

# Soutenance Projet 6 : "Classifier automatiquement des biens de consommation"



# Sommaire

I. Problématique

II. Les données

III. Partie Texte

IV. Partie Image

V. Combinaison Texte + Image

VI. Résumé des résultats

VII. Conclusion



# I. Problématique

- L'entreprise 'place de marché' souhaite lancer une marketplace e-commerce.
- Des vendeurs proposent des articles à des acheteurs en postant une photo et une description.
- L'attribution de la catégorie d'un article est effectuée manuellement par les vendeurs et est donc peu fiable.
- Pour le moment il y a très peu d'articles en ligne.
- *Il devient donc nécessaire d'automatiser la tâche d'attribution des catégories.*
- Étudier la **faisabilité** d'un moteur de classification automatique avec une précision suffisante.
- Effectuer un **clustering** et choisir une **représentation 2D** à déterminer pour présenter les résultats.



# I. Problématique

## Démarche de travail :

- Analyse rapide des données disponibles.
- Récupérer les données associées aux descriptions et aux images.
- Traiter les données textes (descriptions) : pré-traiter, extraire des features et effectuer un clustering puis comparer aux catégories des produits.
- Traiter les données images : pré-traiter, extraire des features (SIFT / ORB / CNN transfer learning) et effectuer un clustering puis comparer aux catégories des produits.
- Combiner les features textes et images et comparer les résultats
- Choisir quelle méthode serait la plus performante.
- Conclusion sur la faisabilité et les possibilités pour aller plus loin.

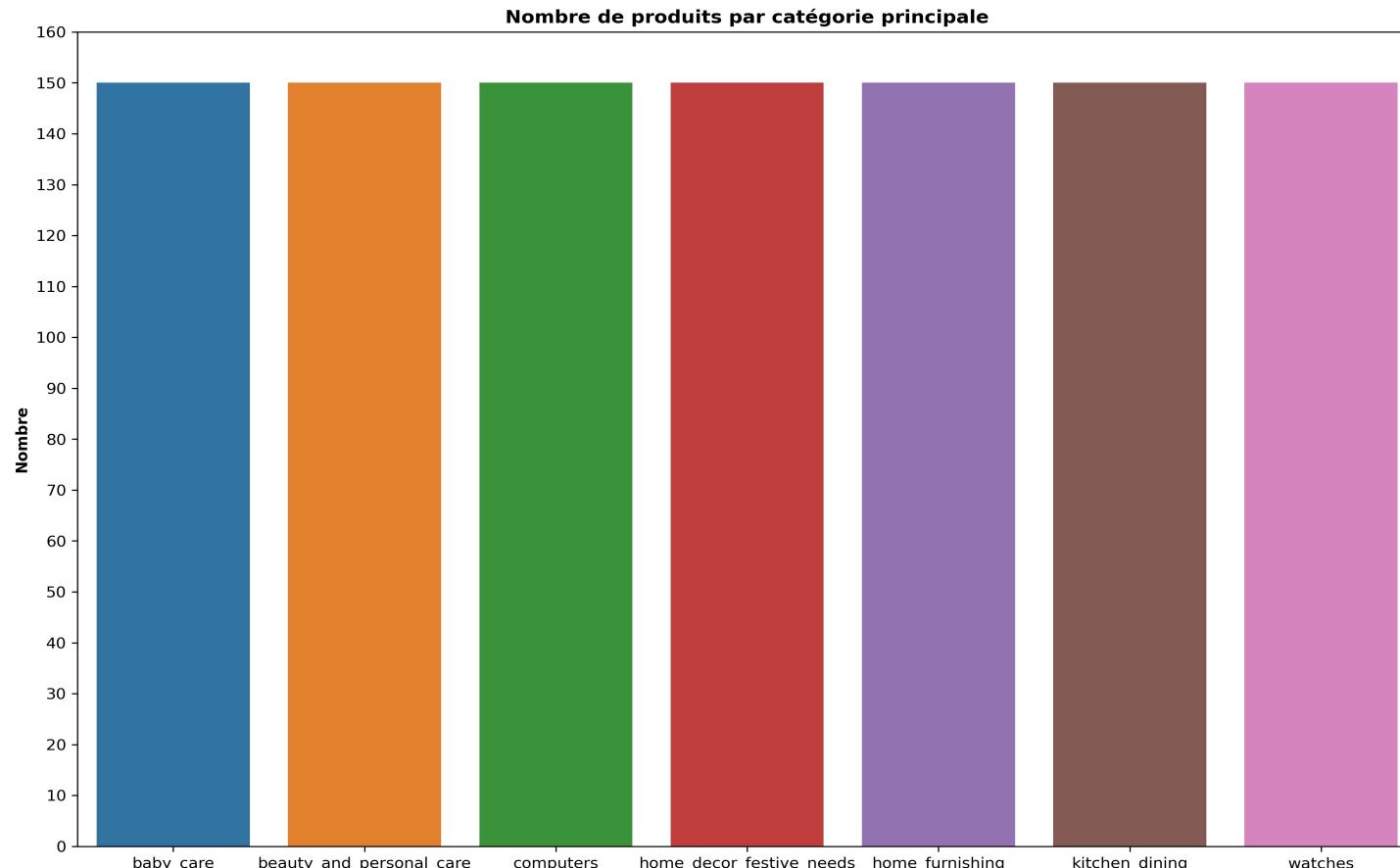


## II. Les données

- Un fichier CSV qui contient des informations par produits:
  - Nom
  - **Descriptions du produit**
  - **Catégories du produit**
  - **Image associée**
  - Url de l'image
  - Informations complémentaires
- Un dossier contenant toutes les images des produits
- 1050 produits disponibles pour l'étude de faisabilité
- Pas de doublons dans les produits.

## II. Les données

→ Répartition des produits par catégorie principale :



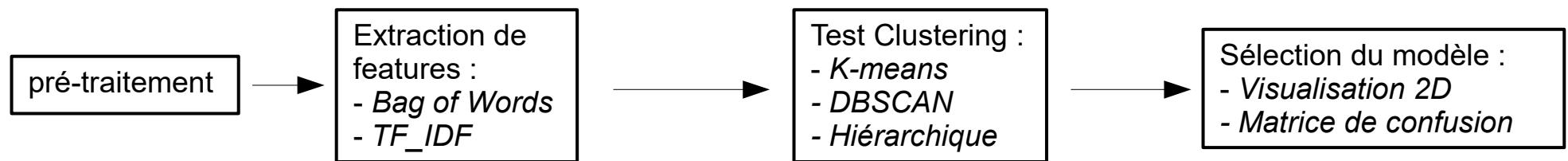
→ Pas de données manquantes dans les images et descriptions des produits.

→ 7 catégories de produits réparties uniformément.



# III. Partie Texte

- Utilisation des descriptions produits
- Overview de la démarche :



# III. Partie Texte

## Pré-traitement :

- Passage en minuscule
- Tokenization :
  - « nltk.RegexpTokenizer(r'[A-Za-z]+') »
- Création des stopwords :
  - « nltk.corpus.stopwords.words('english') »
  - Suppression des 30 mots les plus communs dans le corpus
- Stemming :
  - nltk.stem.snowball.EnglishStemmer()



# III. Partie Texte

## Pré-traitement :

- Exemple sur la première description ('rideau multicolore')

Description brute (1420 caractères): Key Features of Elegance Polyester Multicolor Abstract Eyelet Door Curtain Floral Curtain,Elegance Polyester Multicolor Abstract Eyelet Door Curtain (213 cm in Height, Pack of 2) Price: Rs. 899 This curtain enhances the look of the interiors.This curtain is made from 100% high quality polyester fabric.It features an eyelet style stitch with Metal Ring.It makes the room environment romantic and loving.This curtain is anti-wrinkle and anti shrinkage and have elegant appearance.Give your home a bright and modernistic appeal with these designs. The surreal attention is sure to steal hearts. These contemporary eyelet and valance curtains slide smoothly so when you draw them apart first thing in the morning to welcome the bright sun rays you want to wish good morning to the whole world and when you draw them close in the evening, you create the most special moments of joyous beauty given by the soothing prints. Bring home the elegant curtain that softly filters light in your room so that you get the right amount of sunlight., Specifications of Elegance Polyester Multicolor Abstract Eyelet Door Curtain (213 cm in Height, Pack of 2) General Brand Elegance Designed For Door Type Eyelet Model Name Abstract Polyester Door Curtain Set Of 2 Model ID Duster25 Color Multicolor Dimensions Length 213 cm In the Box Number of Contents in Sales Package Pack of 2 Sales Package 2 Curtains Body & Design Material Polyester

\*\*\*\*\*

Après Tokenization (226 mots): ['key', 'features', 'of', 'elegance', 'polyester', 'multicolor', 'abstract', 'eyelet', 'door', 'curtain', 'floral', 'curtain', 'elegance', 'polyester', 'multicolor', 'abstract', 'eyelet', 'door', 'curtain', 'cm', 'in', 'height', 'pack', 'of', 'price', 'rs', 'this', 'curtain', 'enhances', 'the', 'look', 'of', 'the', 'interiors', 'this', 'curtain', 'is', 'made', 'from', 'high', 'quality', 'polyester', 'fabric', 'it', 'feature', 'an', 'eyelet', 'style', 'stitch', 'with', 'metal', 'ring', 'it', 'makes', 'the', 'room', 'environment', 'romantic', 'and', 'loving', 'this', 'curtain', 'is', 'ant', 'wrinkle', 'and', 'anti', 'shrinkage', 'and', 'have', 'elegant', 'appearance', 'give', 'your', 'home', 'a', 'bright', 'and', 'modernistic', 'appeal', 'with', 'these', 'designs', 'the', 'surreal', 'attention', 'is', 'sure', 'to', 'steal', 'hearts', 'these', 'contemporary', 'eyelet', 'and', 'valance', 'curtains', 'slide', 'smoothly', 'so', 'when', 'you', 'draw', 'them', 'apart', 'first', 'thing', 'in', 'the', 'morning', 'to', 'welcome', 'the', 'bright', 'sun', 'rays', 'you', 'want', 'to', 'wish', 'good', 'morning', 'to', 'the', 'whole', 'world', 'and', 'when', 'you', 'draw', 'them', 'close', 'in', 'the', 'evening', 'you', 'create', 'the', 'most', 'special', 'moments', 'of', 'joyous', 'beauty', 'given', 'by', 'the', 'soothing', 'prints', 'bring', 'home', 'the', 'elegant', 'curtain', 'that', 'softly', 'filters', 'light', 'in', 'your', 'room', 'so', 'that', 'you', 'get', 'the', 'right', 'amount', 'of', 'sunlight', 'specifications', 'of', 'elegance', 'polyester', 'multicolor', 'abstract', 'eyelet', 'door', 'curtain', 'cm', 'in', 'height', 'pack', 'of', 'general', 'brand', 'elegance', 'designed', 'for', 'door', 'type', 'eyelet', 'model', 'name', 'abstract', 'polyester', 'door', 'curtain', 'set', 'of', 'model', 'id', 'duster', 'color', 'multicolor', 'dimensions', 'length', 'cm', 'in', 'the', 'box', 'number', 'of', 'contents', 'in', 'sales', 'package', 'pack', 'of', 'sales', 'package', 'curtains', 'body', 'design', 'material', 'polyester']



# III. Partie Texte

## Pré-traitement :

- Exemple sur la première description

```
Dictionnaire de mots courants (194 mots): ['if', 'above', 'further', 'just', 'delivery', 'are', 'these', 'their', 'of', 'off', 'isn', 'through', 'such', 'shan't', 'will', 'same', "mightn't", 'between', 'cm', "hadn't", 'very', 'were', 'a', 'only', 'against', 'his', 'ours', 'but', 'doesn', 'whom', 'didn', 'when', 'most', 'this', "that'll", 'the', 'had', 'was', 'there', 'flipkart', 'him', "needn't", 'our', "she's", 'shouldn', 'com', 'from', "aren't", 'as', 'haven', 'its', 'replacement', 'because', 'yourselves', 'i', 'you', 'more', 'out', 'until', 'genuine', 'her', "you'll", 'shipping', 'after', 'about', 'nee dn', 'hers', 'having', 'nor', 'cash', "won't", 'he', 'd', 'under', "wouldn't", 'weren', 'your', 'day', 're', 'why', 'down', 'so', 'should', 'up', 'guaran tee', 'o', "weren't", 'yours', 'has', 'we', 'themselves', 'couldn', 'll', 'rs', 'theirs', 'own', 't', 's', 'm', 'is', 'for', 'with', 'they', 'no', "hasn't", 'an', 'to', 'wasn', "doesn't", 'my', 'ain', 'hadn', 'hasn', 'here', 'then', 'both', 'being', 'any', 'some', "you're", 'all', 'does', 'while', 'now', 'products', 'each', 'online', 'wouldn', 'myself', 'again', 'other', 'mightn', 'on', 'and', 'yourself', 'before', 'aren', 'y', 'below', "didn't", 'during ', "shouldn't", 'those', "wasn't", 'himself', 'who', 'that', 'buy', 'once', 'too', "isn't", 'free', 'ma', "couldn't", "mustn't", "it's", 'them', 'itself ', 'few', 'be', 'over', 'or', 've', 'how', 'can', 'where', 'it', 'have', "should've", "don't", "you've", 'in', 'by', 'did', 'into', 'doing', 'me', 'ourse lves', 'won', 'what', 'herself', 'am', 'than', 'which', 'mustn', 'at', "haven't", 'don', 'not', 'she', "you'd", 'shan', 'been', 'do']  
*****
```

```
Après Stemming (148 mots): ['key', 'featur', 'eleg', 'polyest', 'multicolor', 'abstract', 'eyelet', 'door', 'curtain', 'floral', 'curtain', 'eleg', 'poly est', 'multicolor', 'abstract', 'eyelet', 'door', 'curtain', 'height', 'pack', 'price', 'curtain', 'enhanc', 'look', 'interior', 'curtain', 'made', 'high ', 'qualiti', 'polyest', 'fabric', 'featur', 'eyelet', 'style', 'stitch', 'metal', 'ring', 'make', 'room', 'environ', 'romant', 'love', 'curtain', 'ant', 'wrinkl', 'anti', 'shrinkag', 'eleg', 'appar', 'give', 'home', 'bright', 'modernist', 'appeal', 'design', 'surreal', 'attent', 'sure', 'steal', 'heart', 'contemporari', 'eyelet', 'valanc', 'curtain', 'slide', 'smooth', 'draw', 'apart', 'first', 'thing', 'morn', 'welcom', 'bright', 'sun', 'ray', 'want', 'w ish', 'good', 'morn', 'whole', 'world', 'draw', 'close', 'even', 'creat', 'special', 'moment', 'joyous', 'beauti', 'given', 'sooth', 'print', 'bring', 'h ome', 'eleg', 'curtain', 'soft', 'filter', 'light', 'room', 'get', 'right', 'amount', 'sunlight', 'specif', 'eleg', 'polyest', 'multicolor', 'abstract', 'eyelet', 'door', 'curtain', 'height', 'pack', 'general', 'brand', 'eleg', 'design', 'door', 'type', 'eyelet', 'model', 'name', 'abstract', 'polyest', 'd oor', 'curtain', 'set', 'model', 'id', 'duster', 'color', 'multicolor', 'dimens', 'length', 'box', 'number', 'content', 'sale', 'packag', 'pack', 'sale', 'packag', 'curtain', 'bodi', 'design', 'materi', 'polyest']
```

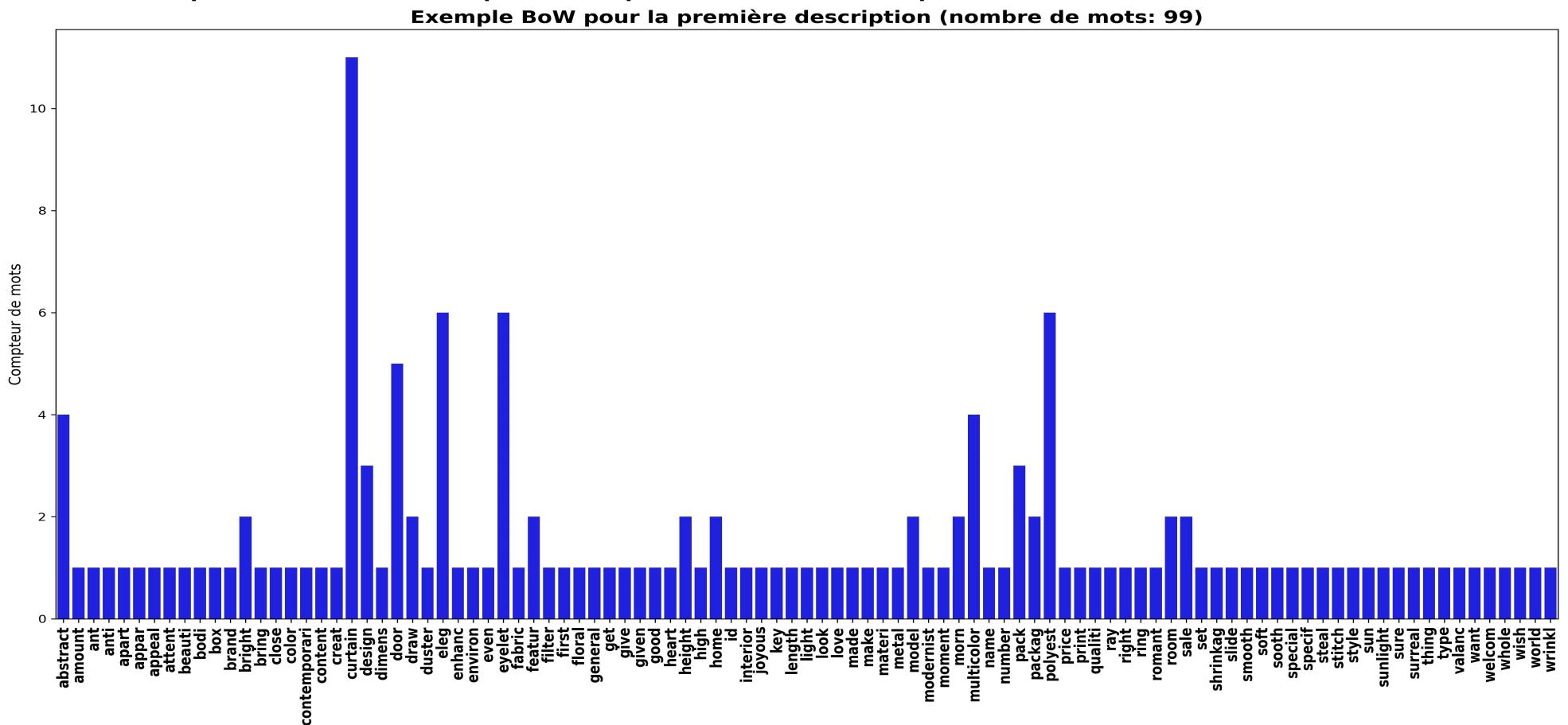
→ 1420 caractères / 226 mots / 148 mots au final



# III. Partie Texte

## Bag of Words (unigramme) / Extraction de features :

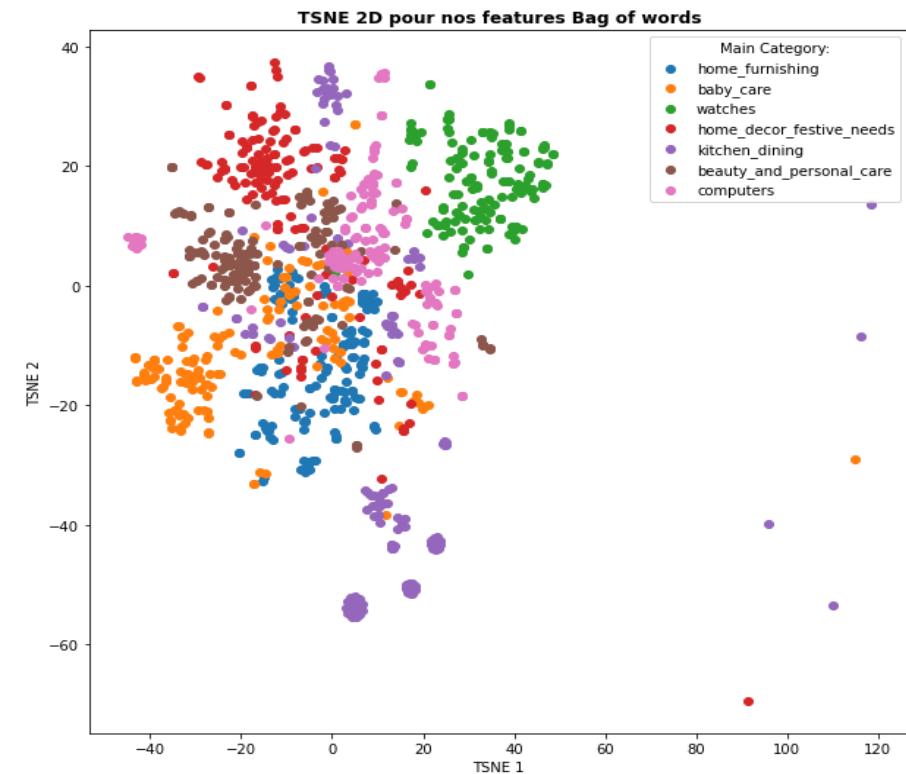
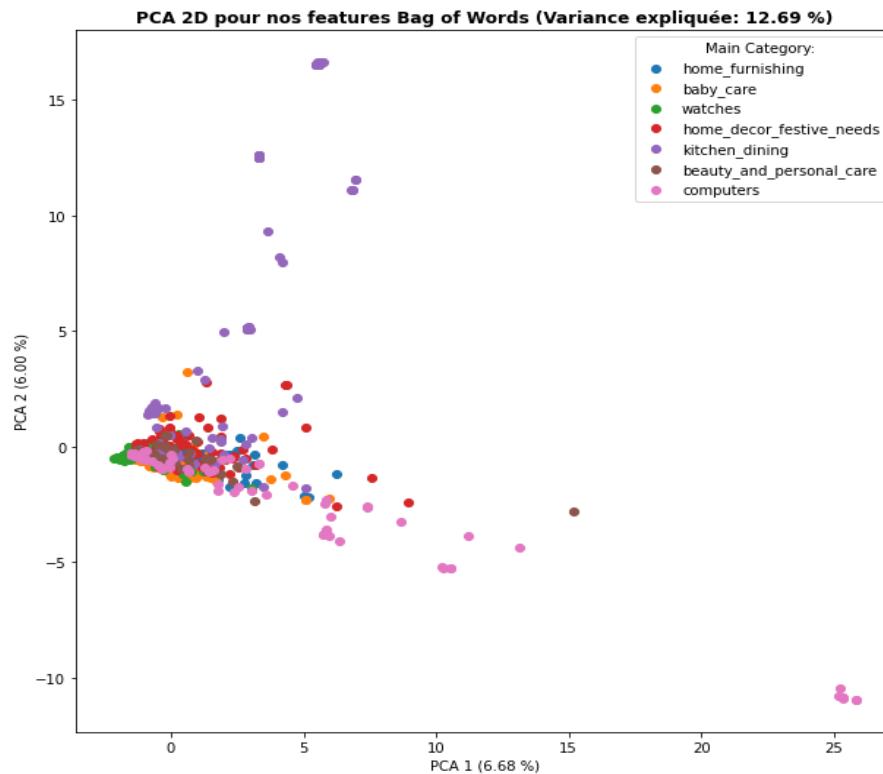
- Utilisation de CountVectorizer() → Matrice (1050, 4079)
- Exemple de features pour la première description :



# III. Partie Texte

## Bag of Words (unigramme) / Extraction de features :

- Visualisations 2D (vraies catégories) :



→ La visualisation via TSNE semble plus intéressante.



# III. Partie Texte

## Bag of Words (unigramme) / Clustering :

- Test sur trois modèles :
  - Kmeans / Hiérarchique → 7 clusters
  - DBSCAN → Minimise le bruit tout en essayant de trouver 7 cluster avec les hyperparamètres epsilon et min\_samples
- Essai avec réduction de dimension PCA(99% de variance) sur Kmeans (features 4079 → 568)
- Résultats :

	Model	nb_cluster	Davies_bouldin_score	Silhouette_score	comment	similarity with category	Test
0	KMeans(n_clusters=7, random_state=4)	7	1.962405	0.308543	raw features Bag of words	0.040611	BoW_Kmeans
1	DBSCAN(eps=17.4, min_samples=3, n_jobs=-1)	7	2.269214	0.401409	raw features Bag of words	0.001634	BoW_DBSCAN
2	AgglomerativeClustering(n_clusters=7)	7	2.651601	0.234225	raw features Bag of words	0.054947	BoW_Clustering_Hiérarchique
6	KMeans(n_clusters=7, random_state=4)	7	2.274031	0.263548	reduce features Bag of words	0.044166	BoW_Kmeans_red99

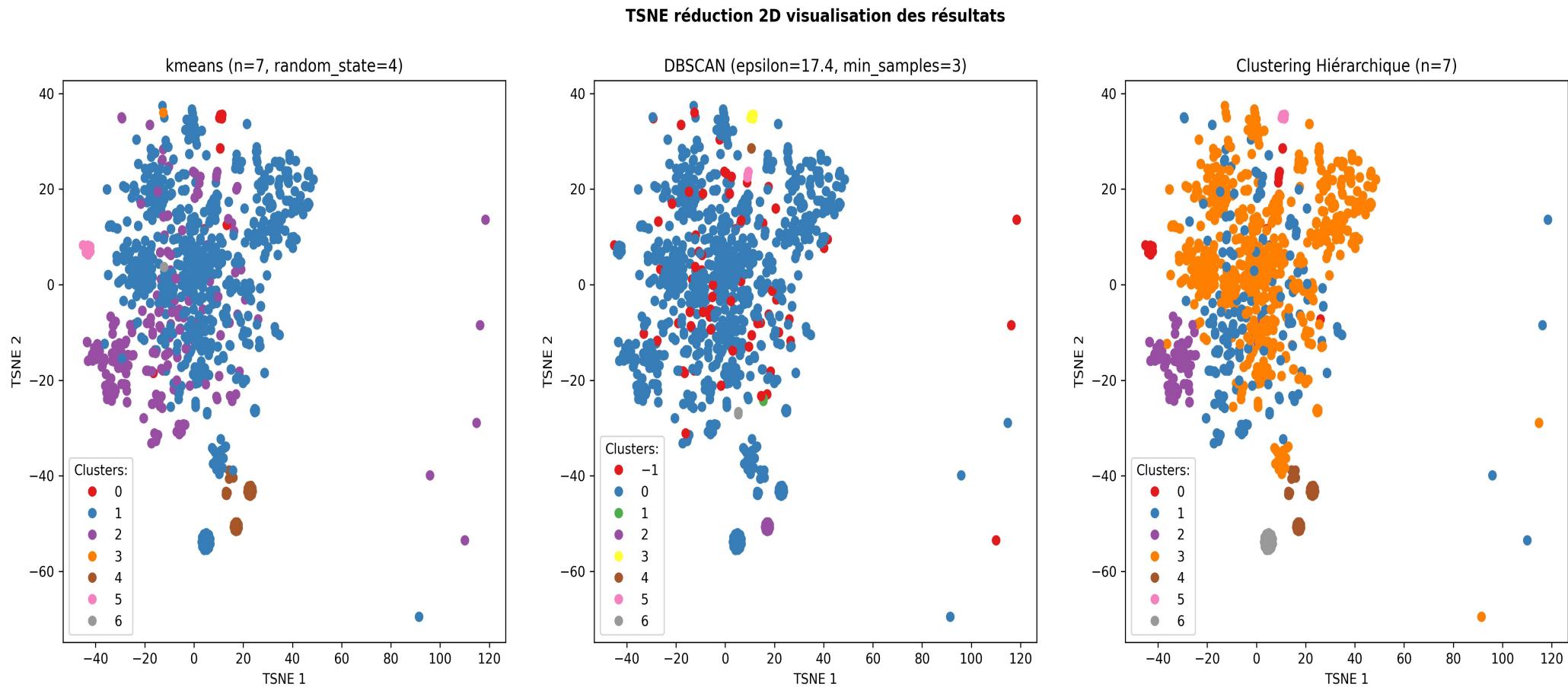
→ On mesure la similarité avec les vraies catégories via le score ARI.



# III. Partie Texte

## Bag of Words (unigramme) / Clustering :

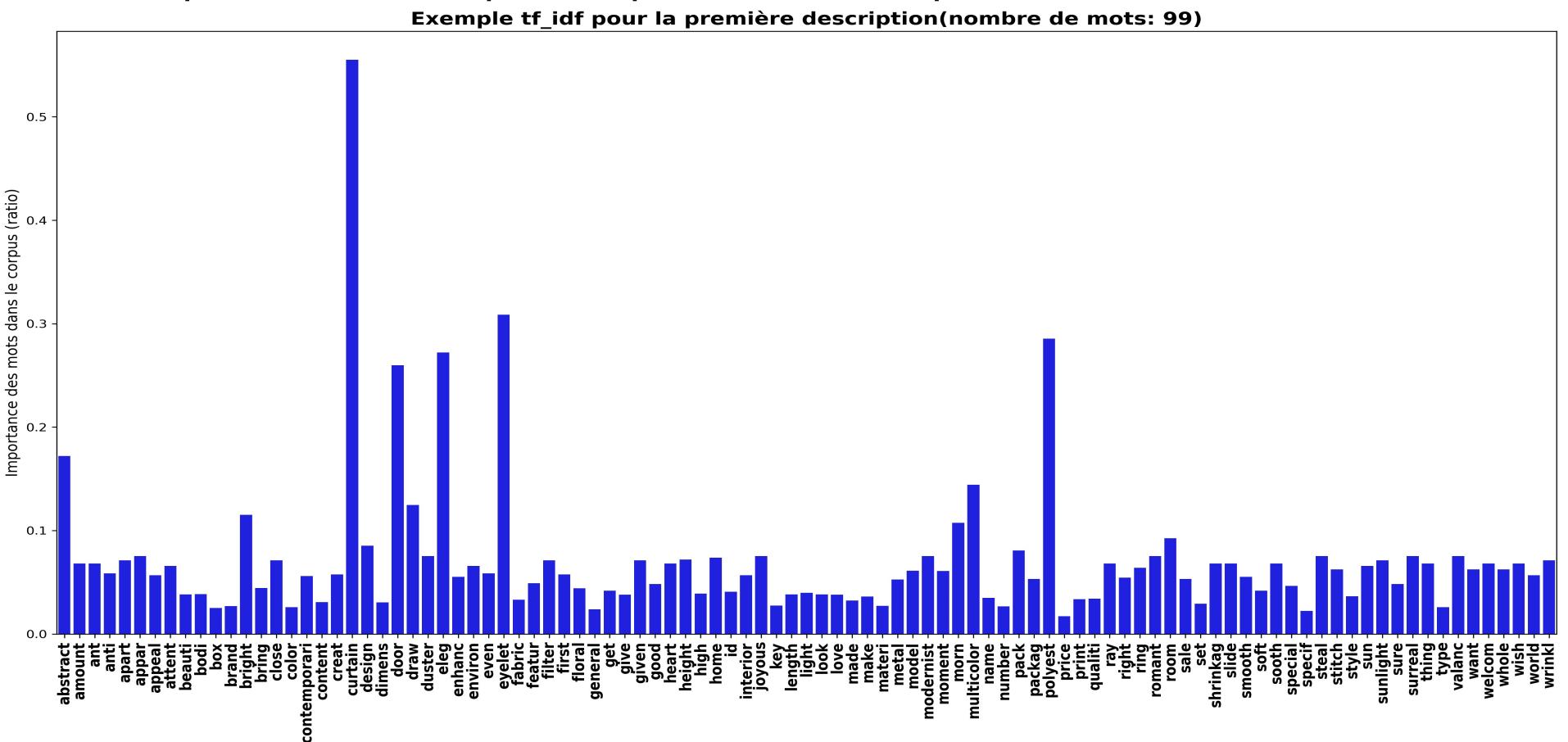
- Visualisations :



# III. Partie Texte

## TF\_IDF(unigramme) / Extraction de features :

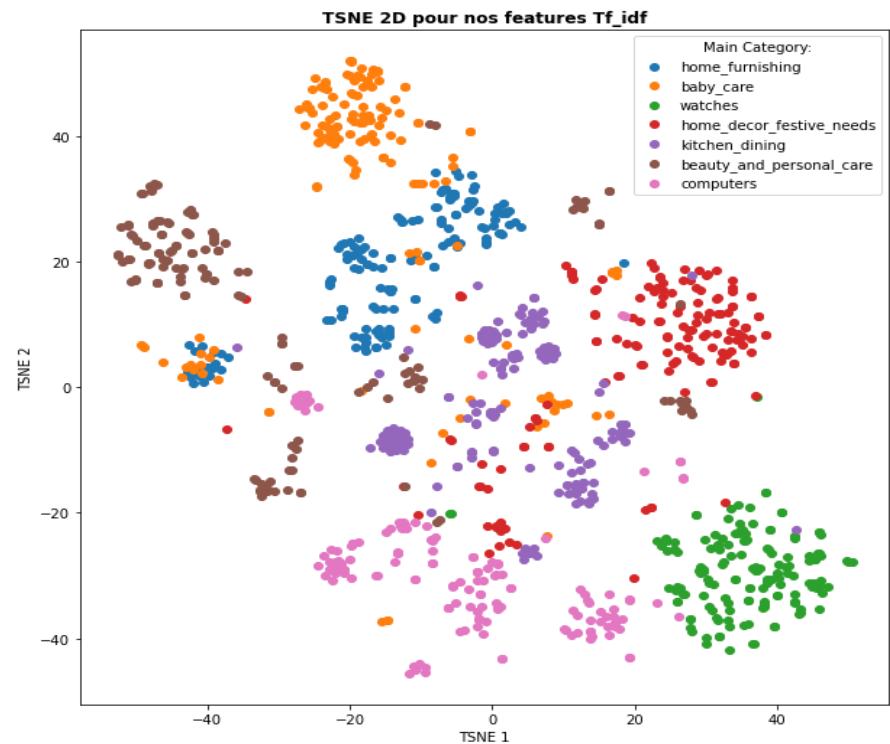
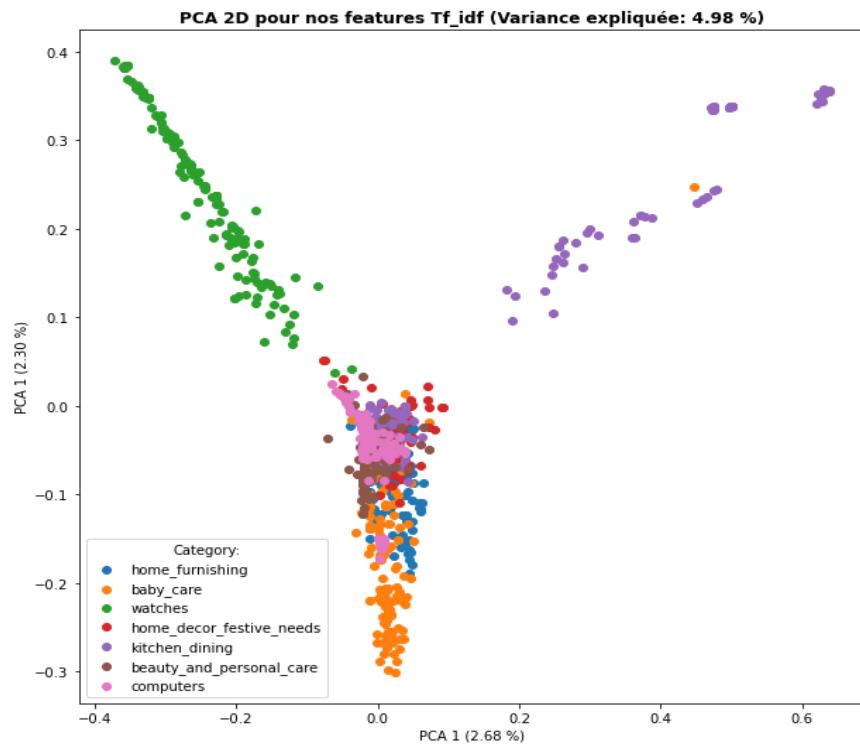
- Utilisation de TfidfVectorizer() → Matrice (1050, 4079)
- Exemple de features pour la première description :



# III. Partie Texte

## TF\_IDF (unigramme) / Extraction de features :

- Visualisations 2D (vraies catégories) :



→ La visualisation via TSNE semble plus intéressante ici aussi.



# III. Partie Texte

## TF\_IDF (unigramme) / Clustering :

- Test de trois modèles :
  - Kmeans / Hiérarchique → 7 clusters
  - DBSCAN → Minimise le bruit tout en essayant de trouver 7 cluster avec les hyperparamètres epsilon et min\_samples
- Essai avec réduction de dimension PCA (99% de variance) sur le Kmeans (features 4079 → 809)
- Résultats :

	Model	nb_cluster	Davies_bouldin_score	Silhouette_score	comment	similarity with category	Test
3	KMeans(n_clusters=7, random_state=4)	7	4.897084	0.047969	raw features tf-idf	0.343890	tfidf_Kmeans
4	DBSCAN(eps=0.6, min_samples=6, n_jobs=-1)	7	1.262782	0.027384	raw features tf-idf	0.005554	tfidf_DBSCAN
5	AgglomerativeClustering(n_clusters=7)	7	4.201250	0.049079	raw features tf-idf	0.270394	tfidf_Clustering_Hiéarchique
7	KMeans(n_clusters=7, random_state=4)	7	4.564332	0.043283	reduce features tf-idf	0.247483	tfidf_Kmeans_red99

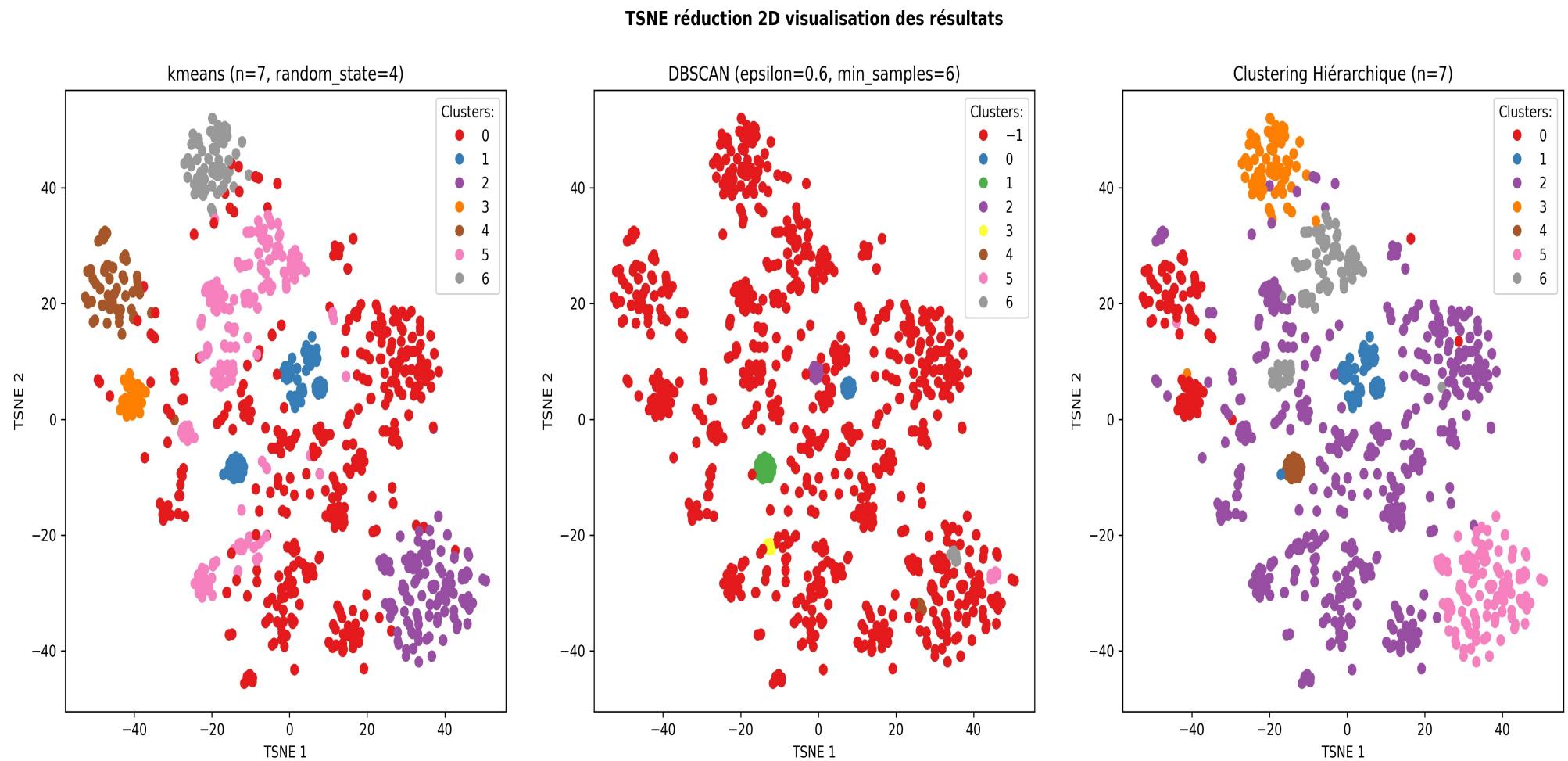
→ On mesure la similarité avec les vraies catégories via le score ARI.



# III. Partie Texte

## TF\_IDF (unigramme) / Clustering :

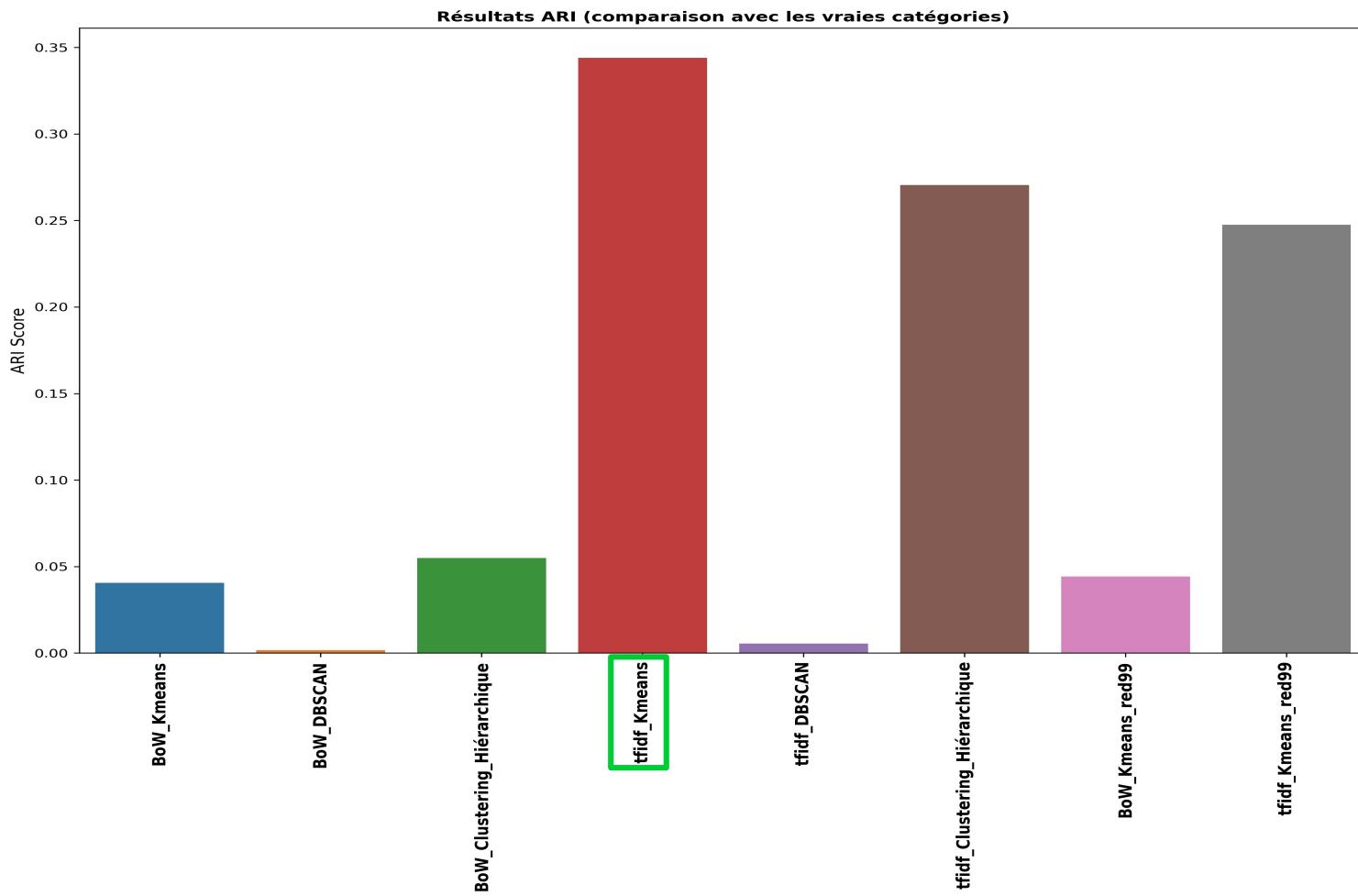
- Visualisations 2D (clusters) :



# III. Partie Texte

## Sélection du modèle :

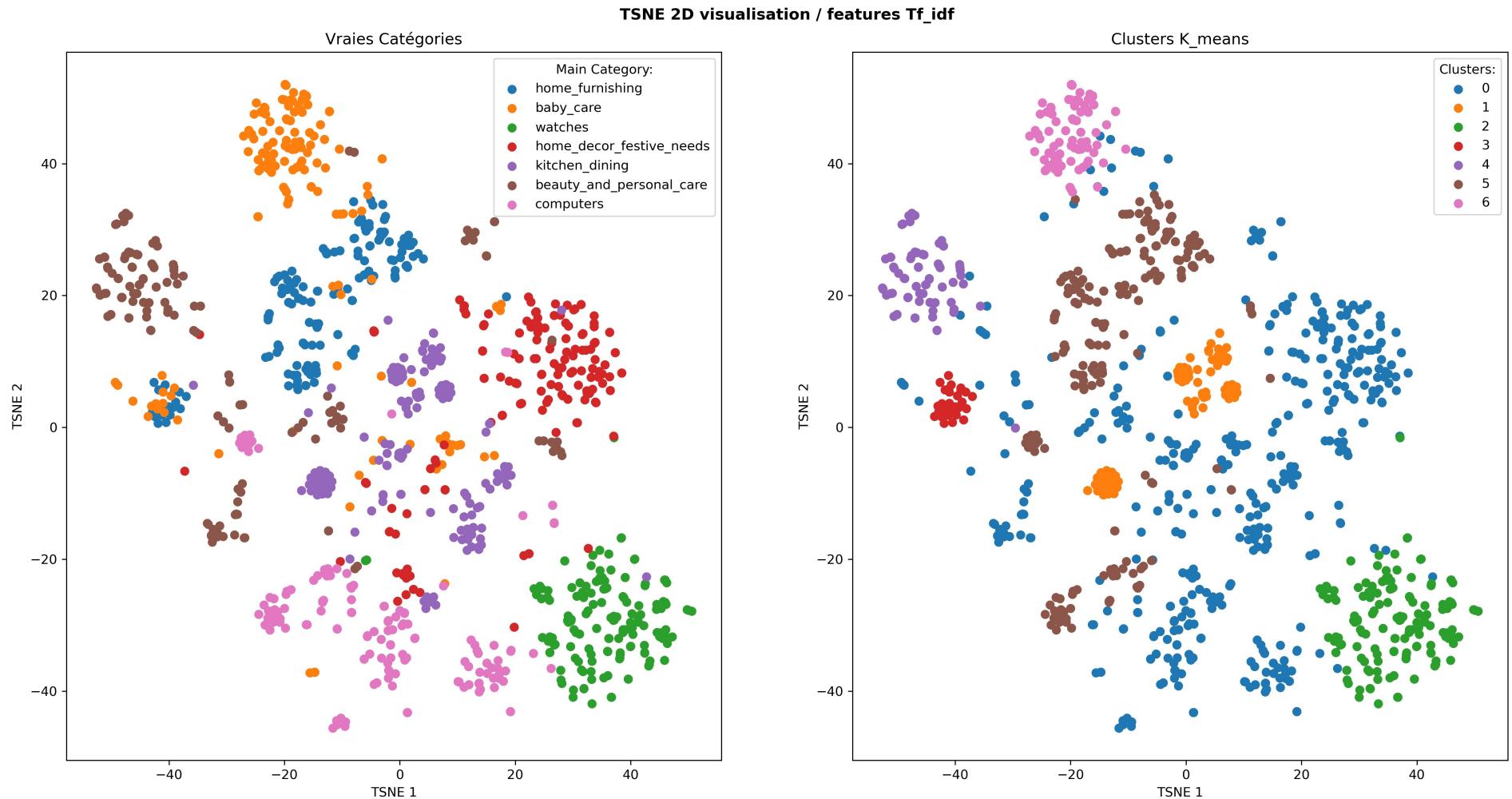
- Résumé des scores de similarité:



# III. Partie Texte

## Sélection du modèle :

- Visualisation 2D :



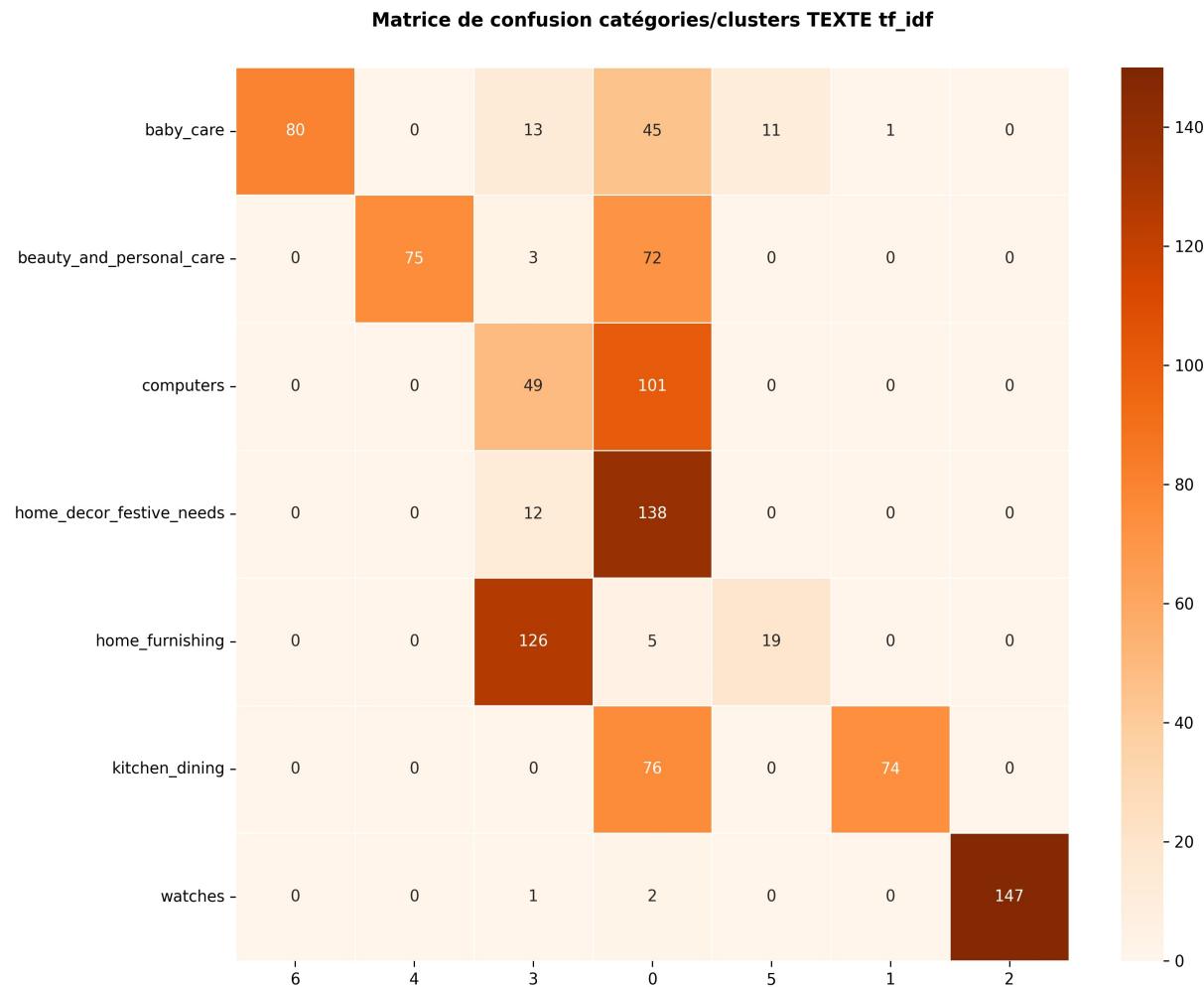
SENECHAL Yannick



# III. Partie Texte

## Sélection du modèle :

- Matrice de confusion :

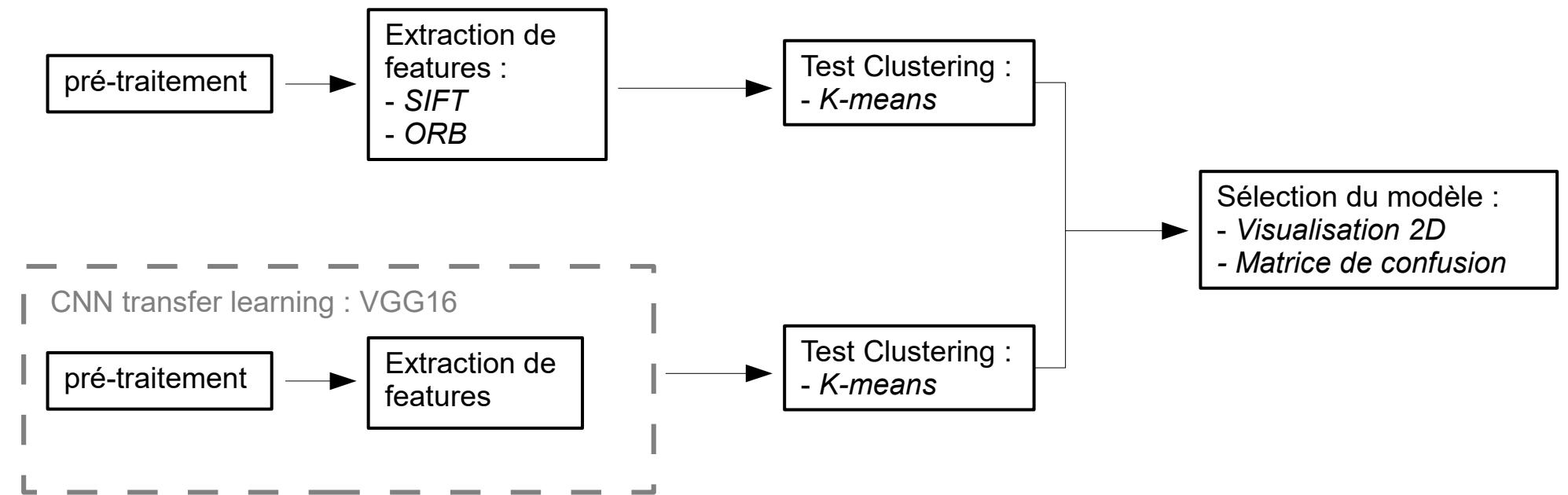


→ On associe manuellement les clusters trouvés aux vraies catégories



# IV. Partie Image

- Utilisation du dossier Images
- Overview de la démarche :



# IV. Partie Image

## Pré-traitement (**SIFT - ORB**) :

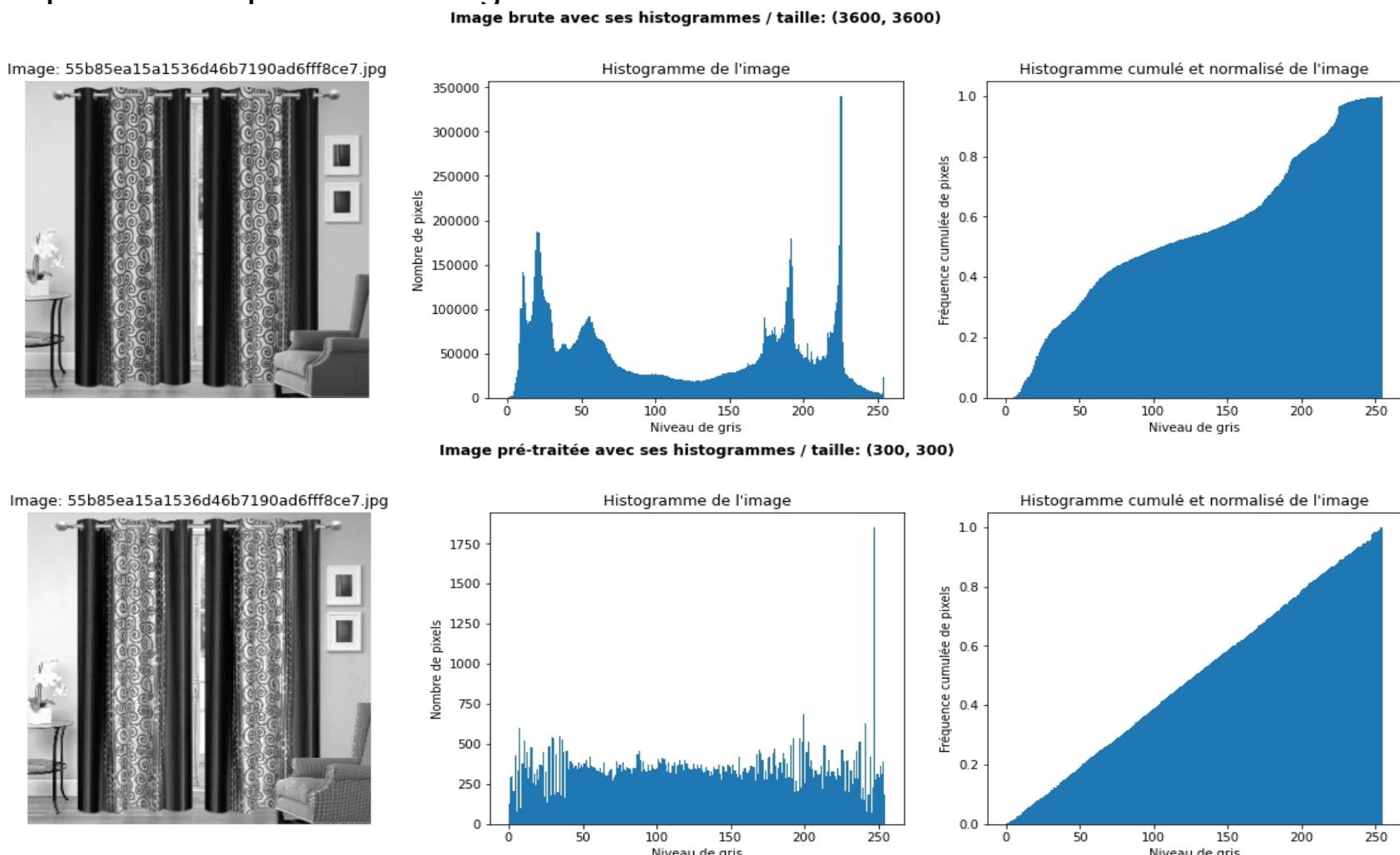
- Import de l'image en noir et blanc
- Traitement du bruit : cv.GaussianBlur()
- Traitement du contraste et de l'exposition : cv.equalizeHist()
- Redimensionnement en 300X300 : cv.resize()



# IV. Partie Image

## Pré-traitement (SIFT - ORB) :

- Exemple avec la première image :



# IV. Partie Image

## SIFT / Extraction de features :

- Récupération des 'descripteurs' pour chaque image (`cv.xfeatures2d.SIFT_create()`).
- Exemple pour la première description :

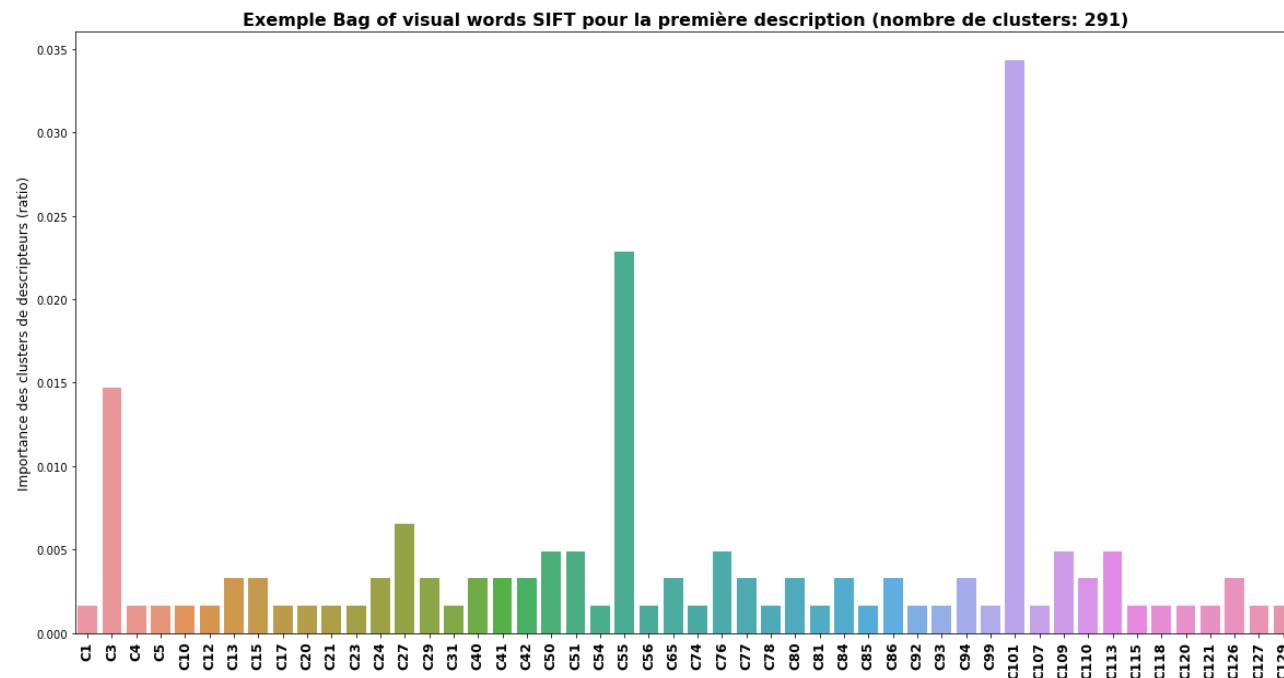


- 612 descripteurs dans l'image d'exemple.
- 575610 descripteurs cumulés sur toutes les images.

# IV. Partie Image

## SIFT / Extraction de features :

- On associe tous les descripteurs de chaque image à un groupe via clustering ( $n_{\text{cluster}} = \text{racine carré du nombre total de descripteurs}$ )
- Création d'un histogramme de comptage pour chaque clusters de descripteurs présents dans l'image (que l'on pondère aux nombres de clusters différents présents dans l'image)
- Exemple :



→ On ne présente que les 50 premiers clusters présents dans l'exemple

→ Pour une image on a 759 features



# IV. Partie Image

## ORB / Extraction de features :

- Récupération des 'descripteurs' pour chaque image (cv.ORB\_create()).
- Exemple pour la première description :

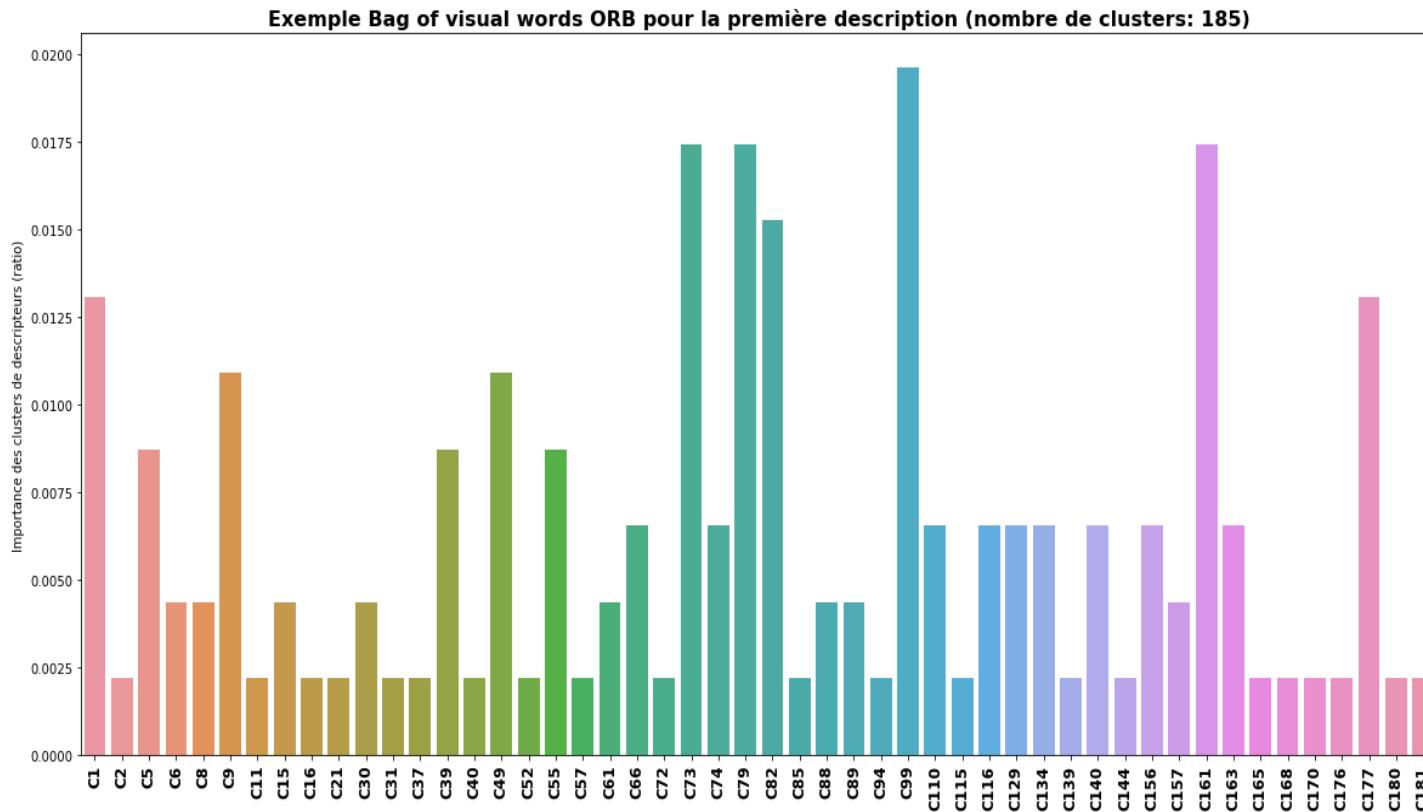


→ 493 descripteurs dans l'image d'exemple.  
→ 467101 descripteurs cumulés sur toutes les images.

# IV. Partie Image

## ORB / Extraction de features :

- Même méthode que pour SIFT, on crée un histogramme de comptage pour chaque clusters de descripteurs présent dans l'image ( $n_{\text{cluster}} = \text{racine carré du nombre total de descripteurs}$ )
- Exemple :



→ On ne présente que les 50 premiers clusters présents dans l'exemple

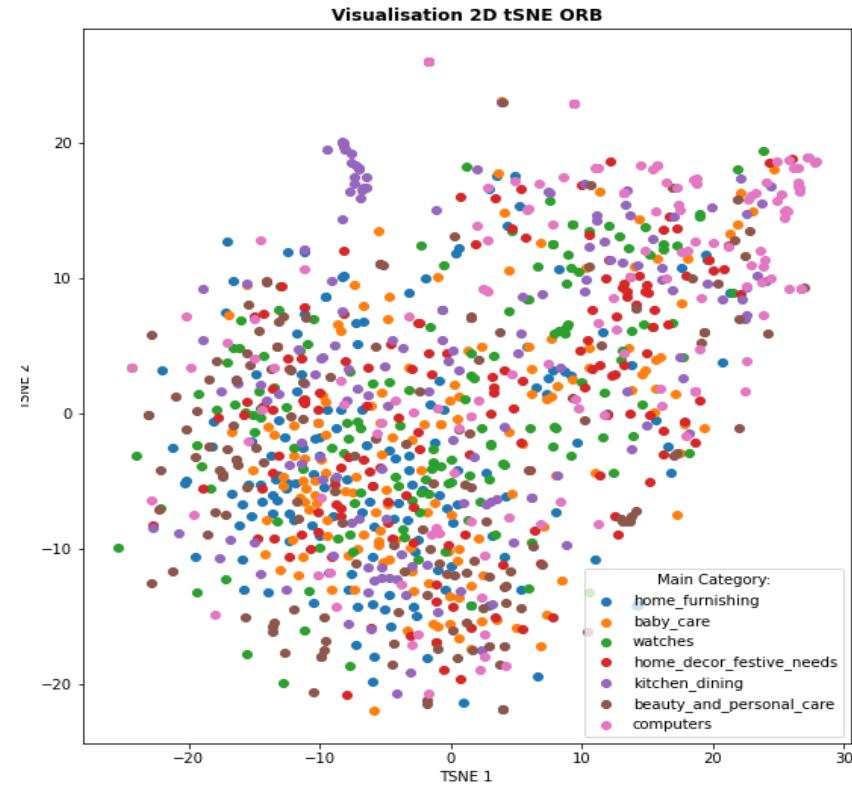
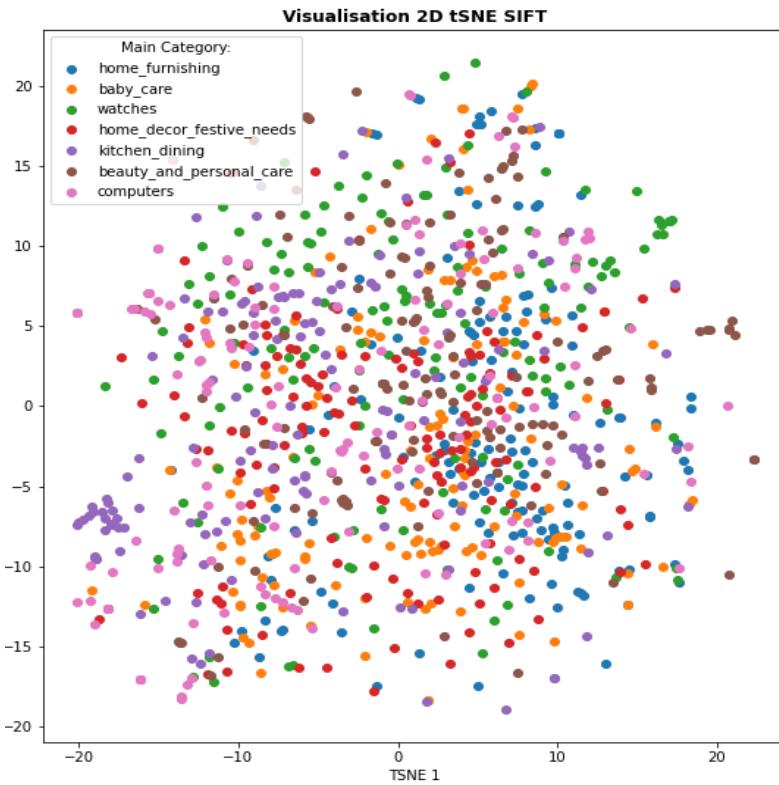
→ Pour une image on a 683 features



# IV. Partie Image

## SIFT - ORB / Extraction de features :

- Visualisations 2D (vraies catégories) :



# IV. Partie Image

## SIFT - ORB / Clustering:

- Essai Kmeans / Clustering Hiérarchique avec n\_cluster = 7
- Essai avec réduction de dimension PCA (99% de variance) sur Kmeans (features SIFT 739 → 556 / features ORB 683 → 547)
- Résultats :

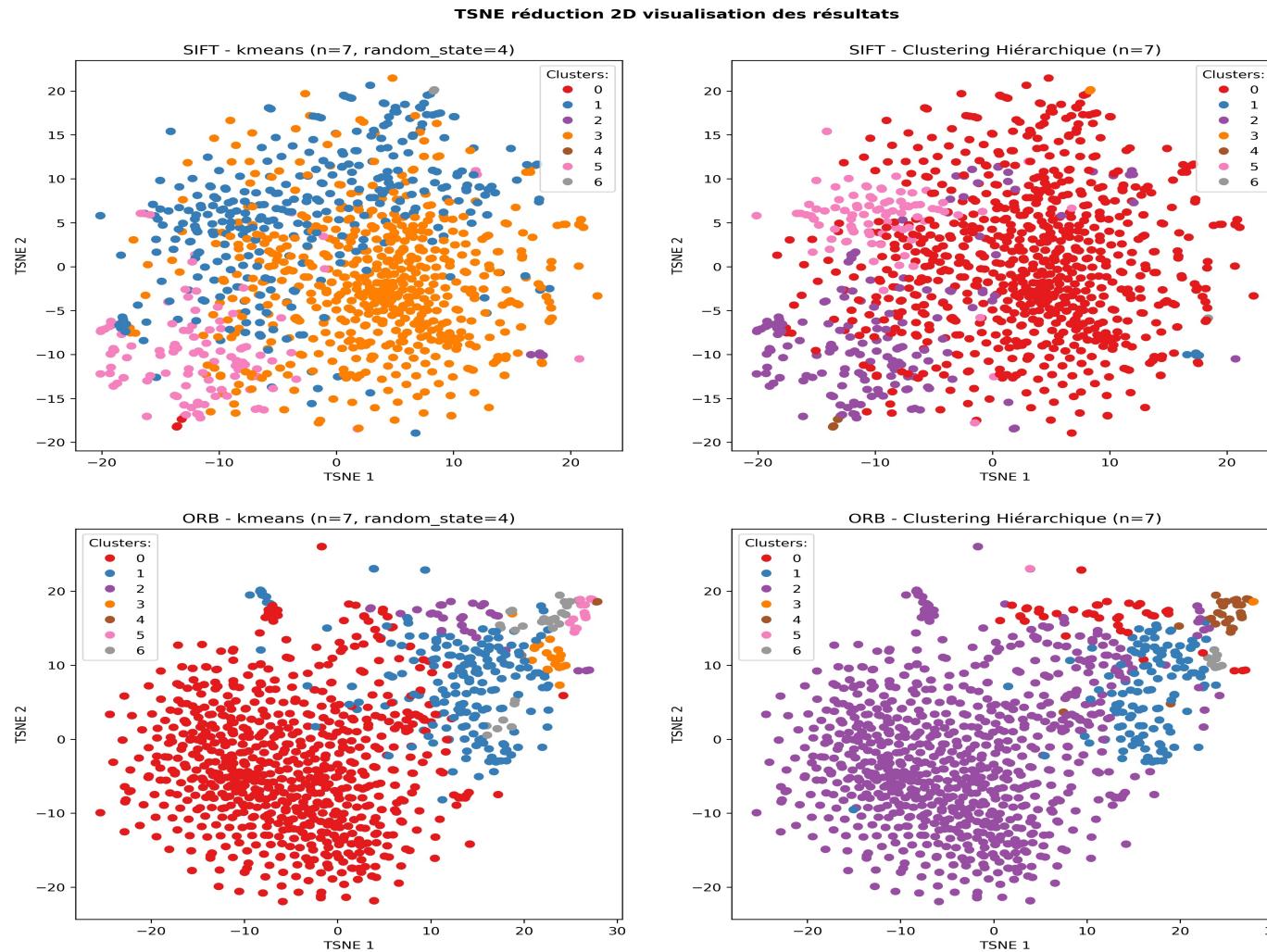
	Model	nb_cluster	Davies_bouldin_score	Silhouette_score	comment	similarity with category
0	Kmeans	7	2.502546	0.060763	features SIFT	0.043956
1	Agglomerative clustering	7	1.952732	0.090348	features SIFT	0.024528
2	Kmeans	7	1.537859	0.197191	features SIFT reduced	0.022229
3	Kmeans	7	2.461319	0.130456	features ORB	0.013844
4	Agglomerative clustering	7	1.927591	0.159531	features ORB	0.012629
5	Kmeans	7	2.199987	0.146239	features ORB reduced	0.013421



# IV. Partie Image

## SIFT - ORB / Clustering:

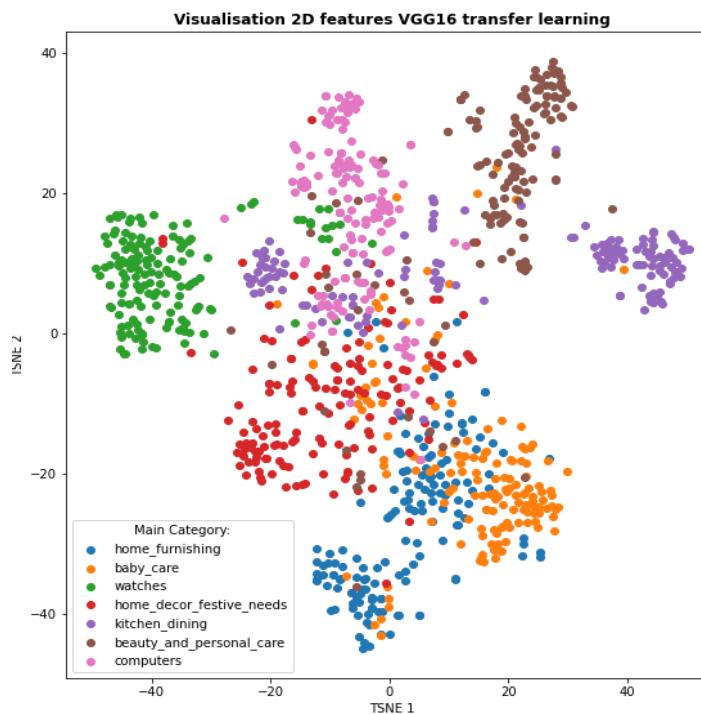
- Visualisations 2D (clusters) :



# IV. Partie Image

## CNN transfer learning / VGG16:

- Utilisation d'un VGG16 pour extraction de features (disponible via Keras).
- Pré-traitement + extraction de features automatique avec le réseau de neurones pré-entraîné 'ImageNet'.
- On garde toutes les couches sauf celle de classification.
- Pour chaque image 4096 features.
- Visualisation 2D (vrais catégories) :



# IV. Partie Image

## CNN transfer learning / VGG16:

- Essai clustering Kmeans n\_cluster = 7.
- Essai avec réduction de dimension PCA (99% de variance) : features 4096 → 940
- Résultats :

Model	nb_cluster	Davies_bouldin_score	Silhouette_score	comment	similarity with category
8 Kmeans	7	5.028498	-0.012876	features extracted VGG16 without classifier	0.42004
9 Kmeans	7	4.980836	-0.012200	features extracted VGG16 without classifier re...	0.42232

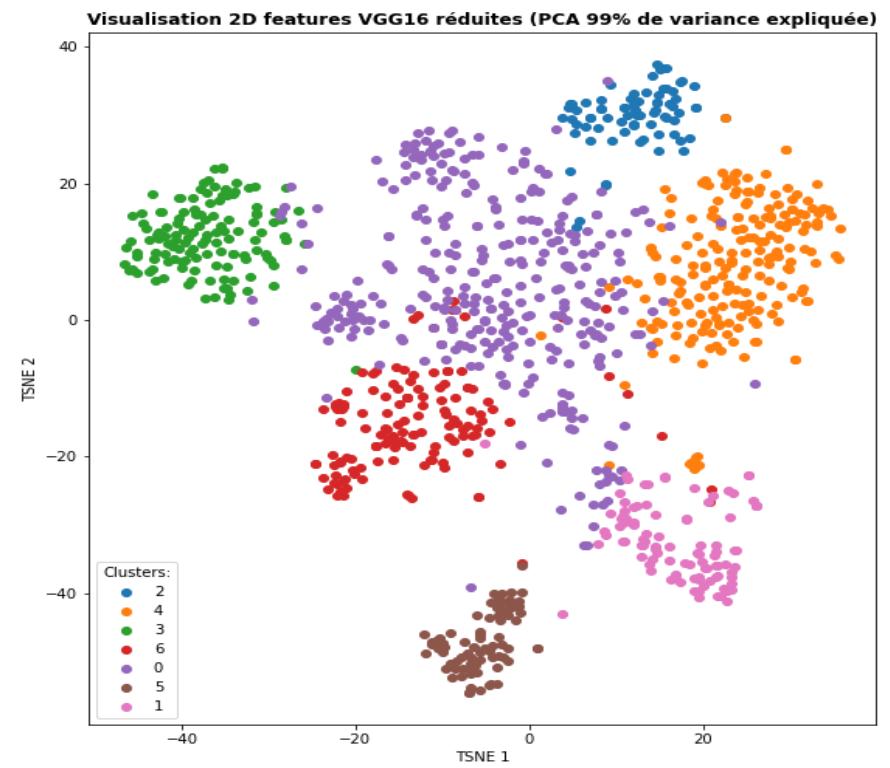
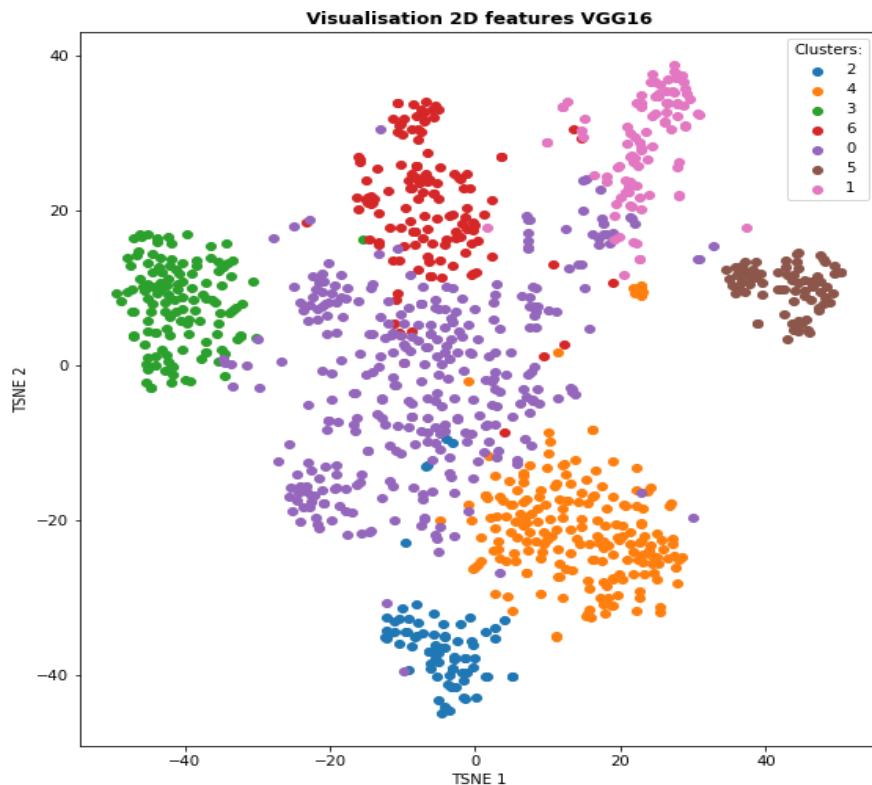
→ Score ARI bien meilleur qu'avec SIFT et ORB.



# IV. Partie Image

## CNN transfer learning / VGG16:

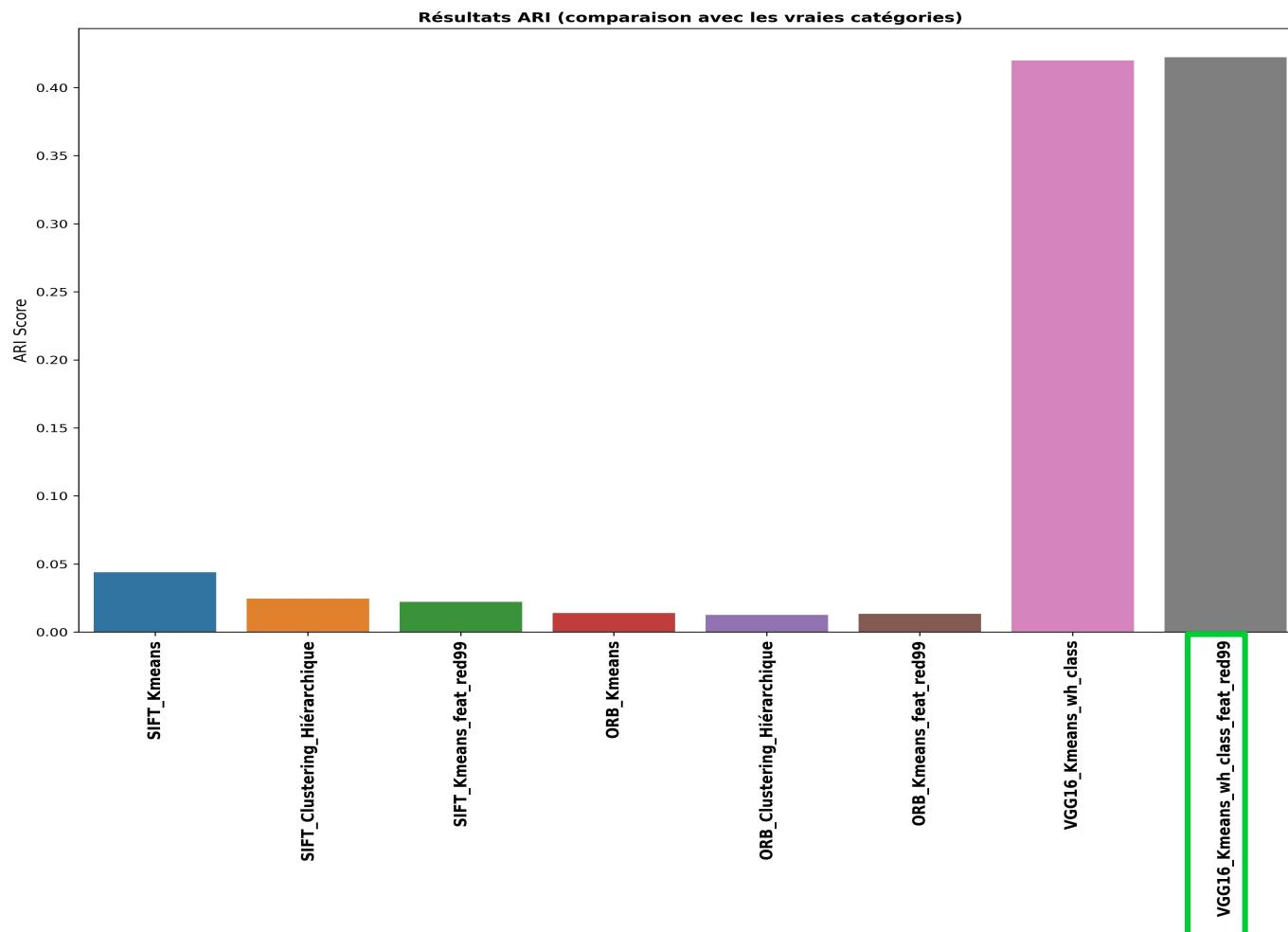
- Visualisation 2D (clustering) :



# IV. Partie Image

## Sélection du modèle:

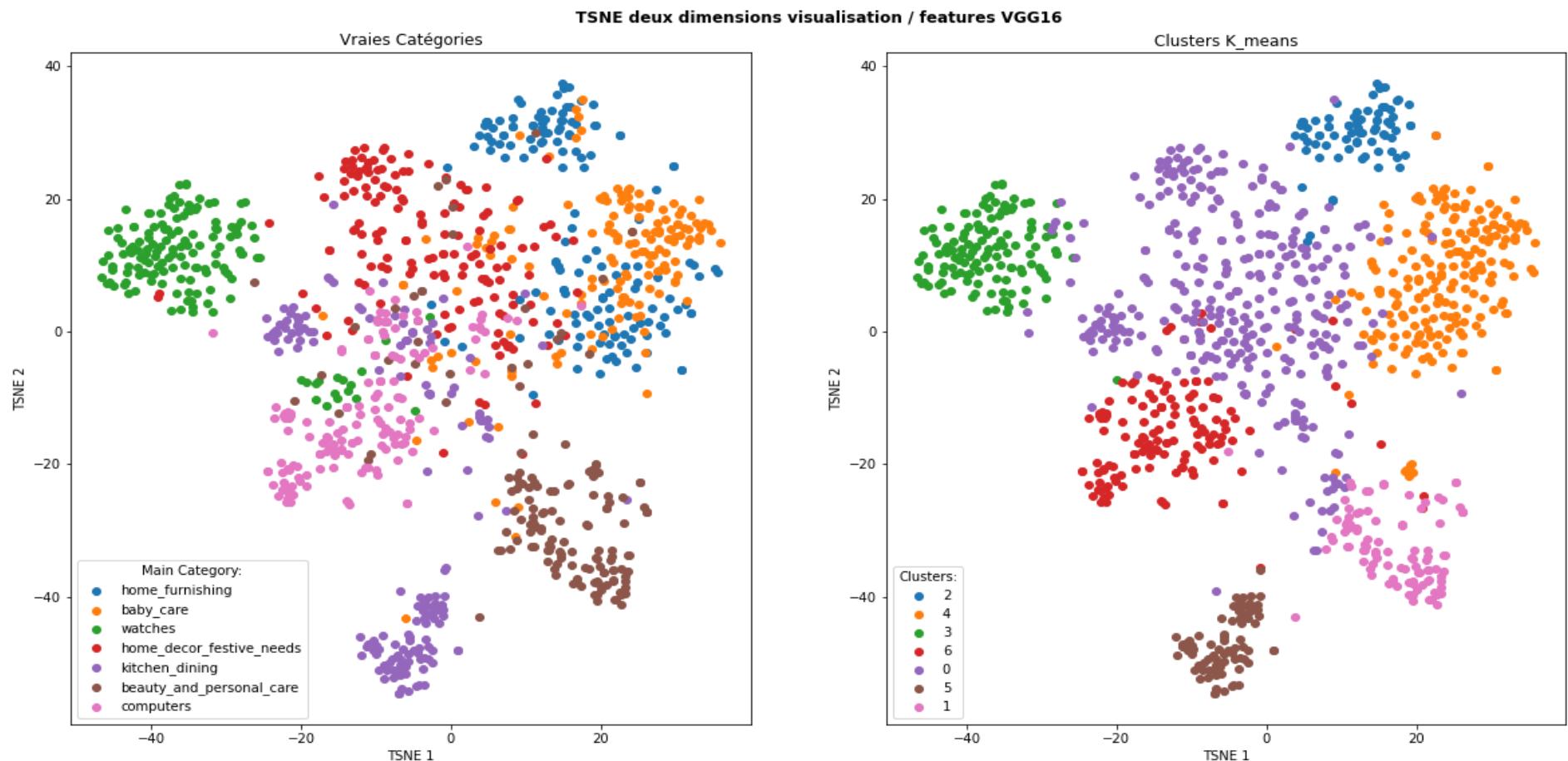
- Résumé des score de similarités :



# IV. Partie Image

## Sélection du modèle:

- Visualisation 2D:

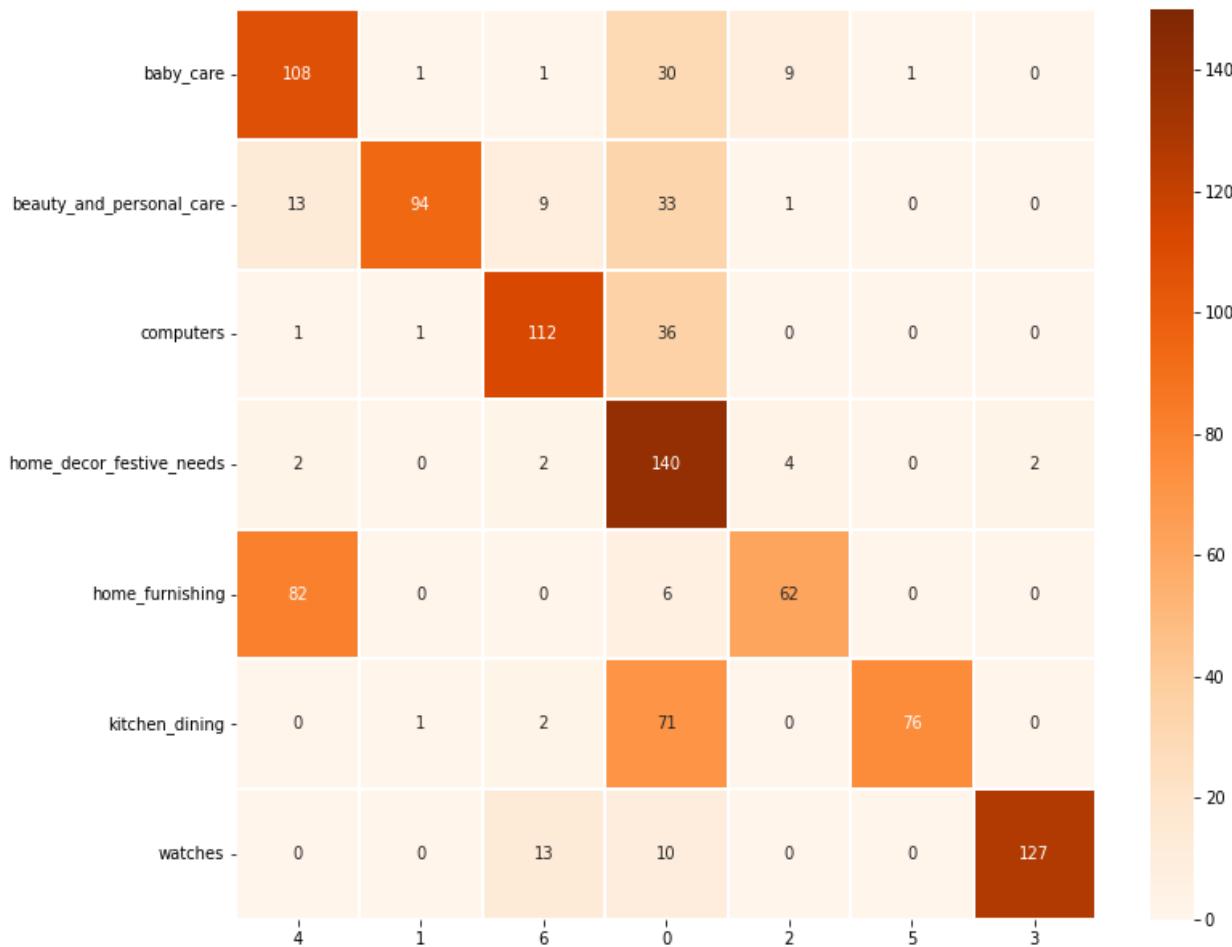


# IV. Partie Image

## Sélection du modèle:

- Matrice de confusion:

Matrice de confusion catégories/clusters Image VGG16 features réduites sans classifieur



→ On associe manuellement les clusters trouvés aux vraies catégories



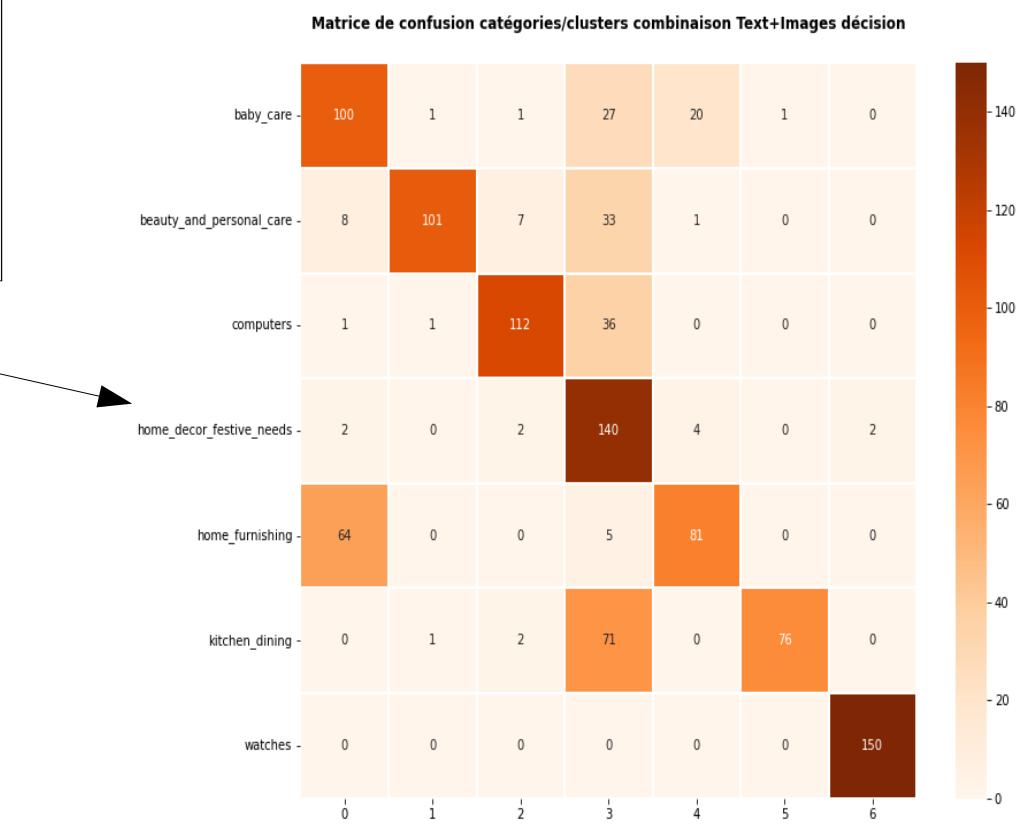
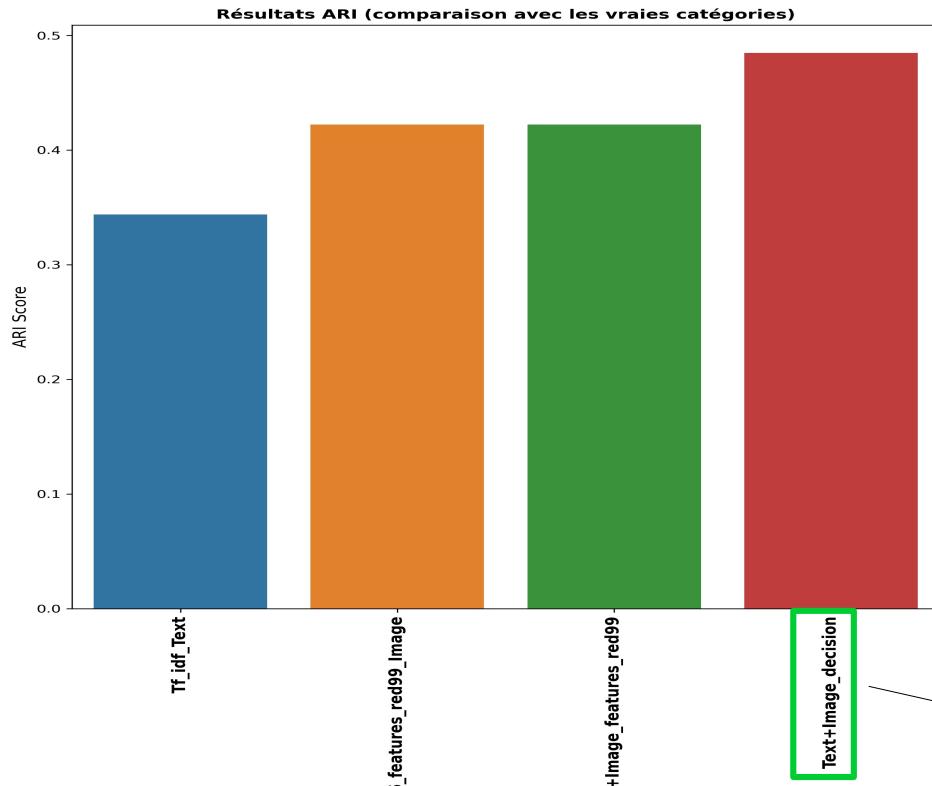
# V. Combinaison Texte+Image

## Deux méthodes explorées:

- Concaténation des features Texte et Image réduites (PCA 99%) puis Kmeans :
  - 1612 features
  - ARI score avec les vraies catégories : **0.42**
- Prise de décision en fonction des meilleurs modèles Texte et Image basé sur la précision des modèles :
  - Évaluation des précisions en fonction des vraies catégories
  - Attribution d'un cluster en fonction du modèle le plus précis
  - ARI score: **0.48**

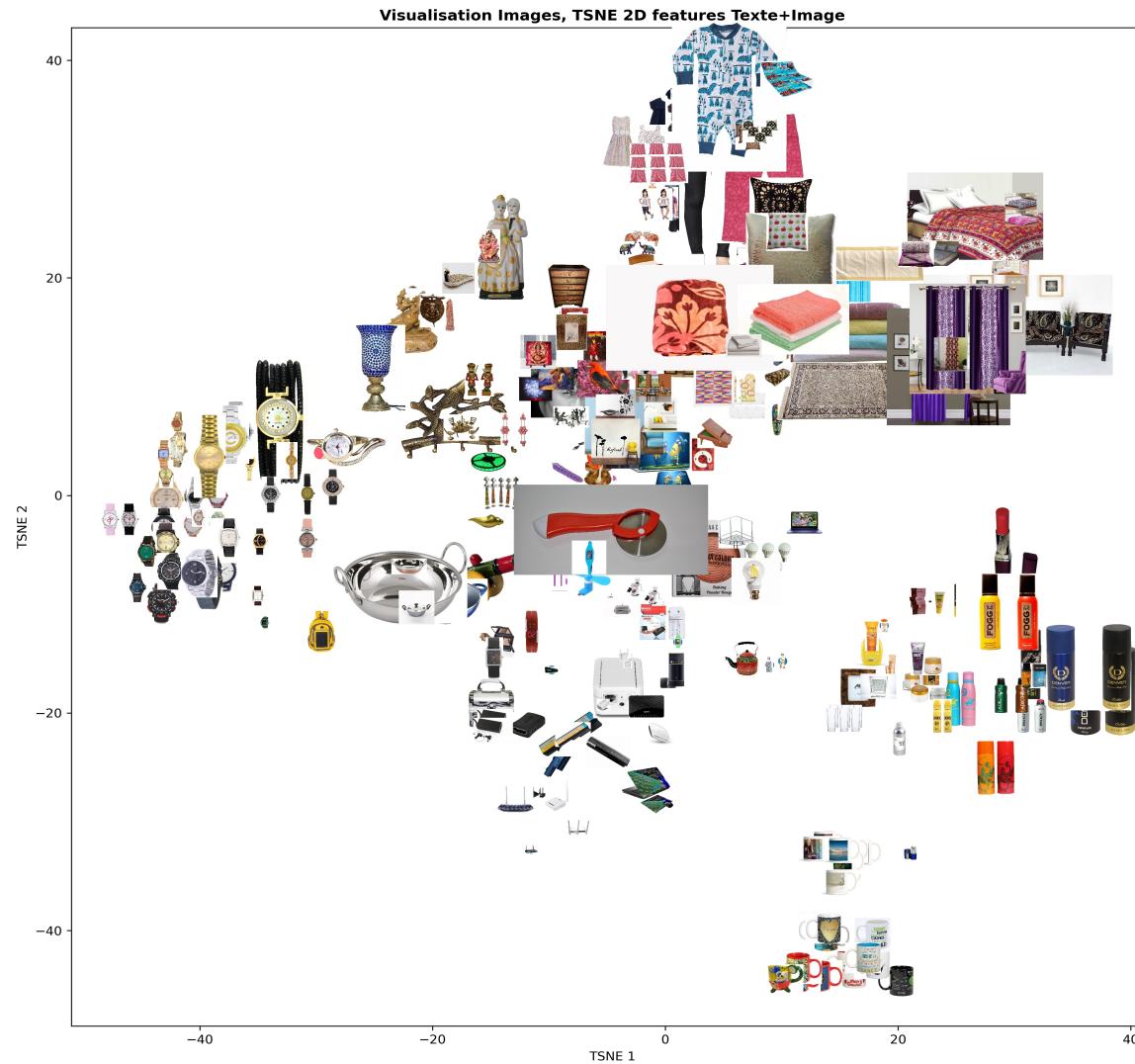


# VI. Résumé des résultats



# VI. Résumé des résultats

## Visualisation 2D des images:



SENECHAL Yannick



# VII. Conclusion

- On peut envisager la faisabilité d'un moteur de classification automatique
- Avec les features Texte+Image (tf\_idf+VGG16) on arrive à identifier de manière clair la catégorie d'une partie des produits
- Recommandations pour aller plus loin :
  - Tester d'autres features n\_gram, word embedding sur la partie texte
  - Envisager aussi le transfer learning sur la partie texte (ResNet50)
  - Utiliser des algorithmes de classification ?



***Merci pour votre attention !***

**Questions ?**

