

**INDICE**

Javi: Lo hago yo cuando esté todo terminado.

# **INTRODUCCIÓ A LA PRACTICA**

Sens planteja resoldre un problema simple d’etiquetatge d’imatges. Donat un conjunt d’imatges d’un catàleg de roba desenvoluparíem els algorismes que permetessin aprendre a etiquetar automàticament imatges per tipus de peça i per color. Com que el problema podia ser molt complex es va limitar a paraules en angles, 8 tipus de roba i 11 colors basics a més de treballar amb imatges de baixa resolució.

Es plantegen 3 problemes a resoldre:

* **Etiquetatge automàtic de color (no supervisat)**

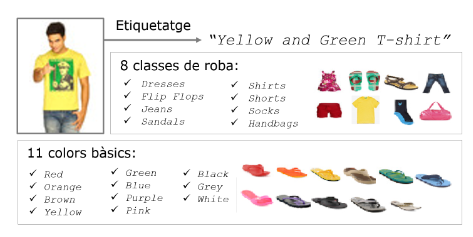
Es va decidir realitzar un agrupament no supervisat de punts utilitzant l’algorisme *K-means,* on es van implementar 10 funcions en les quals destacaven: **[DECIDIR CUALES PONER, PASO DE PONER 10 XD]**

* **Etiquetatge automàtic de forma (supervisat)**

Per a organitzar el tipus de roba, es va decidir utilitzar l’algorisme *KNN,* on es van implementar 3 funcions per a agrupar-les en classes.

* **Implementació del cercador**

Una vegada els mètodes son implementats es combinarien per a crear diferents test d’anàlisis i arribar a tenir una interfície similar a la d’una botiga online.



A continuació es parla amb més detall de cada una d’aquests problemes.

# **METODES D’ANALISI IMPLEMENTATS**

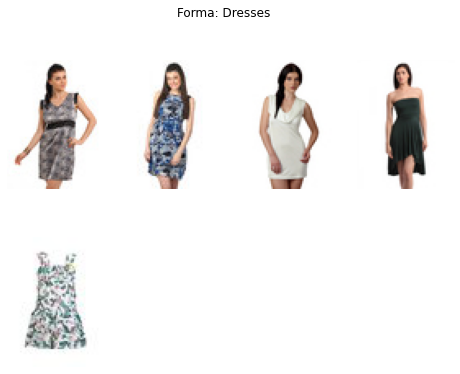
## **METODES D’ANALISIS QUALITATIUS**

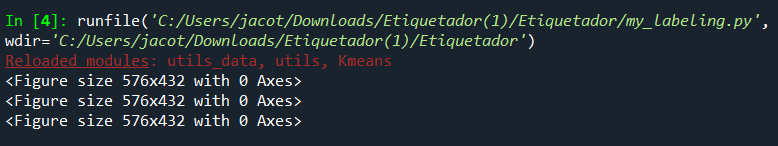
Els mètodes d’anàlisis qualitatius son funcions que ens permeten avaluar d’una manera visual els nostres classificadors. Sens va donar tres funcions a fer: Retrieval\_by\_color, Retrieval\_by\_shape i Retrival\_combined, que retornarien les imatges que contenen les etiquetes de la pregunta que passem per paràmetre.

Vaig començar per ordre amb la funció *‘Retrieval\_by\_color’*, fent un esborrany en *Spyder* i no va costar molt, arribant a un resultat que podria ser el correcte, ja que quan ho vaig fer era incapaç de saber amb exactitud si la meva funció era favorable o no. Com el professor ens va dir que si teníem una funció, teníem les tres, vaig llegir els altres enunciats de les funcions, trobant una diferencia en cada una d’elles:

* **Retrieval\_by\_color:** Vaig posar: ‘if pregunta in clase:’, perquè si hagués posat un == , al tractar-se d’una cadena, aquesta hauria fallat.
* **Retrieval\_by\_shape:** Al contrari que la funció anterior, aquesta si era necessària un == al tractar-se d’una sola etiqueta única.
* **Retrieval\_combined:** Com el seu nom indica, vaig combinar les dos funcions en una sola, amb dos condicions (forma i color) dins un bucle for, sense saber que hi havia un error (que solucionaria dies més tard.

Les funcions no van suposar un problema. Per comprovar que funcionés, vaig decidir fer els meus propis tests per a cada una de les tres funcions. Per fer-ho, vaig informar-me una mica de les funcions a utils\_data, llegit atentament el pdf de presentació de la part i inspiració tomada dels tests proporcionats pels professors en TestCases\_kmeans y knn. Agafant test\_color\_labels, Retrival by color/shape van donar un resultat correcte en base al que se li demana.



El problema va ser amb Retrieval Combined el qual no mostrava cap imatge i es mostrava aquest missatge a la terminal.

El problema es trobava en la condició del color, que només retorna les imatges correctes si es posa amb in, al contrari que la seva part per separat.



En un principi, els resultats han donat correctes amb la primera implementació. Pensava que tot estava acabat, però se’m va comentar que havia d’implementar el meu Kmeans i Knn en comptes dels que ja hem donaven.

Vaig crear una nova funció per tant de crear una llista amb les imatges de test\_imgs parcejades per KMeans, inicialitzant i cridant a fit(), per retornar els colors. Fer això fa que el codi vagi mes lent(1).

Comentarios propios:

Si hay tiempo, añadir a estas funciones el calculo del%

Dudas para el profesor:

1. El programa tiene que tardar menos de 3 segundos? O en este caso puede tardar más? Lo he hecho con un bucle for y estoy intentandolo hacer con funciones numpy para tratar de disminuir el tiempo de ejecución.

Habria que hacer lo mismo del Kmeans para el Knn (algo tipo test\_class\_labels)?

## **METODES D’ANALISIS QUANTITATIUS**

Se encarga Lucas

# **MILLORES SOBRE EL KMEANS I KNN**

*Al pdf de la explicació de la part es mencionen les següents millores del Kmeans i el KNN:*

* *No només inicialitzar la funció init\_centroids amb ‘first’, sinó també amb:*
  + *Random: Inicialitzar els centroides assignant-lis K punts aleatoris de la imatge.*
  + *Custom: L’usuari escolleig una estratègia d’inicialització.*
* *A més de tenir implementat WithinClassDistance, també fer:*
  + *InterClassDistance, que tindrà el paràmetre d’entrada self i en comptes de fer els càlculs per calcular la distancia entre la mateixa classe es faran els de calcular la distancia entre diferents classes.*
  + *CoeficientFisher, que calcularà la distancia de Fisher.*

*Pensem que tindran una estructura similar, però els seus càlculs seran diferents a WCD.*

* *El KNN utilitza la distancia Euclidiana, per tant es vol canviar això. En teoria el resultat serà diferent, però millor respecte a l’anterior ja que s’utilitzarà una distancia més precisa que la que s’utilitza ara. En teoria.*

*Segurament només implementarem 3 d’aquestes 4 millores.*

*Ja hem modificat una: El llindar de find\_best\_K. Diferent de 20, només funcionen els llindar de 19 o 21,*

Dudas para el profe:

¿Como se puede comprovar que find\_best\_K con diferente llindar da un major resultado? ¿Ejecutando el TestCases\_Kmeans o my\_labeling? ¿Si es en my\_labeling como lo podemos comprovar? Es que de Kmeans solo llamamos a fit() y find\_best\_k no se encuentra dentro.

# **CONCLUSIÓ**

# **BIBLIOGRAFIA**

PDF.Introduccio.Proj2