DEPARTAMENTO DE COMPUTACION

Facultad de Ciencias Exactas y Naturales - UBA

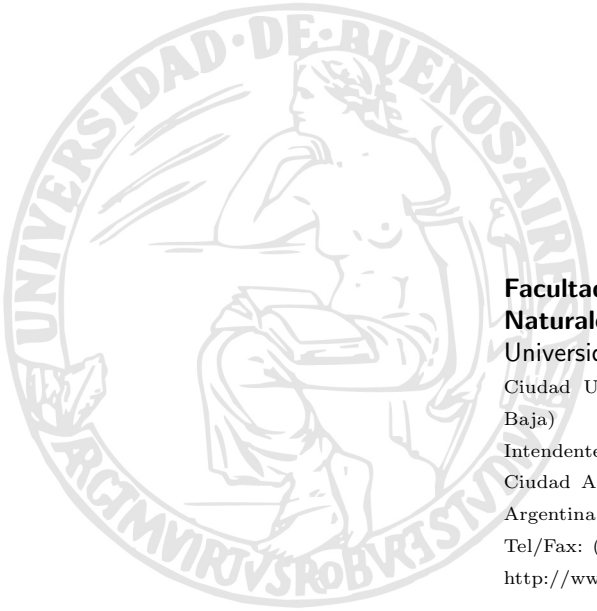
TP2 - Laboratorio de Datos

Fashion - Mnist

Laboratorio de Datos

GLA

Integrante	LU	Correo electrónico
Russo, Agustin	39/23	agus.drum12@gmail.com
Alonso, Lucas	897/23	la1247890@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

Contents

1 Introducción

1.1 Sobre el TP

En este Trabajo Practico trabajamos particularmente la realizacion de 2 modelos de aprendizaje automatico, en este caso ambos supervisados (Arbol de Clasificacion, KNN), donde el objetivo era poder conseguir un modelo de clasificacion que nos permitiera predecir segun una imagen a que clase de ropa corresponde. Tambien evaluar dichos modelos para confirmarnos que son sufiecientemente precisos a la hora de clasificar. Para eso usamos distintas tecnicas como K-folding, Cross Validation, afinacion de los hiperparametros,etc.

1.2 Caracteristicas del Dataset

El Dataset propuesto por la catedra, `fashion-mnist.csv`, cuenta con 60.000 imagenes representadas cada una como filas y excepto la primer columna el resto corresponden a un pixel, siendo en total 784 pixels, representando finalemente una imagen de 28x28. Cada columna de pixel tiene un valor de 0 a 255 correspondiente a la escala de grises, siendo 0 = totalmete negro y 255 = totalmene blanco. La primera columna esta reservada para el id de la clase de ropa que corresponde la imagen en este orden:

- 0 T-shirt/top
- 1 Trouser
- 2 Pullover
- 3 Dress
- 4 Coat
- 5 Sandal
- 6 Shirt
- 7 Sneaker
- 8 Bag
- 9 Ankle boot

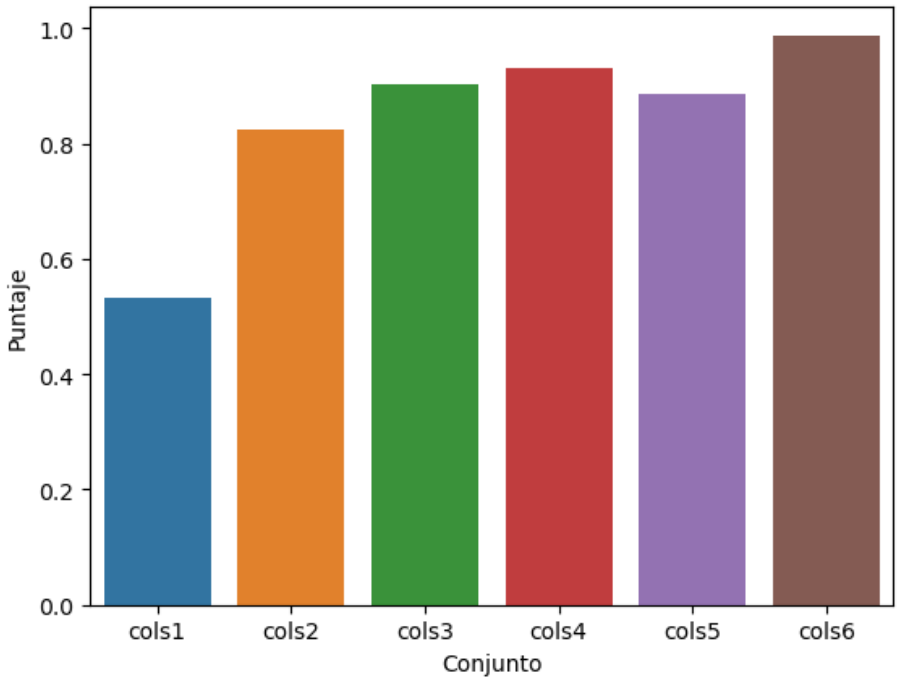
Este primer set de datos es el que tomaremos como el dataset de entrenamiento y validacion de los modelos, que nos permitiran testearlos y calibrarlos. Para evaluarlos tomamos el dataset para testeo que ofrece kaggle.com, a forma de 'holdout' con el cual haremos las predicciones finales.

2 Analisis exploratorio - Ej 1

2.1 Atributos relevantes

En un principio creiamos que al tratarse de imagenes, todos los atributos serian relevantes para la prediccion, pero al ir realizando los puntos siguientes muchas veces, por ejemplo, con conjuntos distintos de 3 atributos conseguiamos una precision aceptable. Nos dimos cuenta tambien que la la relacion entre los pixeles elegidos tambien mejoraban la precision segun como estaban ubicados; por ejemplo, tres pixeles pegados uno al lado del otro en una esquna aportan muy

poca informacion sobre la imagen en comparacion con una diagonal de 4 pixeles equidistantes uno del otro (ver A). Sin embargo no esta de mas decir que cuando se utilizan todos los pixels la precision del modelo aumenta considerablemente, llegado hasta 98% de precision.



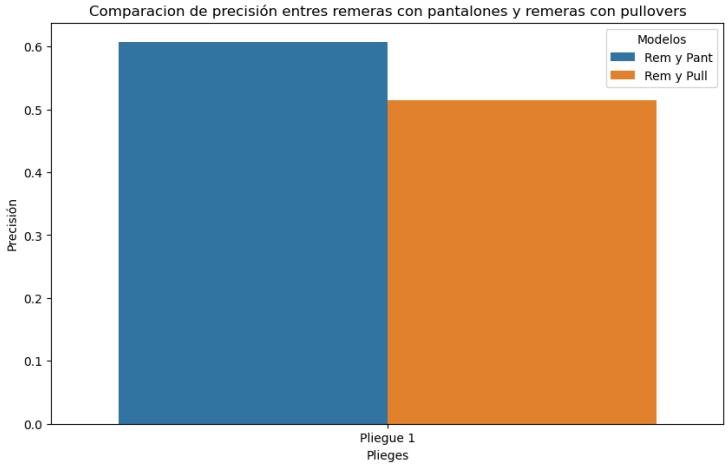
1) *En este grafico se muestran los puntaejes obtenidos segun los conjuntos de atributos dados para el entrenamiento de un modelo KNN siendo:* ***cols1****: los primeros 3 pixels de la esquina izquierda,* ***cols2****: recta vertical en el medio de la imagen,* ***cols3****: diagonal desde esquina superior izquierda hasta esquina inferior derecha,* ***cols4****: diagonal con 1 pixel mas,* ***cols5****: recta vertical con 1 pixel mas,* ***cols6****: todos los pixels*

2.2 Diferencia entre prendas

Armando los modelos de aprendizaje como el KNN, comparamos dos situaciones para corroborar si el modelo tenia mas dificultades en distinguir entre prendas "similares", como una remera y un pullover, que con prendas "distintas" como una remera y un pantalon. Al comparar nos damos cuenta de que sí se le dificulta distinguir segun las prendas que se comparen. Por ejemplo:

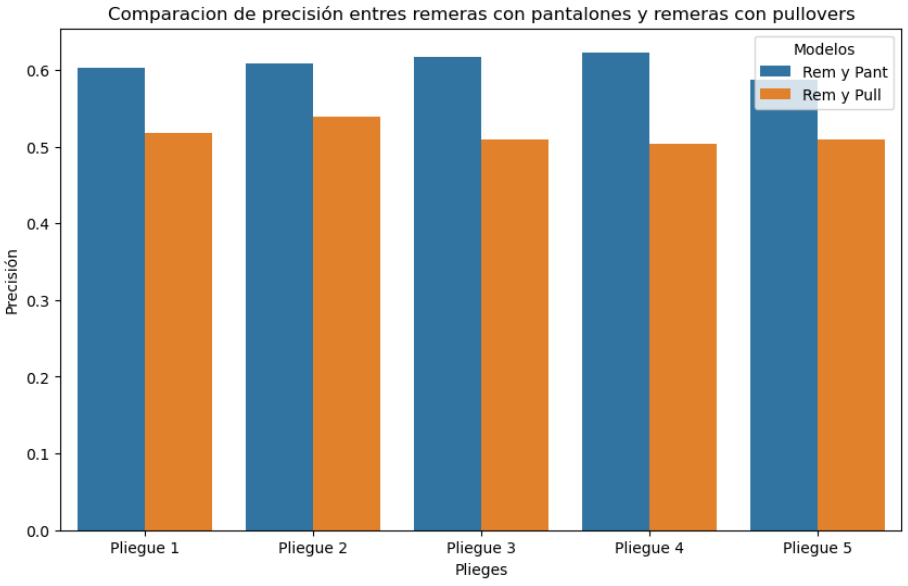
- Al comparar remeras y pantalones (mas distintas) obtuvimos una precision en promedio de: 0.6076

- Y al comparar entre remeras y pullovers (mas parecidas) obtuvimos una presicion en promedio de: 0.5154



2) *Aqui mostramos el puntaje promedio de cada modelo segun su set de datos*

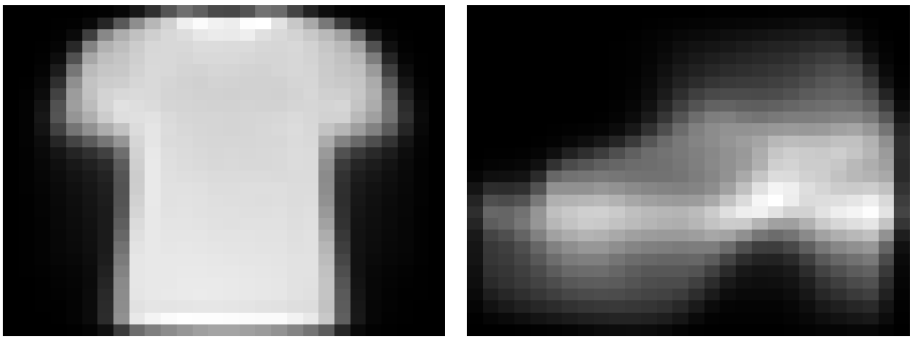
y ademas obtuvimos que le va mejor en cada pliegue haciendo validacion cruzada



3) *Grafico que muestra la precision obtenida sobre cada pliegue del set de datos segun las prendas que se comparen (en el codigo esta mas explicado).*

2.3 Similitud entre prendas

Usando la funcion que armamos `promedioImagen()`, pudimos definir, de manera visual, que las prendas que presentan una imagen promedio mas "definida" en cuanto al contorno de la imagen son las que mas presentan similitudes o varian menos (por ejemplo las remeras). Mientras que las mas "borrosas" son las que mas presentan diferencias entre ellas (por ejemplo las sandalias)



a) *Imagen promedio de Remera* **b)** *Imegen promedio Sandalias*

2.4 Diferencia entre tipos de datos

Al trabajar con un dataset compuesto por imagenes, nos encontramos con que las principales diferencias son:

a) La dimensionalidad de los datos: en este dataset nos manejamos con 784 columnas que representan solamente a los pixeles de la imagen, que a diferencia de otros dataset con los que trabajamos que se constituyen de menos cantidades de columnas.

b) Los tipos de datos: las imagenes prensentadas como columnas de pixeles son variables unicamente numericas, representando el grado de luminosidad del pixel. Otros datasets presentan diversos tipos de datos, tanto categoricos como numericos. Por lo que las condiciones de clasificacion se van a ver afectados por distintas variables.

Por lo que creemos que no presenta una dificultad considerable frente a los otros dataset sino mas bien un acercamiento distinto a la hora de explorar los datos.

3 Experimentos

3.1 Funciones

En el proceso de explorar el dataset y realizar los ejercicios fuimos construyendo distintas funciones que nos permitieron manipular los datos y testear modelos

de una forma mas practica. Las funciones creadas son:

- `mostrarImagen`: devuelve una imagen dentro de la clase seleccionada.
- `imagenPromedio`: devuelve la imagen promedio de una clase.
- `graficarScores`: grafica los resultados entre 2 modelos, segun la validacion cruzada de cada uno dado cierto k para el K-folding.
- `validacionCruzada`: realiza un K-folding de k pliegues y realiza una validacion cruzada y nos devulve un array con los puntajes de precision de cada test hecho a partir de cada pligue.
- `testarModelo`: recibe un modelo y un set X e Y, donde divide el set de training y el set de testing, y nos devuelve la precision de una prediccion.
- `calibrarModelo`: recibe un modelo (que en este caso es un `GridSearchCV()` ya configurada y el set de datos X e Y, y nos devuelve el parametro optimo segun el dataset, y la precision que consigue.

3.2 Evaluacion de Modelos

A traves de los ejerciocios dados para resolver, los modelos tanto el KNN como el Arbol de Clasificacion fueron entrenados y evaluados con el dataset original dado por la catedra. En la seccion **Conclusiones** los resultados mostrados fueron obtenidos a traves del dataset de testeo `fashion-minst_test.csv` que se encuentra en la pagina kaggle.com. El dataset tiene 10000 muestras y lo utilizamos a forma de holdout para tener un set de datos que el modelo nunca haya visto.

4 Conclusiones

Para finalizar mostramos los testeos tanto del KNN como del Arbol de Clasificacion:

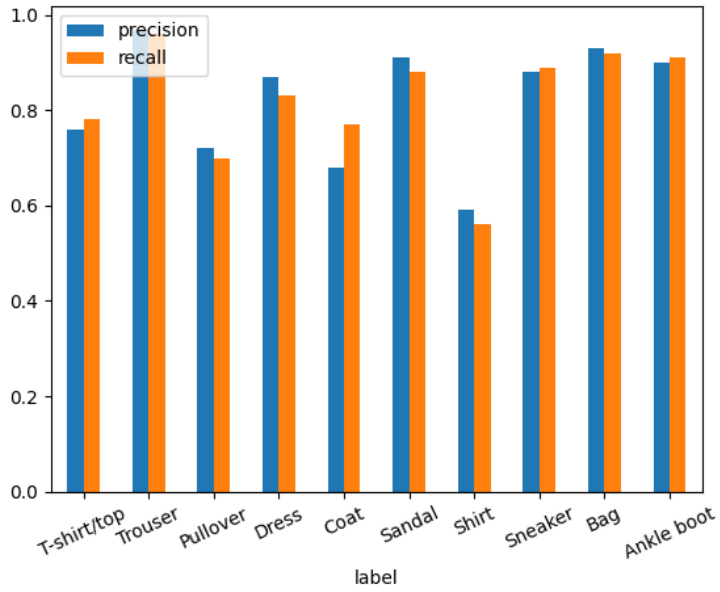
4.1 KNN

El modelo KNN utilizado para el punto 2 del TP fue entrenado con varios conjuntos del dataset de entrenamiento para conseguir el conjunto que mejor precision generaba, dejando de lado el conjunto de todas las columnas de pixeles, decidimos quedarnos con el conjunto de 4 pixeles equisdistante de la diagonal que fue el segundo mejor dandonos una precision de: 0.93125. Luego para calibrar los hiperparametros del modelo se uso el metodo `GridSearchCV()` donde para ese conjunto de datos el mejor numero de vecinos fue de $k = 12$. Con lo que mejoramos la presicion a: 0.9383. Para finalizar, ya con los hiperparametros y el dataset de entrenamiento seleccionados, entrenamos el KNN con $k = 12$ con el set de datos `Xd` e hicimos una prediccion con el `X_test` y obtuvimos una presicion final de: 0.9385.

4.2 Arbol de Clasificacion

Los parametros del modelo fueron elegidos usando el metodo `RandomizedSearchCV()`, en el que los mejores parametros fueron profundidad maxima 13 y criterio "*gini*". Luego de encontrar el mejor modelo y entrenarlo, pasamos a las predicciones finales. En general, el modelo de árboles de clasificación tiene una precisión alta para el formato de datos utilizado. Sin embargo, hay algunas prendas que son difíciles de distinguir, como las remeras y los pullovers, o los sacos y las camisas. En estos casos, la precisión del modelo se reduce al 60%.

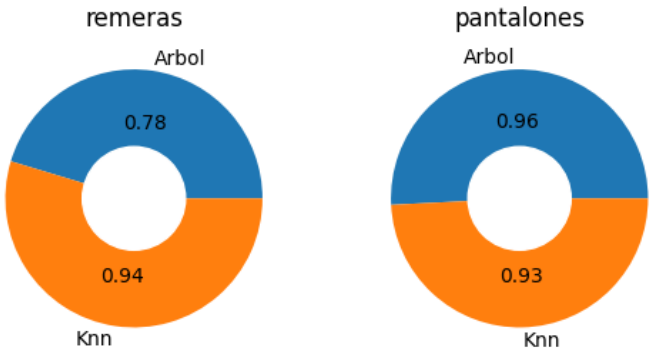
Si se aislan estas cuatro prendas, el modelo tiene una precisión minima del 85%. Esto sugiere que el modelo es capaz de aprender a distinguir entre estas prendas cuando se les proporciona un conjunto de datos específico.



4) *Aqui mostramos las diferencias en precision y recall en cada clase de prenda que obtuvo el arbol*

4.3 Comparación de Modelos

Para terminar graficamos los resultados del modelo KNN y los resultados del modelo de arbol de decision distinguiendo entre remeras y pantalones. Es importante aclarar que ambos modelos fueron entrenados de formas distintas, ya que el modelo de arbol de decisiones fue entrenado utilizando todo el dataset. Y el modelo de KNN, a diferencia del modelo de arboles de decision, fue entrenado solo con remeras y pantalones dando un mejor desempeño en remeras.



5) *Comparacion de la medida recall de los modelos distinguiendo entre pantalones y remeras*