CLASIFICACIÓN Y SELECCIÓN DE MODELOS

LABORATORIO DE DATOS 2025

DOROGOV, PUCCIARELLI, SALTO

UNIVERSIDAD DE BUENOS AIRES

Resumen

En este trabajo se propone la construcción y comparación de distintos tipos de modelos de clasificación. Para ello, se trabajó sobre el DataSet **Fashion-MNIST**[1]. El objetivo perseguido es entrenar modelos de clasificación KNN y árboles de decisión, realizar comparativas exhaustivas de la performance de ambos tipos de modelos, y concluir con aquel que mejor logre clasificar imágenes de prendas de vestir. Para esto, se realizará una limpieza y filtrado del DataSet original, buscando captar aquellos píxeles que mejor describen a cada tipo de prenda. Los análisis que se implementarán tras esto serán conocer la mejor profundidad de árbol para una correcta generalización de resultados y conocer qué píxeles son relevantes para diferenciar a dos de las clases elegidas, 0 y 8.

Introducción

Este trabajo pretende analizar distintos modelos de clasificación y encontrar el más adecuado que mejor generalice. A lo largo de la sección de *Análisis Exploratorio*, se buscó realizar un filtrado de píxeles con el objetivo de reducir el volumen de datos con el que posteriormente se entrenarán los modelos. Presenta distintas métricas de promedio y varianza, trabajando sobre el conjunto completo de datos, y sobre cada clase (en particular); con el fin de exponer estos píxeles de interés. En la sección *Clasificación Binaria,* se trabajó con modelos de clasificación KNN. El análisis se centró en dos clases de prendas particulares: la clase 0 y la clase 8. Se buscó encontrar el mejor número de *k-vecinos* para poder generalizar resultados mediante el promedio de distintas métricas. Por su parte, la sección *Clasificación MultiClase* contempla todo el desarrollo realizado sobre el tipo de modelo Árbol de Decisión mediante el uso de *k-fold cross validation*. Aquí se trabaja con el conjunto de clases completo, sutilmente modificado para el análisis. Se determinan los *hiper-parámetros* que mejor adaptan el modelo a los datos brindados y se hacen los testeos respectivos. Por último, la sección *Conclusiones* contiene las conclusiones del trabajo realizado. En la misma se encuentra que para pocas clases y muy diferentes entre sí, es conveniente usar el KNN mientras que para mayor caudal de información y clases similares es mejor utilizar el árbol de decisión. En anexo se pueden contemplar algunas de las imágenes de las prendas que representan a cada una de las clases.

Análisis Exploratorio

Para empezar, se realizó una exploración general y básica del DataSet, con el objetivo de entender qué información contenía. Se trata de imágenes de 28x28 píxeles donde cada columna es el valor del color que debe tener cada píxel, hay 28 x 28 = 784 columnas en total de píxeles y una de la clase a la que pertenece cada imágen. Además, los valores van entre 0 y 255. Hay 70.000 filas; es decir, 70.000 imágenes en total. La última columna, denominada '*label*', es el tipo de prenda al que representa esa imágen. Esa etiqueta puede tomar valores del 0 a 9 donde cada uno de ellos representa un tipo de prenda diferente. El análisis se centrará en buscar patrones que permitan identificar qué píxeles tienen información importante y cuáles no.

La primera idea es ver si ciertos píxeles tienen valores distintivos en cada tipo de prenda (para cada clasificación). Para empezar, se agrupó la información por Clase y se calculó el valor promedio de cada píxel dentro de cada clase. Esto último con el objetivo de visualizar cuál es la imágen promedio que se esconde detrás de cada clase. A partir de estas imágenes[2], se establece una mejor clasificación para los distintos tipos de prendas. Clase 0: prenda superior con mangas cortas; Clase 1: prenda inferior (pantalón largo); Clase 2: prenda superior con mangas largas; Clase 3: prenda completa (vestido largo, mangas cortas), Clase 4 : prenda superior con mangas largas; Clase 5: calzado; Clase 6: prenda superior con mangas largas; Clase 7:

[1]obtenido de https://github.com/zalandoresearch/fashion-mnist
[2] Ver *Figura 5* a *Figura 14* del Anexo.

calzado; Clase 8: accesorio (tipo bolso); Clase 9: calzado. A partir de esto, se espera que cada Clase va a representar variantes del tipo de prenda que refleja su *imágen promedio*.

Por otro lado, se observa que hay clases muy parecidas entre sí. Las Clases 2, 4 y 6 parecen representar variantes de la misma pieza: prendas de mangas largas. Es difícil diferenciar si se trata de remeras, abrigos o camisas; también si se trata de prendas masculinas o femeninas. Las Clases 5, 7 y 9 queda claro que corresponden a calzados. Mas no se considera que sean lo suficientemente claras como para diferenciar tipos de calzados.

En cuanto a los píxeles con información importante, se empieza trabajando sobre la idea de que los píxeles de los bordes se mantienen siempre en *negro*, y que los píxeles del centro (de cada imagen) se mantienen siempre en *blanco*. Primero se buscarán aquellos píxeles que no aportan información tomando todas las clases en conjunto (más adelante, se trabajará sobre aquellos píxeles de mayor información Clase por Clase). Para ello, se propone el siguiente paso: Calcular la varianza de cada píxel en todo el DataSet. Esto puede dar una mejor idea de cuáles son los píxeles que no aportan información. Ya que puede diferenciarse más claramente cuáles son *casi siempre negro*, cuáles son *casi siempre blancos* y cuáles se mueven en un rango de grises. A este punto, el análisis se centra en píxeles que se tienden a mantener en blanco y negro los cuales se considera que no aportan información alguna para diferenciar clases y son los que se decidió descartar.

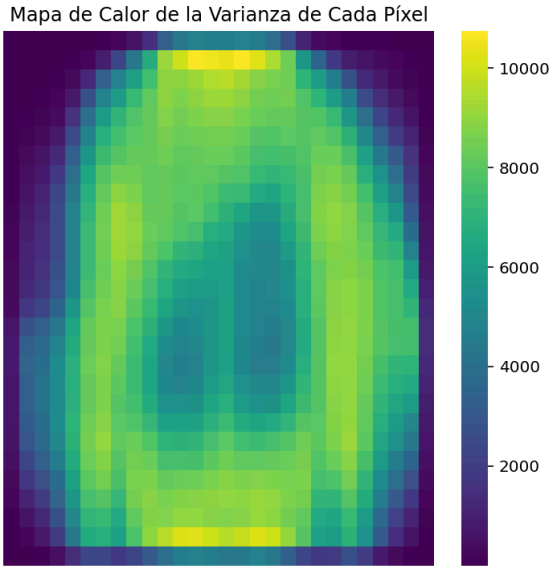

Figura 1: Mapa de la Varianza de cada Píxel.

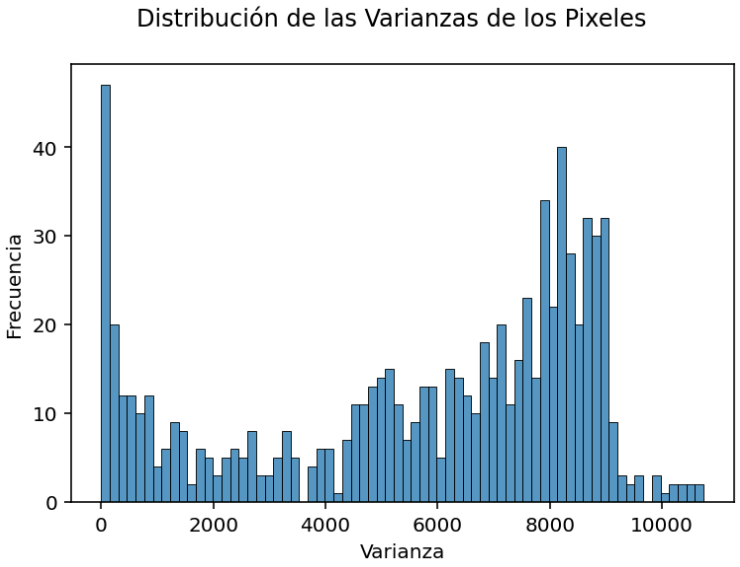

Figura 2: Histograma de la Varianza de cada Píxel.

A partir de la *Figura 2,* se puede visualizar mejor cómo se distribuye la varianza a lo largo de todos los píxeles. Es fácil notar que existen muchos con varianza pequeña. Esto se corresponde con que el primer *bin* más cercano a cero, es el que mayor altura tiene; siendo el mismo el que almacena a aquellos que tienen varianza cercana a cero. Allí se encuentran los píxeles que siempre se mantienen en blanco o negro. Además se observa que en el extremo derecho del histograma, también hay una buena cantidad de datos. Aquellos *bins* mayores valores del eje horizontal, se corresponden con las varianzas de valores altos. Se observa que hay muchos *bins* con alturas considerables, lo que significa que se cuenta con una buena cantidad de píxeles con varianza muy alta. Estos son los datos que se buscarán priorizar para entrenar los modelos, ya que se considera que son los que van a marcar la diferencia a la hora de entrenar y mejorar la tasa de acierto en las predicciones.

La *Figura 1* muestra las zonas donde hay menor variación de píxeles dando así una idea de cuáles son los píxeles a descartar. Las zonas bordes y centrales mantienen colores oscuros lo cual indica que tienen

varianza pequeña, candidatos a ser descartados. Aquellas zonas con colores más claros, contienen a los píxeles de mayor varianza. Estos son los que mejor caracterizan los cambios de Clases. Esta figura permite definir un *umbral* en 8900 de varianza mínima, el cual busca todos los píxeles que tienen varianza mayor al mismo. Este valor parece incluir a los píxeles que pertenecen a los bordes (de las imágenes) y al centro.

Al trabajar clase por clase, resulta ser que los píxeles más representativos de cada una son aquellos que menor varianza tienen. Para explicar mejor la idea, si hay 10 remeras mangas cortas, cada una con un estampado diferente se calcula la varianza a los píxeles de esa clase: los píxeles con mayor varianza van a ser aquellos donde se encuentra la estampa de cada remera, los píxeles con menor varianza van a ser aquellos que moldean la remera. De esa forma, es fácil notar que aquellos píxeles que representan y mejor caracterizan a las remeras mangas cortas, son aquellos que menos cambian entre una imagen y otra (los más representativos). Generalizando la idea para todas las clases, se estaría trabajando con los píxeles que mejor caracterizan la figura dentro de cada una de ellas. Se definieron dos clasificaciones:

Varianza Intra-Clase : ¿Cuánto varía un píxel dentro de una clase?

- Permite identificar píxeles estables para una clase dada.

Varianza Inter-Clase : ¿Cuánto varía el valor promedio de un píxel entre las diferentes clases?

- Permite encontrar píxeles que separan bien las clases: píxeles donde el valor medio cambia significativamente de una clase a otra.

Se considera un *buen píxel predictivo* a aquel que tiene baja Varianza Intra-Clase, o sea siendo estable dentro de cada clase, y tiene alta Varianza Inter-Clase, la cual permite distinguir una clase de otra.

Para ambas clases, se calcula el promedio de varianza de cada píxel en cada clase.

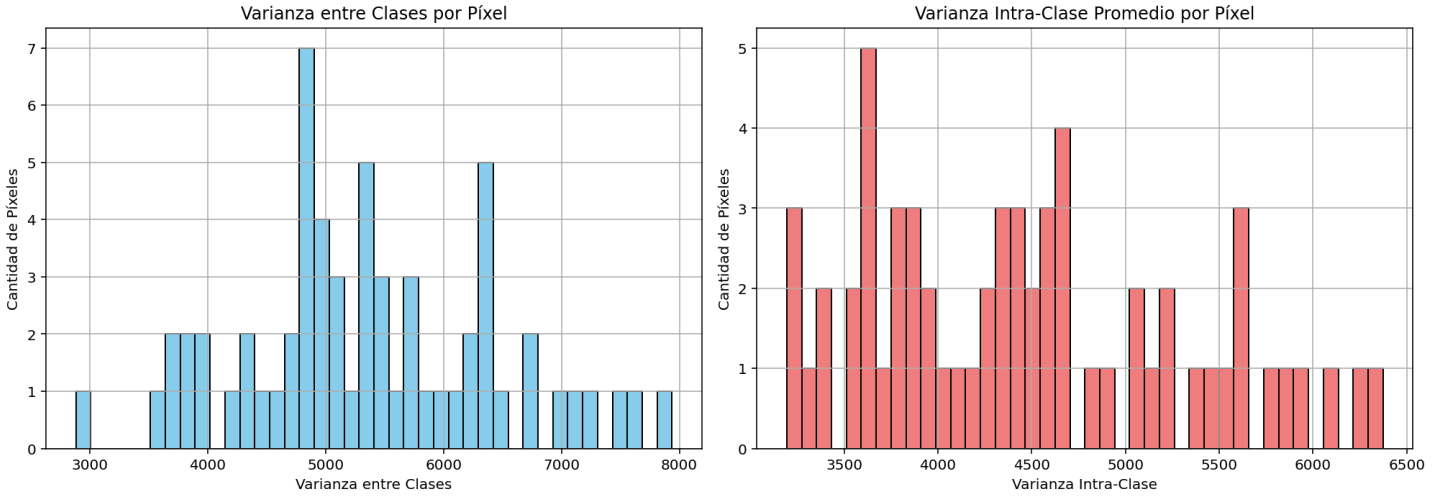

Figura 3: A la izquierda (en azul) el Histograma correspondiente a la *Varianza Inter-Clase.* A la derecha (en rojo) el Histograma correspondiente a la *Varianza Intra-Clase.*

A partir de la *Figura 3*, se establecen nuevos *umbrales*:

- *Umbral Entre Clases* : 4000
- *Umbral Intra Clases* : 4400

A continuación, se filtra aquellos píxeles que tienen *Varianza Inter-Clases* superior al *Umbral Entre Clases* y *Varianza Intra-Clases* menor al *Umbral Intra Clases*.

Llegado este punto, se buscó mejorar un poco más el conjunto de píxeles a considerar. Para ello se propuso aumentar las diferencias entre los píxeles que componen a ciertas clases en específico, en particular a las clases cuyas imágenes son muy similares. Para ello, se calculó los valores promedio de los píxeles para

cada una de esas clases, se sumó las diferencias (en valor absoluto) entre cada par de píxeles y luego se capturó los de mayor distancia (mayor diferencia entre las Clases). Una vez obtenidos los píxeles resultantes de este procedimiento, se añaden al conjunto de píxeles obtenido en el paso anterior[3]. Este procedimiento se desarrolló sobre los conjuntos de clases: 2, 4 y 6; 5 y 7; 0 y 6.

Este análisis exploratorio finaliza con un nuevo dataset que cuenta con 63 píxeles por imágen logrando reducir el volumen de manera significativa respecto a los datos originales, y quedando aquellos que mayor información aportan al fin de entrenar modelos de clasificación.

Clasificación Binaria

Se trabajó únicamente con las clases 0 y 8. Para ello, se filtró la información, la cual previamente atravesó un proceso de limpieza, mediante una consulta SQL en donde se observó que hay 13999 imágenes pertenecientes a esas clases. De ellas, se buscó obtener los píxeles que mejor las diferencian mediante el cálculo de la diferencia absoluta entre las medias de ambas clases y se seleccionaron los 64 píxeles más representativos. Se entrenó un primer modelo con esos atributos, en donde se utilizó 70% de los datos para training y 30% para testing. Este modelo dio una exactitud de 0.97. Luego se realizó el ajuste para el cual se armaron tres subconjuntos de 4 atributos[4] cada uno. Estos 12 atributos en total son los que tienen la mayor diferencia de media obtenida. Los subconjuntos elegidos fueron: ["pixel554", "pixel538", "pixel582", "pixel510"], ["pixel470", "pixel566", "pixel482", "pixel442"] y ["pixel554", "pixel538", "pixel582", "pixel510"]. Otros hiper-parámetros decididos fueron el número de k vecinos y la cantidad de repeticiones a realizar en el split. El k se tomó de 1 a 19 mientras que la cantidad de repeticiones realizadas fue 5.

Tras ejecutar el split con los distintos subconjuntos de atributos se observa que la exactitud del modelo en abstracto es 0.97[5], la cual es la misma en los tres casos. Esto tiene sentido ya que a las clases 0 y 8 se les hizo un trabajo adicional de conocer cuáles píxeles las distinguen mejor y utilizar aquellos seleccionados. La exactitud con atributos reducidos varía levemente, siendo en el primer subconjunto del 0.90, en la segunda elección 0.89 y la última del 0.92. Esta sutil variación permite apreciar que el modelo se mantiene estable para los distintos atributos y que los mismos elegidos para entrenarlo son realmente atributos relevantes para la clasificación. Se decide mostrar a continuación el gráfico correspondiente al subconjunto tres dado que es el que presentó mayor exactitud al reducir atributos. Sin embargo, los tres gráficos son bastante similares.

[3] Se tuvo en cuenta eliminar las ocurrencias repetidas dado que ya podrían estar incluidas en el paso anterior.
[4] Todos los atributos de los subconjuntos provienen de los 12 seleccionados previamente.
[5] Si bien los primeros tres dígitos son iguales, obviamente hay una sútil varianza en los siguientes restantes.

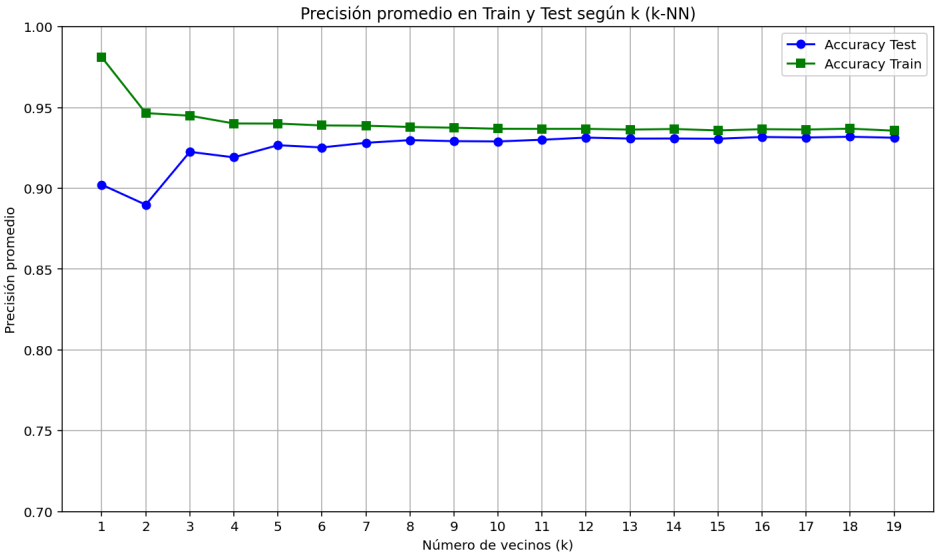

Figura 4: Muestra la precisión promedio del tercer subconjunto para distintas cantidades de vecinos

Clasificación Multiclase

Como primer paso, se decidió reservar el diez porciento de la información, previamente ya filtrada, para el holdout, y utilizar el noventa restante para el desarrollo. Mediante la técnica de k-fold cross validation se entrena el árbol con cinco folds y cinco distintas profundidades siendo las mismas 1, 2, 5, 7 y 10 y se obtiene como resultado el promedio de exactitud para cada uno de ellos. En el orden estipulado, los mismos fueron: 0.1957, 0.3218, 0.6911, 0.7396 y 0.7741. Se nota que a mayor profundidad, mayor es la exactitud del árbol. Por eso, se decide entrenar un modelo con profundidad diez. En esta etapa se realizaron dos evaluaciones, una sobre el conjunto de desarrollo y la segunda, con el conjunto previamente seleccionado para holdout. La primera se tuvo en cuenta información[6] como la exactitud: 0.80, la precisión: 0.81, el recall: 0.80 y la F1: 0.80. Se observa que dieron resultados muy similares y por eso, para evaluar con el conjunto holdout, se decidió únicamente calcular cuán exactos son los resultados. El valor obtenido fue de 0.77, el cual es bastante similar con los resultados previamente obtenidos.

Por último, para tener un panorama más completo del árbol se decidió realizar un último testeo respecto al dataset original. Los resultados alcanzados son muy similares a los mencionados previamente, exactitud: 0.80, precisión: 0.81, recall y F1 : 0.80. Esto permite apreciar que el árbol generaliza bien para mayor cantidad de casos y es estable. Si los resultados fueran muy distintos existiría el riesgo de sobreajuste o subajuste.

Conclusiones

Al principio el trabajo realizado se planteó realizar la comparación de distintos modelos de clasificación. Para la misma fue importante haber realizado un filtrado de píxeles comunes a todas las clases dado que no sería de utilidad entrenar árboles y knns con ellos, no hubiesen aportado información alguna a las predicciones. En lo que respecta al knn, se puede concluir que el modelo entrenado, para dos clases distintas no similares entre sí, tiene una gran exactitud, 97 %. No se cree que para dos clases similares, tales como la 4 y 6, el modelo funcione ya que las prendas pertenecientes a esas clases son muy parecidas, de tipo

[6] Se puede observar la matriz de confusión en el código brindado.

superior de mangas largas. Para el árbol de decisión, se utilizaron todas las clases existentes en la fuente de datos y tuvo una exactitud del 80% para el testeo con el dataset completo. A simple vista, parece ser que el modelo de KNN es el mejor ya que uno quiere tener la mejor exactitud posible. Sin embargo, cabe recordar que KNN es un modelo que funciona mejor para cantidad pequeña de datos y compara imágen a imágen. Si uno desea poder distinguir a qué clase corresponde cada prenda de vestir que hay en el archivo, la mejor elección sería elegir el árbol de decisión ya que para mayor volumen de información es el adecuado. Además a cada decisión tomada se le puede hacer un seguimiento y corregirla en caso equivocado. Si se trabajase todo el dataset en KNN, esto sería imposible de realizar. Entonces, para clases de gran diferencia y poco caudal de información, es preferible usar knn, para mayor volumen de información y clases similares, es mejor utilizar el árbol de decisión.

Anexo

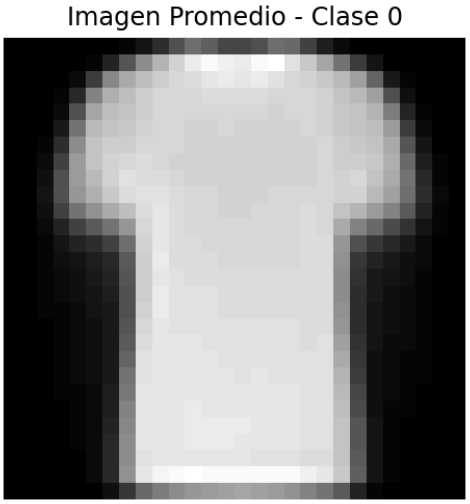

Figura 5: Gráfico de los Píxeles Promedio de la Clase 0.

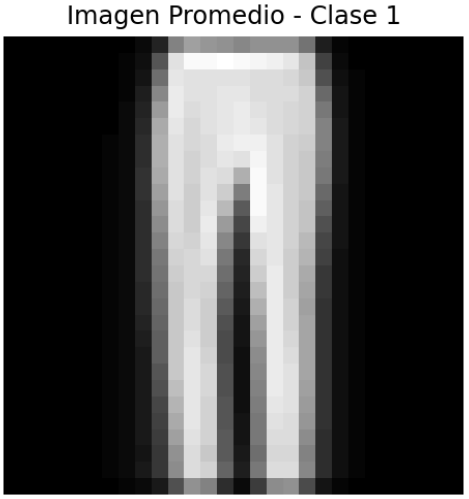

Figura 6: Gráfico de los Píxeles Promedio de la Clase 1.

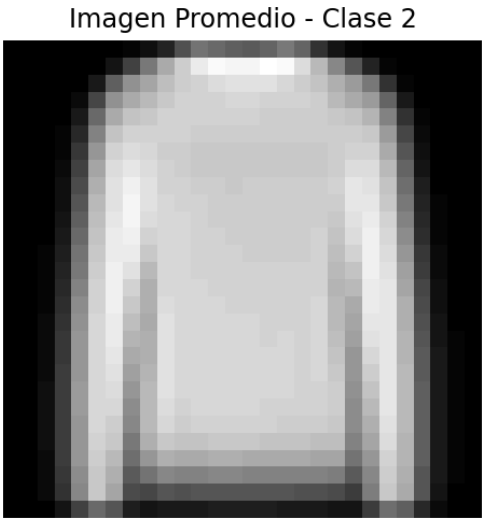

Figura 7: Gráfico de los Píxeles Promedio de la Clase 2.

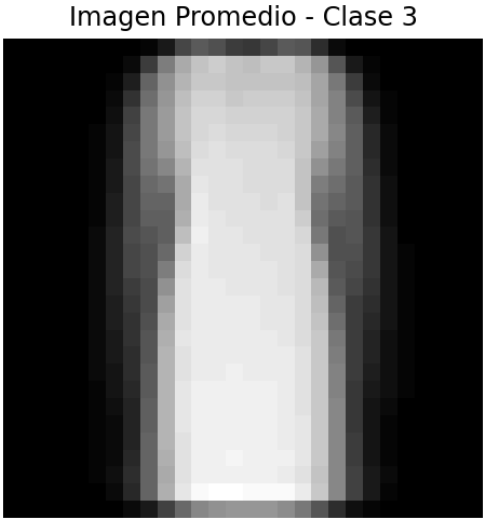

Figura 8: Gráfico de los Píxeles Promedio de la Clase 3.

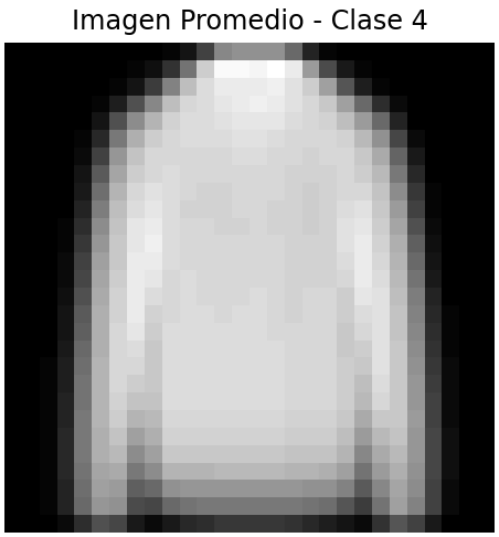

Figura 9: Gráfico de los Píxeles Promedio de la Clase 4.

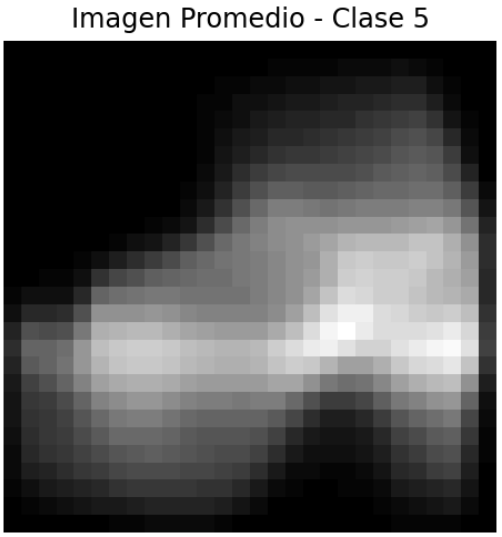

Figura 10: Gráfico de los Píxeles Promedio de la Clase 5.

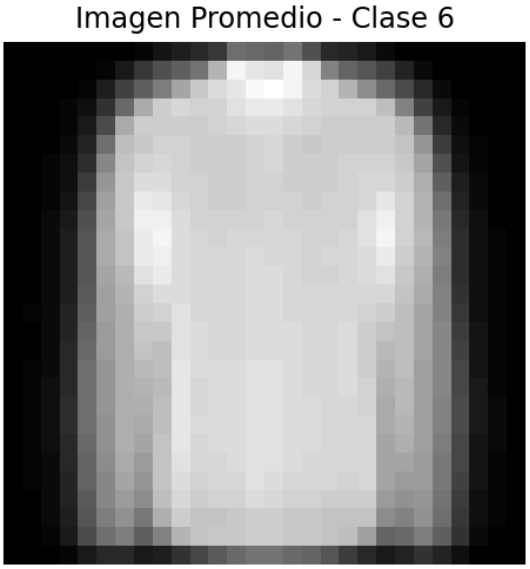

Figura 11: Gráfico de los Píxeles Promedio de la Clase 6.

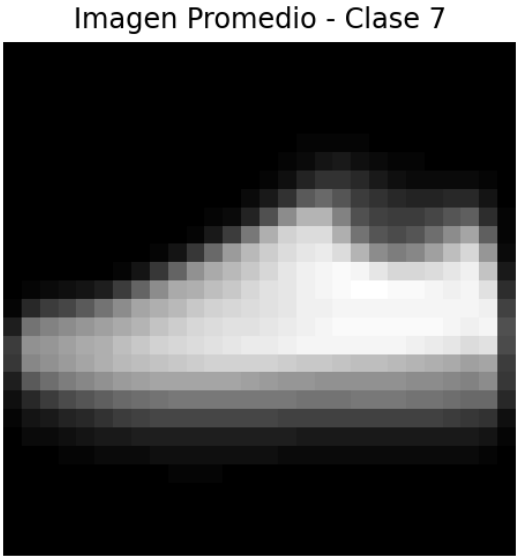

Figura 12: Gráfico de los Píxeles Promedio de la Clase 7.

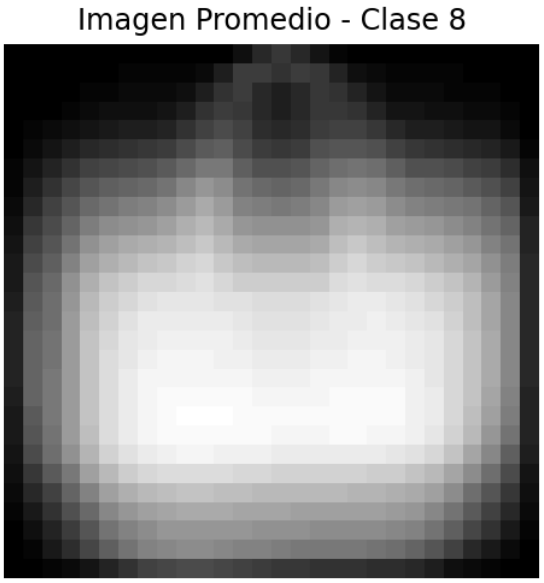

Figura 13: Gráfico de los Píxeles Promedio de la Clase 8.

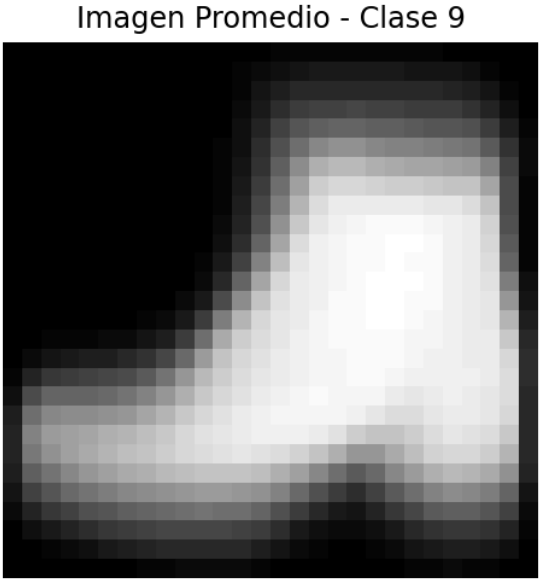

Figura 14: Gráfico de los Píxeles Promedio de la Clase 9.