Trabajo Práctico II:

Fashion

INFORME

**Laboratorio De Datos (DC), Comisión Laje**

**2025**

Autores:

* **Cuestas, Martín Iván**
* **Nakasone, Julián**
* **Poli, Dante**



# Introducción y Análisis Exploratorio:

En el presente trabajo se tiene como objeto de análisis un dataset de imágenes de prendas de ropa de dimensión 28 x 28 llamado “Fashion-MNIST”. El objetivo será aplicar modelos de clasificación, como K-Nearest Neighbors (KNN) y Árboles de decisión, para predecir la clase correspondiente a una prenda a partir de sus píxeles.

Para ello, se realiza un análisis exploratorio que permite comprender la estructura y comportamiento de los datos, identificar atributos relevantes y evaluar su impacto en la clasificación. Posteriormente, se entrenan distintos modelos y se comparan sus rendimientos bajo diversas configuraciones. Finalmente, se propone una solución óptima en base a los resultados obtenidos.

Estructura de Fashion-MNIST:

El conjunto de datos está compuesto por 70.000 instancias, balanceadas uniformemente entre 10 clases (7000 imágenes por clase), donde cada imagen está representada por 784 atributos correspondientes a los valores de los píxeles en escala de grises de 0 a 255. Esto más un atributo 785 correspondiente a una etiqueta con el tipo de prenda, codificada como número entero del 0 al 9, clasificado de la siguiente forma:

0: T-shirt/top - Remera/top

1: Trouser - Pantalón

2: Pullover - Pulover

3: Dress - Vestido

4: Coat - Abrigo

5: Sandal - Sandalia

6: Shirt - Camisa

7: Sneaker - Zapatilla

8: Bag - Bolso

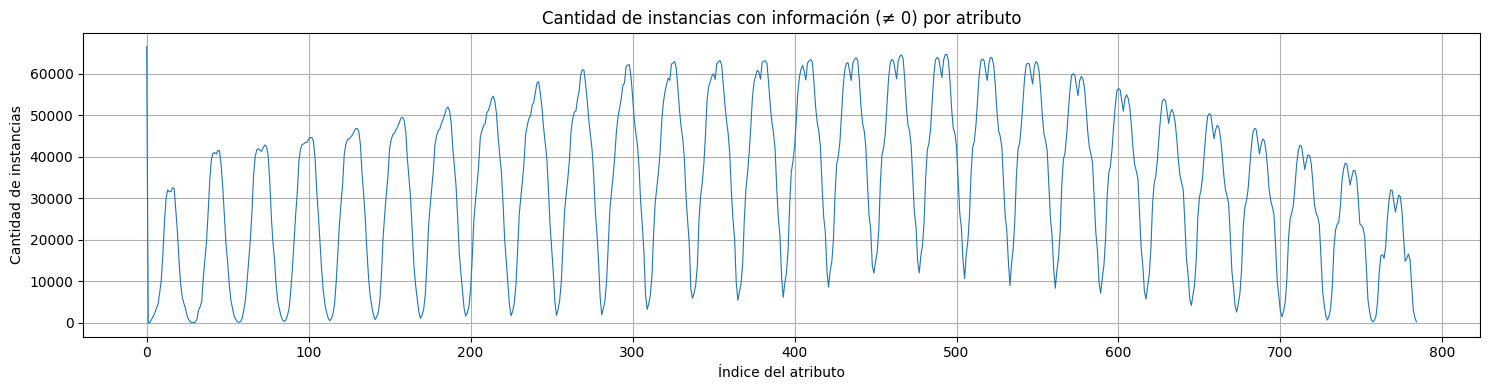
9: Ankle Boot - Botín

(Se utilizarán los nombres en inglés)

El primer paso en la distinción de etiquetas o clasificaciones en base a atributos consta de estudiar mediante distintos enfoques el comportamiento de los mismos; esta idea se fundamenta en encontrar patrones y relaciones en cuanto a cómo se distribuye la información en el conjunto de píxeles, y cómo dichos comportamientos influyen e incluso determinan la etiqueta asignada “label”. Traducido a algo más concreto y visual, que valores deben tomar “en general” los distintos píxeles para generar la imagen de una determinada prenda.

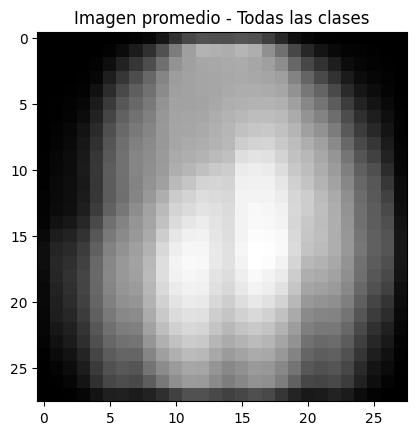
Se buscaron cuáles atributos son los que menos información aportan para la clasificación, es decir, aquellos que permanecen sin variaciones a lo largo de todas sus instancias, para toda prenda.

Comenzando con la búsqueda de aquellos píxeles cuyo valor es siempre 0, debido a que estas serían las zonas vacías que rodean al objeto de cada imagen, se realiza un gráfico que cuenta por cada atributo cuantas instancias tienen un valor mayor a 0.



*Plot Cantidad de instancias con información no vacía, hecho con Seaborn - Figura (I)*

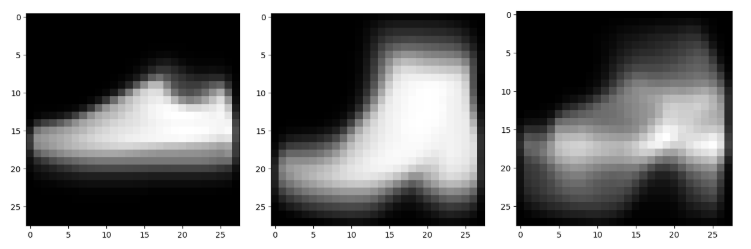
Se puede observar que las zonas de la imagen en donde el color es siempre negro, es decir vacío, se concentran principalmente en los atributos que van del 0 al 300, donde la figura azul comienza a ascender, y del 700 al 800, donde vuelve a descender. Pero esto no sucede constantemente dentro de cada intervalo, si no de a periodos, fenómeno reflejado por las oscilaciones en la figura. Esto se explica debido a que intuitivamente, las zonas vacías representan los bordes, ya que la prenda se encuentra centrada en el recuadro de la imagen, y al ser dicha imagen de 28x28, y estar desglosada horizontalmente en cada instancia, las regiones borde estarán dadas por los atributos que se acerquen a: los primeros 28 atributos para el borde superior, aquellos atributos que sean múltiplos de 28 para los bordes derecho e izquierdo, y los últimos 28 para el borde inferior. Esto se puede ver en la superposición de todas las imágenes (instancias) del dataset, donde las zonas negras yacen en los bordes, particularmente en las esquinas (*Figura (II)*).

**

*Superposición de las imágenes de todas las clases - Figura (II)*

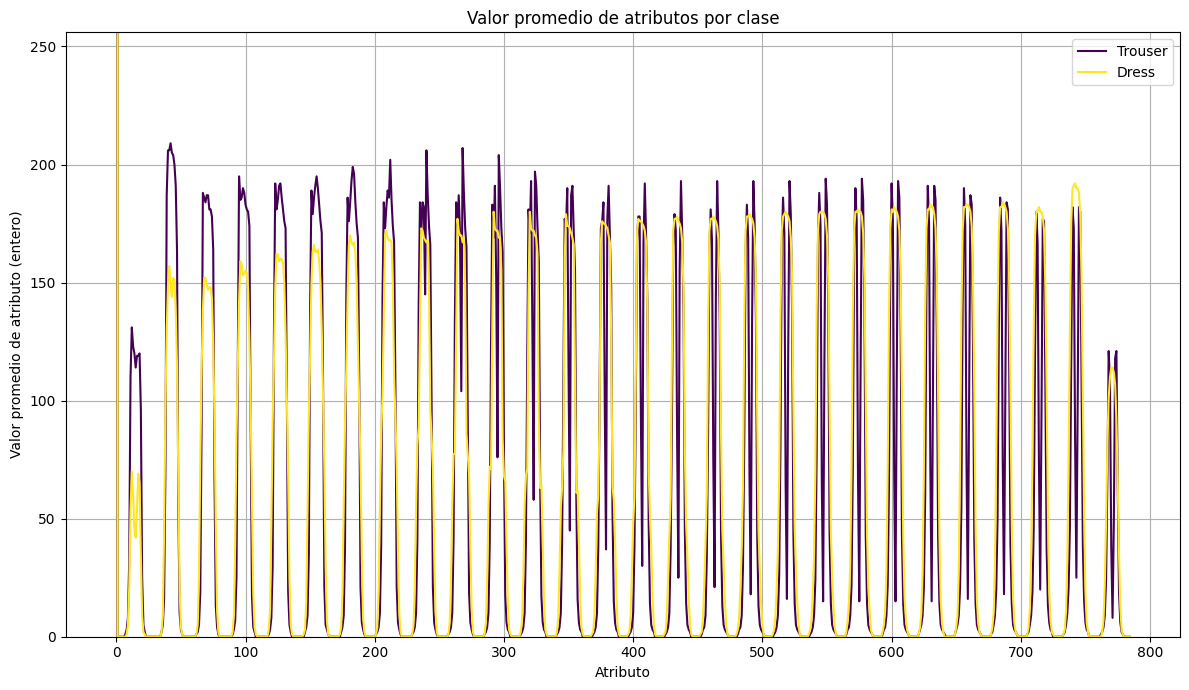
A pesar de esto, en la zona media derecha del gráfico (*figura (I)*) se ve un ascenso en los valores de las oscilaciones, lo que indica que para los atributos pertenecientes a esta zona, existen menos instancias con valor 0, incluso en los bordes. Este comportamiento parece deberse a prendas como las sandalias, las zapatillas y los botines debido a que son más alargadas horizontalmente (*figura (III)*), y tiene sentido a su vez que este comportamiento suceda en la zona media derecha del gráfico, debido a que es la parte que reflejaría la zona media baja de la imagen, aquella donde se encuentran estas prendas.

Por otra parte, los atributos cuyo índice se encuentra entre medio de múltiplos de 28 (14, 42, entre otros) parecen ser los que más información otorgan, o al menos, los que poseen pigmentación y dibujan algún objeto.

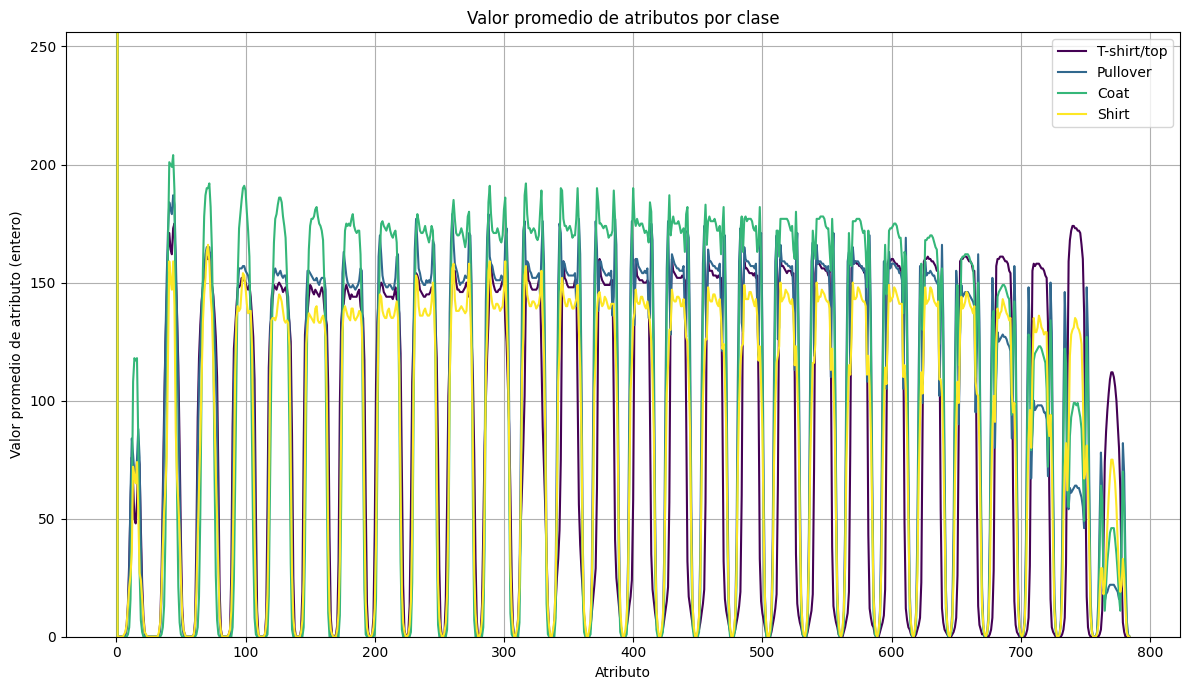


*Superposición de todas las imágenes de: Sneaker, Ankle Boot, Sandal - Figura (III)*

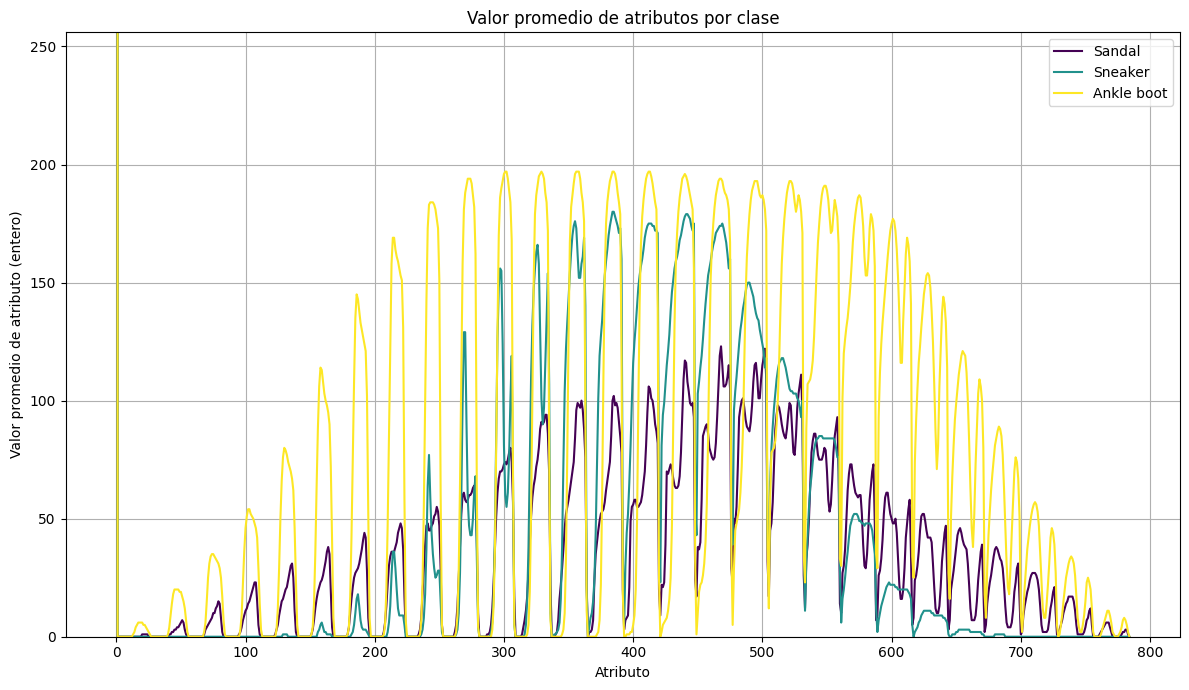
Entre todas las prendas, a su vez, existen algunas que son más parecidas entre sí, y otras que poseen un mayor contraste; donde intuitivamente, las prendas de categoría “similar” tienen mayor semejanza. Esto es algo que se puede ver reflejado en el valor de sus atributos, por esto, se realizan gráficos de líneas que muestran el valor promedio de atributo por clase, agrupados por prendas de tipo “similar” para observar la semejanza de comportamiento en sus patrones.



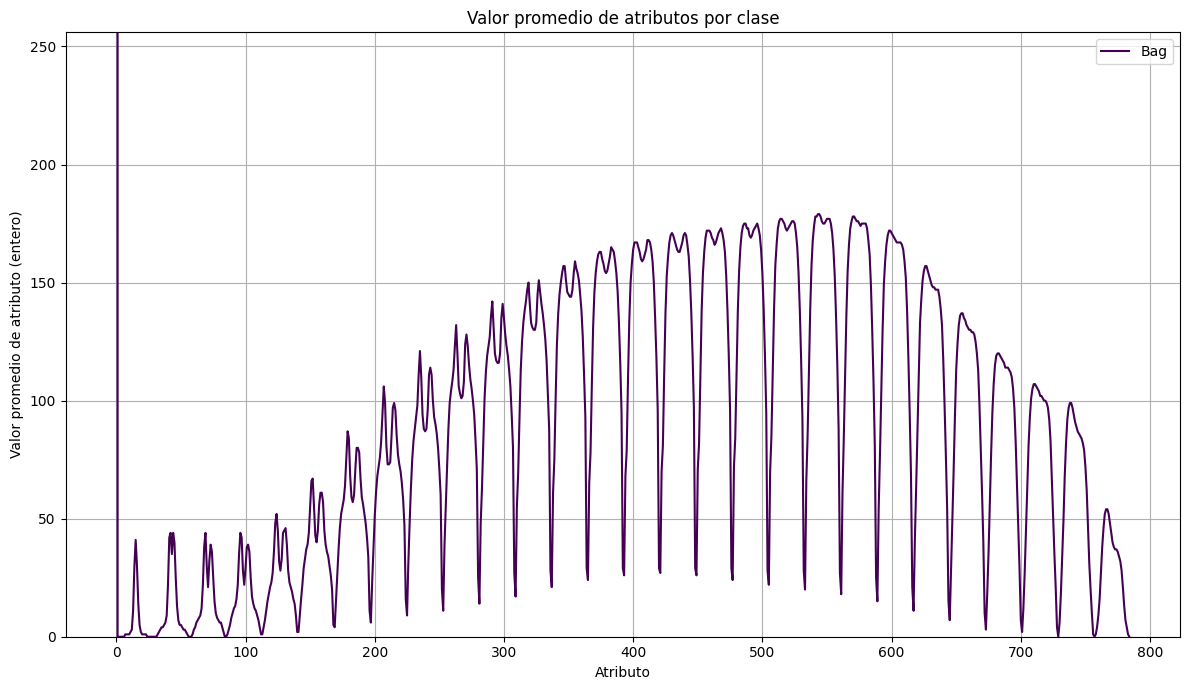
*Comparación Trouser / Dress - Figura (IV)*



*Comparación T-shirt/top / Pullover / Coat / Shirt - Figura (V)*

**

*Comparación Sandal, Sneaker, Ankle Boot - Figura (VI)*

**

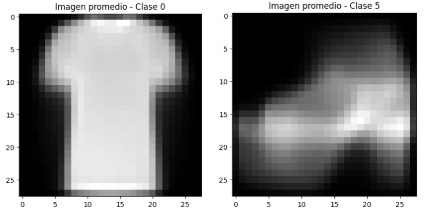
*Bag - Figura (VII)*

Se pueden observar notables comportamientos particulares para cada grupo de prendas. Para trousers y Dress, se ve un patrón rectangular aplanado que abarca desde el extremo derecho al izquierdo, muy similar al de T Shirt, Top, Pullover, Coat y Shirt. Esto indica que una gran parte del espacio de la imagen visto desde el borde superior hacia el inferior está “coloreado” por pixeles con valor distinto de 0, es decir, que el objeto abarca la mayor parte del recuadro desde el borde inferior hacia el superior. Sin embargo, en el primer grupo los picos se ven más delgados, lo que indica un mayor grosor en los espacios vacíos de los bordes, es decir pixeles con valor 0. Esto tiene sentido ya que al tratarse de pantalones y vestidos, la forma en la imagen generada va a ser más rectangular y angosta que en el caso de una remera o similar, cuyas mangas se extienden hacia ambos lados. También es notable que Coat es el que tiene valores promedio mayores, por lo que tendrá una pigmentación más clara en general.

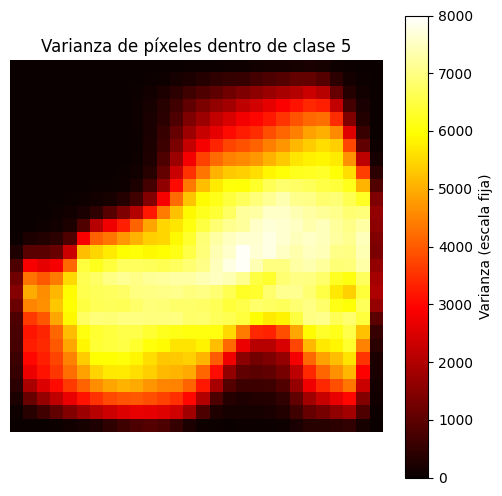
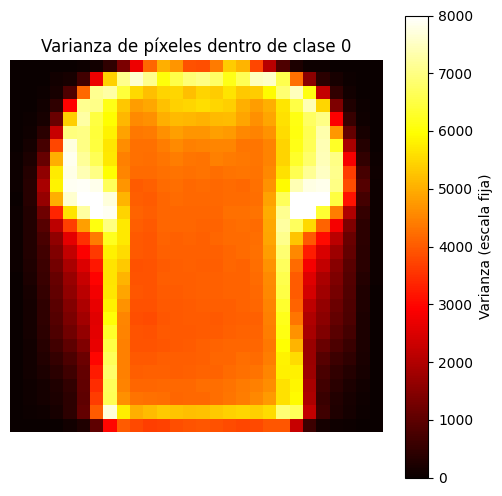
En el caso de los calzados, Sandal, Sneaker y Ankle Boot, se puede ver una concentración de valores altos en el medio, y a los lados valores bajos. Esto sucede ya que estas prendas se encuentran en la zona media de la imagen, dispuestas horizontalmente, por lo que en la parte superior e inferior el valor de los píxeles generalmente será nulo o en promedio bajo. También se ve que los picos son anchos, lo que indica que los bordes laterales ocupan poco espacio, debido a la forma alargada propia del calzado. Comparando ahora entre cada prenda dentro de esta clase, se puede notar que Ankle Boot es la que que posee la figura que más se extiende a ambos lados, lo que da a entender que su imagen se extiende más hacia arriba y hacia abajo en el recuadro de pixeles; mientras que Sneaker es la que posee la figura más angosta, pasando así lo contrario en la imagen. Sandals por otro lado se extiende de igual forma que Ankle Boot hacia los costados, pero tiene la particularidad de poseer valores promedios de atributos más bajos, lo que indica una pigmentación más oscura.

Finalmente, en Bag, los valores más altos se encuentran concentrados hacia la derecha del gráfico, lo que indica que el objeto en la imagen va a encontrarse en la parte inferior, y va a aproximarse bastante a los bordes laterales, hecho que se puede deducir del alto grosor de los picos, y de el elevado piso de valores promedio entre los atributos 200 y 650.

Dentro de cada etiqueta se pueden tener distintos grados de variabilidad en sus atributos para distintas instancias, coloquialmente, cada prenda de alguna categoría puede parecerse en mayor o menor medida a otra semejante. Un caso particular de una clase con un gran parecido a lo largo de todas sus instancias es TShirt/Top (clase 0), y para el caso opuesto, es decir que sus diferentes instancias tienen una gran variación, se tiene a la clase Sandal (clase 5).



*Comparación de imágenes promedio, superposición de todas las instancias de Clase 0 y Clase 5 - Figura (VIII)*

**

*Comparación de varianza de cada pixel, Clase 0 y Clase 5 - Figura (IX)*

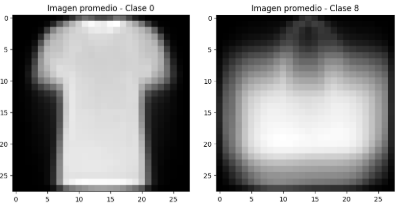
Este fenómeno es observable a simple vista en las imágenes promedio (*Figura (VIII)),* donde la clase TShirt/Top tiene una forma con contornos definidos, y la clase Sandals tiene una apariencia más “ruidosa”, en donde no se distingue con tanta claridad una forma definida; y en (*Figura(IX)*), donde la superficie de tonalidad amarrillo brillante es evidentemente mayor en la Clase 5 que en la Clase 0, indicando una mayor variabilidad en sus imágenes.

A fines prácticos, en cuanto al objetivo del presente trabajo, una menor variabilidad en los valores de los atributos de cada clase permite obtener rangos más precisos para su clasificación.

# Experimentos:

## Clasificación Binaria

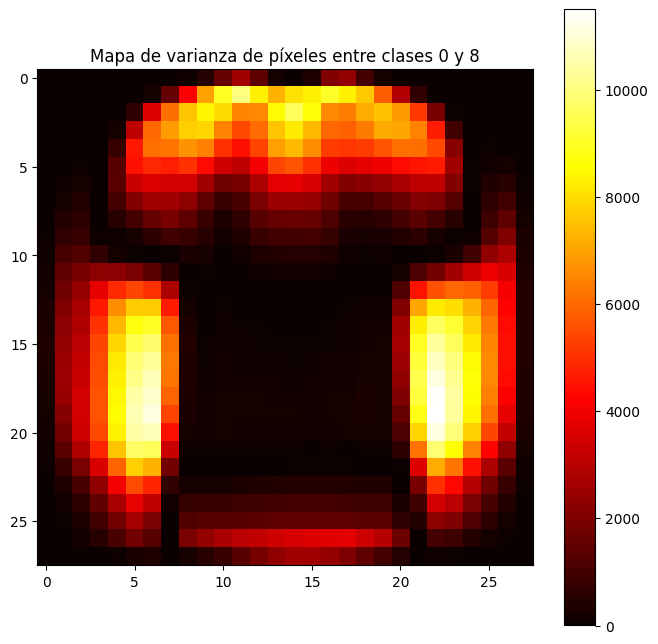
Esta sección consta de la implementación de un modelo clasificador kNN sobre dos clases: 0 y 8 (TShirt/Top y Bag). Ambas prendas son distinguibles fácilmente a simple vista.



*Comparación de imágenes promedio, superposición de todas las instancias de Clase 0 y Clase 8 - Figura (X)*

Para ello, se separan las instancias del dataset cuyo “label” sea igual a 0 o a 8, y del resultado se selecciona el 90% de los datos como “train”, para entrenar al modelo, y el 10% como “test” (al tener tantos datos, esta cantidad es suficiente para testear), equilibrados en términos de instancias por clase, para evaluar la performance del modelo.

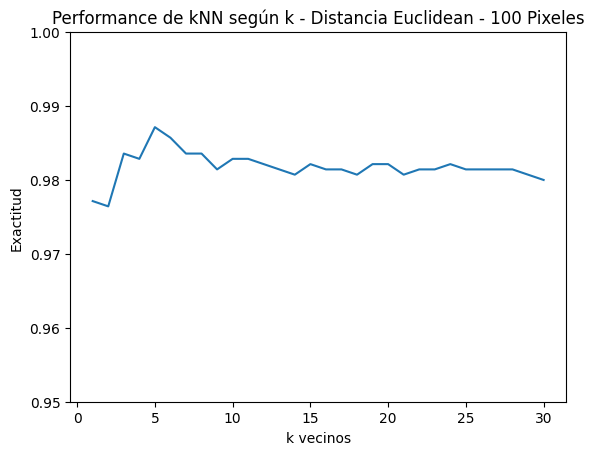
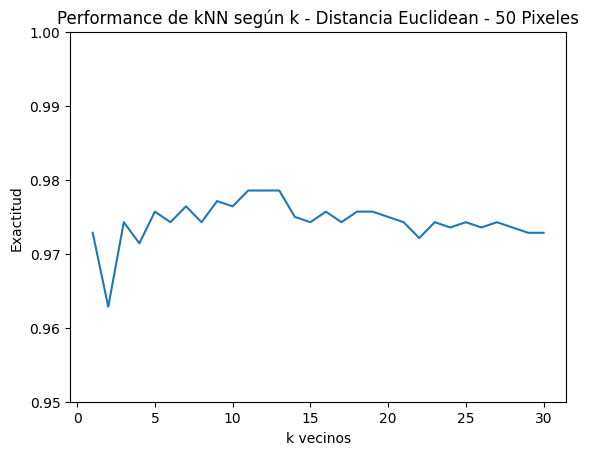
Lo que se busca finalmente es obtener el kNN con la mayor exactitud posible, y la pregunta que surge para conseguirlo es ¿que atributos e hiperparametros utilizar?. Como se tienen 784 atributos, es menester para optimizar el proceso seleccionar aquellos que ayuden a distinguir más entre distintas clases, para ello se realiza una comparación entre la clase 0 y la clase 8, donde se obtiene la varianza de cada uno de sus píxeles.



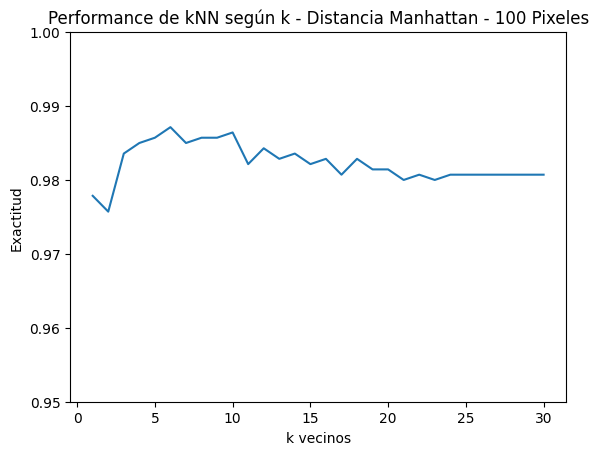
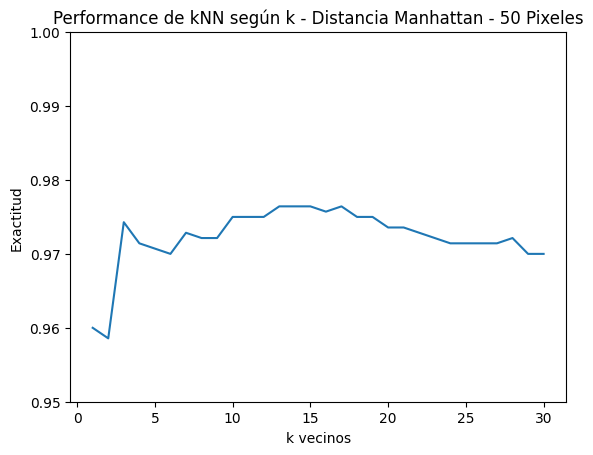
*Comparación de clase 0 y 8 en cuanto a la varianza del promedio de sus píxeles - Figura (XI)*

Los atributos que más información aportan resultan ser los que están pintados en tonalidades más claras; luego en los subconjunto train y test, se decide utilizar solamente los 100 atributos con la varianza más alta obtenida en esta métrica. Con los datos seleccionados, se entrena el modelo probando distintos hiperparámetros, como *“k nearest neighbours”* (Hasta k = 30)y medidas de distancia (Euclidean y Manhattan)*.* También, se prueba la utilización de los primeros 50 atributos, para ver si existe un cambio significativo a utilizar 100, y por consiguiente, determinar si este último vale la pena.

Se realizan gráficos en el cuál se puede observar la exactitud obtenida para cada profundidad y cada cantidad de atributos, según medida de distancia.



*Comparación de exactitudes de kNN según k, 50 vs 100 pixeles, con distancia ‘Euclidean’ - Figura(XII)*



*Comparación de exactitudes de kNN según k, 50 vs 100 pixeles, con distancia ‘Manhattan - Figura (XIII)*

En los gráficos se ve una clara mejora en cuanto a la exactitud en ambos modelos entrenados con 100 píxeles sobre los de 50, por lo que dentro de estos se pasa a comparar entre Manhattan y Euclidean.

Las exactitudes máximas obtenidas en ambos gráficos, junto con el valor de k correspondiente, son las siguientes:

* (Euclidean) Exactitud máxima con 100 atributos: 0.9871428571428571- En k: 5
* (Manhattan) Exactitud máxima con 100 atributos: 0.9871428571428571En k: 6

Aunque la mayor exactitud es la misma, Euclidean la alcanza con un k menor, lo que podría ser preferible en términos de rendimiento. Por otro lado, es importante considerar también las demás exactitudes obtenidas con otros k; se puede observar en los gráficos que la curva de exactitud en función de k con distancia Euclidean es más “constante” a lo largo de su recorrido, es decir, que tiene menos cambios abruptos o significativos al variar el k; en cambio los valores del análisis con distancia Manhattan reflejados en la curva de su gráfico se ven menos constantes, es decir, que varían en mayor medida al cambiar k. Esto sugiere que, para este conjunto de datos, la distancia Euclidean podría ser más robusta; su desempeño es más consistente ante cambios en k y menos sensible a la parte del proceso correspondiente a la aleatoriedad, como la variabilidad en los subconjuntos de entrenamiento y prueba. En contraste, Manhattan muestra mayor oscilación, lo que podría indicar una mayor sensibilidad a la distribución de los datos, y sus picos de exactitud deberse por lo tanto tal vez a algo más cercano a la “casualidad”. Cabe aclarar que en esta sección no se utiliza k-folder, por lo que todo está más sujeto a la aleatoriedad.

Por estas razones, se decide que el modelo óptimo entre los planteados es aquel que utiliza 100 atributos (los seleccionados anteriormente), k = 5, y distancia Euclidean, con un 99% (redondeado a dos cifras) de exactitud en base a los datos de test. Para ampliar el detalle de la performance:

True 0, False 8, False 0, True 8: [685, 15, 9, 691]

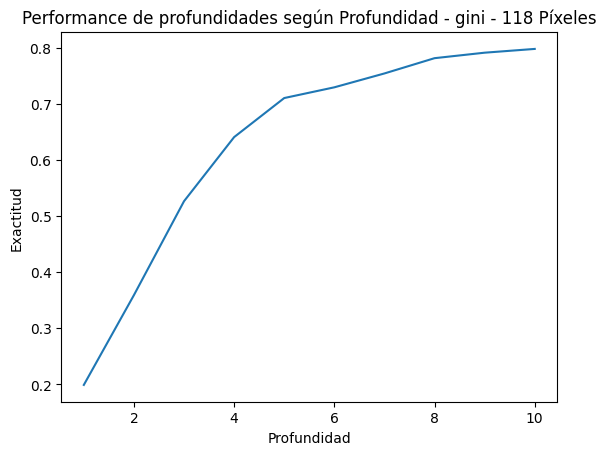
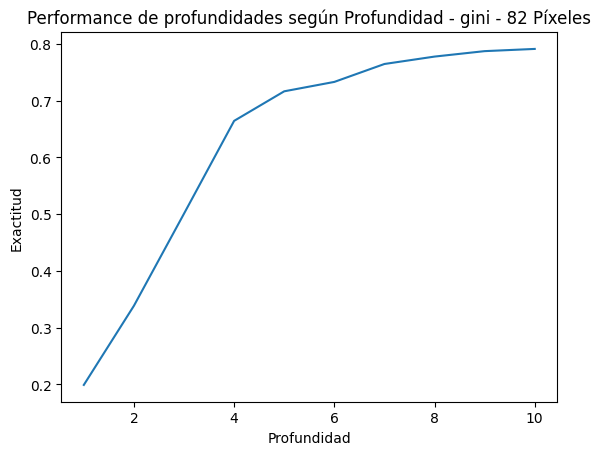
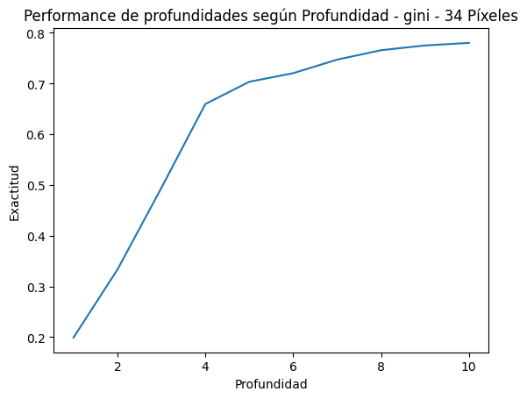
## Clasificación Multiclase

En esta sección se busca entrenar un modelo de clasificación “Decision Tree” sobre todas las clases (0 a 9).

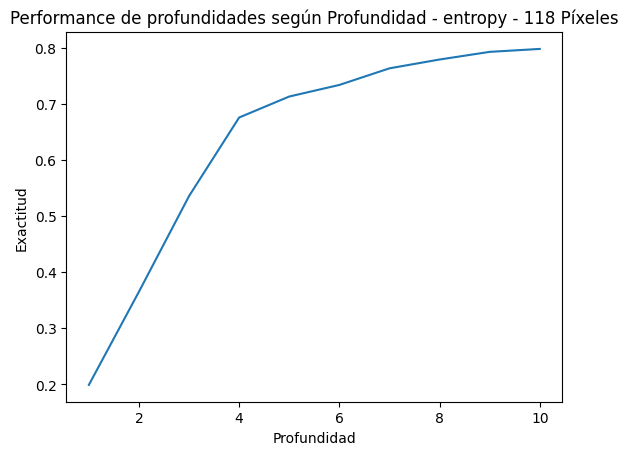
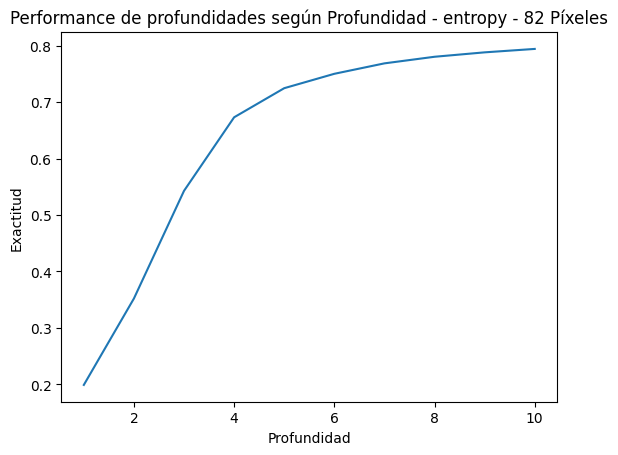
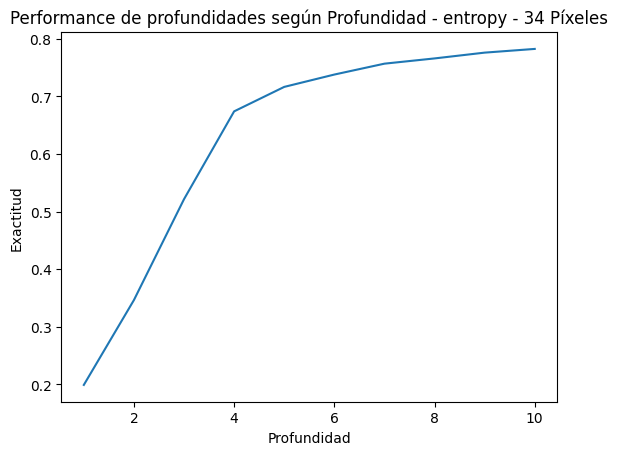
Lo primero a realizar es la separación de los datos: un 90% para desarrollo, y un 10% holdout, el cual está equilibrado en cuanto a cantidad de instancias por clase, y será utilizado al final del trabajo para determinar la exactitud del modelo definitivo.

Para aprovechar de la mejor forma la información disponible, es necesario seleccionar los atributos que más información aportan en cuanto a la diferenciación de las clases. Para ello, se realizan todas las comparaciones posibles entre pares de clases, donde se observa la varianza en cada pixel (como en *Figura(XI)*), y se obtiene un “top n píxeles más importantes” (eliminando los repetidos ).Es decir, aquellos atributos que poseen valores promedio más distanciados entre un par de clases. Como resultado, se obtiene una lista de píxeles, que serán los utilizados para entrenar los distintos árboles.

Se quiere encontrar el árbol óptimo, por lo que se comparan las exactitudes en función de la profundidad (hasta 10) utilizando n = 5 (115 atributos), n = 3 (82 atributos) y n = 1 (34 atributos), y considerando tanto Entropy como Gini como criterios de partición . Todos los entrenamientos son llevados a cabo con k-fold cross-validation, respetando el equilibrio de instancias por clase en cada partición, con el fin de reducir la influencia de la aleatoriedad en los resultados.

**

*Comparación de exactitudes de Decision Tree según profundidad,34 vs 82 vs 118 píxeles, con criterio Gini - Figura (XIV)*

**

*Comparación de exactitudes de Decision Tree según profundidad, 34 vs 82 vs 118 píxeles, con criterio Entropy - Figura (XV)*

Todos los resultados son en extremo similares. Viendo los valores máximos numéricamente:

* (Entropy) Exactitud máxima con 118 atributos: 0.7983492063492064- En profundidad: 10
* (Entropy) Exactitud máxima con 82 atributos 0.7945396825396825 - En profundidad: 10
* (Entropy) Exactitud máxima con 34 atributos: 0.7825873015873015 - En profundidad: 10
* (Gini) Exactitud máxima con 118 atributos: 0.7975555555555557 - En profundidad: 10
* (Gini) Exactitud máxima con 82 atributos: 0.7913650793650794 - En profundidad: 10
* (Gini) Exactitud máxima con 34 atributos: 0.7802380952380953 - En profundidad: 10

Las máximas exactitudes alcanzadas, todas en profundidad 10, son prácticamente idénticas, a pesar de que se reducen los atributos de 118 a 34, lo que parece indicar que dentro del primer grupo hay mucha redundancia en información; en Entropy, al pasar de n = 5 a n = 3 se pierde un 0,38% aproximadamente de exactitud, y en Gini un 0,62%; pero también se utilizan 36 atributos menos, lo que es bueno para el rendimiento; al pasar de n = 3 a n = 1, se pierde un 1,19% de exactitud en Entropy, y 1,12% en Gini; a la vez que 48 atributos. Entre ambos criterios de particiones no existe una diferencia considerable, por lo que cualquiera sería aceptable. Se decide, por lo tanto, tomar como modelo final al árbol de profundidad 10, entrenado con los 82 atributos previamente seleccionados, debido a ser un punto medio entre eficiencia por cantidad de atributos y resultados en exactitud, y con criterio Entropy (por tener un resultado ligeramente mejor que Gini).

Finalmente, se verifica la exactitud del modelo definitivo utilizando el 10% del dataset original correspondiente a los datos reservados holdout. El valor obtenido es de 80.09% (redondeado a dos decimales).

Conclusión:

En conclusión, en el dataset Fashion-MNIST se pudieron encontrar patrones claros e información valiosa para el entrenamiento de modelos de clasificación, y se obtuvieron buenos resultados en cuanto al rendimiento de los mismos. El Análisis Exploratorio permitió identificar a través de múltiples gráficos aquellos atributos que son constantes a lo largo de todas las instancias en general ubicados en los bordes (particularmente las esquinas), y aquellos que varían más a la hora de comparar distintas prendas, los cuales son los más valiosos para el fin planteado en este trabajo. Se realizaron análisis en donde se agruparon prendas según similitud, comparando los valores promedio de sus atributos, y se pudo observar que grupos son más fáciles de distinguir entre sí (pantalones vs remeras vs calzados), y también las comparaciones que serían más problemáticas, así como el comportamiento de los píxeles según la prenda. Se pudo observar a su vez, la presencia de clases que poseen una mayor variación en cuanto a los valores de sus atributos para distintas instancias, como “Sandals” y otras que permanecen más estáticas, como “Tshirt/Top”. Todos estos resultados fueron importantes en tanto al entendimiento general del dataset con el cual se trabaja, ya que es vital conocer las distintas dinámicas que poseen los atributos para luego poder definir cuál es la mejor forma de entrenar los distintos modelos.

En Clasificación Binaria, se entrenó el mejor modelo kNN posible para clasificar aquellas clases pertenecientes a los “label” 0 y 8 (Tshirt/Top, y Bag). Para ello, se guardó un 90% del total de las instancias para el entrenamiento de los modelos, y un 10% para testearlos; luego se seleccionaron los 100 píxeles más “importantes” estratégicamente, utilizando la técnica vista en el análisis exploratorio de la diferenciación en la variación de los valores de los datos al comparar dos imágenes promedio, y se realizaron pruebas comparando modelos con distintos hiperparametros, como distancia Euclidean y Manhattan, y variando el valor de k entre 1 y 30, buscando la mejor exactitud posible. El mejor resultado obtenido en cuanto a la exactitud fue de 99%, con 100 atributos, y k = 5 en euclidean y k = 6 en manhattan; sin embargo Euclidean mostró mayor robustez y constancia en sus valores al variar k, por lo que se llegó a la conclusión de que la mejor opción es un modelo de k = 5, con 100 atributos y distancia Euclidean.

En Clasificación Multiclase, se realizó un modelo clasificador Decision Tree para las 10 clases existentes. A diferencia de lo anterior, se guardó un 10% de las instancias totales como “Holdout”, equilibrado en cuanto a instancias por clase, y se utilizó la técnica de k-fold, también equilibrada, para evaluar los distintos modelos, con el fin de reducir la incidencia de la aleatoriedad en el proceso. Los atributos a utilizar fueron seleccionados al comparar todas las clases de a pares, y seleccionar los primeros n píxeles con mayor variación en cuanto al promedio de su valor en ambas clases, lo que significa que son los que más información aportan en cada comparación binaria. Se probaron distintos modelos variando los n píxeles a utilizar (1, 3 y 5), el criterio (Gini o Entropy), y la profundidad del árbol. Todos los resultados obtenidos fueron en extremo similares, obteniendo la mejor exactitud en profundidad 10, por amplia diferencia, y siendo esta de un valor similar en todos los casos, por lo que se concluyó que era pertinente descartar información redundante, es decir, píxeles utilizados para el entrenamiento. Por estas razones se eligió como modelo definitivo aquel con criterio Entropy (por resultado infimamente mejor), profundidad = 10, y n = 3 (82 píxeles), sacrificando solo un 0,38% de exactitud a cambio de reducir los atributos en 36, lo cual mejora la eficiencia del modelo sin una pérdida sustancial de rendimiento. Finalmente, utilizando el Holdout, se registró que la exactitud de este modelo es de 80.09%.