Products' Classification Engine with text & image *feasibility study*

Goal of "Place de Marché" is to enhance users' experience
sorting products ***reliably*** in categories
through a ***scaled*** and ***automated*** products classification engine

Project's scope:
Feasibility study expectations

1. Global EDA : inputs & target analysis
2. Feature Engineering : process text // image
3. Feature Engineering : create {BoW, BoV}
4. Reduce & Combine
5. Classify
6. Evaluate & Optimize

Optimize: Only *final evaluation* matters

Key points

- Scaling
- Alternate… processors
- extractors
- reducers
- Illustration ≠ Evaluation

Robustness & Scope

Classify

Expected decisions :

Feasibility study **validation** Y N

Engage next steps Y N

Feasibility study contents

1. Global EDA : inputs & target analysis
2. Feature Engineering : process text // image
3. Feature Engineering : create features {BoW, BoVW}
4. Reduce & Combine
5. **Cluster and visualize**
6.

P6_01_Notebook

jupyter

*P6_**01a**_Explorations_NB*

jupyter

*P6_**01b**_CNN_Dedicated_NB*

jupyter

***.csv files for intermediate results**

Global EDA

- **Inputs :**
  - One *.csv files with **1050 products descriptions** & image filename
  - **1050 jpg files**
- **Target :**
  - Category : *product_category_tree*
- **14 features:**
  - Among 6 *possible* primary keys : keep "**image**" and use *df index*
  - Remove 10 useless features
  - Build an extended description (brand, name, description)
  - Filling by ' ' the 32 % missing "brand" information
- Consistency checked between 'image' & image files folder

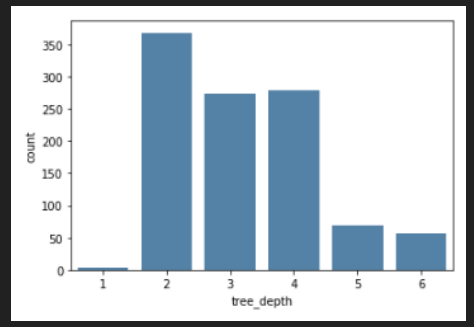
uniq_id
crawl_timestamp
product_url
product_name
product_category_tree
pid
retail_price
discounted_price
image
is_FK_Advantage_product
description
product_rating
overall_rating
brand
product_specifications

Global EDA: Products Categories

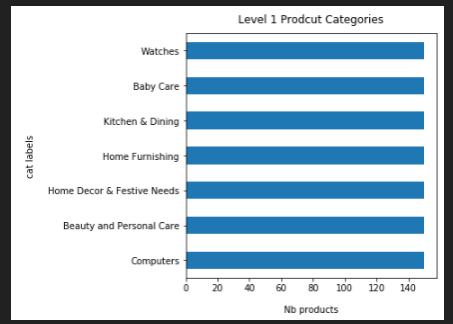
Clip or Collect ?

- Pattern is: *1st level >> 2nd level >> … >> n level >> concat(Product_name, ProductDescription) ending with '…'*
- Extraction of tree depth, level's labels & observation of products count's balance :

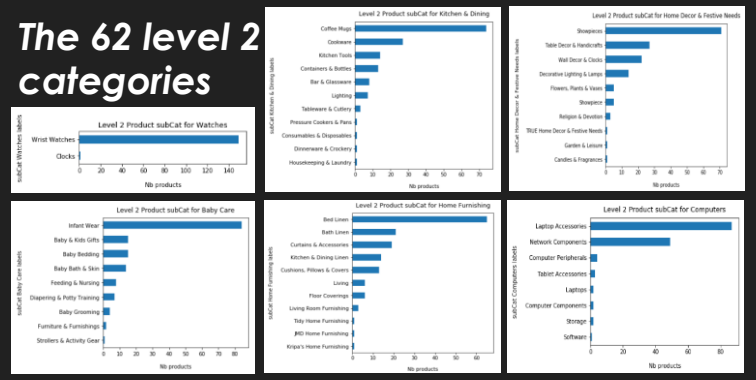
Mostly in 2 to 4 levels



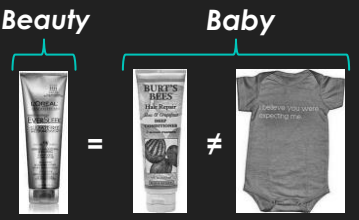
The 7 level 1 categories



The 62 level 2 categories



- **Key point:**
  - Remedy **unbalanced classes,**
    - *e.g. with similar amount of similar products at a given category level*
    - Dealing with **cross category similarity** and **internal category dissimilarity**



- **Alternate approaches**:
  - Either ***clip*** data to same-sized classes (e.g. by sampling), but **not enough inputs**,
  - Or ***collect*** additional descriptions and images e.g. through an API : **purpose of the company's project.**

Feature Engineering : process text

Text processing : pre-process

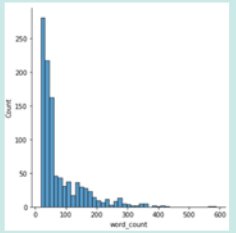
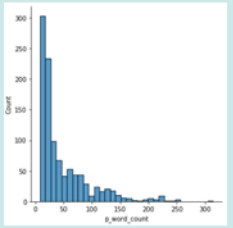
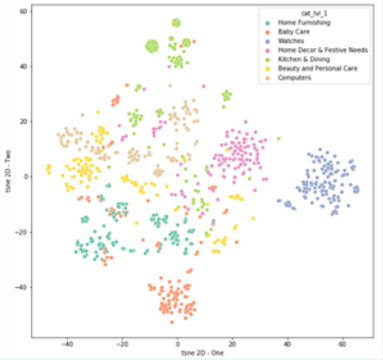
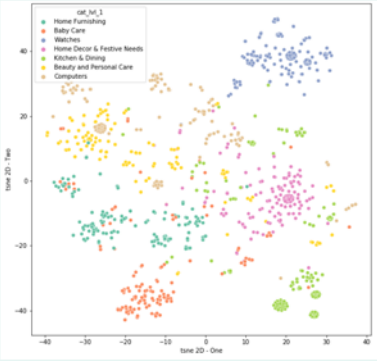
please refer to Explorations NB : section 2. jupyter

- Product's "**description**" is a string, with average **80 words count**.
- It appears being a concatenation of many inputs, with repeated info and details such as size, price, etc.
- To get an adequate corpus, we pre-process text :

1. Switching to lower case,
2. Getting rid of punctuation, numbers, < 3 words length,
3. Tokenizing
4. Stemming
5. Removing **default's stop words** & **custom's** (iteratively)

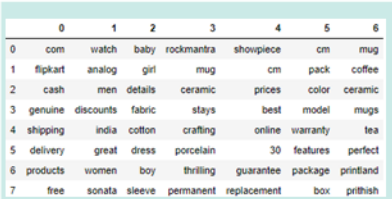
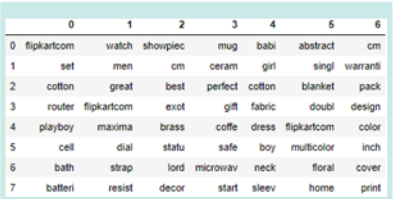
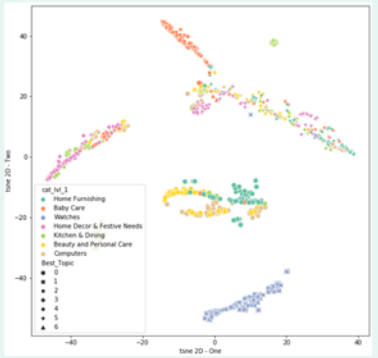
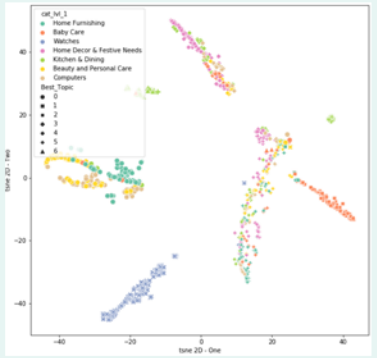
- Result is a **processed_description** for each product

- We compute **tf-idf matrix**, illustrated through t-sne 2D projections

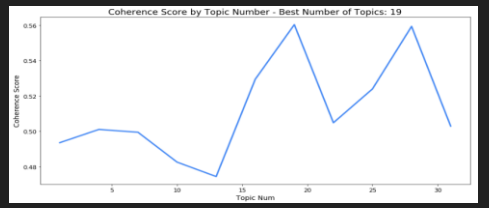
	"raw"	pre-processed
Word count distribution		
Rough t-sne 2D projection *Caution : Neither classified nor clustered & stochastic*		
Vocabulary size	**2442**	**1242**

Text processing : Bag of Words

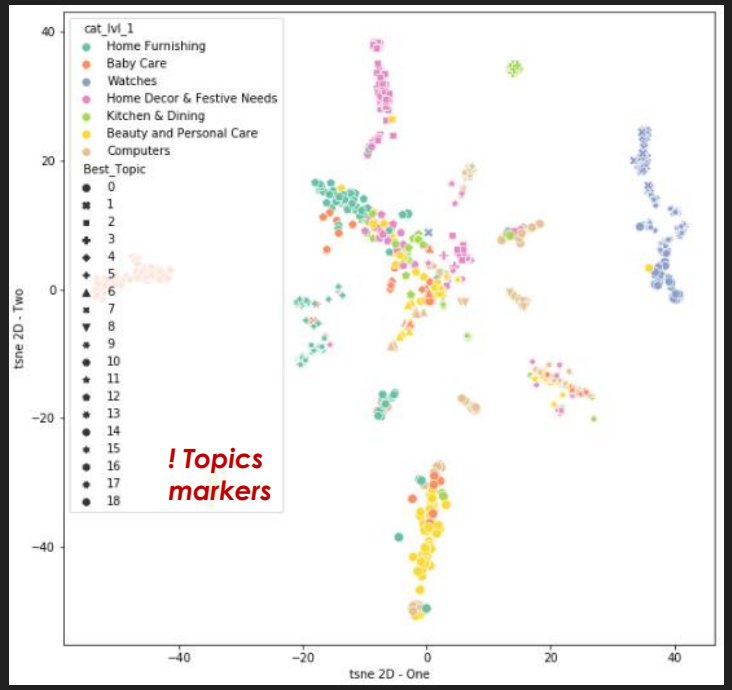
- **NMF with "as few" topics as lvl 1 categories**:

NMF approach	"raw"	processed
Topics table		
Rough t-sne 2D projection *Warnings : Neither classified nor clustered & stochastic*		

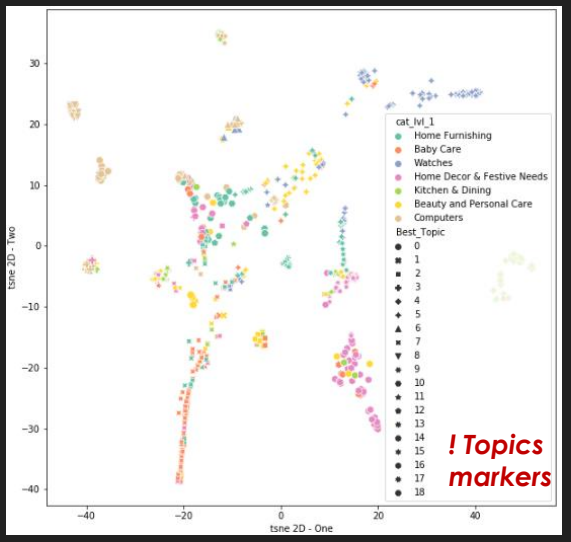
- **Coherence score** to get an optimal* topics numbers (NMF)



- **NMF with "optimal"* topics number:**
  - Reduces the size of **"carryall"** topic,
  - Highest **sparsity**, with few topics for a single 1st level category,
  - "Mixed topics & categories" area split into **smaller areas**,
  - Still a high level of confusion for products with "weak" coordinates.



- **LDA "optimal" as alternate approach**



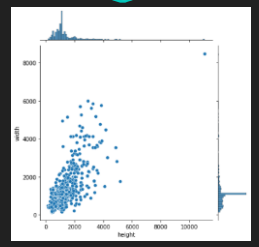
** **Caution** : optimization is only valid considering final classifier*

Feature Engineering : **process image**

Image processing: pre-process

please refer to Explorations NB : section 3. jupyter

Pictures of various sizes are stored in BGR mode



- **Computer** vision is not **Human** vision: we decided to **explore processors from a feature extraction and matching perspective (SIFT)**

1. Alternate processors → 2. Visualize results → 3. Explore* extraction & matching → 4. Create images set

- 1 -

Color space / Contrast / Brightness

Adjust size & keep ratio

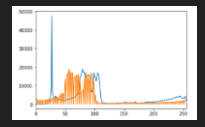
Enhanced Contrast (through L of LAB space)

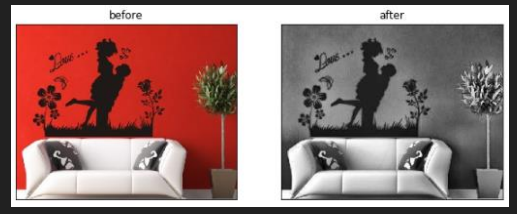
Emphasize shape (through binarize & dilate)

Get the area of interest?

- 2 -

Visualize results : before & after processing



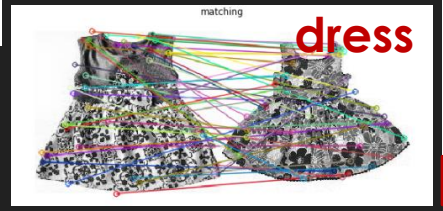


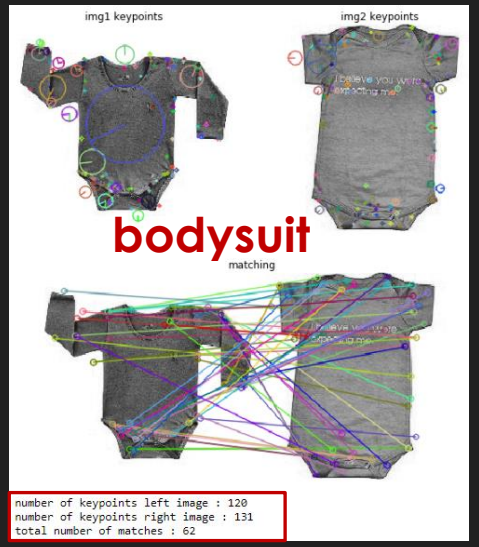
Guess what is the product here?

- 3 -

Explore* through samples SIFT features extraction & matching :

Keypoints and matching





- 4 -

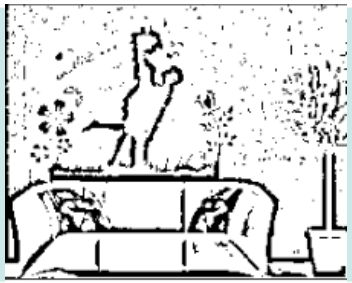
Process all images to fill folders of **image sets**

Feature Engineering : **process image**

Image processing: pre-process

please refer to Explorations NB : section 3. jupyter

Bodysuit - Dress details

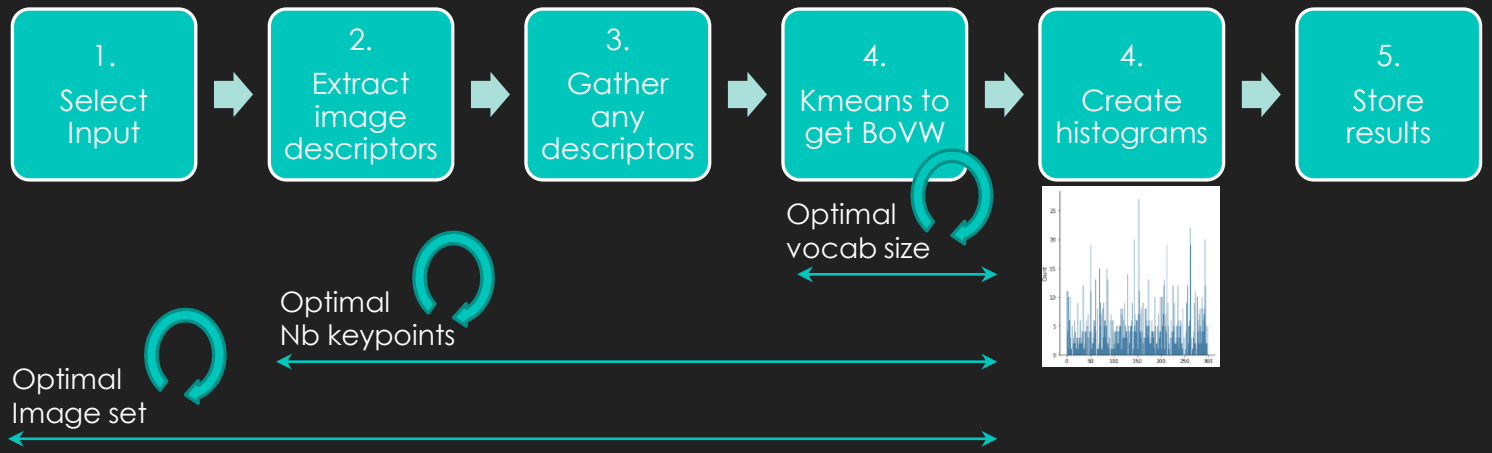
Processors	"raw"		resized		equalized		enhanced contrast		enh. contrast & resized		emphasize shape	
sample												
test	Bodysuit	Dress	Bodysuit	Dress	Bodysuit	Dress	Bodysuit	Dress	Bodysuit	Dress	Bodysuit	Dress
nb keypts	medium	massive	low	medium	medium	massive	high	massive	low	medium	high	massive
match rate	33 %	40%	44%	41%	28%	36%	45%	39%	49%	43%	48%	41%
Improve	baseline	baseline	++	same	-	-	++	same	+++	+	+++	same
50 keypts	*low*	*low*	*low*	*low*	*low*	*low*	*low*	*low*	*low*	*low*	*low*	*Low*
match rate	38%	22%	42%	44%	32%	30%	48%	50%	46%	36%	32%	52%
improve	+	---	++	+	same	--	+++	++	++	-	same	+++

- **matching rate** is "nb matching / avg(keypoints)" and trends observed **differs** from on sample to another.
- while **scale** shouldn't matter in SIFT case, keypoints total number **depends on it**.
- Introduces idea of a **minimum viable threshold** for keypoints number

** **Caution**: proceed statistically and consider classifier results to get valid results.*

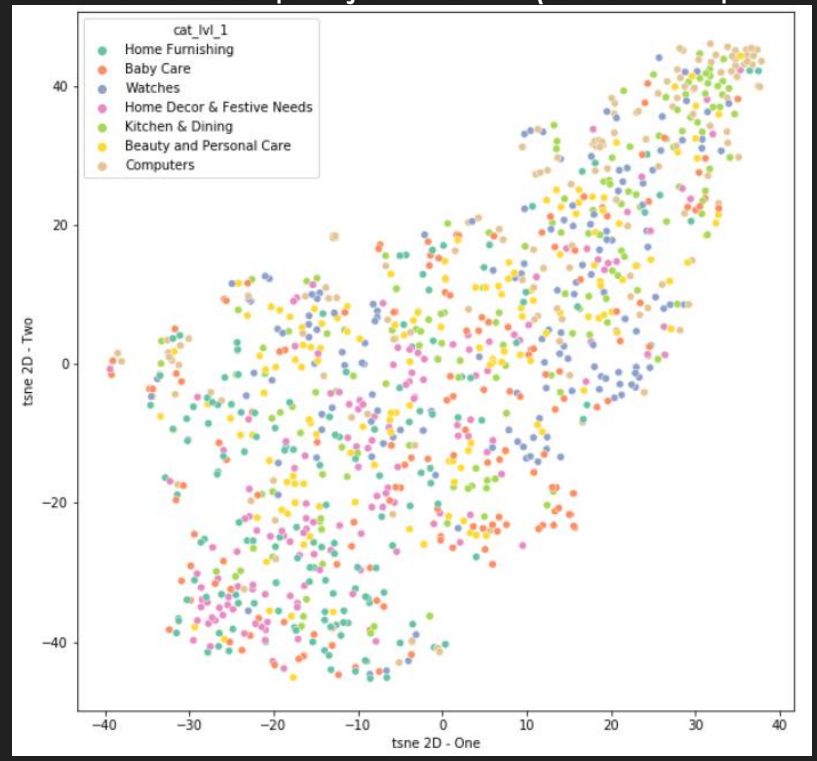
Image processing : Bags of Visual Words

- Again a process, with 3 interdependent steps : 1., 2. and 4.



- **Key points :** we identify 3 steps where an optimum could be searched :
  - Find an optimal image set
  - Find an optimal nb of keypoints
  - Find an optimal vocab size
- We proceed with :
  - Enhanced constrast images and medium size (for easy data handling)
  - 2 alternate image features :
    - **50 keypoints & 150 BoVW**
    - **"Free" keypoints & 300 BoVW**

- t-sne 2D projection* (case "optim")



With same rough t-sne parameters, we don't see a nice projection.

** **Caution**: consider classifier results to get valid results.*

Image processing : CNN transfer learning

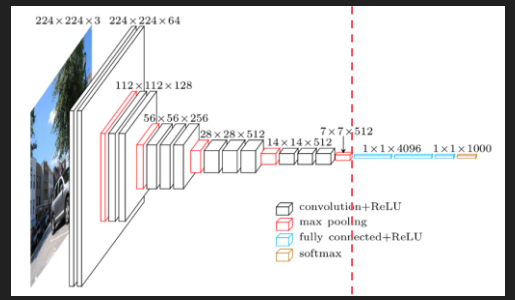
please refer to P6_01_CNN_Dedicated_NB.ipynb jupyter

1. Explore ability to **create features** from pre-trained VGG16 ImageNet model

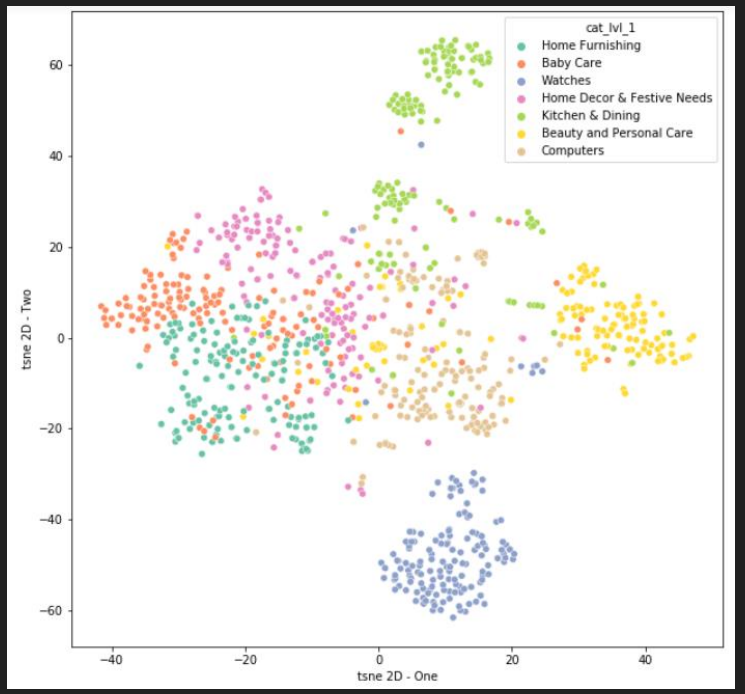
2. First, the full original CNN provides top classes for an image sample:

Top 3 classes reach 98%
44% Sweatshirt
35% Jersey
19% Bulletproof Vest !

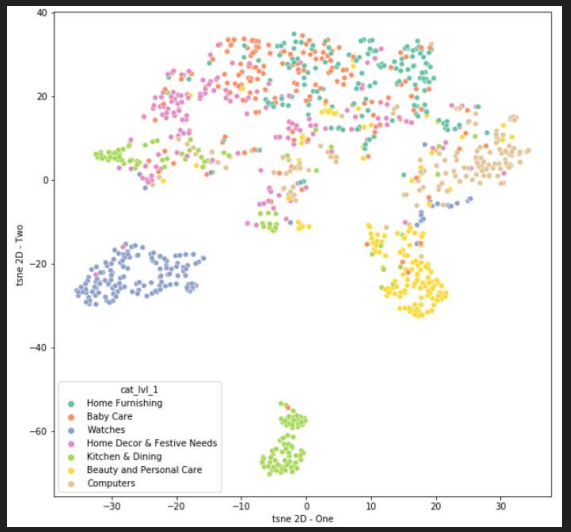
3. we cut at last pooling layer: goal is to get a flatten vector, here dim 512.



4. Make the usual t-sne 2D projection observations



Using rough t-sne after alternate dim reducers (nfm 7)



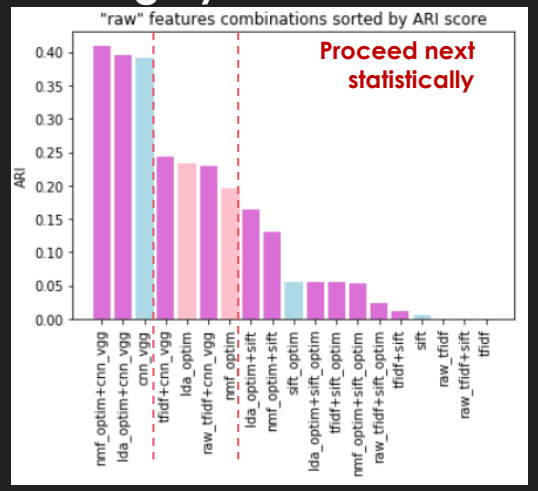
5. Further work about CNN : training of the model with our dataset (img generator & augmentation)

Browse combinations of text / image features

Observations of rough clustering ARI score

- We build alternate **combinations** of **text** or **image** features (*incl. single text nor image*)
- After a PCA 80% reduction, it consists in an assembly of 4 **text** and 3 **image** inputs of different resulting [sizes]
  - **raw td-idf** [**343**] or **tf-idf with processed text** [**276**]
  - **NMF BoW** [**14**] or **LDA BoW** [**14**]
  - **SIFT BoVW** [**98**] **optimum** [**72**] **cnn_vgg** [512]

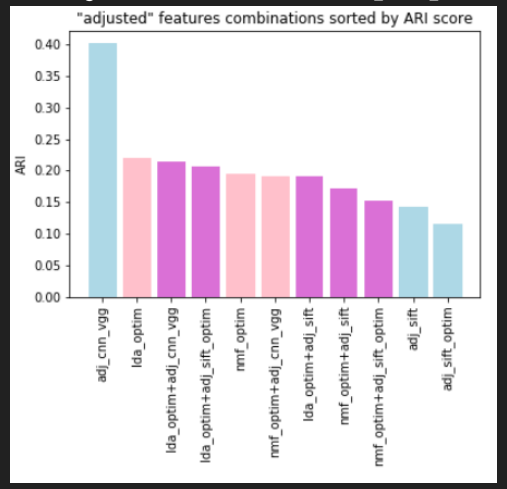
Ranking by ARI score:



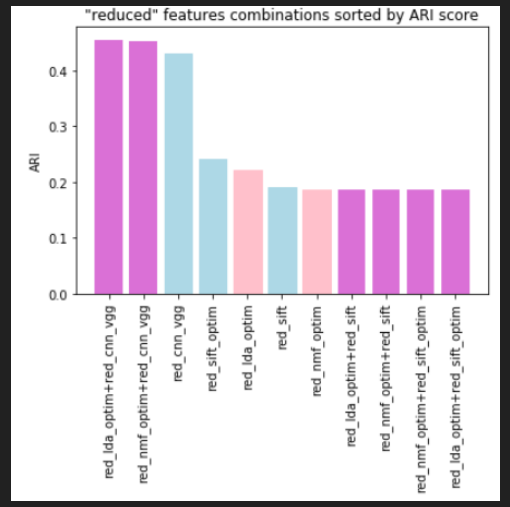
Observations:

- Results are **unstable** and we shall proceed **statistically**
- **cnn_vgg** is on top, enhance **txt**, opposite to **sift BoVW**
- Confirm this trend with **reduced** or **adjusted** inputs dimension.

***"Adjusted"* dim to [14]:**



***"Reduced"* dim to [7]:**



We "chained" PCA and MNF sliding negative to positive data with an offset.

Dead end in case of [14]

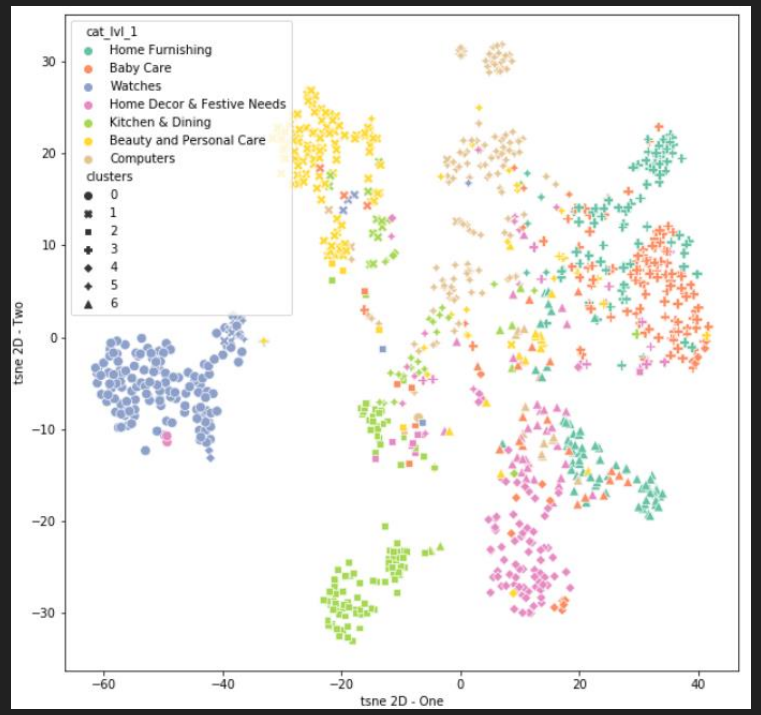
- Mainly lower results
- Ranking seems **irrelevant**
- Trend is **not confirmed**

Promising in case of [7]

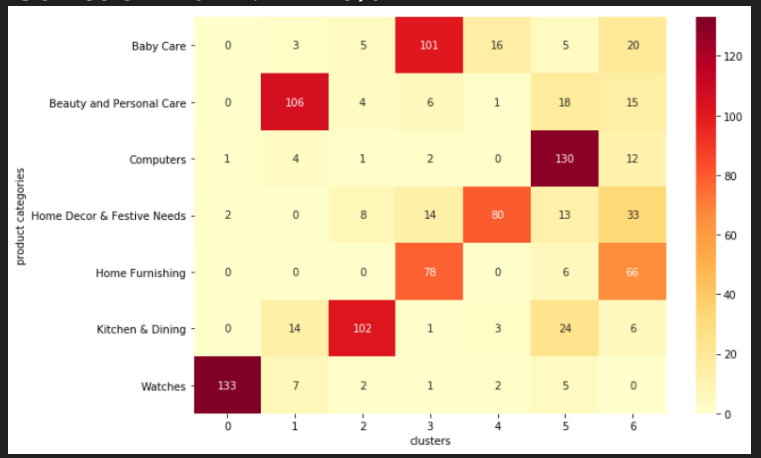
- Being aware of **instability** of our rough results
- Being aware of our **ability** to tune any method
- Combination of **cnn_image** and **text BoW** could even improve results
- We should keep **both** feature and reduction and balance between **image** and **text** is a key point.

"Best" clustering visualizations: lda_optim & cnn_vgg

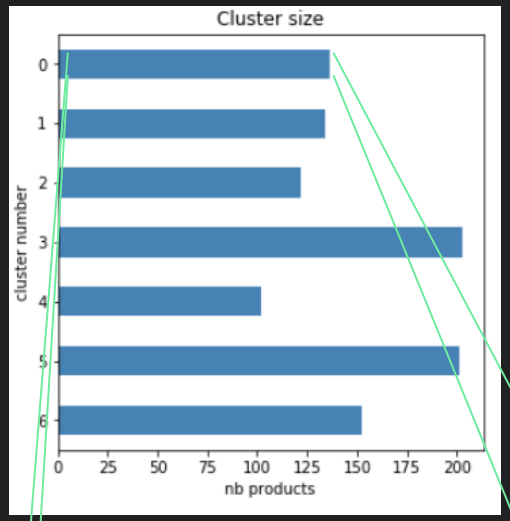
t-sne 2D projection



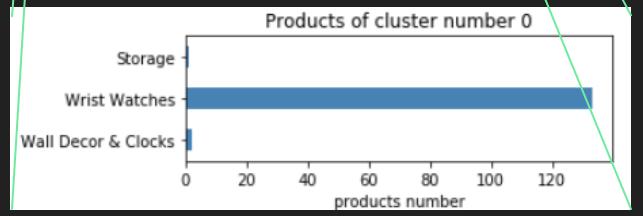
Confusion matrix: ARI 46%



With such combination, clusters are almost balanced:



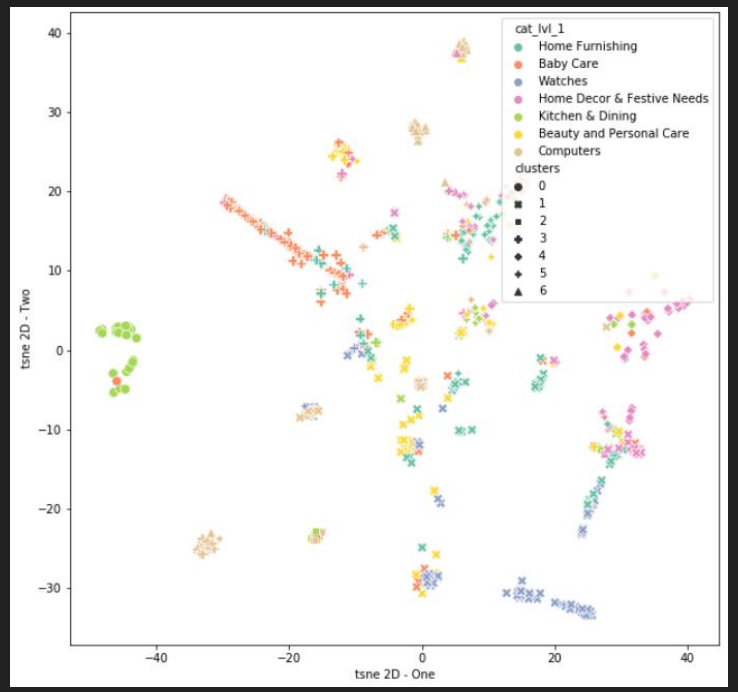
Digging deeper (Cat level 2), we find only one restricted product's panel. Strong similarity has been recognized through our study.



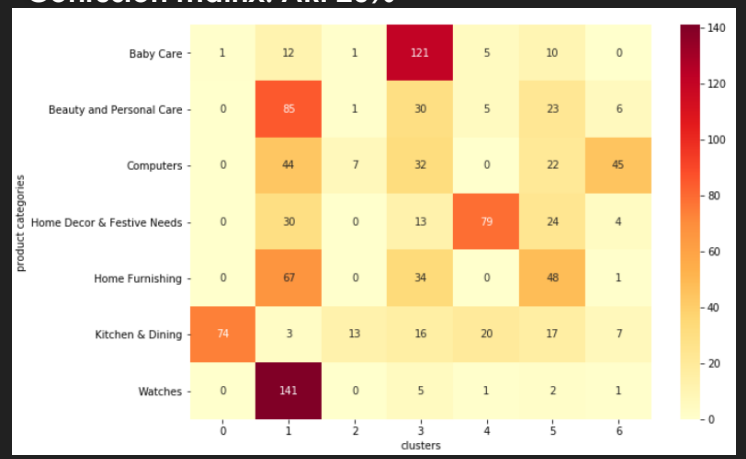
Guess what kind of "storage" product we have?

Alternate case to understand combination: lda_optim only

t-sne 2D projection

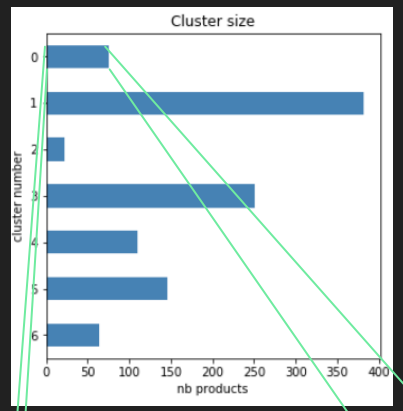


Confusion matrix: ARI 20%

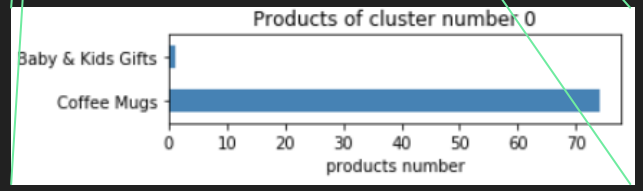


With text LDA BoW only, clusters are more imbalanced:

Nb 1 is our "carry-all" cluster: il contains "Watches" and a wide range of other categories.



Digging deeper (Cat level 2), we find restricted product's panel. Strong similarity has been recognized through our study.



Guess what the product of "Baby & Kids Gifts" is about?

Further steps

- Our recommendation is to engage further work **on both text & image topics**, with key points:
  - Enhance the hierarchical decomposition through **multi-labelling** of products
  - Remedy imbalanced classes (through API)
  - Refine the scope & target, in order to dig deeper only on right and valuable directions, meaning
    - Refine the search space, choose among alternate techniques (processors, extractors, reducers)

Expected decisions :	Feasibility study **validation**	**Y** ☐ ☐ **N**
	Engage next step : risk & opportunity matrix, project plan	**Y** ☐ ☐ **N**

Time for Q&A, Thank you for your attention!