DEPARTAMENTO DE COMPUTACION
Facultad de Ciencias Exactas y Naturales - UBA

Trabajo Práctico 02

Clasificación y selección de modelos

Laboratorio de Datos

Grupo Ctrl

Integrante	LU
Arango, Joaquín	342/24
Cardinale, Dante	593/24
Herrero, Lucas	179/24

Introducción

Fashion-MNIST es un dataset que consiste en un conjunto de 70000 imágenes, de tamaño 28x28 y en escala de grises; donde cada una está asociada a una clase de prenda, etiquetadas del 0 al 9. Las clases en cuestión son: remera/top, pantalón, vestido, abrigo, sandalia, camisa, zapatilla, cartera y bota (ordenadas acorde a su label, de menor a mayor).

En este trabajo, nos proponemos desarrollar modelos predictivos capaces de lograr la identificación de las prendas. Al contar con los labels correspondientes a cada registro, utilizaremos modelos de aprendizaje supervisado, en donde nuestros algoritmos usarán información de la tabla para entrenar modelos que asignen outputs a inputs.

Al ser estos outputs variables categóricas, los modelos son, entonces, de clasificación.

Para lograr nuestro objetivo, llevaremos a cabo un análisis exploratorio de los datos, cuyo propósito es la realización de un conjunto de observaciones previas, a fin de conocer y comprender el dataset descripto, observando patrones y diferencias entre las clases, evaluar relevancia de los atributos presentados, discernir las características identificadoras de cada clase y más. Para ello, hemos utilizado herramientas como la realización de consultas SQL, armado de funciones programadas en lenguaje python, construcción de visualizaciones gráficas, etc.

A partir de este análisis, elaboraremos, por un lado, modelos de clasificación binaria, en el cual nos propondremos configurar modelos predictivos que diferencien correctamente las prendas pertenecientes a las clases remera/top y cartera, siendo el algoritmo KNN (K-Nearest Neighbors) el que implementaremos para tal función; y por otro lado, desarrollaremos modelos de clasificación multiclase, cuya motivación es la discriminación de la totalidad de las clases presentadas, es decir, la misión será la construcción de modelos predictivos que logren clasificar los outputs acorde al tipo de prenda que representan. Para lograr el cometido, serán utilizados árboles de decisión, con técnicas extra de evaluación y construcción como lo es KFolding.

Finalmente, presentaremos los resultados en función de métricas como accuracy, recall, precisión o f1 score; acompañado de representaciones visuales que contribuyan a la exhibición y esclarecimiento de los datos.

Análisis exploratorio de Datos

Como se ha mencionado, Fashion-MNIST es un dataset compuesto por 70000 imágenes de prendas de vestir distribuidas/categorizadas en diez clases distintas, con 7000 muestras por cada categoría. Cada imagen, de dimensiones 28x28 píxeles, se encuentra en escala de grises, donde cada píxel está representado por un valor de intensidad de entre 0 y 255 (bordes incluidos).

Desde una perspectiva estructural, cada fila del dataset corresponde a una prenda específica, en la que los atributos son los valores de la intensidad de los 784 píxeles (28x28) que conforman su representación visual. Además, se incluye el atributo adicional ‘label’, que es número que identifica a la clase a la cual pertenece la prenda. Este formato no solo nos permite visualizar las prendas como imágenes (ver figura I), sino también procesarlas numéricamente para realizar las tareas de clasificación deseadas.

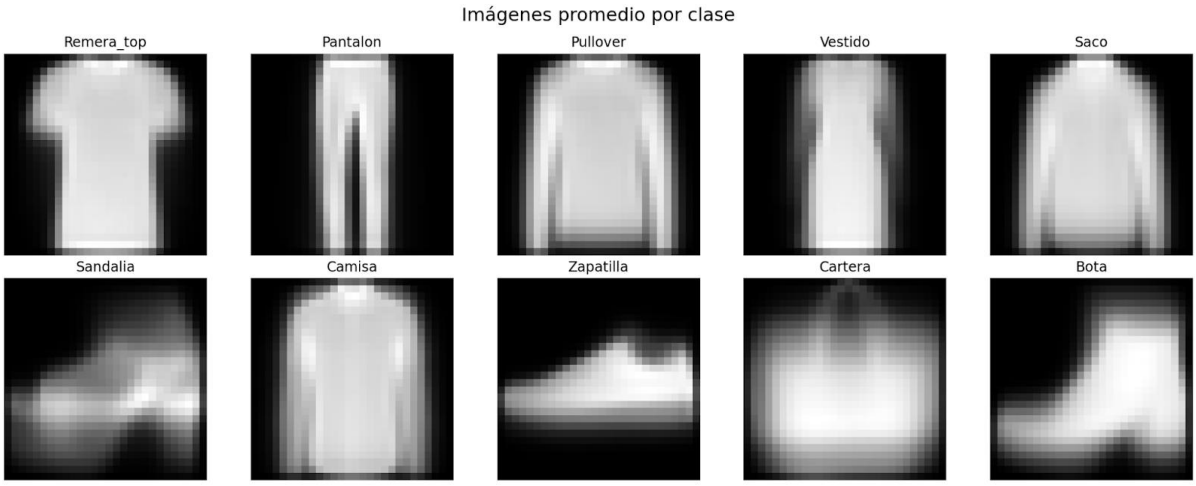


Figura 1 : Imágenes promedio de cada prenda

La composición por imágenes del dataset nos obliga a realizar distintos tipos de medidas para observar el comportamiento de los datos. La única variable categórica explícita que ofrece Fashion-MNIST es la columna “label”, que indica el tipo de prenda, y permite separar las observaciones en 10 clases distintas. El resto de los atributos representan valores de píxeles en escala de grises, todos del mismo tipo y sin un significado individual evidente fuera de su contexto espacial. Por este motivo, consideramos inadecuado enfocar el análisis en atributos individuales (como un solo píxel), ya que dos imágenes de la misma clase pueden presentar variaciones locales sin que eso implique una diferencia significativa en su categoría. Asimismo, debido al amplio rango de valores posibles por píxel (0–255), segmentar imágenes basándonos en coincidencias de intensidad en posiciones específicas resultaría poco informativo y computacionalmente ineficiente.

Si tomamos el dataset de Titanic como comparación, las diferencias en la estructura de los datos son evidentes. Más allá de la menor cantidad de atributos, cada variable en ese caso toma un conjunto limitado de valores discretos (por ejemplo, el sexo o la clase del pasajero), lo que facilita la identificación de patrones relevantes para la clasificación. Además, los atributos del dataset presentan una semántica que el público general maneja y comprende (todos identificamos lo que define la edad, sexo, clase, etc, de una persona), lo que permite, aún sin conocer los datos, tener una idea de la relevancia que va a presentar cada característica en cuanto a la clasificación deseada (‘sobrevivió’, ‘no sobrevivió’). Un píxel de una imagen, sin embargo, no logra trasmitir una idea de lo que ocurre en la completitud de las imágenes de nuestra tabla.

Ante esta diferencia de naturaleza entre los datasets, decidimos incorporar técnicas específicas que nos permitan comparar imágenes dentro de una misma clase y entre clases distintas. Esto incluye el uso de imágenes promedio, análisis de variabilidad por píxel y métricas robustas como el IQR. A través de estos enfoques, nos propondremos extraer conclusiones significativas sobre los patrones de las prendas de cada clase.

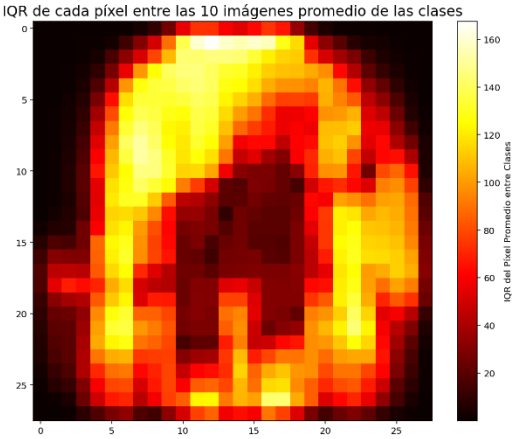


Figura 2 : Imagen del IQR de cada pixel entre imágenes promedio

En cuanto a la relevancia de estos píxeles, hemos observado que los de mayor incidencia corresponden a aquellos que, al momento de graficar la imagen (es decir, en términos de su esquema visual), se sitúan en la región central de la misma, en especial los que forman una circunferencia alrededor de este centro (ver figura II), ya que son los que poseen mayor diferencia en sus valores según el tipo de prenda que representen.

Por otro lado, los atributos que denominamos entonces 'irrelevantes', son aquellos que se encuentran cercanos a los bordes -especialmente en las esquinas- pues presentan valores de gran similitud (mayormente oscuros, es decir, con baja intensidad) independientemente del tipo de prenda. Tras un análisis del IQR de cada píxel por cada una de las clases, observamos que había ciertos píxeles (ubicados en las esquinas de las imágenes) cuyo IQR era exactamente 0 en todas las clases (ver figura III) y, además, en todas las clases el valor promedio de ese píxel era cercano a 0 (como se observa en la figura IV), por lo que decidimos no tenerlos en cuenta al momento de modelar (en cuanto se pueda despreciar atributos), ya que no aportan información alguna que pueda ser útil. En la figura V se pueden apreciar dichos píxeles.

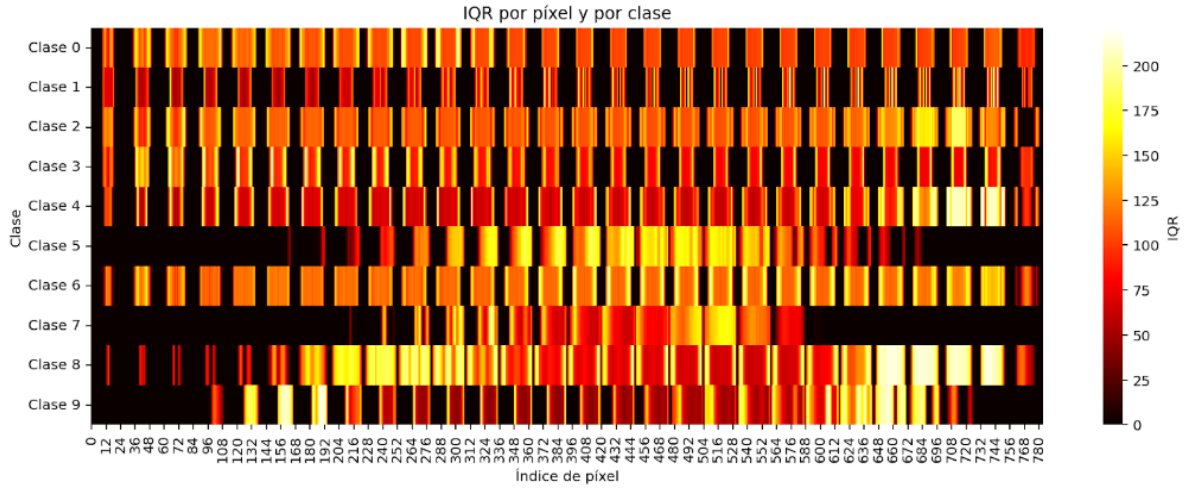


Figura 3 : Heatmap con IQR por pixel en cada clase

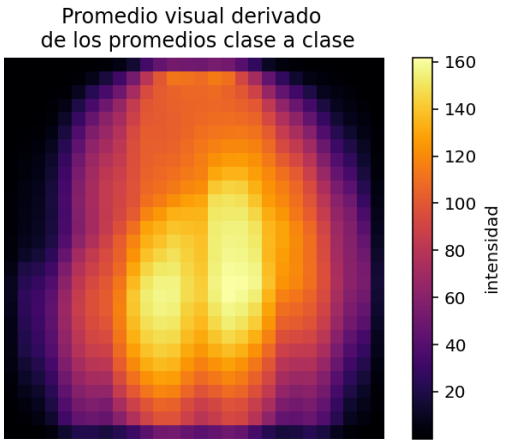


Figura 5 : Imagen con el promedio visual derivado de cada clase

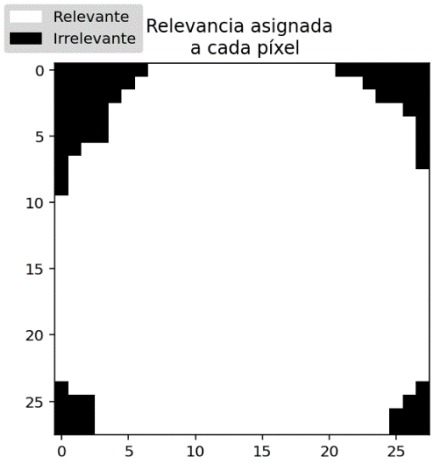


Figura 4 : Gráfico con los pixeles descartados

Para analizar las diferencias entre los distintos tipos de prenda, llevamos a cabo un procedimiento sistemático. En primer lugar, dividimos el DataFrame original en 10 subconjuntos, uno por cada clase. Luego, calculamos la imagen promedio por clase: para cada píxel, obtuvimos el valor promedio que toma en todas las imágenes pertenecientes a esa clase, lo que permite representar visualmente la intensidad típica de cada píxel y comprender su comportamiento dentro de una categoría específica (el promedio tiene la particularidad fundamental de que es el valor más cercano a todos los que representa, o como dice Walter Sosa en su libro 'Big Data', los promedios son 'los Beatles de la estadística'; por lo que nos resulta una medida importante a tener en cuenta). Posteriormente, para comparar dos clases, restamos entre sí las imágenes promedio correspondientes (píxel a píxel), obteniendo un nuevo array que refleja las diferencias de intensidad entre ambas clases. Este array se puede graficar para visualizar en qué regiones de la imagen las prendas tienden a diferenciarse más. A continuación, exponemos ciertos gráficos para diferenciar prendas.

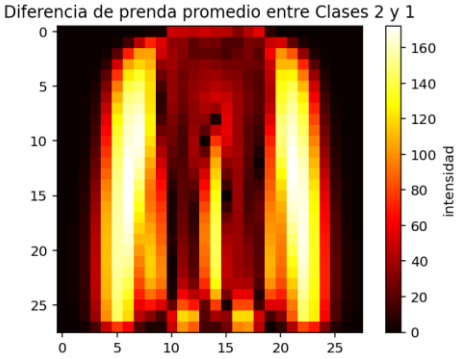


Figura 6 : Imagen con la diferencia entre clases 2 y 1

Al restar las imágenes promedio de pullover y pantalón, podemos observar (en la figura VI) que por los costados se encuentran un gran número de píxeles con alta intensidad, mientras que en la región central las diferencias son menores. Esto puede explicarse por la forma típica de ambas prendas: el pantalón ocupa principalmente el área central de la imagen, mientras que el pullover incluye, además, mangas que se extienden hacia los laterales. Debido a esto, se puede inferir que estas clases presentan patrones espaciales sumamente distintos. También se puede apreciar que el extremo de los bordes laterales se encuentra 'vacío', como se ha dado a ver anteriormente.

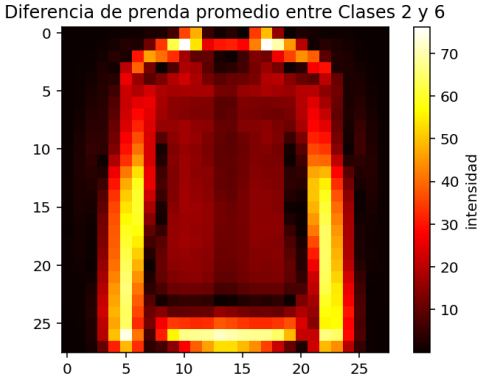


Figura 7 : Imagen con la diferencia entre clases 2 y 6

Por otro lado, comparando los promedios de la clase pullover y camisa, se observa en la figura VII una diferencia notable con respecto al contraste previamente analizado con pantalón. Si bien los píxeles con mayor diferencia de intensidad también se ubican en el sector de las mangas, en este caso abarcan principalmente la zona inferior de las mismas. Además, la magnitud de estas diferencias es menor que en la comparación anterior, lo que indica que ambas prendas presentan valores más altos (mayor intensidad) en regiones similares. Considerando la forma y disposición espacial de las prendas, es razonable suponer que este comportamiento se debe al parecido entre pullovers y camisas, ya que ambas incluyen mangas y ocupan áreas similares dentro de la imagen. En consecuencia, estas clases muestran representaciones visuales más cercanas/parecidas, lo que puede dificultar su correcta diferenciación por parte de un modelo de clasificación.

Y, nuevamente, los bordes laterales de la imagen poseen intensidad nula (o prácticamente nula).

Para profundizar en el tema, graficamos y evaluamos también las diferencias entre la imagen promedio de la clase remera (0) y las clases cartera (8) y vestido (3) (figura VIII y figura IX respectivamente). En ambos gráficos, se observa que las mayores diferencias de intensidad se concentran en los extremos de las prendas, lo que indica que esos píxeles son los que más varían entre clases. Sin embargo, los patrones de diferenciación no son iguales en ambos casos.

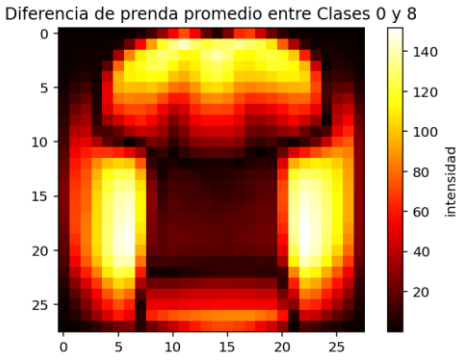


Figura 8 : Diferencia entre clases 0 y 8

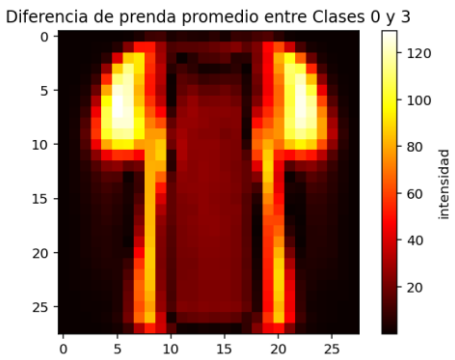


Figura 9 : Diferencia entre clases 0 y 3

Al comparar remera con cartera, se destaca una mayor cantidad de píxeles intensos tanto en estos extremos (pues el ancho de la cartera supera el de la remera) como en la parte superior de la prenda, posiblemente asociada al poco espacio que ocupa la manija del bolso en comparación al torso superior de la remera.
Por otro lado, contrastando remera con vestido, las diferencias se concentran mayormente en los bordes superiores de la remera, lo cual puede vincularse a la presencia de mangas en ella que no están presentes en todos los vestidos.

En función de estas observaciones, podemos inferir que la clase remera es más fácilmente diferenciable de cartera que de vestido, debido a la mayor magnitud y tamaño de las diferencias entre sus respectivas imágenes promedio.

Hemos mencionado que el promedio es el valor más cercano a todos los que representa (lo cual es cierto), pero éste es influenciable por valores atípicos/outliers (valores extremadamente chicos/grandes). Además, no nos da una idea de la distribución de los datos, lo cual es interesante poder analizar si queremos ser capaces de identificar las características particulares de cada clase.
Es por eso que, en cuanto al análisis de similitud entre imágenes de una misma clase, y con el objetivo de estudiar la variabilidad por píxel sin que los valores extremos distorsionen los resultados, definimos una métrica basada en la representación gráfica de un boxplot, a la que denominamos pseudo-rango.
Esta medida toma como valores mínimo y máximo aquellos que corresponden a los extremos de los *whiskers* del boxplot, lo cual permite capturar el rango típico de variación de cada píxel, excluyendo los outliers. Matemáticamente, tiene la siguiente forma:

$$Pseudorango = Límite\ superior\ whisker - Límite\ inferior\ whisker$$

siendo

$$Límite\ superior\ whisker = \max\{valor_{pixel} : valor \leq Tercer\ quartil + 1{,}5 * IQR\}$$

$$Límite\ inferior\ whisker = \min\{valor_{pixel} : valor \geq Primer\ quartil - 1{,}5 * IQR\}$$

De esta forma, obtenemos una representación más robusta de la dispersión de los valores dentro de la clase, lo que nos permite evaluar la similitud entre las imágenes que la conforman.
A partir de esta métrica, realizamos gráficos de dos clases de ejemplo, cartera y bota, y analizamos la distribución de las intensidades de los píxeles en la clase, para determinar la similitud entre las imágenes que poseen un mismo label.

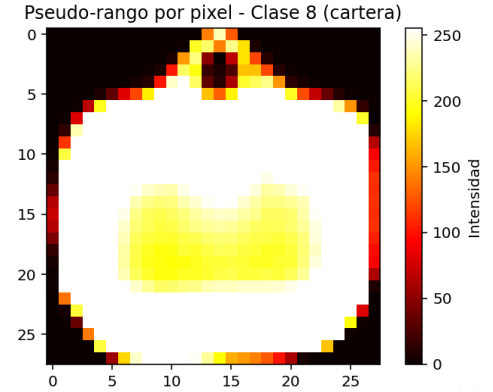


Figura 10 : Imagen con pseudo-rango por pixel de clase 8

Al analizar la figura X correspondiente a la clase *cartera*, se observa que todo el centro de la figura presenta niveles de intensidad elevados, lo cual indica una alta variabilidad en los valores de esos píxeles entre las distintas imágenes de la clase. En contraste, en el contorno de la imagen, los valores de intensidad son considerablemente más bajos, lo que sugiere una mayor consistencia en esa área. A partir de esta distribución, se puede inferir que las imágenes de carteras presentan diferencias notables entre sí, especialmente en el centro, lo que refleja una mayor diversidad en la forma o el diseño de los objetos representados.

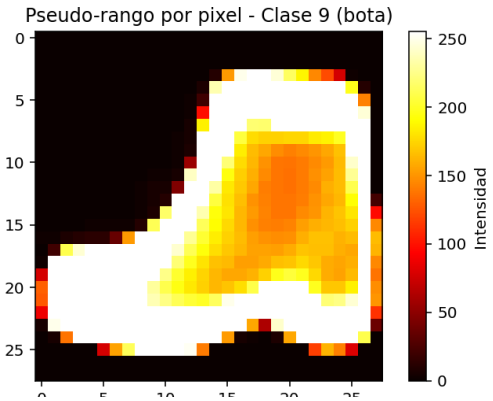


Figura 11 : Imagen con pseudo-rango por pixel de clase 9

Por otra parte, al observar la figura XI correspondiente a la clase bota, se identifican diferencias notables con respecto al gráfico anterior. En este caso, la variación de intensidad es considerablemente menor, y la cantidad de píxeles con grandes diferencias en intensidad se reduce significativamente. Se equiparan la cantidad de valores medios y altos a lo largo de la imagen, lo que sugiere una menor dispersión en los valores de cada píxel en contraste con la situación de la clase cartera. Esta menor variabilidad indica una mayor similitud entre las imágenes de las botas, lo que permite inferir que, en comparación con las carteras, las instancias de esta categoría presentan una estructura visual más consistente, aunque es observable también que hay varios sectores cuyas grandes intensidades indican presencia de registros de la clase con formatos heterogéneos en esas zonas.

Clasificación binaria

En esta sección, nos enfocaremos en la construcción de modelos para clasificación binaria, siendo las clases remera-top (clase 0) y cartera (clase 8) aquellas a predecir. Como se tiene 7000 ejemplares por prenda, hay cantidades uniformes por clase a predecir.
No obstante, el conjunto de los 14000 datos resultante lo dividimos en *conjunto de entrenamiento* y *conjunto de testeo*, donde cada uno tendrá la finalidad que indica su nombre. La distribución será 80-20 respectivamente.
Los modelos a construir se basan en KNN, un algoritmo de aprendizaje supervisado (es decir, contamos con un conjunto de entrenamiento del que conocemos las etiquetas) en el que la clasificación de una nueva instancia se determina tras buscar los k-puntos más cercanos dentro del conjunto de entrenamiento, ver a qué clase están asignados y elegir finalmente la clase de presencia mayoritaria.
Es decir, que teniendo una serie de atributos $\{X_1, \dots, X_n\}$, la etiqueta Y a asignar será aquella la cual tienen asignados la mayoría de los k-puntos de nuestro conjunto de entrenamiento 'más similares/cercanos' a la serie definida (es decir, los registros con atributos de valores más próximos).
Es entonces que surge la necesidad de seleccionar aquellos atributos que mejor diferencian a una clase de la otra. Si bien es cierto que es posible usar la totalidad de los píxeles, ésto trae consigo diversas dificultades extras: en primer lugar, al usar la totalidad de los atributos, la cantidad de datos que debe procesar el modelo es exorbitantemente grande si se toma en cuenta que, como visto en el análisis exploratorio, no todos los píxeles poseen la misma relevancia para la distinción de una clase; por otro lado, estos datos "extra" causan una gran ralentización en el desarrollo del modelo, consumiendo mucho tiempo en tareas que pueden ser optimizadas.
Luego, ante esta necesidad descrita, hemos determinado que los factores por los cuales elegiremos los píxeles a comparar son el promedio, la mediana e IQR; y la suma máxima de atributos a evaluar será 10.
Es entonces que el proceso de selección de estos atributos es:

- Para promedio y mediana, se realizan procesos independientes pero análogos: calculamos la métrica correspondiente (promedio o mediana) para cada atributo en ambas clases, obtenemos la diferencia absoluta entre esas métricas y extraemos los 10 píxeles cuya diferencia sea mayor.
- Para IQR, la heurística se modifica ligeramente:
    - Primero, descartamos los píxeles cuyos valores presenten intersección en estos rangos. Es decir, al momento de ver el IQR de los píxeles, pediremos inicialmente que los rangos sean disjuntos.
    - Una vez aplicado este filtro, identificamos los diez atributos cuyos rangos presentan la mayor distancia entre sí. La distancia mencionada es la diferencia entre el límite inferior del IQR más alto y el límite superior del IQR más bajo.

En principio la idea inicial era, a partir del mismo mecanismo que el usado para selección de atributos en base a IQR, utilizar al pseudo-rango como otra medida óptima. Sin embargo, al llevarlo a cabo, nos hemos encontrado con que los conjuntos de atributos seleccionados eran los mismos. Es por eso que, por practicidad, hemos optado por mantener uno de los dos factores de selección, pero nótese en todo momento que el resultado es igual de válido para ambos.

Es así como finalmente desarrollamos y entrenamos diversos modelos de clasificación KNN, en función de los distintos atributos, sus cantidades y el número de vecinos (k).
La presentación de resultados la dividimos en dos partes, en línea con ciertos intereses de visualización que detallaremos. Estos resultados son evaluados sobre el conjunto de testeo separado previamente.

Clasificación binaria: dos atributos y veinte vecinos

Por un lado, nos interesa plantear y visualizar modelos formados a partir de una cantidad fija de vecinos y atributos. Hemos decidido que estas cantidades fijas sean 2 píxeles y 20 vecinos.
Esta elección se basa en que, al tener dos atributos, nos es posible la realización de gráficos que exhiban la "frontera de decisión" del modelo; es decir, gráficos en donde se pueda analizar, ante determinado input, cómo lo clasificará el sistema. Por otro lado, la elección de veinte vecinos está influenciada por las diversas actividades y trabajos que hemos realizado en materia de modelos predictivos, y por la gran cantidad de los datos con los que contamos para entrenar el modelo, lo que nos impulsa a estimar como sensata y adecuada la suma elegida. Los resultados fueron evaluados en función del accuracy y f1 score.

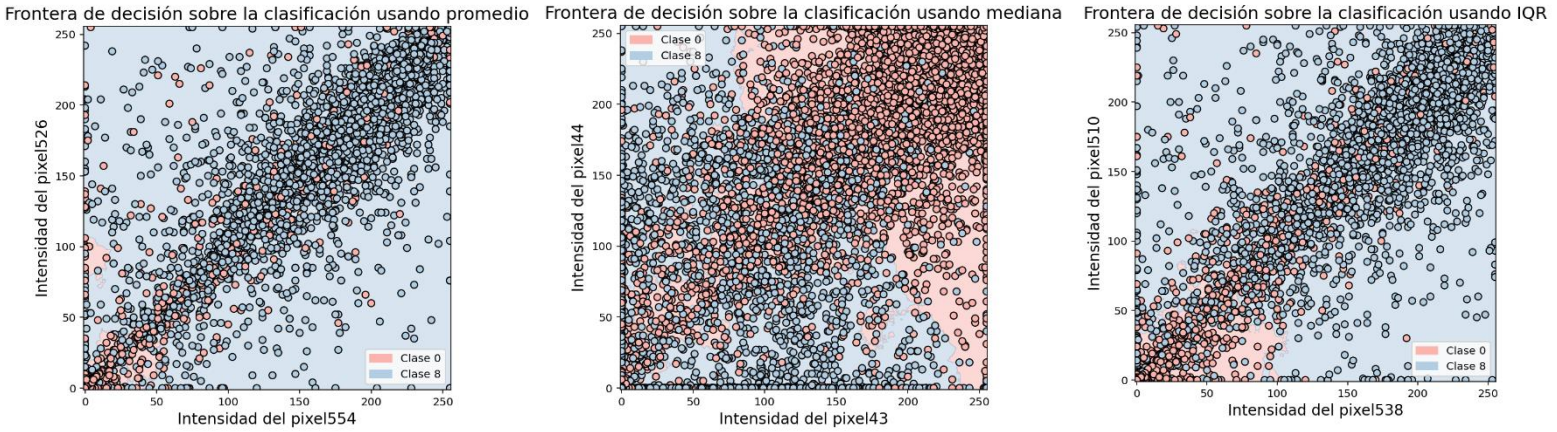


Figura 12 : Fronteras de decisión del algoritmo KNN en función de dos píxeles y veinte vecinos; por cada par de atributos de los distintos criterios de selección valorados

En la figura XII (fronteras de decisión) se puede apreciar lo detallado respecto del por qué el modelo clasifica en la forma en que lo hace. Es decir, los gráficos logran explicitar el fundamento detrás de las etiquetas asignadas; pues es posible observar la asignación en base a la visualización de la distribución de los distintos registros de cada clase en la figura.

Los resultados de estos modelos son:

MÉTRICA	ACCURACY SCORE	F1 SCORE
Promedio	0.9175	0.9174999894770395
Mediana	0.8428571428571429	0.8428532141874976
IQR/Pseudo-rango	0.9021428571428571	0.9021388128795118

La extracción de los atributos cuya diferencia absoluta de promedios era mayor, resultó ser el mejor factor de selección de atributos, teniendo en cuenta las cantidades respectivas de los parámetros fijados.

Cantidad de atributos y vecinos variable

Nos interesa apreciar ahora la variación de cierta métrica en función del volumen de atributos y vecinos valorados. No obstante, por pragmatismo, establecemos ciertas limitaciones en cuanto a estos montos, acotándolos a valores enteros comprendidos entre 3 y 10, inclusive; y la métrica estudiada es el accuracy. La figura XIII (matrices) muestra la fluctuación del accuracy score (redondeada en dos decimales) en base a las distintas cantidades de atributos y vecinos; diferenciando en base al mecanismo de selección de los píxeles evaluados.

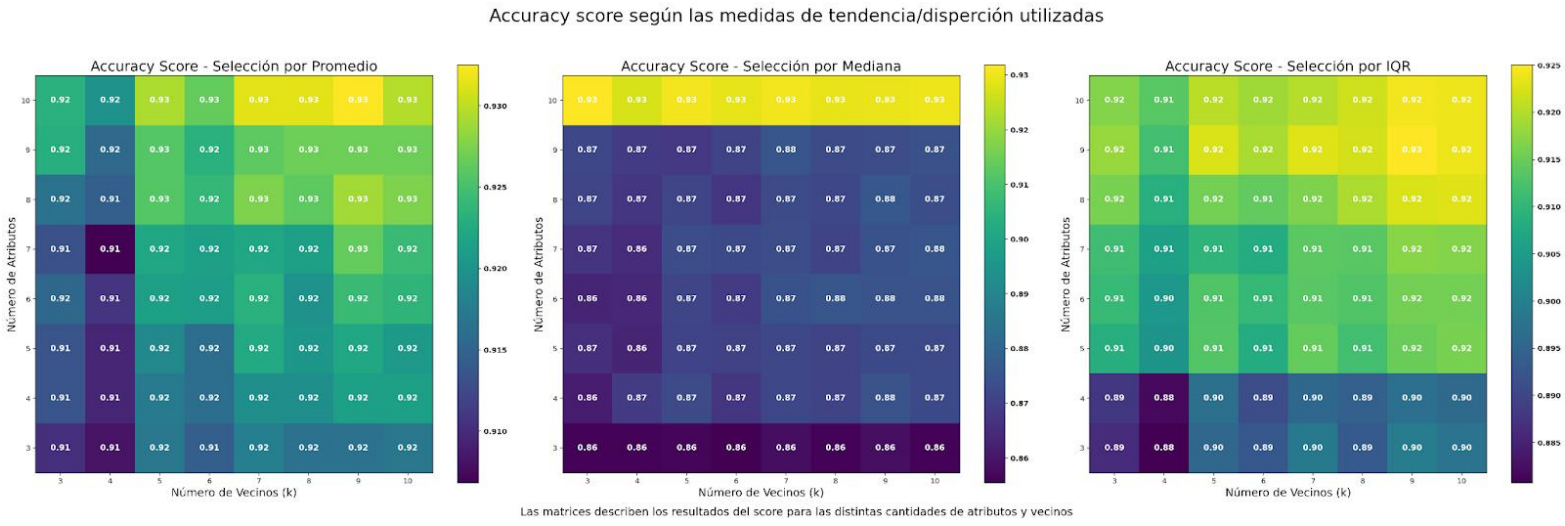


Figura 13 : Matrices según los resultados del accuracy score con distinta cantidad de vecinos y atributos para los distintos criterios de selección de píxeles

Clasificación multiclase

En esta sección, nos enfocaremos en la construcción de modelos para clasificación multiclase, en la que se busca predecir la categoría correspondiente a cada una de las distintas prendas incluídas en el conjunto de datos del dataset. Al contar con 7000 ejemplares por prenda, hay cantidades uniformes por clase a predecir.
Para el entrenamiento de los modelos y su posterior evaluación, separamos un *conjunto de desarrollo* y un *conjunto de evaluación*, siendo la proporción 90-10 respectivamente.
A partir del conjunto de entrenamiento, optamos por el desarrollo de modelos predictivos de árboles de decisión.
Los árboles de decisión son modelos de aprendizaje supervisado utilizado principalmente para clasificación, basados en el armado de una jerarquía de reglas a partir de disyunciones de conjunciones sobre valores de atributos.
Es por estas características que son un modelo altamente interpretable. Es decir que, dada una predicción particular, podemos entender por qué el modelo la generó. Sólo hay que mirar la rama de la hoja correspondiente a la predicción.
Los árboles presentan dos hiperparámetros de gran relevancia: profundidad y criterio. Nos interesa, a partir de diferentes combinaciones de hiperparámetros, generar distintos modelos y seleccionar finalmente aquel con el que se obtenga mejor métrica, la cual determinamos que sea el accuracy score.
Dada la naturaleza del problema (pasamos de una clasificación binaria a multiclase), hemos optado por la utilización de técnicas más sofisticadas de evaluación y selección de modelos, envolviendo métodos de validación cruzada (o cross-validation): KFolding y función GridSearchCV.
KFolding, o K-Fold cross-validation es un algoritmo el cual se basa en realizar una partición del conjunto en k-folds disjuntos del mismo tamaño (con, quizá, un previo 'desordenamiento' de los datos) en el que, para $i = 1, \dots, k$, se entrena el modelo usando todos los folds menos el número i, en el cual finalmente será evaluado/testeado el modelo. Una vez finalizado el proceso para cada fold (es decir, todo fold fue parte del conjunto de entrenamiento y de testeo), se realiza un promedio de las métricas obtenidas.
Por otra parte, GridSearchCV es una herramienta que permite encontrar la mejor combinación de hiperparámetros para un modelo, utilizando validación cruzada. Para ello, se le proporciona un conjunto de valores posibles para cada hiperparámetro. El algoritmo evalúa todas las combinaciones posibles de estos valores, y para cada una realiza una validación cruzada. Análogamente al proceso anterior, calcula el promedio de la métrica elegida (como el *accuracy*) en los distintos folds y selecciona la combinación de hiperparámetros que arroje el mejor rendimiento promedio.

En primer lugar, nos interesa ver la variación de las métricas de estimación de performance en base al aumento de la profundidad permitida al árbol. Para ello, hemos realizado un lineplot (figura XIV) en el que se puede observar el incremento de los valores arrojados por las medidas a partir del desarrollo de árboles más profundos, tomando como criterio 'Gini'.

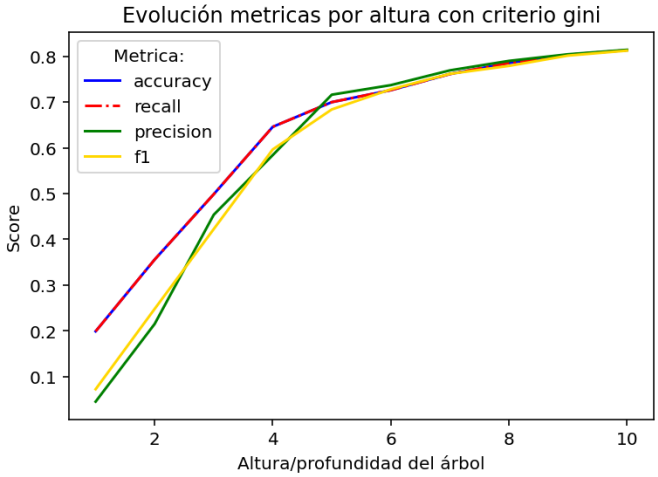


Figura 14 : Lineplot con la evolución de cada métrica en función de la altura

Por otra parte, aplicando KFolding con cuatro folds uniformemente balanceados (para asegurarnos que en cada una de las particiones haya la misma cantidad de prendas por clase), hemos podido observar la

evolución del nivel del accuracy score en base a las distintas alturas (entre 6 y 10) y criterios considerados (gini y entropía). Esta progresión la graficamos en la figura XV.

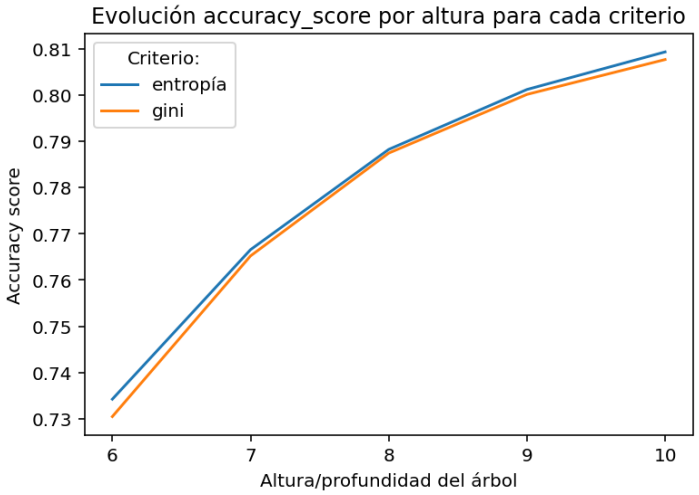
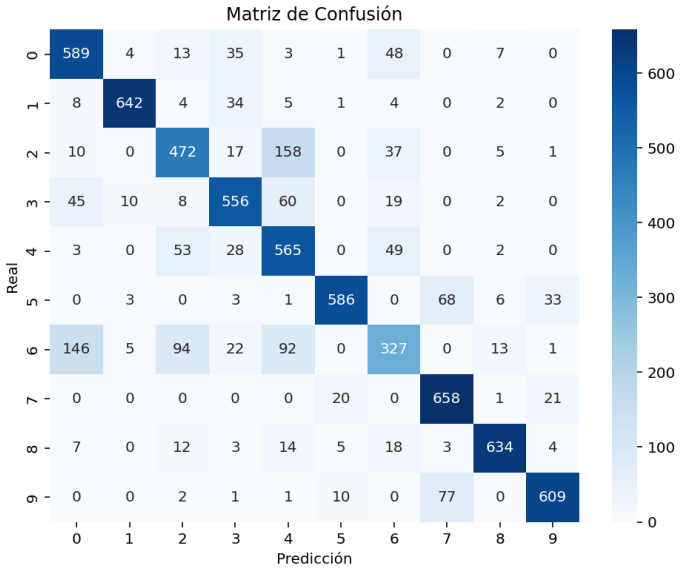


Figura 15 : Lineplot con la evolución del accuracy obtenido en función de las distintas alturas y criterios

Es entonces que podemos determinar que la mejor combinación/el mejor conjunto de hiperparámetros para el desarrollo del árbol que se destaque en la clasificación de las prendas del dataset (sin analizar alturas superiores a diez) es el árbol de profundidad 10 y criterio entropía.
Finalmente, utilizamos el conjunto de evaluación para reportar la performance que tiene nuestro modelo, en base a las distintas métricas utilizadas a lo largo del informe, y con el añadido de una matriz de confusión (figura XVI) en el que se logra observar los aciertos del modelo, así como los desaciertos (evidenciando, en este caso, la etiqueta incorrecta asignada):

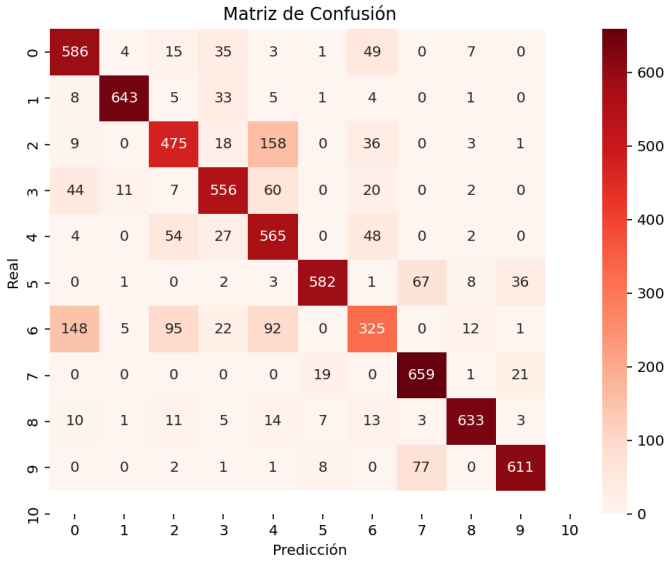


Resultados métricas:
Accuracy score = 0.8054285714285714
Recall score = 0.8054285714285714
Precision score = 0.8099199037564777
F1 score = 0.8036896521847856

Figura 16 : Matriz de confusión de la mejor combinación de hiperparámetros para el árbol entrenado con la totalidad de los píxeles

A partir del análisis realizado, propusimos, de forma complementaria, la construcción de un nuevo árbol de decisión. Esta iniciativa surge de la observación de que el empleo de la totalidad de atributos presentes en el dataset constituye un procedimiento computacionalmente costoso. Con el objetivo de agilizar y optimizar los procesos, decidimos reducir la dimensionalidad de los datos bajo ciertos criterios.
En primera instancia, recurrimos a la función tree.features_importantes para identificar cuáles son los píxeles considerados 'relevantes' por nuestro árbol de mejor rendimiento. Con base a esta información, procedimos a excluir aquellos atributos que no ejercen influencia en el proceso de clasificación (es decir, eliminamos los píxeles que el árbol no valora al momento de identificar las prendas).

Adicionalmente, descartamos los atributos previamente definidos como “inútiles” durante el análisis exploratorio de los datos; pues determinamos que la distinción entre clases no podía sostenerse razonablemente sobre tales características. Luego de esta depuración, donde se descartaron 383 píxeles de los 784 iniciales, se reestructuraron nuevamente los datos en conjuntos de desarrollo y de evaluación. Con el subconjunto resultante de atributos, desarrollamos lo que denominamos nuestro ‘*árbol alternativo/optimizado’*. Para ajustar sus parámetros, utilizamos la herramienta GridSearchCV, con el objetivo de hallar la combinación de hiperparámetros que arrojara el mejor valor de accuracy score. En particular, evaluamos profundidades máximas de entre 6 y 10 niveles, y los criterios Gini y Entropía.
El procedimiento retornó como mejor configuración la de un árbol de profundidad 10 y criterio entropía, replicando así el desempeño óptimo observado en la versión inicial.
Como cierre, evaluamos el rendimiento del nuevo árbol en el conjunto de evaluación reservado a tal fin, consiguiendo la matriz de confusión de la figura XVII, y los siguientes resultados en las métricas elegidas:



Resultados métricas:
Accuracy score = 0.805
Recall score = 0.805
Precision score = 0.8097803743104336
F1 score = 0.8032526121679229

Figura 17 : Matriz de confusión de la mejor combinación de hiperparámetros del árbol entrenado con menos atributos

Conclusiones

Habiéndose realizado todos los procesos de exploración de los datos, análisis de las distintas clases del dataset, identificación de patrones de comportamiento de cada una de las prendas del mismo; y tras haber desarrollado distintos modelos predictivos para la identificación y clasificación de las distintas clases, basándonos en algoritmos como KNN y árboles de decisión, hemos podido evaluar la performance y examinar con detalle las diferencias entre las distintas estrategias implementadas. Asimismo, apreciamos los grandes beneficios de un análisis exploratorio exhaustivo para la obtención de aquellas características particulares de cada elemento del dataset, que permiten, en la mayoría de los casos, su correcta identificación.
Esto resultó en niveles de rendimiento muy altos, donde la tasa en la clasificación binaria fue superior al 90% mientras que la clasificación multiclase consiguió un porcentaje mayor al 80%.

En suma a los altos números presentados, fuimos capaces de la construcción de modelos optimizados de efectividad similar pero con grandes diferencias de costo computacional que se traduce en menor cantidad de tiempo de espera, agilización por parte del árbol en la selección de atributos relevantes y mayor interpretabilidad de los resultados que retorna los modelos KNN al contar con cierto grado de comprensión en torno a la fundamentación de las clasificaciones efectuadas.

En conjunto, esto nos indica que, teniendo conciencia respecto al objetivo que se desea cumplir, y llevando a cabo mecanismos integrales de investigación, procesamiento, filtrado y organización de datos a emplear; es posible la construcción modelos predictivos capaces de cumplir en gran medida el cometido inicial.