DEPARTAMENTO DE COMPUTACION

Facultad de Ciencias Exactas y Naturales - UBA

Trabajo práctico 2

13 de noviembre de 2023 — Laboratorio de datos

Integrante	LU	Correo electrónico
Llop, Tobias	871/22	tobiasllop@gmail.com
Pasquet, Felipe Luc	1084/22	felipe.pasquet@gmail.com
Stabile, Delfina	819/22	delfistabile18@gmail.com

Facultad de Ciencias Exactas y Naturales
Universidad de Buenos Aires
Ciudad Universitaria - (Pabellón I/Planta Baja)
Intendente Güiraldes 2610 - C1428EGA
Ciudad Autónoma de Buenos Aires - Rep. Argentina
Tel/Fax: (++54 +11) 4576-3300
http://www.exactas.uba.ar

Introducción

En el presente trabajo vamos a trabajar con la base de datos de imágenes de Fashion MNIST.
El objetivo es generar, comparar y mejorar diferentes modelos predictivos, que cuando se les entregue una nueva imagen decidan qué prenda es.

Análisis de los datos

Poseemos un archivo CSV en el que se encuentran 60000 representaciones de imágenes en blanco y negro de 784 pixeles. Cada píxel está representado en una columna en la que se encuentra el nivel de oscuridad que posee cada imágen en ese píxel. El nivel de oscuridad varía entre 0 y 255. Definimos como nuestra variable de interés a la primera columna 'label' que nos indica que tipo de prenda es cada imágen a partir de una clasificación numérica. La clasificación es la siguiente:

- 0 Remera
- 1 Pantalón
- 2 Pullover
- 3 Vestido
- 4 Saco
- 5 Sandalias
- 6 Camisa
- 7 Zapatillas
- 8 Bolso
- 9 Tacos

El primer experimento que realizamos fue ver qué atributos eran relevantes para predecir el tipo de prenda. Para eso, calculamos la varianza de las imágenes en cada píxel utilizando la función var() de numpy y luego los ordenamos de mayor a menor varianza. Por último, los graficamos en un histograma.

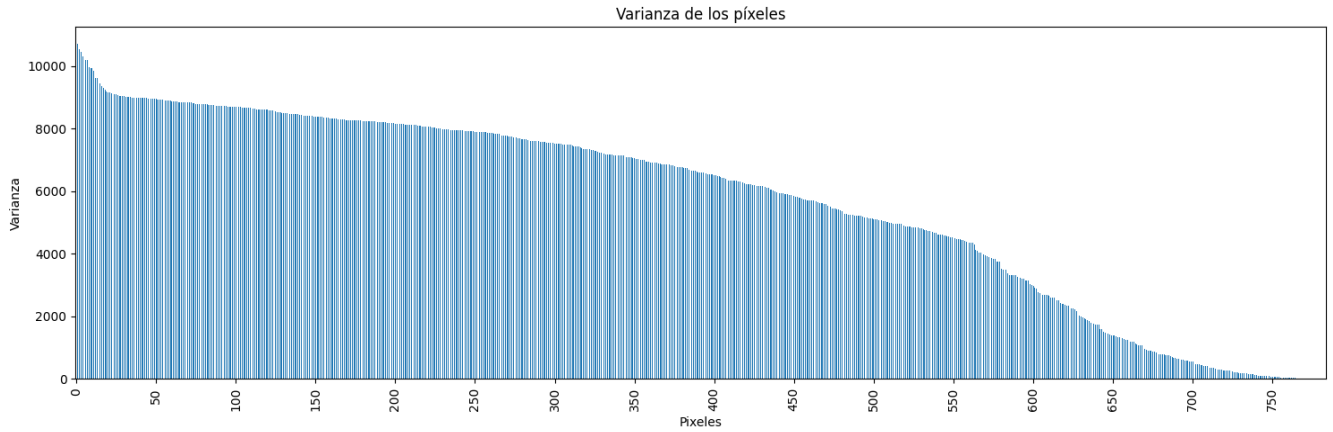


Gráfico 1: Histograma de píxeles ordenados
de mayor a menor según su varianza.

Observando el gráfico 1, observamos que hay píxeles con mucha varianza que son profundamente relevantes a la hora de predecir el tipo de prenda, mientras que, otros con poca varianza no tienen tanta relevancia para la predicción. Decidimos hacer un “zoom” a las últimas 20 barras, para ver cuáles eran los pixeles con menor varianza:

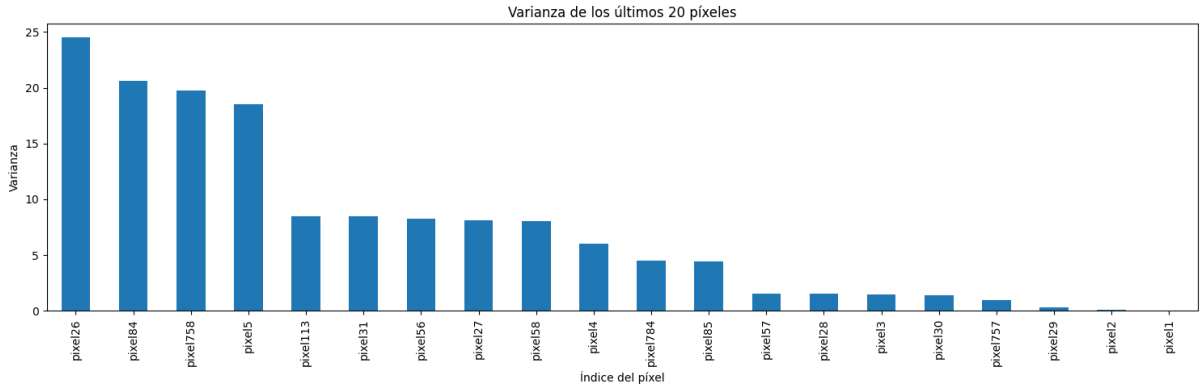


Gráfico 2: Histograma de los 20 píxeles con menor varianza.

Observando el gráfico 2, pudimos ver que algunos de los píxeles con menor varianza resultaron ser los ubicados en las esquinas de las imágenes, como por ejemplo, el píxel 1, 2, 757, 784, 28. Concluimos que estos, son muy poco relevantes a la hora de realizar nuestras predicciones, ya que tienen una varianza menor a 10, por lo que los podríamos descartar.

El segundo experimento que realizamos fue ver si había algún tipo de prenda más fácil de diferenciar que otra. Para eso utilizamos el método PCA, que reduce nuestras variables a dos componentes.

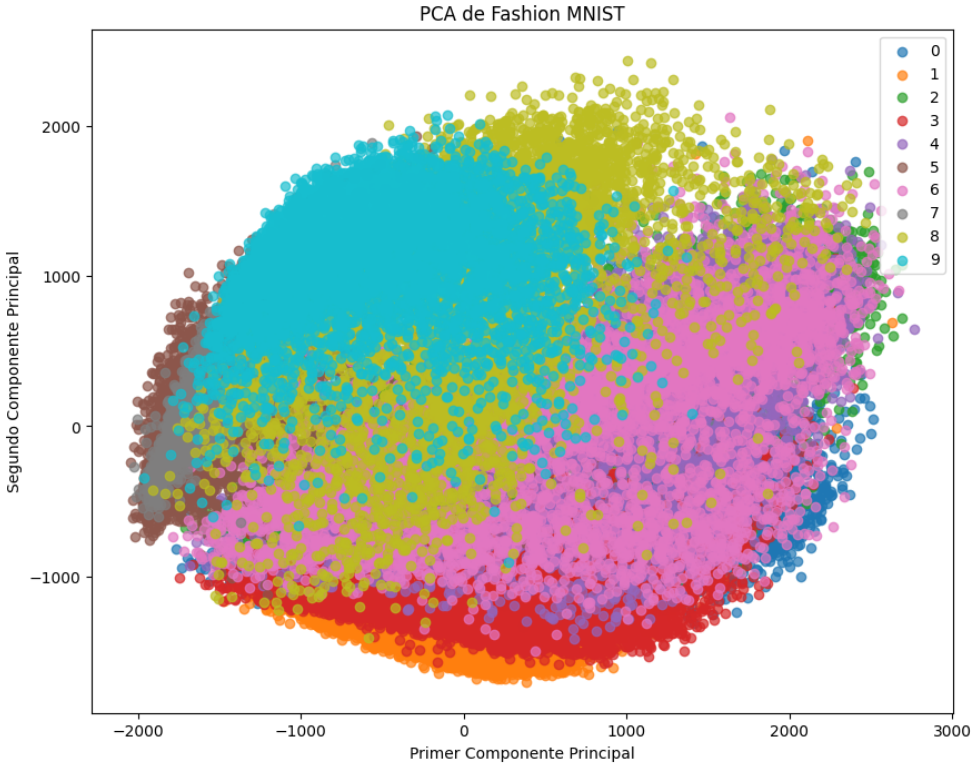


Gráfico 3: Gráfico de PCA
diferenciando a cada clase por tipo de prenda

Observando el gráfico 3 deducimos que definitivamente hay tipos de prendas que se pueden diferenciar de una forma más sencilla entre sí. Por ejemplo, es más fácil diferenciar remeras (0) de pantalones (1), que de pullovers (2).
El siguiente desafío al que nos enfrentamos era ver qué tan diferentes eran las imágenes de una clase. Para ello, tomamos todas las imágenes de la clase 3 perteneciente a los vestidos. Luego calculamos el promedio del nivel de cada oscuridad para cada pixel y graficamos una imágen con todos los valores calculados correspondientes a cada pixel.

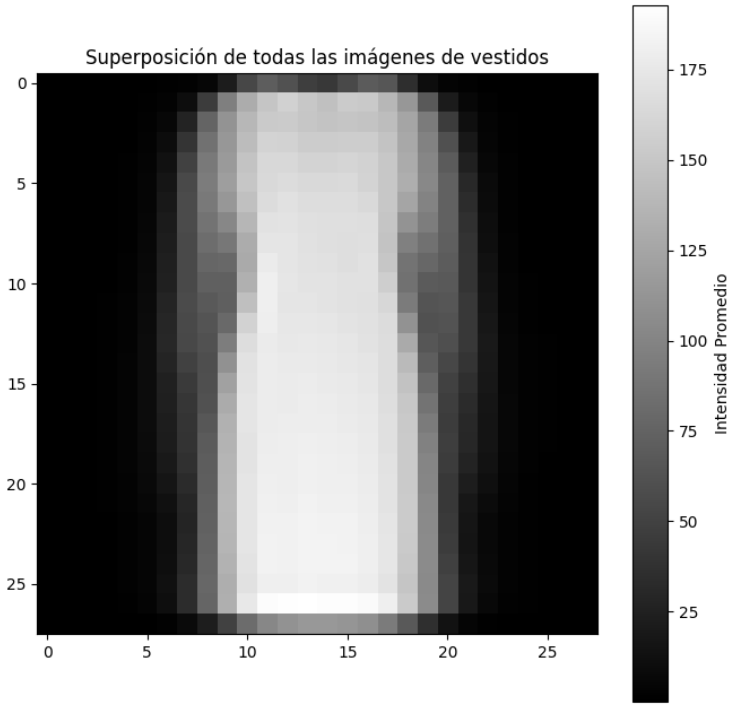


Gráfico 4: Imagen de 28x28
en la que cada pixel posee una oscuridad promedio
correspondiente a los valores de la clase 3.

Observando el gráfico 4, podemos concluir que las imágenes de vestidos parecen ser bastante similares entre sí.

La exploración de datos de nuestro dataset fue diferente a otras exploraciones realizadas, como por ejemplo a la de Titanic. Ya que este dataset se encontraba compuesto por imágenes de 28x28, en la que cada columna de nuestro dataset representa a un pixel. Lo que cambia con respecto al dataset de titanic es que cada pixel está relacionado con su píxel vecino, por lo que las columnas son dependientes una de la otra, mientras que en el dataset de titanic la mayoría de las columnas eran variables independientes. Además en nuestro dataset hay columnas en la que los datos se mantienen constantes, como vimos en el gráfico 2, hecho que no ocurría en el dataset de titanic.

Clasificacion binaria (2 clases)

El siguiente problema al que nos enfrentamos es responder a la pregunta de si una prenda corresponde a una remera o a un pantalón, para ello, seguimos los siguientes pasos:

a)Primero, generamos el data frame que contenga solo las remeras y los pantalones, es decir, seleccionamos la filas en las cuales el atributo label era 1 o 0.
Luego, dividimos nuestros datos en test y train para continuar con la consigna.

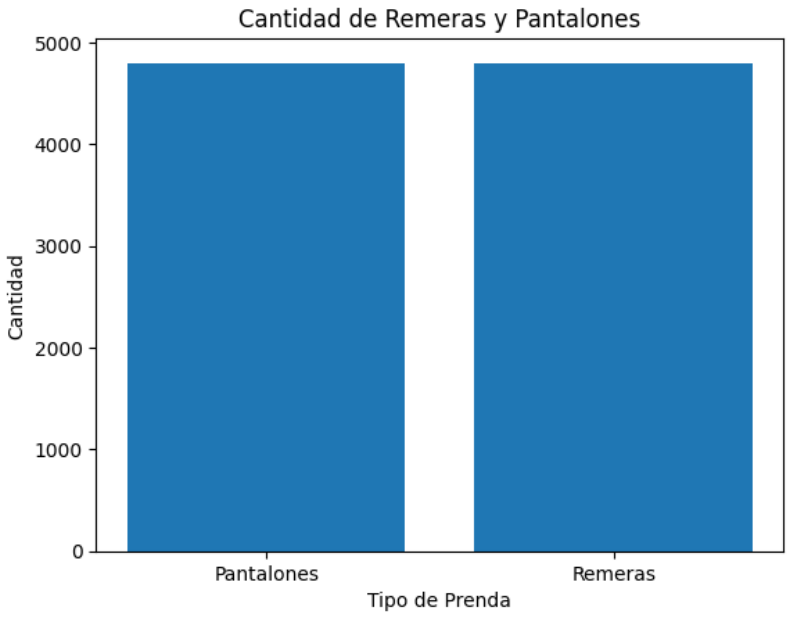


Gráfico 5: Histograma
de cantidad de imágenes de remeras y pantalones
en nuestro conjunto de datos de entrenamiento.

b) Analizando el gráfico anterior, determinamos que nuestro conjunto que previamente destinamos a train, está balanceado con respecto a las variables que queremos predecir. Es decir, hay una cantidad similar de pantalones y remeras en nuestro conjunto de train.

c)
Ordenamos los pixeles por su varianza, y elegimos los 3 tres primeros. Si generamos un modelo **KNN con estos 3 atributos logramos un $R^2 = 0.955$**, el cual está muy bien (cuanto más cercano a 1 mejor), y es muy cercano al R^2 del **modelo KNN generado con todos los atributos disponibles, cuyo $R^2 = 0.994$**.

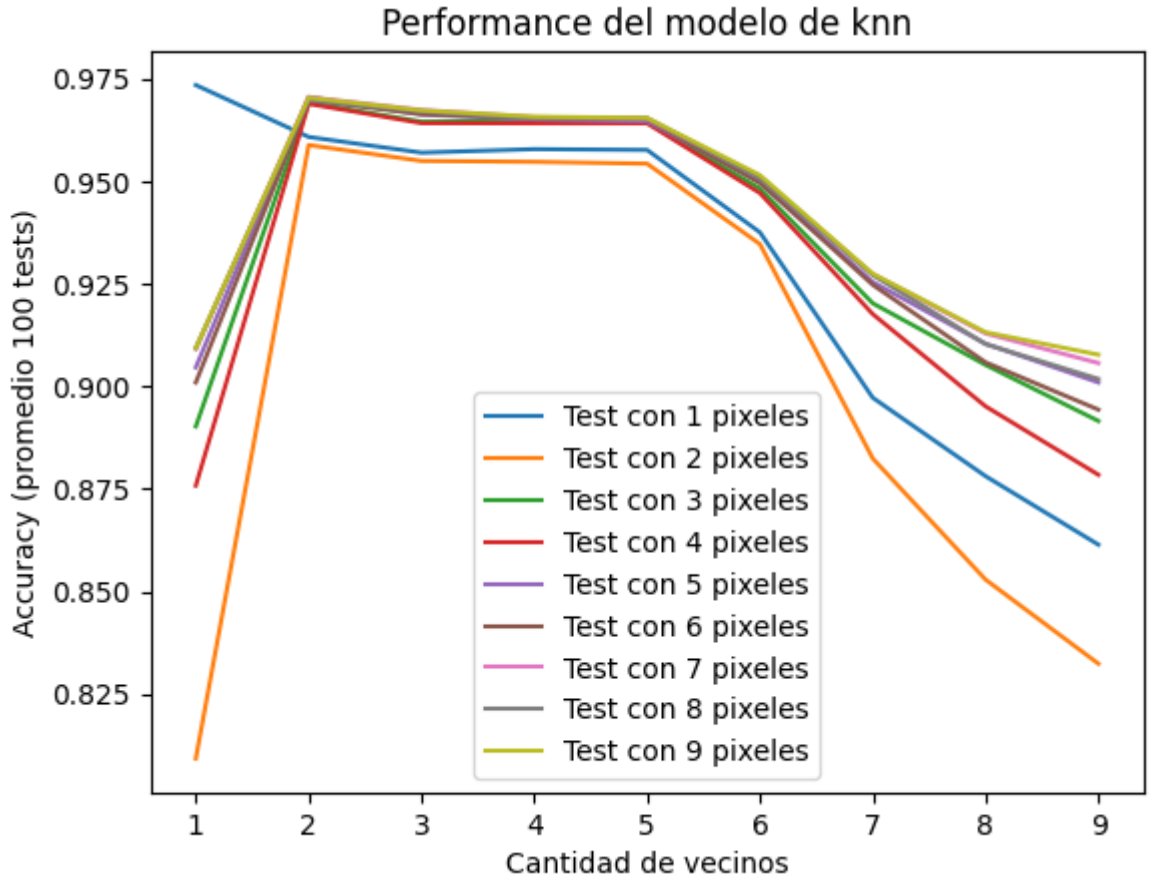
También vemos que si elegimos los siguientes 3 atributos con mayor varianza, el modelo generado por esta tripla es peor. El $R^2=0.94$

Sin embargo, comparando todas las triplas posibles, encontramos otras triplas de pixeles que si bien tienen menos varianza/desviación estándar, tienen una mejor performance. No sabemos por qué sucede esto.

Al ajustar modelos KNN con los datos de los tres pixels con mayor varianza y la mejor tripla que obtuvimos anteriormente, pudimos ver que ambos modelos tienen una buena performance ya que poseen una accuracy de 0.96, valor muy cercano a la accuracy de un modelo que toma en cuenta todos los atributos con una accuracy de 0.99.

Calculamos la accuracy tomando en cuenta nuestros datos del conjunto de test y analizando cómo funcionaban nuestros modelos en datos que no eran los de entrenamiento.

d)



En este gráfico podemos observar los resultados de un pequeño experimento que realizamos. Vemos 2 cosas interesantes:

1. Que cuando hacemos el test de accuracy sobre nuestro **modelo de predicción KNN con 1 solo vecino y entrenado con 1 solo atributo** es el que tiene **mejores resultados**.
2. **Para el resto de casos**, exceptuando el modelo entrenado con 1 y 2 atributos, **los mejores resultados son** obtenidos cuando comparamos **con los 2 vecinos más cercanos**. Hay una gran mejora frente a 1 solo vecino, pero luego el rendimiento disminuye a medida que aumentamos los vecinos que comparamos. También **vemos que casi no hay diferencia entre usar 3 o más atributos**, y esto se empieza a notar más a partir de los 7 vecinos más cercanos.

Clasificación multiclase

La parte final de nuestro trabajo es responder a la siguiente pregunta: Dada una imágen, ¿a qué tipo de prenda corresponde?.

Para responder esta pregunta diseñaremos un modelo de árbol de decisión.

Nuestro primer paso a realizar es dividir el dataset correspondiente al conjunto de todas las prendas en train y test. Los datos de train los utilizaremos para entrenar el modelo y los datos de test los utilizaremos para medir la performance de nuestro modelo.

El siguiente paso para el diseño de nuestro modelo es la elección de los mejores parámetros. Para ello utilizaremos cross validation para comparar alturas de nuestro árbol en un rango de 1 a 45 y diferentes criterios (gini y entropía).

Utilizando Random Search, que compara utilizando cross validation distintas combinaciones de hiper parámetros y nos da aquella que tiene mejor accuracy en cross validation, obtuvimos que los mejores parámetros para nuestro árbol eran: altura = 14 y entropía como criterio utilizado.

Una vez que obtuvimos los mejores hiper parámetros para nuestro árbol, realizamos Cross Validation con 5 folds. y obtuvimos los siguientes resultados para cada fold:

- D5: 0.81208333
- D4: 0.8009375
- D3: 0.80708333
- D2: 0.8084375
- D1: 0.809375

Observamos una accuracy mayor al 80% en todos los folds, lo que evidenció una buena performance en nuestro modelo y confirmó la buena elección de nuestros parámetros en el paso anterior.
Por último evaluamos utilizamos otras métricas como precision, recall y f1-score para evaluar el desarrollo de nuestro modelo para cada tipo de prenda. Además para tener una métrica general de nuestro modelo observamos el accuracy y calculamos el promedio no-ponderado de precision, recall y f1-score para todas las clases.

Tipo de prenda	precision	recall	f1-score	Cantidad de muestras
Remera	0.73	0.77	0.75	1150
Pantalón	0.96	0.94	0.95	1191
Pullover	0.71	0.68	0.69	1249
Vestido	0.82	0.83	0.82	1217
Saco	0.67	0.74	0.71	1144
Sandalias	0.91	0.87	0.89	1215
Camisa	0.59	0.53	0.56	1224
Zapatillas	0.85	0.90	0.87	1144
Bolso	0.92	0.94	0.93	1217
Tacos	0.91	0.89	0.90	1249

En la tabla anterior podemos observar como nuestro modelo predice mejor algunos tipos de prendas como pantalones, con una precisión del 95% y tiende a tener más error en predicciones de otro tipo de prendas como los sacos, con una precisión del 67%.

Accuracy	0.81
Precisión (promedio)	0.81
Recall (promedio)	0.81
f1-score (promedio)	0.81

Concluimos que estas medidas son bastante buenas y que **el mejor modelo es el árbol de decisión con altura 14 y criterio entropía.**