análisis para decidir qué atributos iban a ser los más importantes a la hora de lograr nuestro objetivo, considerando que para algunos modelos íbamos a usar un número reducido de estos. Finalmente, mediante modelos KNN y de árboles de decisión, usando distintas métricas y métodos para evaluarlos, conseguimos resultados satisfactorios, considerando las similitudes que tenían algunas clases entre sí.

1) Análisis exploratorio

a) El dataset consiste de 70.000 tuplas conformadas por su índice y 785 atributos diferentes, dentro de estos últimos están los 784 píxeles (‘pixel0’, ‘pixel1’, … , ‘pixel230’, … ‘pixel783’) que indicarán el valor de brillo de cada uno, y que en conjunto, formarán una imagen. El atributo restante es el ‘label’, que será un diferenciador entre clases. Las clases de ropa son 10, van de 0 hasta 9, y están distribuidas de manera equitativa, es decir, hay 7.000 imágenes para cada una de las 10 clases.

Si vamos a querer predecir el tipo al que corresponde una imagen, vamos a necesitar el ‘label’ para entrenar el modelo y que las predicciones concuerden con el valor de este atributo en la mayor cantidad de casos. Por otro lado, vamos a necesitar los valores de los píxeles como datos que van a permitir al modelo aproximar su respuesta a la correcta. Sin embargo, dentro de estos, hay algunos atributos para los que su valor cambia muy poco en todo el dataset, por ejemplo, los que tienen brillo 0 para la mayoría de casos (píxeles en negro), estos seguramente no aportarán mucho a la hora de diferenciar una imagen de otra puesto que tienen valores parecidos sin importar la clase.

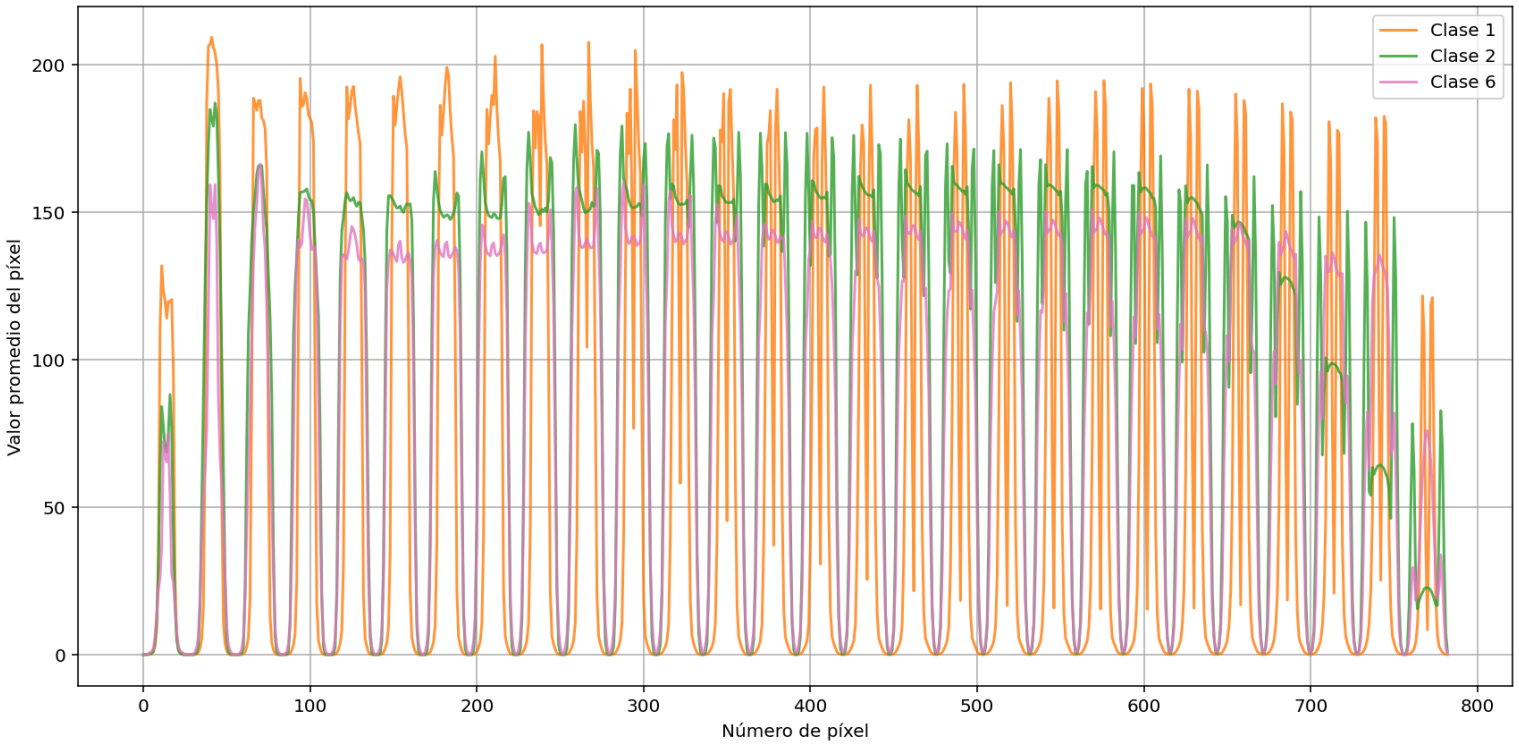


- *Comparación de píxeles con alta y baja variación a lo largo de todo el dataset*

Lo que estamos representando con este gráfico es cuánto varían los dos píxeles que menos y que más lo hacen a lo largo de todas las imágenes (a la izquierda los que menos, a la derecha los que más). Si la variación es (casi) nula en 70.000 imágenes, podemos decir que ese píxel es irrelevante a la hora de hacer un modelo predictivo de cualquier tipo, mientras que en los atributos (o grupos de atributos) que varían más, será al menos posible detectar patrones de los que el modelo pueda sacar conclusiones.

b) Al analizar a simple vista el aspecto de las imágenes por cada clase, pudimos tener una idea de qué tipos de vestimenta representa cada una. El tipo 0 parecieran ser remeras y chombas; el 1 pantalones; el 2 buzos, sweaters y quizás remeras de manga larga; el 3 vestidos, batas o delantales; el 4 camperas o buzos; el 5 sandalias y tacos, con la particularidad de ser calzados que dejan el pie al descubierto; el 6 camisas y remeras; el 7 zapatos y zapatillas; el 8 bolsos y carteras; y el 9 tacos, botas y zapatillas. A partir de esto podemos decir que la clase 2 y 6 se parecen bastante entre sí, así como las 5, 7 y 9; y que las clases 1 y 8 parecieran ser las que menos similitudes comparten con el resto.

Luego del análisis que hicimos viendo el 'ploteo' de las imágenes (para hacernos una idea de con qué nos íbamos a encontrar), seguimos un análisis a partir del valor promedio de cada píxel, para obtener resultados más generales y objetivos. Para esto vamos a comparar las clases 1, 2 y 6.

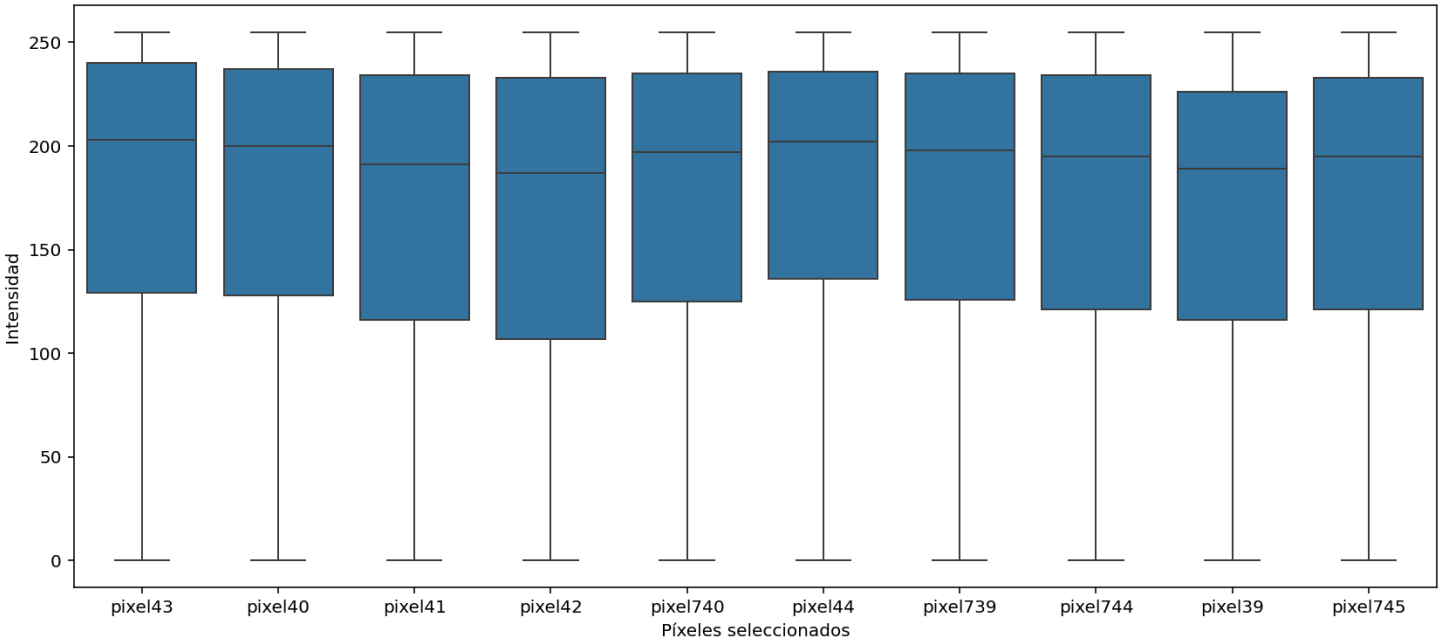


- *Promedio de valores de píxeles por clase*

Lo que representamos con este gráfico es el valor promedio de intensidad por cada uno de los píxeles, asociado a cada una de las tres clases a comparar. Lo que pudimos observar fue que las formas que adoptan las clases 2 y 6 son similares, mientras que la clase 1 se comporta de manera distinta (exceptuando algunos mínimos). Estas observaciones nos pueden dar una idea de cómo son las imágenes dentro de cada clase. Si bien no siempre coinciden en valores, al tener estos mismos patrones de comportamiento podemos decir que las clases 2 y 6 son más parecidas entre sí, mientras que la clase 1 se diferencia más.

Esto tiene coherencia con el análisis previo que hicimos donde notábamos que la clase 1 eran pantalones y la 2 y la 6 eran ambas de remeras, buzos y prendas similares.

c) Para medir similitudes dentro de una misma clase tomamos en cuenta el mismo factor que hacía que un píxel fuese o no relevante, evaluamos cuánto variaba a lo largo de todos los datos dentro de esa misma categoría. Entonces decidimos que si los píxeles que más variaban tenían una variación baja (para este contexto, consideramos variación baja a que esté dentro de un rango de 100), quería decir que el comportamiento de esos píxeles es parecido imagen a imagen, lo que concluimos que implicaría semejanza entre imágenes de una misma clase.

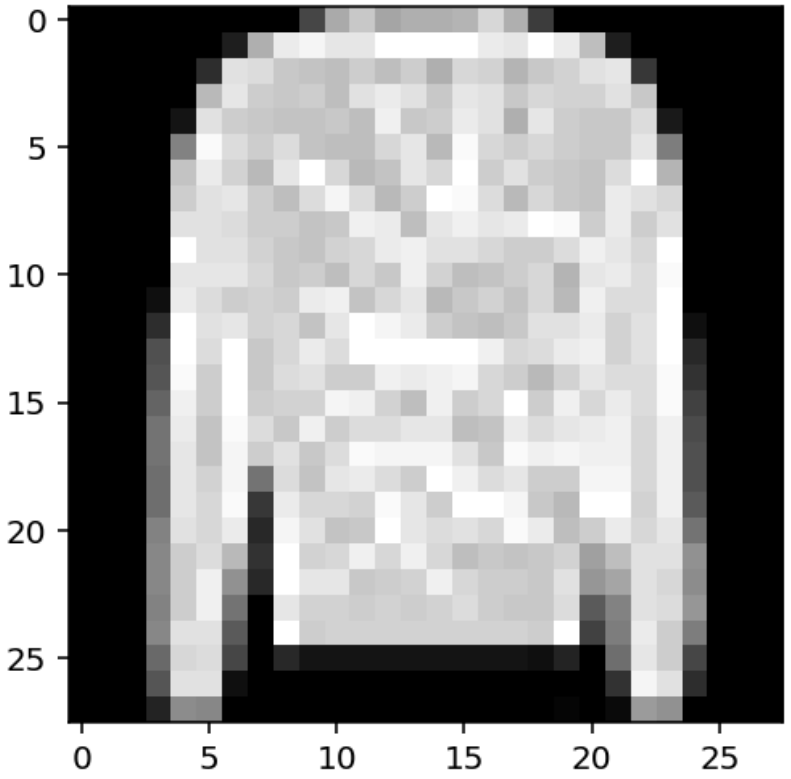


- *Variabilidad de los 10 píxeles más cambiantes en la clase 0*

Tomamos como ejemplo a la clase 0. La mayoría de los píxeles más variantes cumplen con que “varían poco”, por lo que consideramos que en esta clase las imágenes serían relativamente semejantes (considerando que el resto de píxeles también varían relativamente poco, ya que varían menos que estos) Si por ejemplo, tuviésemos muchos valores que variasen mucho, indicaría que para algunas imágenes la forma de la prenda cambia mucho. Para entenderlo visualmente con un ejemplo: si tuviésemos un subconjunto de píxeles en los que suelen aparecer las mangas de las remeras, podríamos identificar que hay varianza entre las remeras de manga larga y de manga corta, ya que en algunas estos píxeles tienen valor 0 y en otras todo lo contrario. Este fue el criterio que utilizamos para encontrar similitudes entre las imágenes de una misma categoría.

d) Creemos que al ser imágenes, la información más clara sobre estas se encuentra no en la tabla con el valor de cada píxel sino en la representación visual que se genera con todos los valores a la vez. Debido a la gran cantidad de atributos y a la naturaleza de los mismos, el valor de brillo de un píxel individual nos aporta poco a la hora de entender las características de lo que quieren representar, a diferencia de otros tipos de atributos que describen de manera más concisa la información que representan (como por ejemplo, la edad de una persona).

Sin embargo la estructura presentada sirve a la hora de hacer un análisis general, numérico y objetivo sobre los datos mediante los modelos de clasificación y otras herramientas. Esta representación de los datos es lo que en definitiva nos permitió abstraer las características de una imagen y sacar conclusiones a partir de estas.



- *Ejemplo de visualización de una de las imágenes*

2) Clasificación binaria

Para esta parte entrenamos varios clasificadores KNN. Probamos variando cantidad de datos, cantidad de vecinos a tener en cuenta para cada uno y también variamos la selección de atributos. Esta selección fue en base a lo que creíamos que iban a resultar en datos clave para los modelos, teniendo en cuenta el análisis previo.

Como una aproximación inicial tomamos aquellos píxeles que más valor promedio tenían para asegurarnos de no tener píxeles negros, aunque luego encontramos criterios con mejores resultados. Luego agregamos un subconjunto de datos aleatorios para tener un contrapunto útil a la hora de comparar las otras selecciones[1]. Finalmente calculamos el promedio de valores de todos los píxeles, y calculamos las diferencias que tenían entre una clase y otra. A partir de esto elegimos los que más cambiaban. La cantidad de vecinos fueron 2, 5, 7 y 10 y la cantidad de píxeles, 3, 5 y 8.

Los resultados fueron los siguientes:

k	cant_datos	Más Brillantes Train	Más Brillantes Test	Random Train	Random Test	Más Distintos Train	Más Distintos Test
2	3	0.8	0.638929	0.904018	0.866786	0.931786	0.870357
5	3	0.7725	0.676786	0.907768	0.91	0.926696	0.915
7	3	0.751607	0.684286	0.909464	0.916071	0.924643	0.920357
10	3	0.729911	0.695	0.908839	0.916786	0.922857	0.9175
2	5	0.824018	0.669286	0.954821	0.9225	0.933036	0.882857
5	5	0.795714	0.701071	0.949196	0.9425	0.929821	0.918929
7	5	0.783125	0.704643	0.94375	0.941071	0.926786	0.920357
10	5	0.761786	0.703571	0.942321	0.941429	0.925268	0.923571
2	8	0.818661	0.663929	0.950089	0.8975	0.934732	0.879643
5	8	0.795804	0.702143	0.941786	0.9275	0.933304	0.922857
7	8	0.779554	0.711786	0.937143	0.927143	0.929732	0.922143
10	8	0.754196	0.7025	0.932411	0.921071	0.928036	0.923929

- *Resultados de los modelos para distintas selecciones de píxeles.*

Como era de esperar, los modelos más precisos coincidieron con los que tomaban mayor cantidad de datos para entrenar, aunque esta no fue la única característica importante ya que también notamos que los modelos se comportaron muy distintos en relación con la cantidad de vecinos tenidos en cuenta.

Si vemos los resultados de los *más brillantes* podemos notar que el train empeora con el aumento de los vecinos, mientras que el test mejora ligeramente. Es el peor de los criterios que tomamos.

---

[1] Antes de este criterio, hicimos un análisis que tomaba valores de filas y columnas aleatorias de la imagen, como también uno que seleccionaba cuadrados en esta. Como vimos que todos estos tenían resultados peores a los de seleccionar píxeles de manera aleatoria (y por una cuestión de procesamiento y cómputo), decidimos dejar este último método.

En el criterio *más distintos*, notamos que los valores de exactitud no tuvieron una relación tan directa con la cantidad de píxeles tomados, sino que fue mucho más relevante diferenciar la cantidad de vecinos tenidos en cuenta. Para 2 vecinos se dieron los peores resultados, mientras que para 5, 7 y 10 , en líneas generales, se mantuvieron en valores similares. Nos sorprendió que aumentar la cantidad de atributos tomados no implicase una mejoría. Tuvo una *accuracy* un poco menor respecto al mejor criterio.

Para el conjunto de datos tomados al azar obtuvimos valores muy altos, incluso superiores al criterio que elegía los píxeles que más cambiaban entre clases. Este resultado nos llamó la atención al tratarse de una elección que toma atributos aleatorios. Sospechamos que elegir los píxeles de esta manera, al otorgarnos una mejor distribución en cuanto a su ubicación en la imagen, permitió capturar mejor las formas y diferencias entre las clases. Notamos que los otros dos criterios elegían píxeles muy cercanos entre sí. A partir de esto, decidimos elegir los píxeles que más variación tenían, pero separando las imágenes en regiones, para lograr distribuir los píxeles elegidos en toda la imagen. Haciendo esto obtuvimos los siguientes resultados:

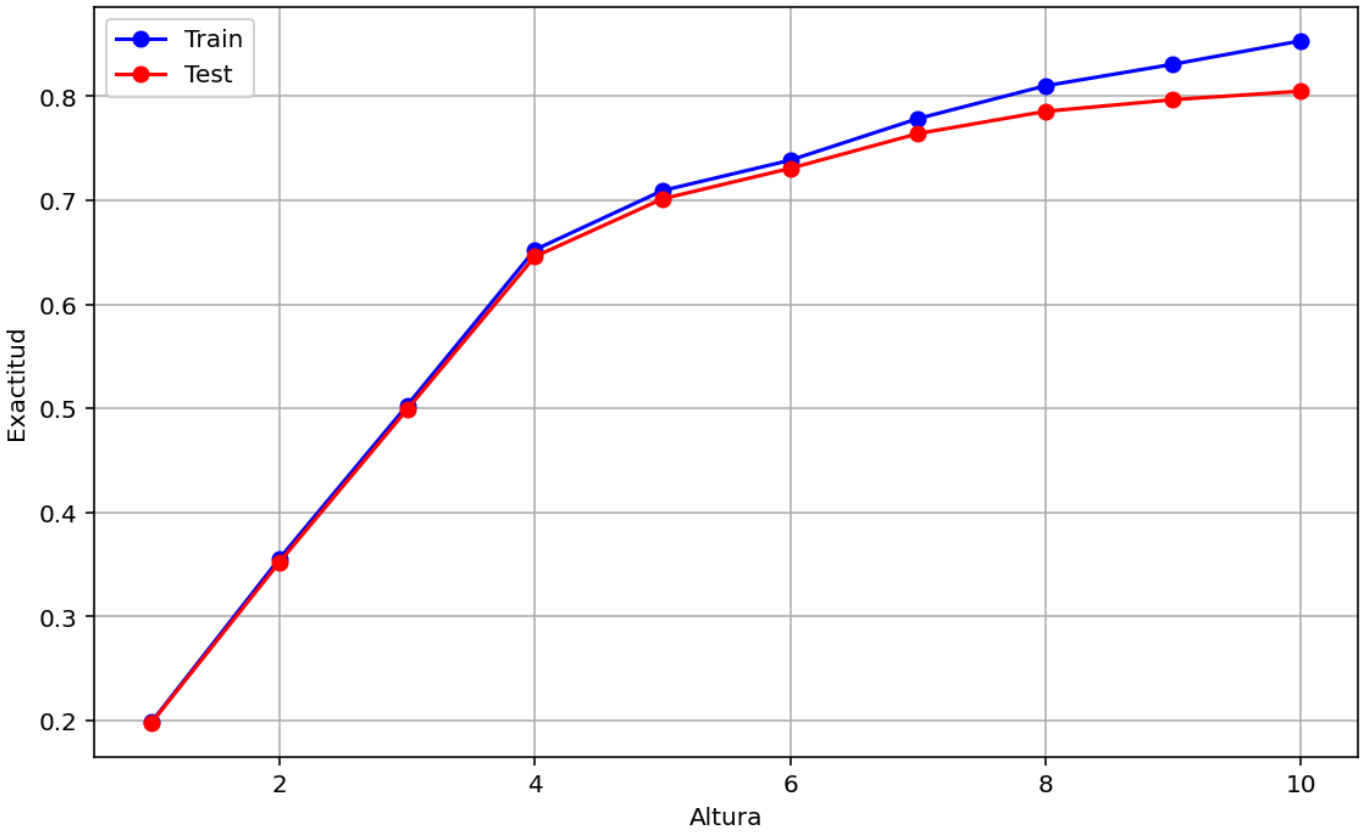
k	cant_datos	Más Distintos Sep. Train	Más Distintos Sep. Test
2	3	0.939643	0.9025
5	3	0.935714	0.932857
7	3	0.932411	0.933571
10	3	0.931786	0.935357
2	5	0.957232	0.932143
5	5	0.953482	0.947857
7	5	0.95375	0.9475
10	5	0.952411	0.948214
2	8	0.973929	0.946071
5	8	0.965982	0.96
7	8	0.964018	0.958929
10	8	0.961607	0.958571

- *Selección de “Más distintos” pero forzando separación entre atributos.*

El mejor modelo siguiendo este método tuvo una *accuracy* de exactamente 96%, con 5 vecinos y 8 píxeles, superando la performance de todas las selecciones anteriores.

3) Árboles de decisión

En este último apartado nos centraremos en la selección de un árbol de decisión que mejor prediga la clase a la que pertenece una imagen dada. Aquí graficamos la exactitud del modelo en función de la altura del árbol:



- *Resultados de comparar distintas alturas con una cantidad de muestras fijas*

En este gráfico observamos que la exactitud mejora a medida que aumenta la altura, aunque empieza a crecer más lento a partir de la altura 4, tendencia más marcada en los resultados del conjunto de *test*, comparando con los del conjunto de entrenamiento. Si bien para la altura 10 los resultados en el test no empeoran, empieza a crecer la distancia con el conjunto train. Creemos que esto es un caso de overfitting, puesto que si bien la performance no empeora, el modelo está encontrando patrones en el entrenamiento que no generalizan a otros conjuntos de datos.

Ahora vamos a ver una comparación entre distintos criterios, a distintas profundidades de árboles (mostramos las 10 mejores) y tomando distintas cantidades de datos. Probamos alturas del 1 al 10, criterios entropy y gini, y el uso de 200, 300 y 784 píxeles.

Para que la comparación sea más estricta vamos a usar k-folding (con 5 splits), teniendo en cuenta que a partir de esto tendremos nuestro modelo final.

Criterio	Cant_Atributos	Altura	Accuracy_promedio	Recall_promedio	Precision_promedio	F1_promedio
entropy	300	10	0.807143	0.80756	0.808318	0.806916
entropy	784	10	0.807125	0.80765	0.807813	0.806688
gini	784	10	0.806339	0.806692	0.808303	0.806032
gini	300	10	0.803214	0.803711	0.804993	0.802772
entropy	200	10	0.800321	0.800965	0.801671	0.799888
gini	200	10	0.799518	0.800057	0.800957	0.799358
gini	784	9	0.799125	0.799597	0.801856	0.799045
entropy	784	9	0.799036	0.79962	0.800114	0.798419
entropy	300	9	0.793857	0.794501	0.795923	0.792287
gini	200	9	0.793446	0.794098	0.796534	0.792839

- *Mejores 10 promedio en los folds, de ambos criterios a distintas alturas y distintas cantidades de atributos*

Con estos resultados elegimos como mejor opción al árbol de profundidad 10 que utiliza el criterio ‘entropy’, tomando 300 píxeles.

Vamos a utilizar este último árbol para evaluarlo en el conjunto Held-Out:

	0	1	2	3	4	5	6	7	8	9
0	1198	1	31	67	12	1	117	0	18	1
1	10	1345	11	48	4	1	6	0	0	0
2	25	6	991	15	255	1	91	0	8	1
3	64	29	13	1121	92	1	19	1	7	0
4	7	5	183	62	1001	0	108	0	10	0
5	1	2	2	8	1	1250	1	89	14	50
6	293	7	205	52	138	2	628	0	26	1
7	0	0	0	0	0	43	0	1271	3	109
8	13	3	23	9	17	17	21	10	1312	6
9	2	3	2	0	0	25	1	63	1	1289

- *Matriz de confusión del árbol seleccionado*

Conseguimos una accuracy de 81,47% para este árbol como resultado final. La matriz de confusión nos permite ver de manera más precisa lo planteado al principio del trabajo, la similitud entre clases. Obtuvimos resultados ligeramente distintos a los que habíamos llegado con nuestras observaciones sobre los ploteos. Si bien las clases 1 y 8 (pantalones y carteras/bolsos) fueron correctamente identificadas en la mayoría de los casos por el modelo (algo que esperábamos porque visualmente se distinguen mucho de las otras), las clases 5, 7 y 9, de las que esperábamos peor rendimiento por el hecho de ser calzados, tuvieron mejor rendimiento del que pensábamos. Por otro lado, la clase 6 tuvo un mal rendimiento como esperábamos, se confundió mucho con las clases 0, 2 y 4; todas prendas del tren superior como abrigos, remeras, sweaters, etc. Las clases 2 y 4 también se confunden entre sí y con la 6. La clase 0 se confunde con la 6 también. Podemos explicar esto teniendo en cuenta que las clases 2 y 4 corresponden a prendas de manga larga, la 0 de manga corta, y la 6 de ambas.

Conclusiones

En este trabajo analizamos un dataset de imágenes de ropa y probamos distintos modelos para predecir de qué prenda se trataba. Primero hicimos un análisis visual y numérico para entender mejor los datos, notar similitudes entre clases y entender de qué forma encontrar los atributos más relevantes para nuestro objetivo. Luego probamos modelos KNN y árboles de decisión, cambiando parámetros y formas de elegir los subconjuntos de píxeles.

A partir de esto pudimos encontrar qué parámetros y atributos funcionaban mejor, de acuerdo a diferentes métricas. La selección aleatoria de píxeles para el ejercicio 2 nos dio buenos resultados, superando al criterio que pensábamos era mejor. Esto nos permitió entender la importancia de la distribución de los píxeles seleccionados y, a partir de esto, implementar un método que encontrase los píxeles más distintos en valor entre las dos clases (el criterio que pensábamos que obtendría mejores resultados) pero mejor distribuidos, separando la imagen en franjas y obteniendo uno de cada una. Con este criterio obtuvimos los mejores resultados.

El mejor modelo para el ejercicio 3 fue un árbol de decisión de altura 10, usando el criterio ‘entropy' y seleccionando solo 300 píxeles de los 784. Este tuvo buenos resultados con algunas clases y problemas con otras, pero al analizar la matriz de confusión, pudimos ver como la mayoría de errores eran cometidos al no poder diferenciar clases que hasta visualmente se parecían mucho. En conclusión, tuvimos resultados satisfactorios teniendo en cuenta las categorías de ropa y las similitudes que estas tenían entre sí.