Trabajo Práctico 2

Clasificación de imágenes en distintos tipos de prenda

Materia: Laboratorio de Datos

Grupo: How DER You?

Azul Barracchia
Augusto Guarnaccio
Mario Sigal Aguirre

Introducción

El objetivo de este informe será documentar los pasos que se siguieron para trabajar en la selección y evaluación de modelos de clasificación sobre el data set Fashion MNIST. Sin embargo, antes de hablar de esto debemos referirnos al análisis exploratorio que fue necesario realizar para así poder ganar un mayor conocimiento de los datos.

En primer lugar, fue necesario conocer la descripción del data set. El mismo cuenta con 60000 filas o tuplas y 785 atributos. Dentro de los atributos, el primero se corresponde al label, una clasificación por tipo de prenda. La clasificación dada significa: 0 - Remera/top, 1 - Pantalones, 2 - Suéter, 3 - Vestido, 4 - Abrigo, 5 - Sandalias, 6 - Camisa, 7 - Zapatillas, 8 - Cartera y 9 - Botas. A su vez, por cada label contamos con 6000 prendas.

Los siguientes 784 atributos se corresponden al valor de cada uno de los píxeles de la imagen. Este valor se encuentra en una escala de 0 a 255.

En segundo lugar, quisimos conocer si había atributos más importantes que otros para predecir el tipo de prenda y cuáles podrían ser descartados. Para esta tarea decidimos superponer las diferentes prendas de una misma clase y, luego, superponer todas las prendas.

Superponiendo las prendas de una misma clase pudimos visualizar los píxeles más importantes dentro de cada tipo de prenda. Estos son fáciles de distinguir ya que son las zonas más oscuras. Además nos permiten ver que tan diferentes son las prendas de una misma clase ya que un coloreado homogéneo nos habla de mucha similitud entre las distintas prendas mientras que uno no homogéneo habla de varias diferencias.

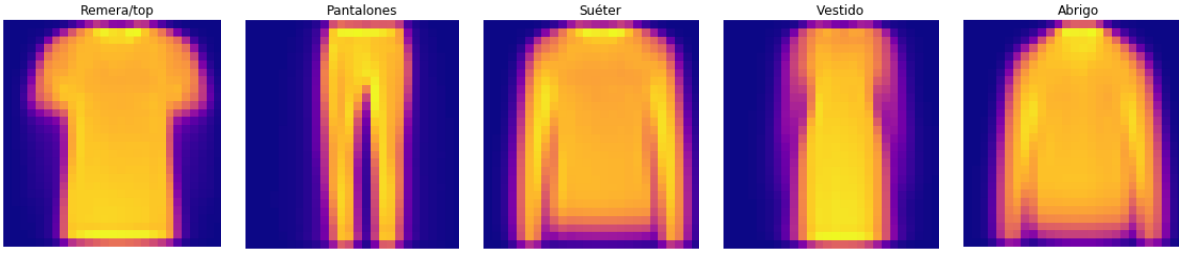


Imagen de todas las remeras/tops superpuestas

Imagen de todas los pantalones superpuestas

Imagen de todas las suéters superpuestas

Imagen de todas los vestidos superpuestas

Imagen de todas los abrigos superpuestas

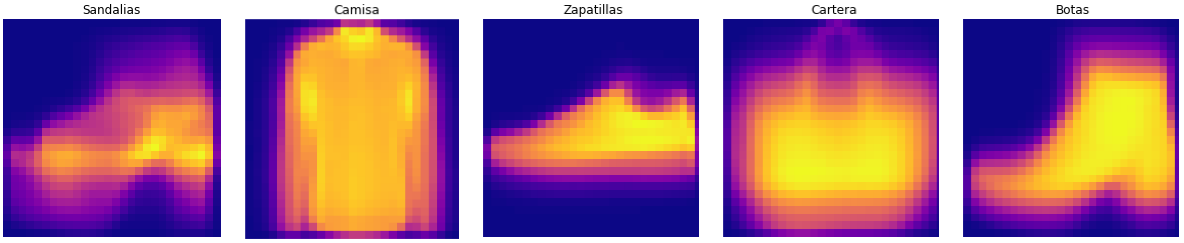


Imagen de todas las sandalias superpuestas

Imagen de todas las camisas superpuestas

Imagen de todas las zapatillas superpuestas

Imagen de todas las carteras superpuestas

Imagen de todas las botas superpuestas

Superponiendo todas las prendas pudimos visualizar los píxeles que más información tienen en general. Estos son fáciles de distinguir por el mismo motivo expuesto

en el párrafo previo y son un buen indicador acerca de qué píxeles no deberíamos elegir si queremos predecir o clasificar los distintos tipos de prenda ya que son compartidos por la mayoría de las imágenes.

Este procedimiento nos ayudó a comprender que los atributos de mayor relevancia son los que se encuentran en el centro de la imagen ya que los de los bordes en general valen 0 y no aportan información (esto se puede apreciar en la imagen promedio donde los bordes son blancos o apenas grises). A su vez, de este análisis pudimos concluir que se podría trabajar con menos atributos ya que son demasiados y varios no aportan información (como es el caso de los bordes). No obstante, para recrear la imagen consideramos que es más fácil dejar los 784 píxeles.

En tercer lugar, nos interesó saber si había clases de prendas similares entre sí para así anticiparnos a posibles problemas con los modelos de clasificación y entender posibles errores. Para poder respondernos esta pregunta decidimos seleccionar solo dos pares de prendas: pantalones-zapatillas y suéteres-camisas. Para estos pares de prendas decidimos superponer sus imágenes promedio.

En el primero de los dos casos, se puede ver como si bien hay algunas áreas compartidas y píxeles superpuestos, las prendas son considerablemente diferentes. Sus imágenes promedio no tienen la misma forma, lo que era esperado. Sin embargo, para terminar de asegurarnos esta diferencia decidimos hacer un scatterplot para cada prenda que graficara la tonalidad promedio (valor de píxel promedio) por fila de la imágen (es decir, cada 28 píxeles). Cuando los superpusimos pudimos ver curvas considerablemente diferentes que confirman estas diferencias.

Imágenes promedio de pantalones y zapatillas superpuestas

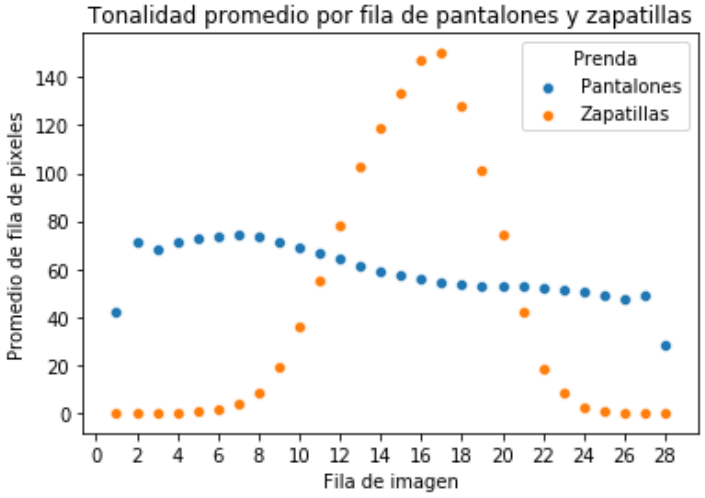
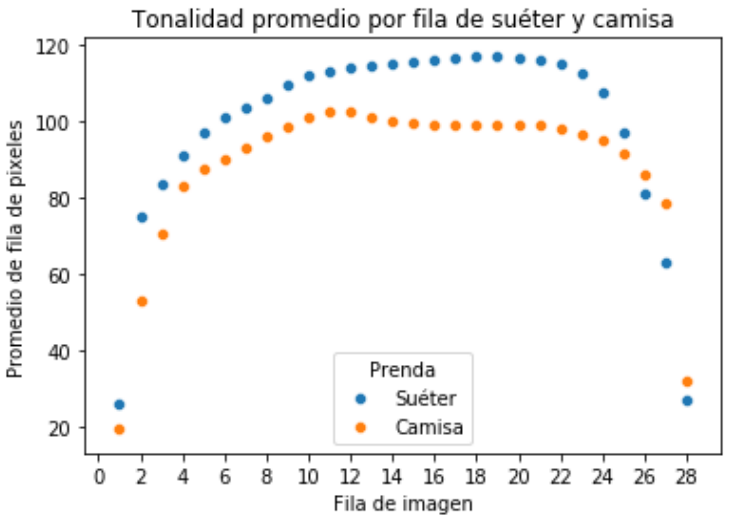


Gráfico de valor promedio de fila de píxeles por fila de imagen

En el segundo de los casos, se puede ver que las prendas son indistinguibles una vez superpuestas sus imágenes promedio. Esto sucede porque las imágenes promedio tienen una forma casi idéntica lo que nos habla de que van a ser difíciles de distinguir. Para

terminar de confirmar esta diferencia decidimos hacer el mismo scatterplot que en el caso anterior. Este scatterplot nos muestra curvas muy similares que además comparten un mismo comportamiento lo que termina de reafirmar su parecido.



Imágenes promedio de suéters y camisa superpuestas

Gráfico de valor promedio de fila de píxeles por fila de imagen

Por lo tanto, esperamos que prendas con formas similares sean parecidas.

En cuarto lugar, evaluamos la varianza entre las diferentes prendas de una misma clase. Es decir, qué tan similares o diferentes son las prendas de una misma clase. Con esto en mente graficamos el desvío estándar para cada una de las prendas.

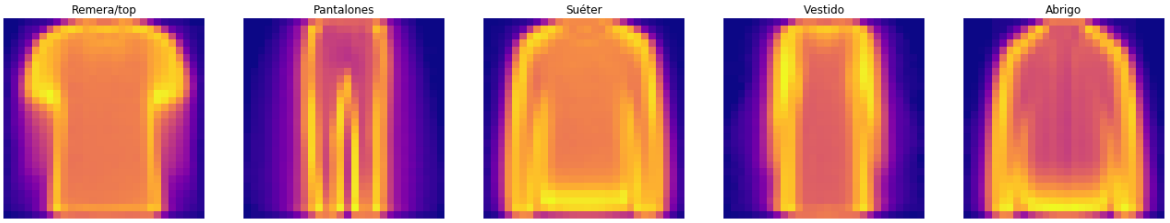


Imagen de los desvíos estándares de remera/top superpuestos

Imagen de los desvíos estándares de pantalones superpuestos

Imagen de los desvíos estándares de suéteres superpuestos

Imagen de los desvíos estándares de vestidos superpuestos

Imagen de los desvíos estándares de abrigo superpuestos

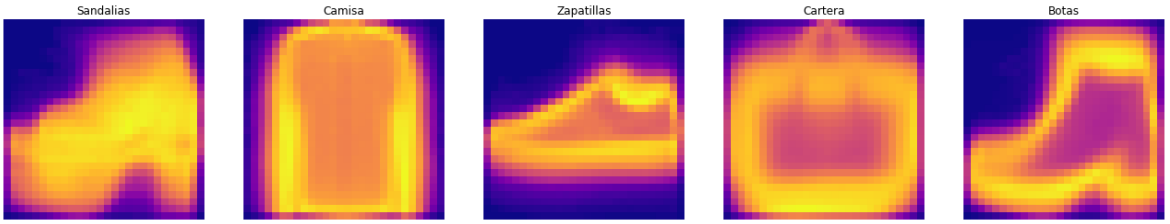


Imagen de los desvíos estándares de sandalias superpuestos

Imagen de los desvíos estándares de camisa superpuestos

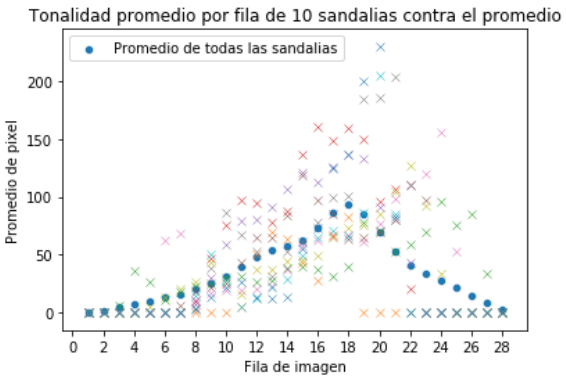
Imagen de los desvíos estándares de zapatillas superpuestos

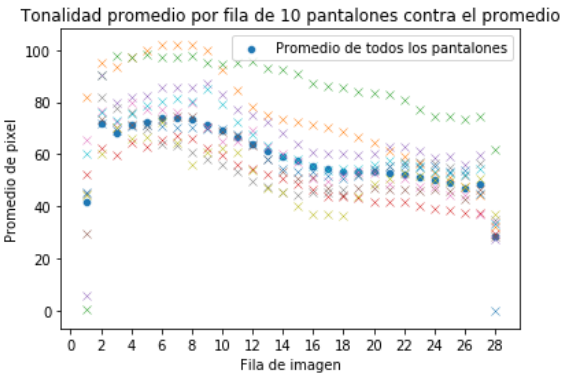
Imagen de los desvíos estándares de carteras superpuestos

Imagen de los desvíos estándares de botas superpuestos

De estos gráficos, podemos ver cuáles son las clases de prenda con la mayor varianza y dónde se da esta. Aquellas zonas más brillantes son las zonas donde el desvío estándar es mayor y, por lo tanto, donde hay una mayor varianza respecto del promedio. Análogamente, las zonas menos brillantes son las zonas donde el desvío estándar es menor y hay una menor varianza. Por lo tanto, aquellas clases de prendas que tengan mayor cantidad de zonas oscuras van a ser clases de prenda donde las prendas sean todas muy parecidas entre sí, por ejemplo, el pantalón. Caso contrario, van a ser clases de prenda donde las prendas sean todas muy diferentes entre sí, por ejemplo, las sandalias. Para obtener una mejor visualización de esto, decidimos analizar tanto sandalias como pantalones.

Para ello, graficamos el promedio de píxeles por fila de la imagen promedio de las sandalias junto con el promedio de píxeles por fila de 10 imágenes random de sandalias. Para pantalones hicimos lo mismo.





Las cruces de un mismo color indican una misma sandalia

Las cruces de un mismo color indican un mismo pantalón

En el primer gráfico podemos ver como las curvas que corresponden a sandalias random presentan comportamientos diferentes a los del promedio de todas las sandalias y tienen una distancia muy variable. En el segundo gráfico podemos ver como las curvas que corresponden a pantalones random presentan comportamientos muy similares a la curva promedio y como la distancia no es tan marcada. Esto confirma que las imágenes de clases de prenda más brillantes son las clases de prenda donde las prendas son menos parecidas

entre sí mientras que las que son más oscuras son las clases de prenda donde las prendas son más parecidas entre sí. Por lo tanto, podríamos afirmar que clases prendas que tienen una forma o un diseño poco variable como lo son los pantalones van a tener elementos parecidos entre sí. También valdría el razonamiento opuesto.

En último lugar, creemos que el hecho de que el data frame esté compuesto por imágenes complicó el análisis exploratorio pero no porque haya agregado una dificultad a los métodos sino que nos obligó a pensar las cosas de otra manera (como por ejemplo el desvío estándar). Tuvimos que pensar cuidadosamente qué era lo que queríamos buscar y qué era lo que íbamos a graficar para poder entender mejor lo que pasaba. No obstante lo cual, una vez realizado el análisis se comprendió muchísimo mejor los datos con los que contábamos lo que facilitó considerablemente entender como poder aplicar los modelos sobre este data set que tanto difiere con los que trabajamos en clase.

Experimentos

Una vez realizado el análisis exploratorio nos vimos capaces de desarrollar modelos de decisión que nos permitieron clasificar prendas según su label. Para ello, primero realizamos un modelo kNN donde fuimos variando la cantidad de vecinos y los atributos; y luego, realizamos un árbol de decisión donde fuimos variando la profundidad del árbol y el criterio de corte.

KNN

En este experimento, se deseaba clasificar imágenes según si correspondían a un pantalón o a una remera. Para ello, se entrenó un algoritmo de aprendizaje supervisado (K-Nearest Neighbors) para que brindara un modelo predictivo de clasificación.

Para evaluar estos modelos de predicción, primero se redujo el data frame principal a sólo las filas con label igual a 0 o 1 (remeras/tops y pantalones) y se dividió el data set en data train y data test. Para este último, se apartaron una selección del 15% de los datos y se trabajó con el 85% de los datos restantes. A su vez, se verificó que esta división mantuviera la distribución de ambas clases de datos para así asegurar que el modelo se estaba entrenando con una misma proporción tanto de datos de remeras/tops como de pantalones.

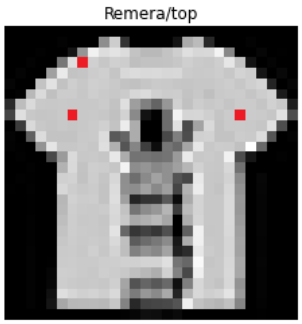
En primer lugar, se entrenó al mismo modelo distintas veces con diferentes subconjuntos de 3 atributos elegidos tanto de manera específica como aleatoria.

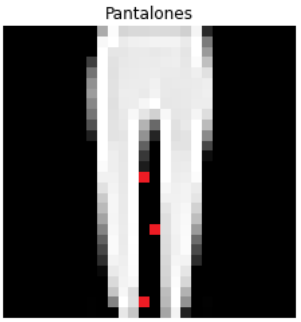
A continuación se expresará una breve descripción de cada subconjunto y una tabla con los resultados de exactitud del modelo para cada subconjunto usado como entrenamiento.

Para el subconjunto de “max_medias”, se seleccionaron 3 atributos, el pixel con el valor más alto en promedio para pantalones, el del valor más alto para remeras, y el pixel con la variaciáción mayor entre las dos prendas.

El criterio para “max_diferencia” fue elegir los 3 atributos con la variación más grande entre los promedios de pantalones y remeras. Esto se hizo restando los data frames de promedios.

Para los subconjuntos de “espacio_pantalones” y “mangas_remera", se seleccionaron pixeles específicos de esas zonas que consideramos que podían diferenciar ambas prendas. Se seleccionaron los siguientes pixeles:





En los subconjuntos “primeros” y “últimos”, quedaron los primeros y últimos pixeles respectivamente, (los pixeles 1, 2 y 3, y los pixeles 784, 783, 782). En “pixeles_random”, se encuentran 3 pixeles generador al azar.

Con la función GridSearchCV, se buscó cuál era la mejor k y su score, para cada selección de atributos.

nombre_subconjunto	mejor_k	mejor_score
max_medias	20	0,6913201852722384
max_diferencias	18	0,867662002496165
espacio_pantalones	13	0,8187045663512451
mangas_remera	15	0,7526847491292428
primeros_pixeles	18	0,020828409276163962
ultimos_pixeles	18	0,01735721124858376
pixeles_random	20	0,401222197903443

Tabla con los resultados del GridSearch

Para hacer una comparación del método con distintas cantidades de atributos, decidimos trabajar con la mejor k y el mejor criterio de selección de atributos obtenidos de la tabla anterior. Al varíar la cantidad de atributos, con k fija igual a 18, la figura 1 muestra los distintos scores obtenidos.

Por último, se hizo KNN para distintas cantidades de atributos (5, 15, 25, … , 775), y distintos valores de k (1, 2, 3, … , 9). Los atributos se seleccionarion al azar con cada iteración. En la figura 2 se puede ver como el score varía en relación a estas dos variables mencionadas. Se puede apreciar cómo, con pocos atributos, al variar la k, el score sube

considerablemente; y como con muchos atributos el valor de k no modifica tanto el score ya que hay una nube de datos muy densa con todos los datos muy cercanos entre sí.

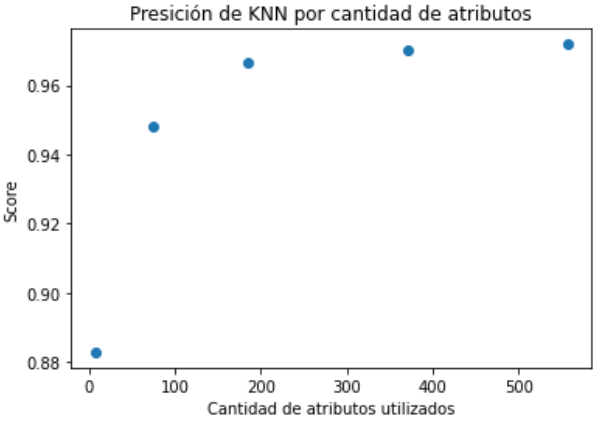


Figura 1

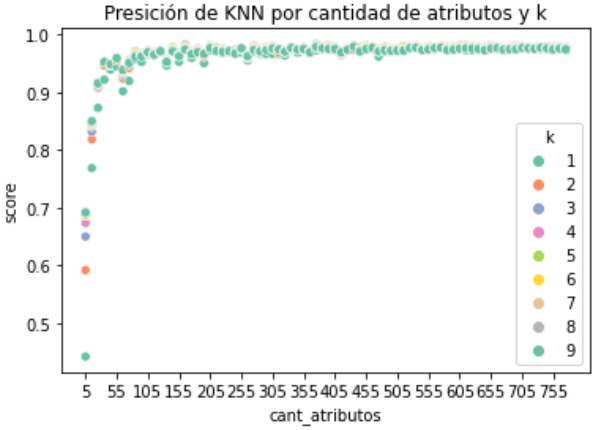


Figura 2

Árbol de decisión

A la hora de realizar el experimento, como ya sabíamos que los datos estaban balanceados por clases simplemente nos aseguramos de hacer una división que los mezclara pero que mantuviera una distribución pareja de todas las clases. La proporción que utilizamos para separar entre data train y data test fue la misma que en los modelo de kNN: 85% para train y 15% para test.

Una vez hecho esto, nos fijamos cuál era el árbol de altura infinito que se podía generar con nuestros datos para así saber cuál era la máxima profundidad que podía llegar a alcanzar nuestro modelo y poder tener un rango dentro del cual buscar la altura óptima. Los resultados de este árbol nos dieron una profundidad de 51.

Con este resultado en mente decidimos hacer un grid search para elegir el mejor árbol posible para nuestro modelo. Los parámetros que nos interesaron evaluar y conocer fueron: las profundidades de 4 a 51 (decidimos descartar las profundidades 1,2,3 porque creemos que no aportaban nada) y los criterios ‘entropy’ y ‘gini’. Este grid search fue acompañado de un k-fold cross validation para el cual decidimos usar 5 folds dada la gran cantidad de datos con los que contábamos.

El mejor modelo que este grid search devolvió fue el árbol de profundidad 12 y criterio ‘entropy’ para el cual los datos de entrenamiento devolvieron un accuracy igual a 0.819, es decir, un accuracy del 82%.

Nos gustaría aclarar que correr este grid search llevo un tiempo considerable de aproximadamente 3 horas con la función optimizada de la librería por lo que desaconsejamos correr la parte del código del grid search si no se tiene una GPU potente.

Ahora que contamos con el mejor modelo posible para nuestros datos decidimos evaluarlo con el data test. No obstante, además de interesarnos por medir el accuracy, también nos interesó medir la precisión aunque también incluiremos las métricas del recall (solo por completitud). Nos interesamos por la precisión ya que dada la gran cantidad de

datos nos parece que es más conveniente conocer cuán útiles son los resultados antes de cuán completos puesto que debido a la magnitud de la base de datos sería ineficiente chequear a ojo y detectar cuales deberían haber sido positivas.

A su vez, nos interesamos por estas otras métricas ya que el accuracy no dice nada sobre aciertos o errores que pueda llegar a tener nuestro modelo.

El accuracy obtenido habiendo evaluado en los data test fue de 82% mientras que la precisión fue de 82% y el recall de 82%. Estas métricas son buenas ya que tenemos un buen porcentaje de precision y un buen porcentaje de recall, lo que nos habla de que nuestro clasificador tiene una buena performance. Estos números reflejan que de las instancias clasificadas como positivas un 82% efectivamente lo son como también vemos que de las instancias positivas que teníamos se clasificaron correctamente un 82%. Por lo tanto, se clasificaron como positivas un 18% de instancias que no lo eran.

Vale aclarar que, en un principio desconfiamos de esta coincidencia tanto de valores de accuracy como de precision y recall y evaluamos nuestro modelo con otras profundidades (dejamos fijo el criterio de corte) y con un data test que no mantuviera la distribución de las clases. Sin embargo, hecho esto, vimos como nuestro modelo mantenía estos tres valores iguales. Por ello, es que podemos afirmar la buena performance de nuestro clasificador.

Otra cuestión que nos gustaria aclarar es que nos resultó un tanto extraño que la performance con los datos de test sea la misma que la alcanzada con los datos de train pero cuando decidimos ver los números sin redondearlos vimos que la diferencia eran apenas unos decimales lo que creemos que se explica con el hecho de que se entrenó con 51000 datos y se testeó con 9000. Es decir, como tuvo que clasificar considerablemente menos datos no tuvo tantas chances de realizar una clasificación incorrecta.

Conclusiones

Fueron varias las conclusiones que pudimos obtener durante el desarrollo del presente trabajo.

En primer lugar, entendimos la importancia de realizar un buen análisis exploratorio ya que este permite una buena comprensión de los datos con los que se cuenta, lo que influye directamente en la selección de los modelos. Sin ir muy lejos, fue gracias a este análisis que en el kNN pudimos elegir buenos conjuntos de píxeles para armar el clasificador.

En segundo lugar, pudimos comprender la variabilidad y sensibilidad del kNN ya que es un método que varía muchísimo de acuerdo a la cantidad de vecinos como la cantidad de atributos que se elija para predecir.

En tercer y último lugar, pudimos comprender las diferencias y los diferentes alcances tanto del árbol de decisión como del kNN. Si bien ambos modelan la realidad con una precisión similar siempre y cuando se entrenen con una buena elección de hiperparametros y bases de datos de Train (los distintos subconjuntos de pixeles en este caso), notamos que se debe ser cuidadoso al trabajar con el kNN ya que resulta muy sencillo desarrollar un modelo que overfitee los datos. Esto se ve por ejemplo cuándo tenemos un k alto y una gran cantidad de atributos. En este caso, se puede llegar a tener una precisión del 97%. En cambio, con el árbol de decisión resulta más sencillo evitar overfitear el modelo.

De todos modos, no creemos que se pueda afirmar que hay uno mejor que el otro sino que depende de lo que uno busca de acuerdo a cada modelo y a lo que necesita. Desarrollar y explorar los distintos parámetros de kNN lleva un tiempo y se hace necesario conocer bien los datos mientras que el árbol de decisión se puede desarrollar de una forma más sencilla y rápida. Además, es necesario tener cuidado de no overfitear cuando uno explora y desarrolla kNN.

Por eso, creemos que si se necesita un modelo con una buena precisión pero uno lo quiere desarrollar de una forma rápida y eficiente debería elegirse el árbol de decisión. En cambio, si se tiene tiempo de entender y explorar los datos además de probar distintas combinaciones el kNN podría ser muy útil dada su altísima precisión. Pero de vuelta, hay que tener cuidado con no overfitear.

Todos estos resultados sugieren que es importante tener en cuenta la complejidad de los modelos y la selección de atributos para obtener un buen rendimiento en la clasificación de imágenes de prendas de vestir o en general para cualquier algoritmo de clasificación de aprendizaje supervisado.