

Classifiez automatiquement des biens de consommation



Sommaire



1 Introduction

- Définition du problème
 - présentation des données
-



2 Traitement des données textuelles

- Lower case
- Stopwords
- Lemmatizer
- Extraire features des textes
 - word2vec (transfer learning)
 - TF-IDF
 - Bag of words

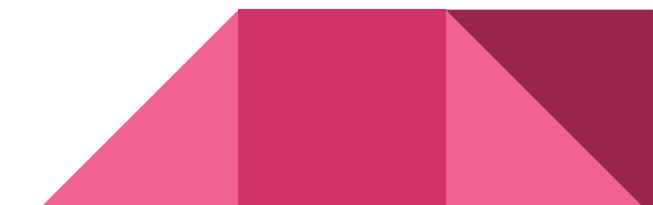


3 Traitement des données images

- SIFT
- CNN
- EfficientNetB0
- VGG-16
- ResNet50



4 Conclusion



Introduction

- Une plateforme d'e-commerce proposant des produits à la vente
- Les données des produits issus de la base FlipKart incluent des descriptions textuelles et des images
- Attribution manuelle des catégories : fastidieuse et peu fiable
- Catégories déjà renseignées pour un petit volume de produits mais le volume de produits non catégorisés est destiné à s'accroître

Problématique:

- Est-il possible de classifier automatiquement les produits de manière pertinente ?



A screenshot of a Flipkart product page for a CLOTHOLOGY 144 TC Cotton Double Printed Flat Bedsheet (Pack of 1, Multicolor). The page shows a green and blue geometric patterned bedsheet and two matching pillows. The price is ₹395 (₹2,999 86% off). It includes a 3.5-star rating with 587 reviews and an 'Assured' badge. Delivery information states it's usually delivered in 7 days. Buttons for 'ADD TO CART' and 'BUY NOW' are visible at the bottom.

Objectifs

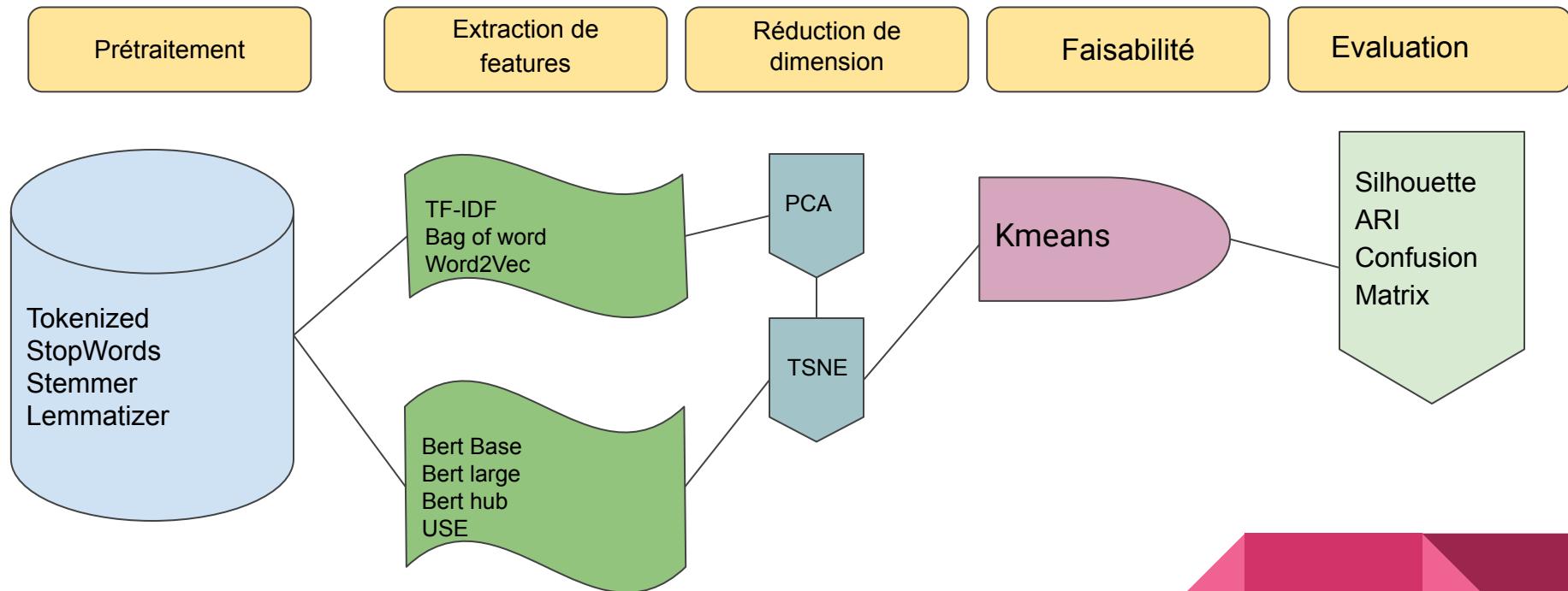
Objectifs:

- Améliorer l'expérience des utilisateurs.
- Fiabiliser la catégorie des articles avec pertinence et précision.

Mission :

Réaliser une étude de faisabilité d'un moteur de classification d'articles basé sur une image et une description pour l'automatisation de l'attribution de la catégorie de l'article pour l'entreprise Place de Marché.

Démarche traitement des textes



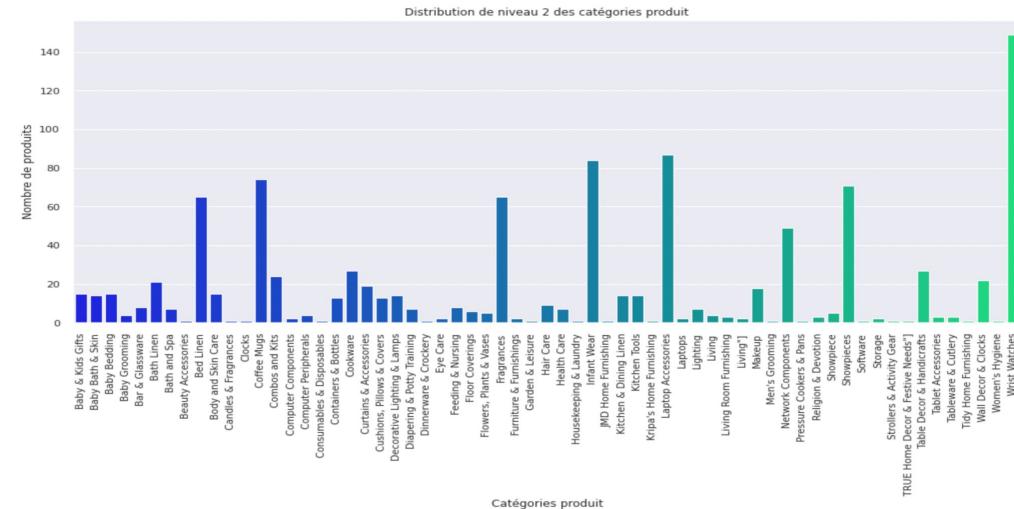
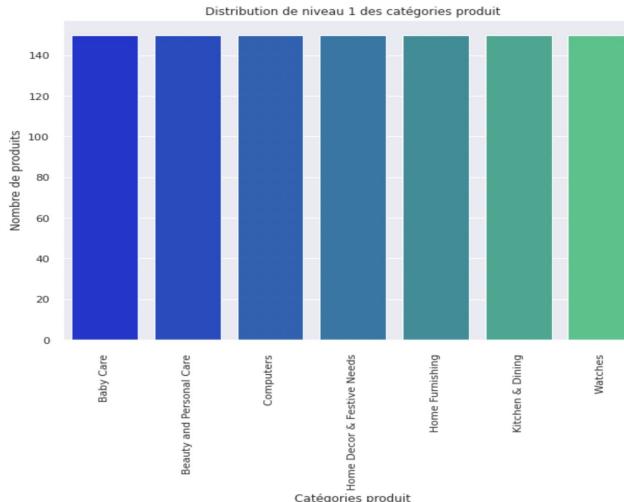
Découverte des données

Récupération des données : Le jeu de données est composé :

- D'un fichier CSV « flipkart_com-ecommerce_sample_1050.csv » : 1050 lignes et 15 colonnes.
- D'un dossier « Images » : 1050 images des produits mentionnés dans le fichier CSV.
- Les columns : « 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»

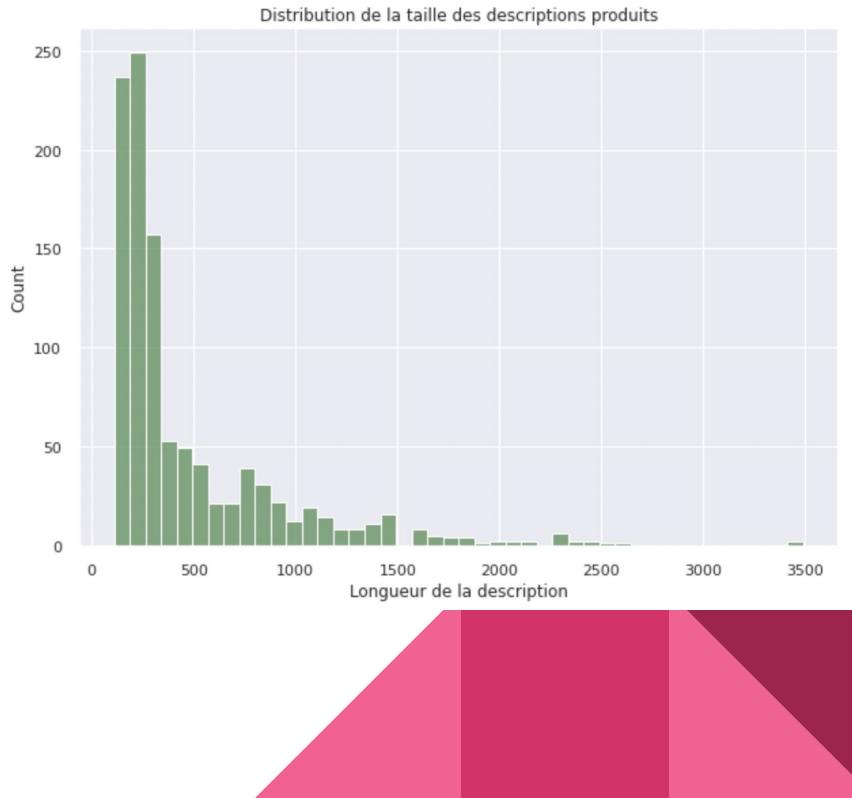
Prétraitement des données

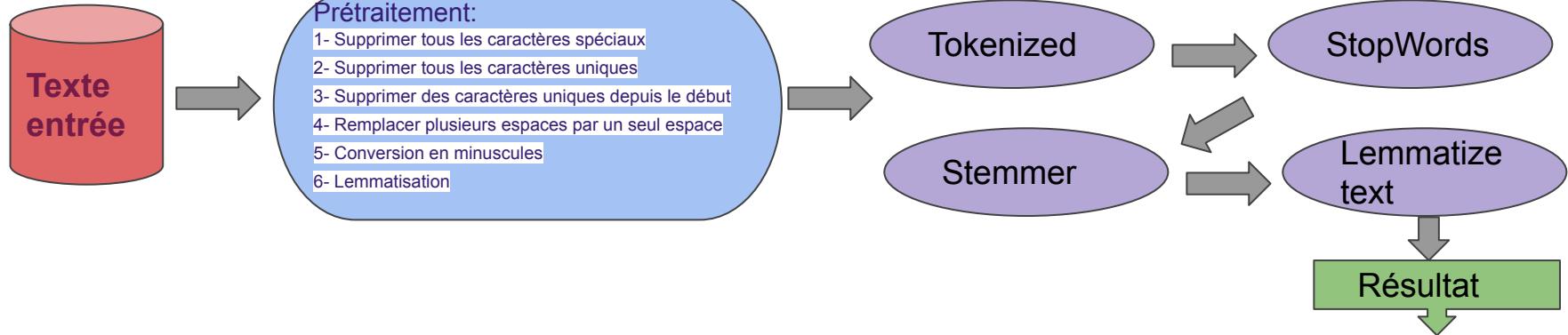
- Extraction de la catégorie à partir de product_category_tree(7 catégories, 62 catégories secondaires)



NLP: Traitement des données textuelles

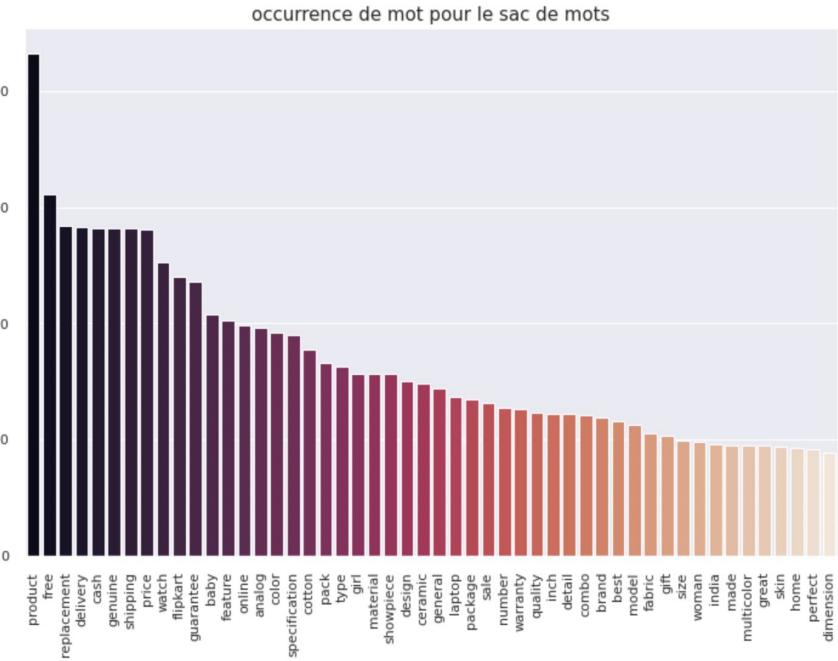
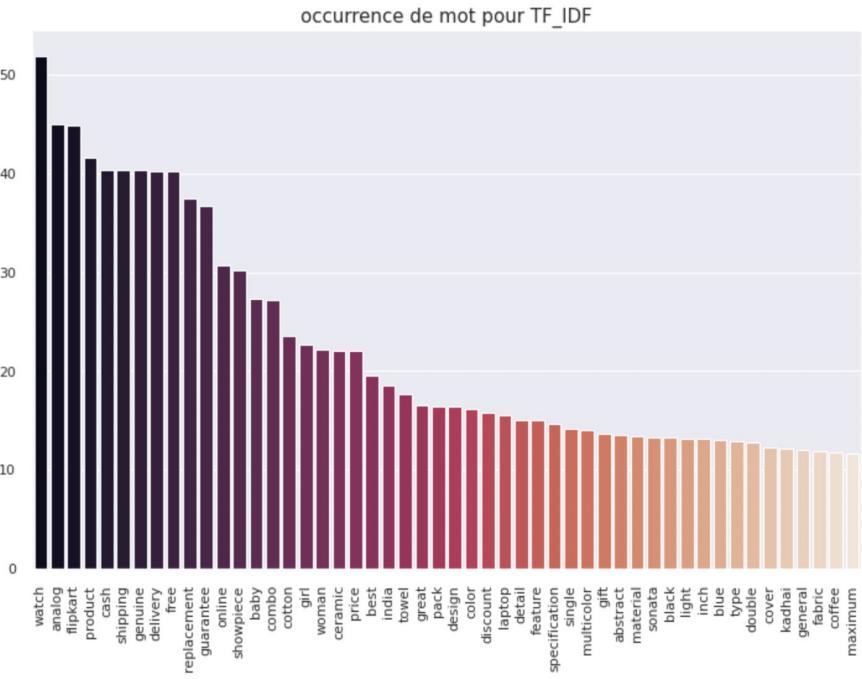
- ❑ Concaténation product_name + description
- ❑ Prétraitement du texte (Tokenization, stopwords, ponctuation, racine, caractères alphas, lower, Stemming (PorterStemmer, snowballStemmer, Lancaster Stemmer, pos_tag / DefaultTagger, UnigramTagger, BigramTagger, Lemmatization))
- ❑ Text Features Extraction (Uni-gram, Bi-gram, Tri-gram)
- ❑ Encodage du texte avec Bag of words et TF-IDF
- ❑ Exploiter une approche se basant sur les word vectors pour encoder le document (word2vec)
- ❑ Extraction des features avec un modèle BERT entraîné
- ❑ Extraction des features avec un modèle USE entraîné
- ❑ Étude de la faisabilité (pour chacune des méthodes d'extraction de descripteurs) :
 - ❑ Visualisation TSNE des descripteurs textuels + coloration des points par catégorie
 - ❑ KMeans sur les descripteurs textuels + indice de rand





corpus	tokenized	stopwords_removed	porter_stemmer	snowball_stemmer	lancaster_stemmer	combined_positag_wnet	lemmatize_word_wo_pos	lemmatize_word_w_pos	lemmatize_text
elegance polyester multicolor abstract eyelet ...	[elegance, polyester, multicolor, abstract, ey...]	[elegance, polyester, multicolor, abstract, ey...]	[eleg, polyest, multicolor, abstract, eyelet, ...]	[eleg, polyest, multicolor, abstract, eyelet, ...]	[eleg, polyest, multicol, abstract, eyelet, do...]	((elegance, n), (polyester, n), (multicolor, n...)	[elegance, polyester, multicolor, abstract, ey...]	[elegance, polyester, multicolor, abstract, ey...]	elegance polyester multicolor abstract eyelet ...
sathiya... cotton bath towel.specifications of s...	[sathiya..., cotton, bath, towel.specifications,...]	[sathiya..., cotton, bath, towel.specifications,...]	[sathiya, cotton, bath, towel.specif., sathiya,...]	[sathiya, cotton, bath, towel.specif., sathiya,...]	[sathiya, cotton, bath, towel.specifications, ...]	((sathiya..., n), (cotton, n), (bath, n), (towel...)	[sathiya..., cotton, bath, towel.specifications,...]	[sathiya..., cotton, bath, towel.specifications,...]	sathiya... cotton bath towel.specifications sath...
eurospa cotton terry face towel set.key featur...	[eurospa, cotton, terry, face, towel, set.key,...]	[eurospa, cotton, terry, face, towel, set.key,...]	[eurospa, cotton, terri, face, towel, set.key,...]	[eurospa, cotton, terri, face, towel, set.key,...]	[eurosp, cotton, terry, fac, towel, set.key, f...]	((eurospa, n), (cotton, n), (terry, n), (face,...)	[eurospa, cotton, terry, face, towel, set.key,...]	[eurospa, cotton, terry, face, towel, set.key,...]	eurospa cotton terry face towel set.key featur...
santosh royal fashion cotton printed king size...	[santosh, royal, fashion, cotton, printed, kin...]	[santosh, royal, fashion, cotton, printed, kin...]	[santosh, royal, fashion, cotton, print, king,...]	[santosh, royal, fashion, cotton, print, king,...]	[santosh, roy, fash, cotton, print, king, siz,...]	((santosh, n), (royal, a), (fashion, n), (cott...)	[santosh, royal, fashion, cotton, printed, kin...]	[santosh, royal, fashion, cotton, print, king,...]	santosh royal fashion cotton print king sized ...
jaipur print cotton floral king sized double b...	[jaipur, print, cotton, floral, king, sized, d...]	[jaipur, print, cotton, floral, king, sized, d...]	[jaipur, print, cotton, floral, king, size, do...]	[jaipur, print, cotton, floral, king, size, do...]	[jaip, print, cotton, flor, king, siz, doubl, ...]	((jaipur, n), (print, n), (cotton, n), (flor...)	[jaipur, print, cotton, floral, king, sized, d...]	[jaipur, print, cotton, floral, king, sized, d...]	jaipur print cotton floral king sized double b...

Occurrence de mot pour TF_IDF et BOW



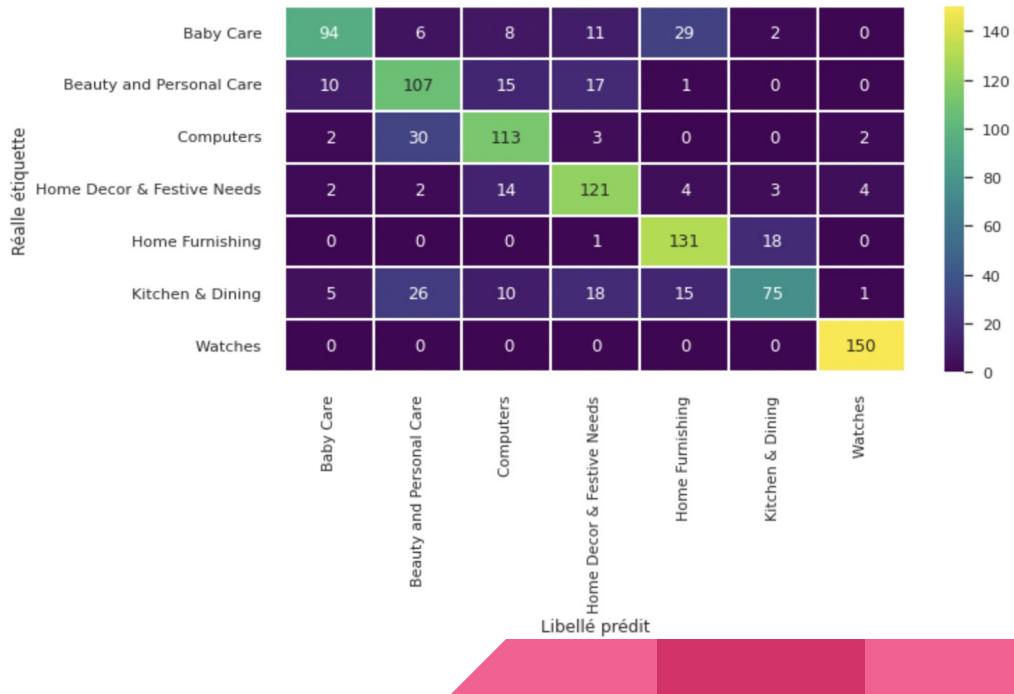
Faisabilité Encodage TF_IDF

- Réduction de dimension(PCA/T-SNE)
- Métriques(ARI, Silhouette, Confusion Matrix)



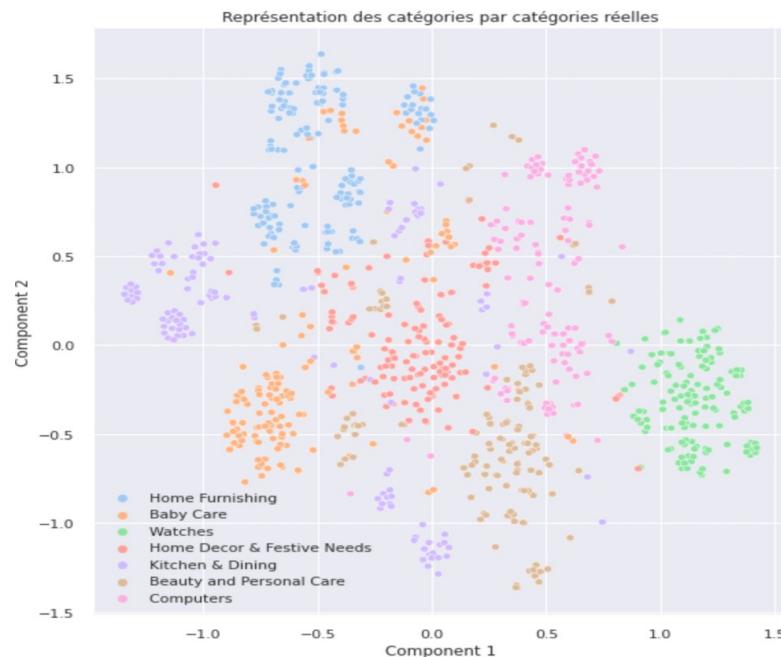
```
ARI_tfidf_pca_tsne = 0.538030699981047
```

Correspondance des clusters : [1 4 3 5 2 6 0]



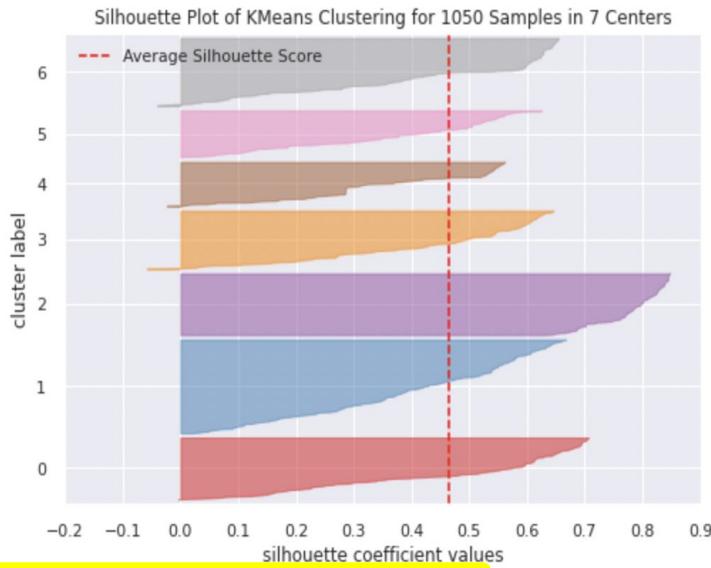
Représentation 2D de notre corpus par catégorie _ TFIDF

TSNE visualization _tf-idf



Faisabilité Encodage Bag of words

- Réduction de dimension(PCA/T-SNE)
- Métriques(ARI, Silhouette, Confusion Matrix)



ARI_bow_pca_tsne = 0.38893726310068794

Correspondance des clusters : [3 2 6 1 5 2 0]

Réelle étiquette	Baby Care	3	35	10	0	3	0
Beauty and Personal Care	0	89	48	11	0	2	0
Computers	0	18	132	0	0	0	0
Home Decor & Festive Needs	0	1	38	106	0	5	0
Home Furnishing	65	30	31	0	0	24	0
Kitchen & Dining	0	0	55	22	0	73	0
Watches	0	0	1	0	0	0	149

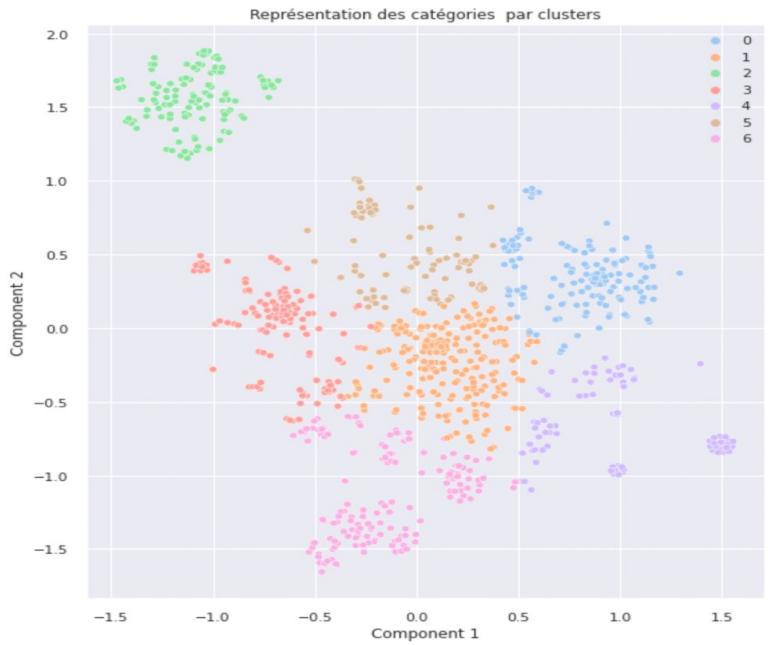
Libellé prédit

Baby Care
Beauty and Personal Care
Computers
Home Decor & Festive Needs
Home Furnishing
Kitchen & Dining
Watches

140
120
100
80
60
40
20
0

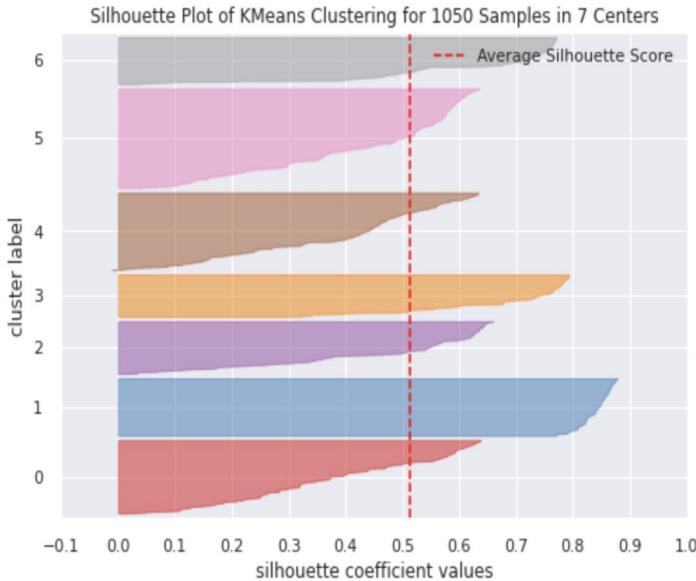
Représentation 2D de notre corpus par catégorie _ BOW

TSNE visualization _bow



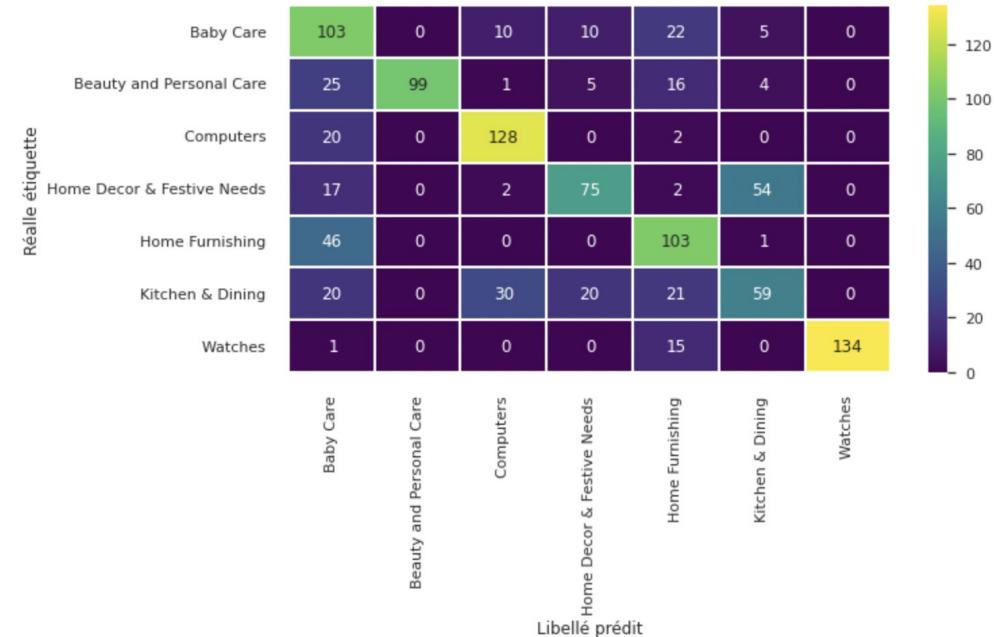
Faisabilité Encodage Word2Vec

- Réduction de dimension(PCA/T-SNE)
- Métriques(ARI, Silhouette, Confusion Matrix)

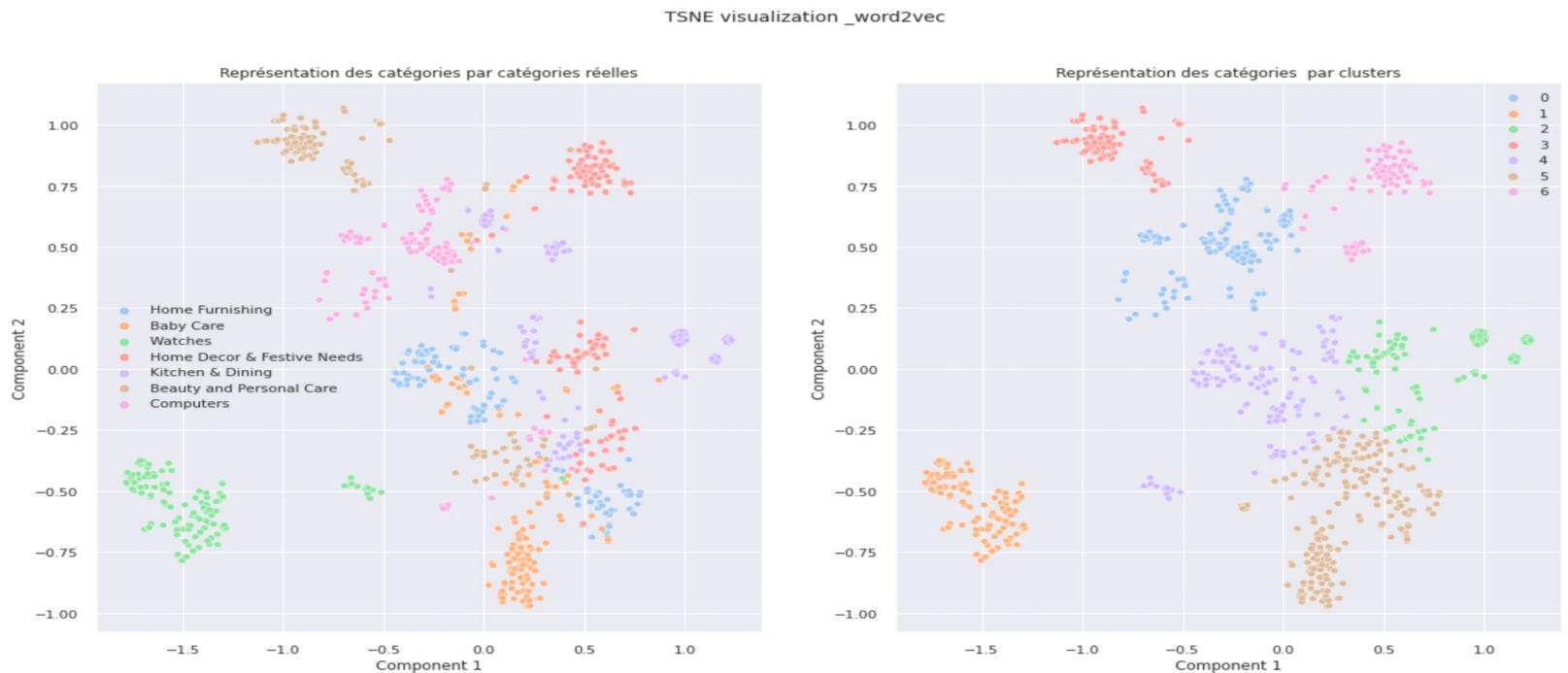


ARI_word2vec_tsne = 0.4247336157332022

Correspondance des clusters : [2 6 5 1 4 0 3]



Représentation 2D de notre corpus par catégorie _ Word2Vec



Faisabilité Encodage Modèle BERT

Bert-base-uncased

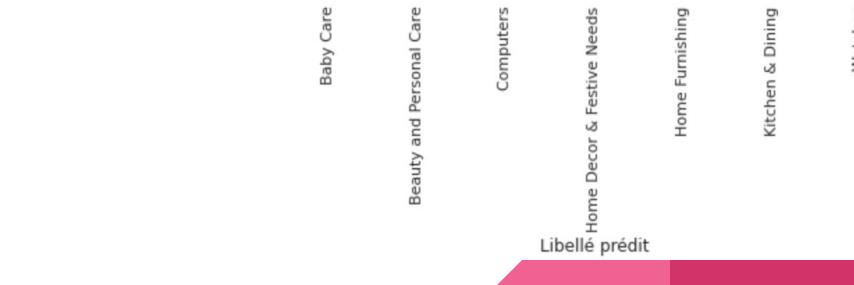
- Réduction de dimension(T-SNE)
- Métriques(ARI, Silhouette, Confusion Matrix)



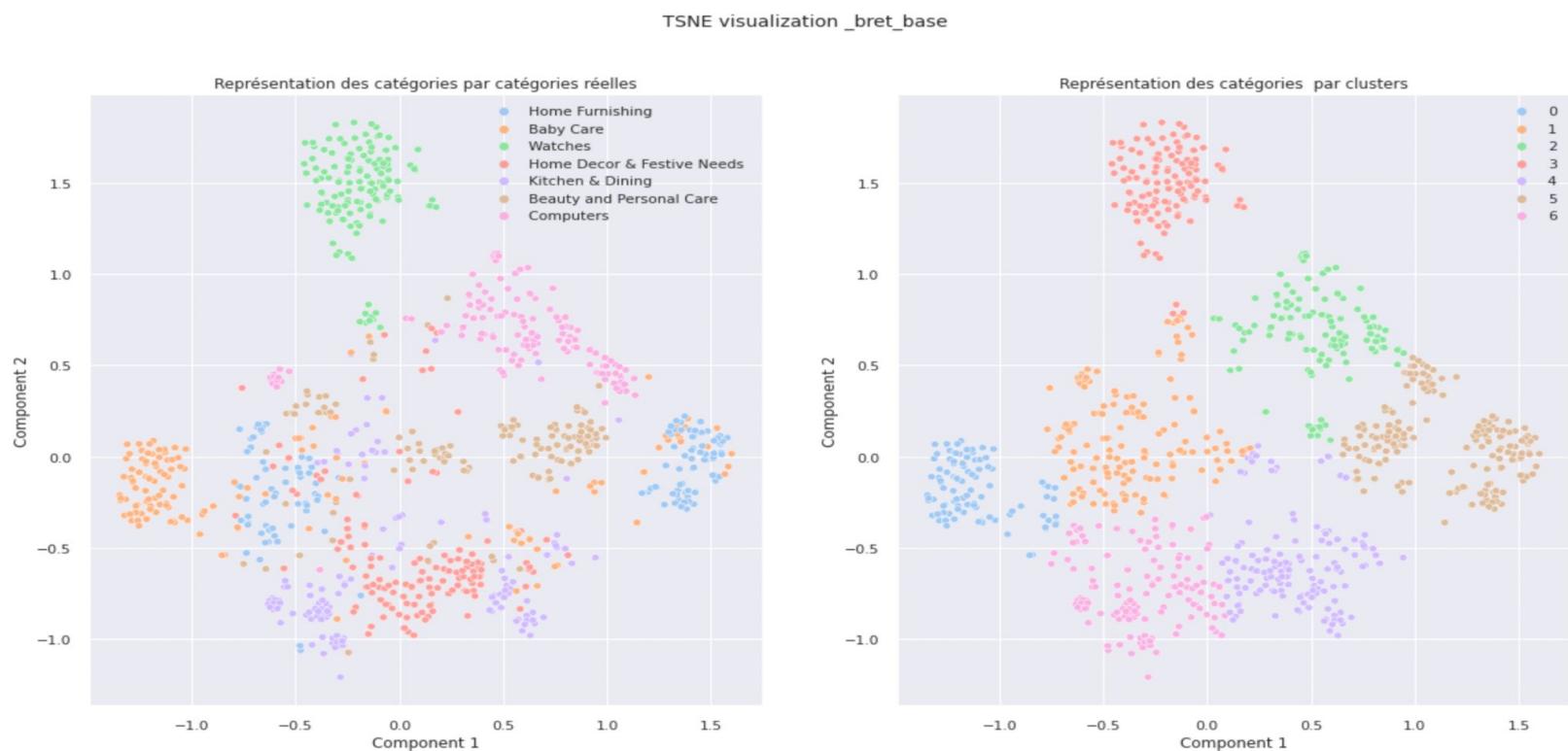
ARI_bret_base_tsne = 0.3672071027784623

Correspondance des clusters : [0 4 2 6 3 4 5]

Réelle étiquette	Baby Care	0	0	14	41	3	0
Baby Care	92	0	0	14	41	3	0
Beauty and Personal Care	0	0	12	28	103	7	0
Computers	0	0	110	0	39	1	0
Home Decor & Festive Needs	1	0	8	70	19	52	0
Home Furnishing	8	0	0	0	122	20	0
Kitchen & Dining	0	0	2	49	28	71	0
Watches	0	0	0	0	7	0	143

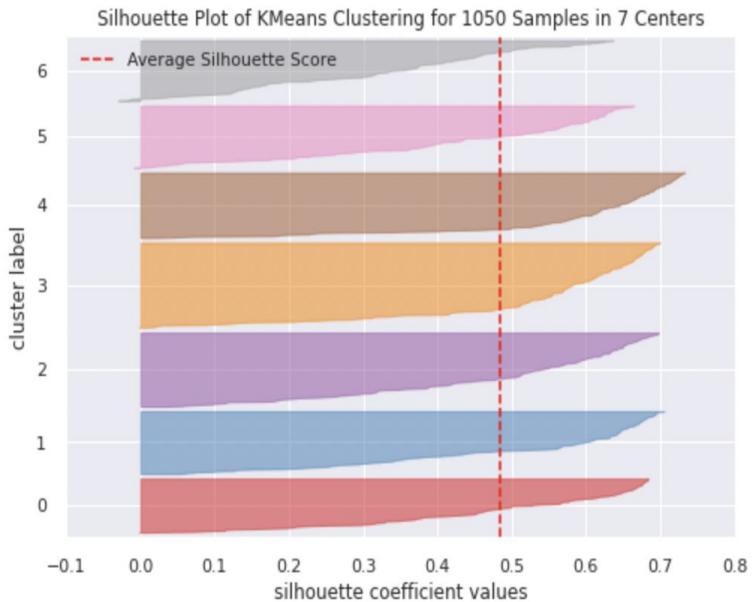


Représentation 2D de notre corpus par catégorie _ Bert Base



Faisabilité Encodage Bert-large-uncased

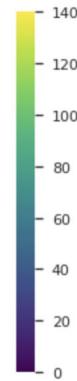
- Réduction de dimension(T-SNE)
- Métriques(ARI, Silhouette, Confusion Matrix)



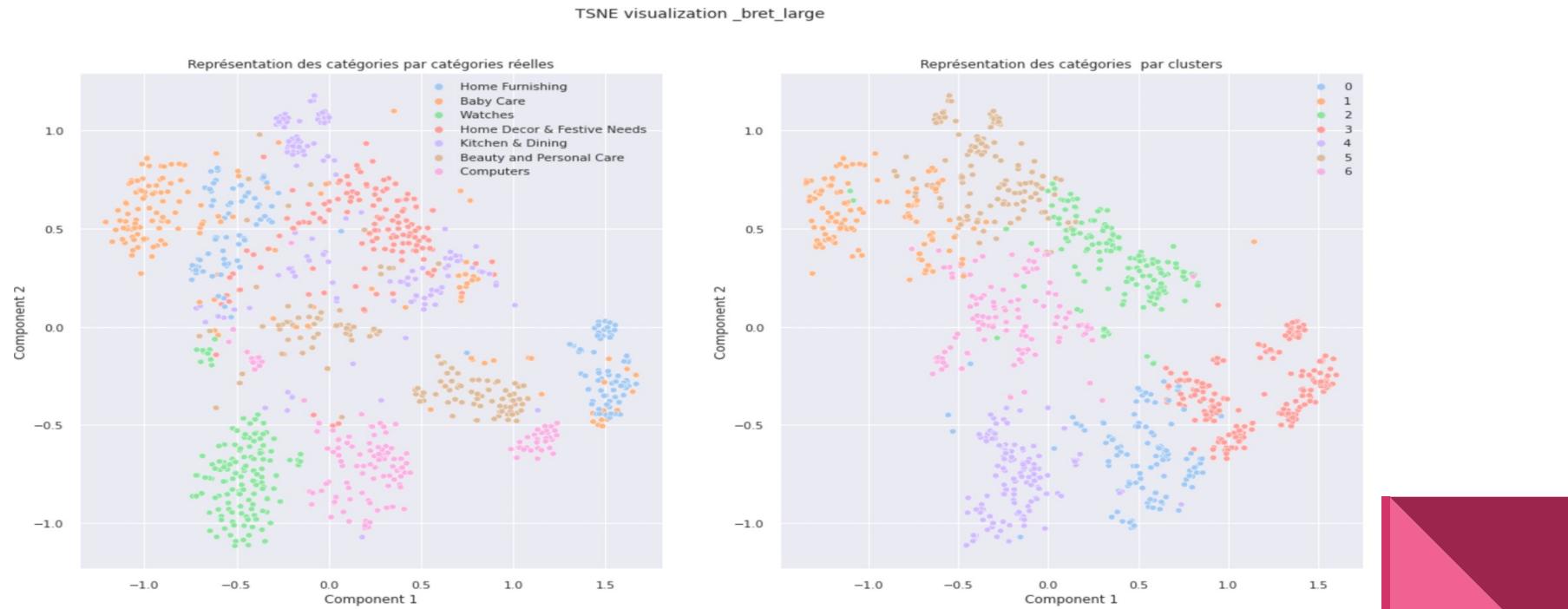
ARI_bret_large_tsne = 0.3565586281863283

Correspondance des clusters : [2 0 3 4 6 5 1]

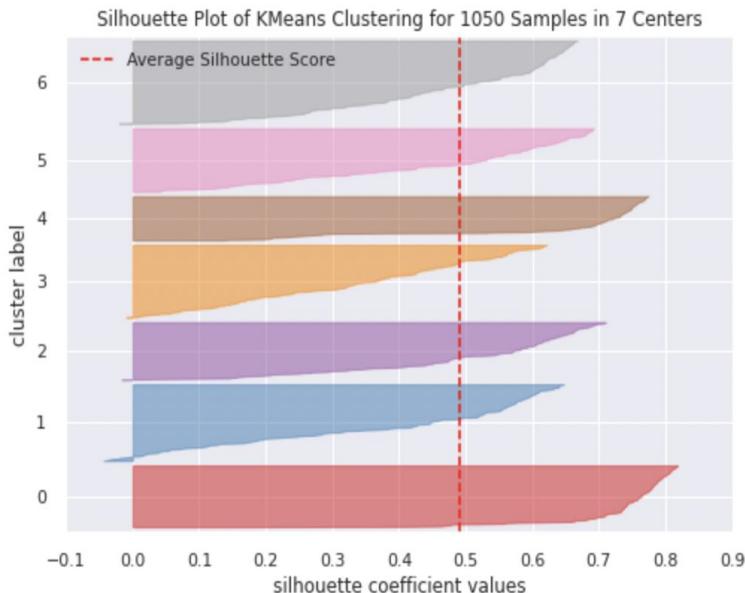
Reelle étiquette	Baby Care	10	1	16	21	7	0
Baby Care	95	10	1	16	21	7	0
Beauty and Personal Care	1	47	16	17	62	6	1
Computers	0	14	99	0	32	1	4
Home Decor & Festive Needs	0	17	3	85	0	45	0
Home Furnishing	46	10	0	0	74	20	0
Kitchen & Dining	0	29	3	49	4	62	3
Watches	0	10	0	0	0	0	140



Représentation des catégories par cluster et catégories réelles - Bert large



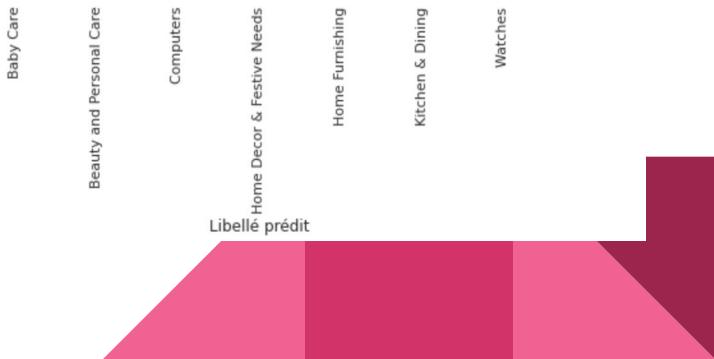
Faisabilité Encodage BERT hub tensorflow



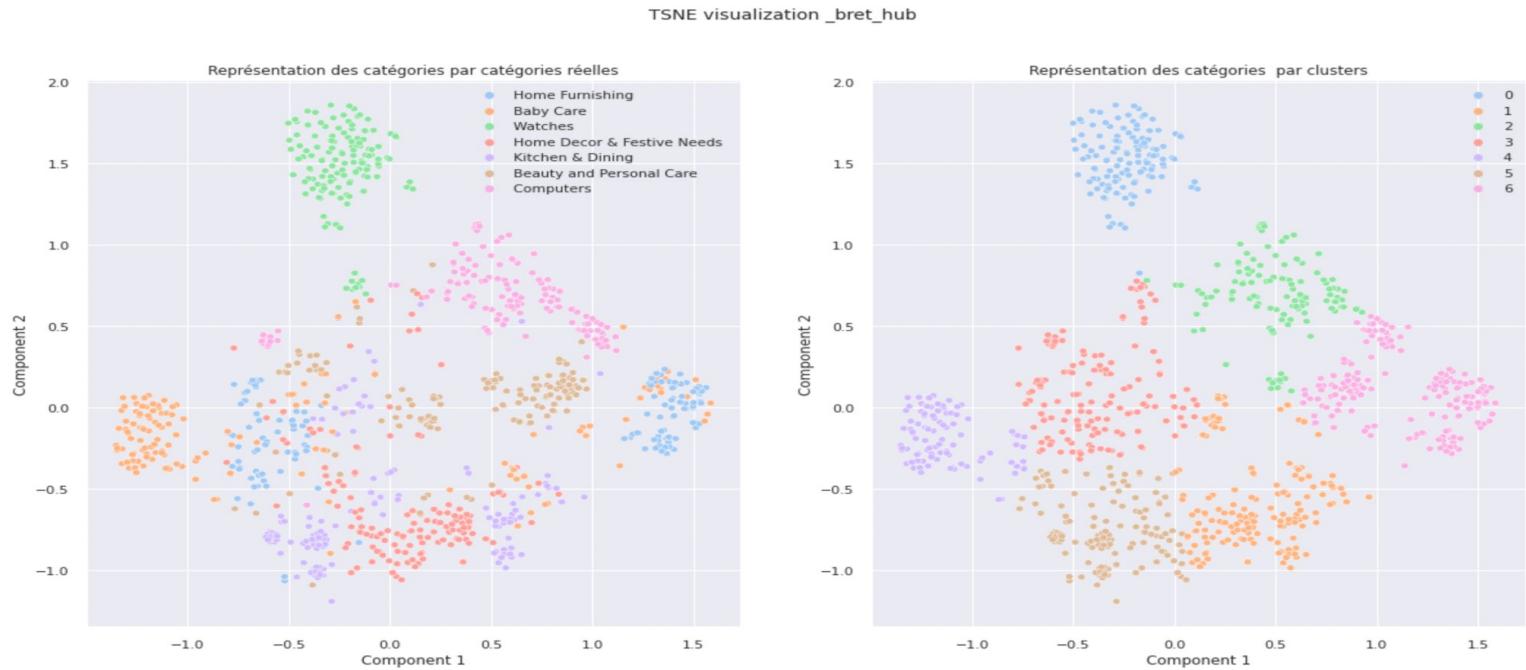
ARI_bret_hub_tsne = 0.3611075085503007

Correspondance des clusters : [6 3 2 4 0 5 4]

Réelle étiquette	Baby Care	0	0	15	40	3	0
Beauty and Personal Care	0	0	12	32	99	7	0
Computers	0	0	109	0	40	1	0
Home Decor & Festive Needs	1	0	8	75	19	47	0
Home Furnishing	8	0	0	0	122	20	0
Kitchen & Dining	0	0	2	53	28	67	0
Watches	0	0	1	0	8	0	141



Représentation des catégories par cluster et catégories réelles - Bert Hub



Faisabilité Encodage USE

- Réduction de dimension(T-SNE)
- Métriques(ARI, Silhouette, Confusion Matrix)



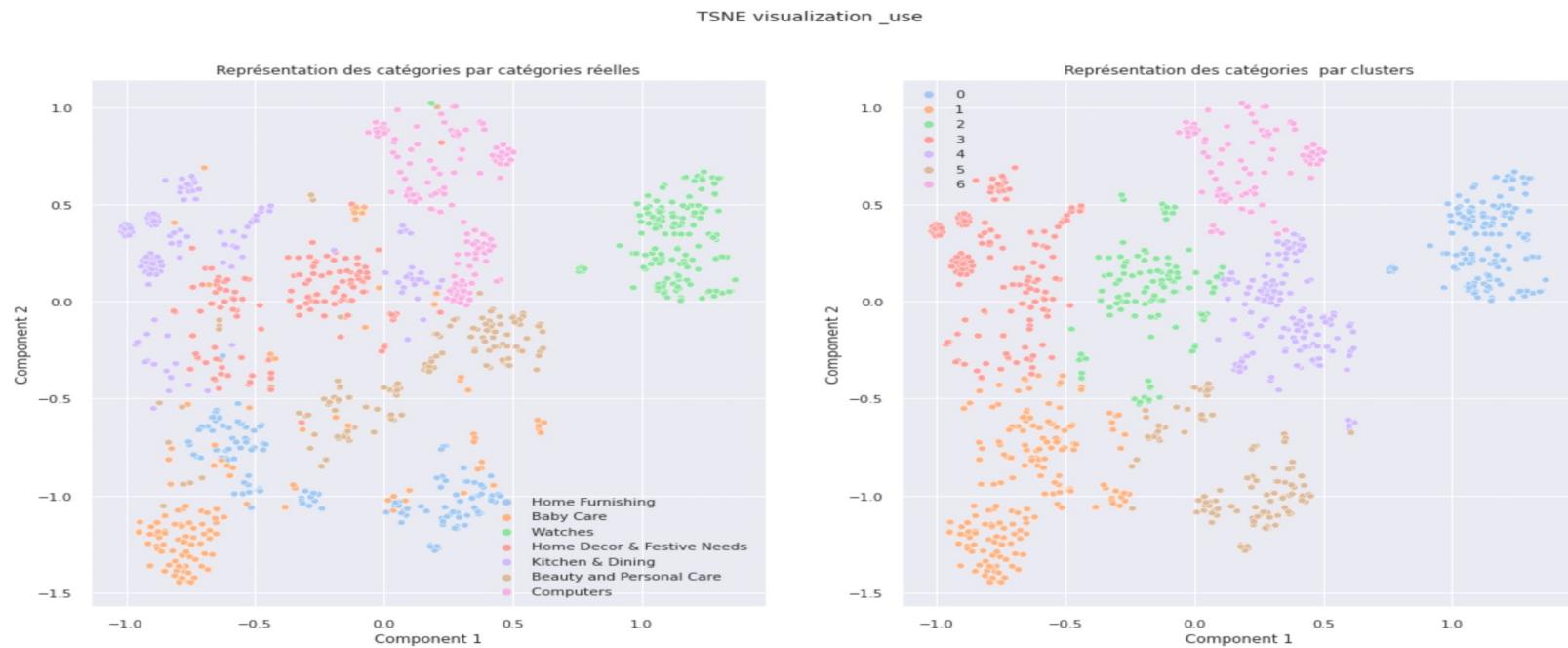
```
ARI_use_tsne = 0.4799329104142017
```

Correspondance des clusters : [6 0 3 5 1 4 2]

Réelle étiquette	Baby Care	104	10	0	13	20	3	0
	Beauty and Personal Care	14	80	2	16	35	3	0
	Computers	0	48	102	0	0	0	0
	Home Decor & Festive Needs	10	1	1	87	0	51	0
	Home Furnishing	74	0	0	0	75	1	0
	Kitchen & Dining	5	13	6	10	0	116	0
	Watches	0	0	1	0	0	0	149

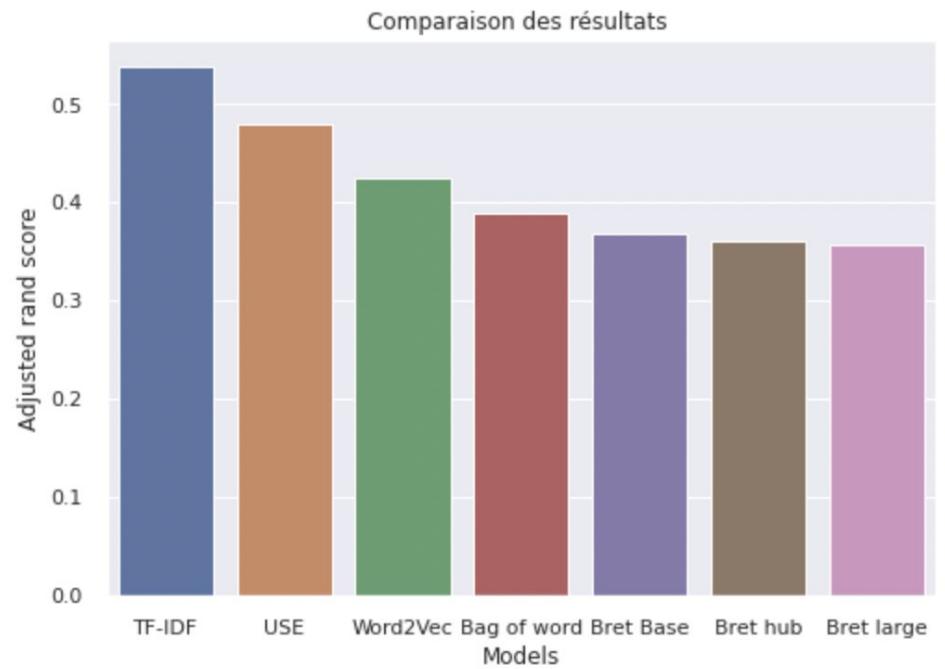
Libellé prédit

Représentation 2D de notre corpus par catégorie USE



Comparaison des résultats

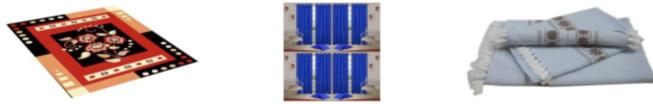
model_name	adjusted_rand_score
0 TF-IDF	0.538031
1 Bag of word	0.388937
2 Word2Vec	0.424734
3 Bret Base	0.367207
4 Bret large	0.356559
5 Bret hub	0.361108
6 USE	0.479933



Traitement des données images

Exemple d'images par catégorie

Home



Baby



Watches



Decor



Kitchen



Beauty



Computers



Démarche traitement des images

Prétraitement

Extraction de features

Réduction de dimension

Faisabilité

Evaluation

Prétraitement

Conversion RGB to GRAY
Egaliser l'histogramme de l'image
Détection des points clés

SIFT



CNN Transfer Learning:
EfficientNetB0
VGG-16
ResNet50

K Keras

PCA

TSNE

MiniBatchKMeans
Kmeans

Grid
Search

Silhouette
ARI
Confusion
Matrix

Meilleure correspondance des clusters aux catégories « vraies »

Processus de transformation et classification des images

Création des clusters de descripteurs

- Utilisation de MiniBatchKMeans pour obtenir des temps de traitement raisonnables

Création des features des images

- Pour chaque image :
- prédiction des numéros de cluster de chaque descripteur
- création d'un histogramme = comptage pour chaque cluster du nombre de descripteurs associés

Réduction de dimension PCA

- La réduction PCA permet de créer des features décorrélées entre elles, et de diminuer leur dimension, tout en gardant un niveau de variance expliquée élevé (99%)
- L'impact est une meilleure séparation des données via le T-SNE et une réduction du temps de traitement du T-SNE

Réduction de dimension T-SNE

- Réduction de dimension en 2 composantes T-SNE pour affichage en 2D des images

Analyse des clusters

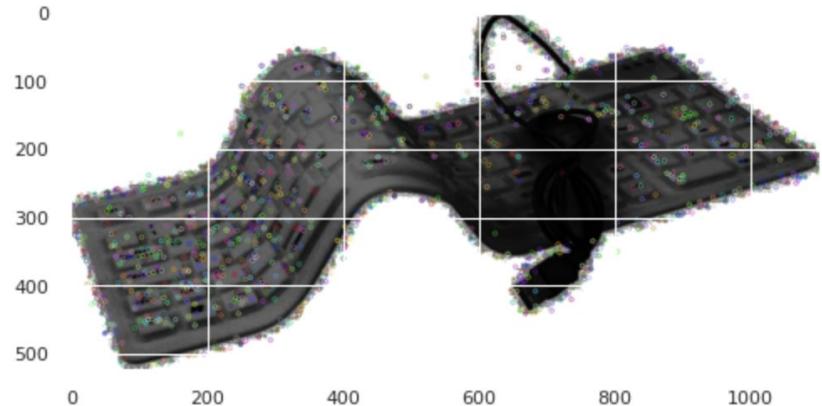
- Optimisation des modèles avec GridSearchCV

Détermination et affichage des descripteurs SIFT

Pré-traitement des images via SIFT

Créations des descripteurs de chaque image

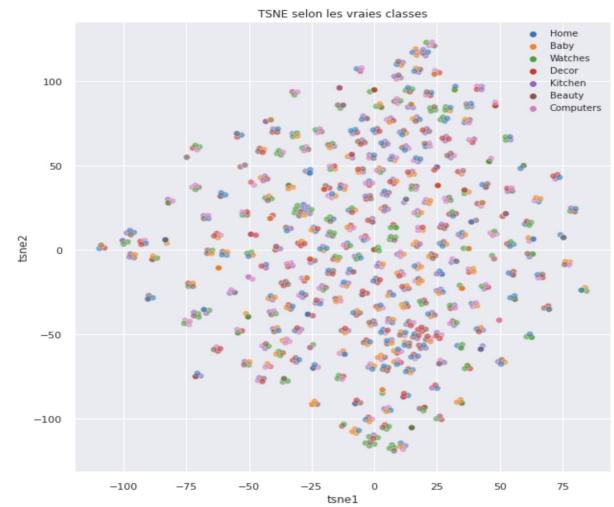
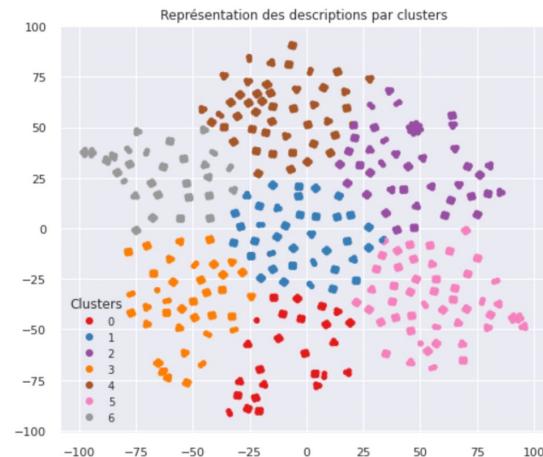
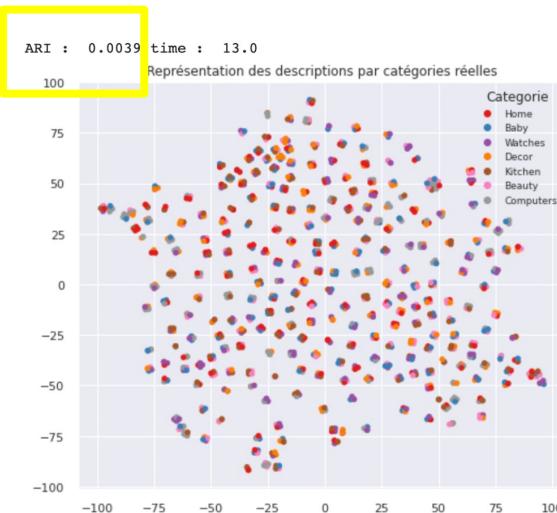
- Pour chaque image passage en gris et égalisation
- création d'une liste de descripteurs par image ("sift_keypoints_by_img") qui sera utilisée pour réaliser les histogrammes par image
- création d'une liste de descripteurs pour l'ensemble des images ("sift_keypoints_all") qui sera utilisé pour créer les clusters de descripteurs



Descripteurs : (1464, 128)

```
[[ 1.  16. 117. ...   0.  22.  76.]  
 [ 0.   0.   11. ...  12.   1.   1.]  
 [ 5.  20.  20. ...   0.   0.  10.]  
 ...  
 [ 83.  68.  10. ...   0.   0.   8.]  
 [ 0.   0.   0. ...   3.   8.  15.]  
 [ 10.   1.   1. ...   0.   0.   4.]]
```

Analyse visuelle : affichage T-SNE selon catégories d'images



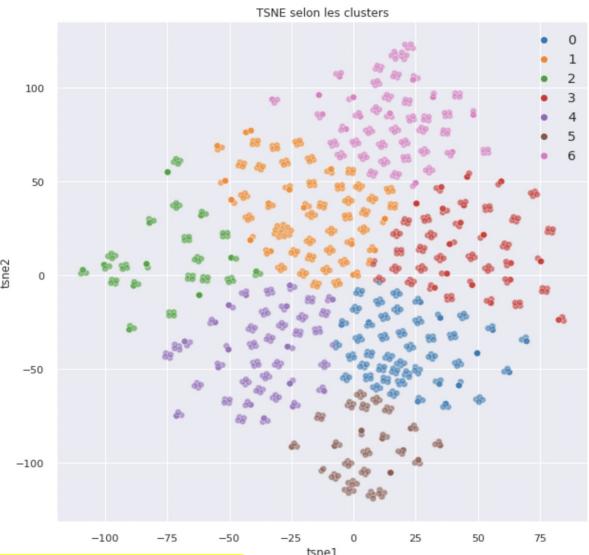
Analyse mesures SIFT: similarité entre catégories et clusters

Création de clusters à partir du T-SNE

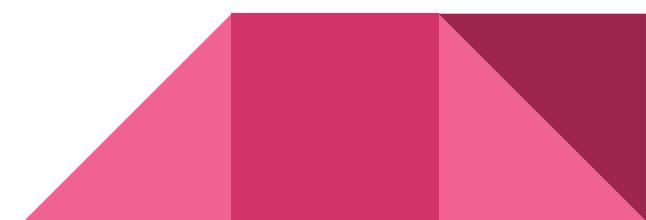
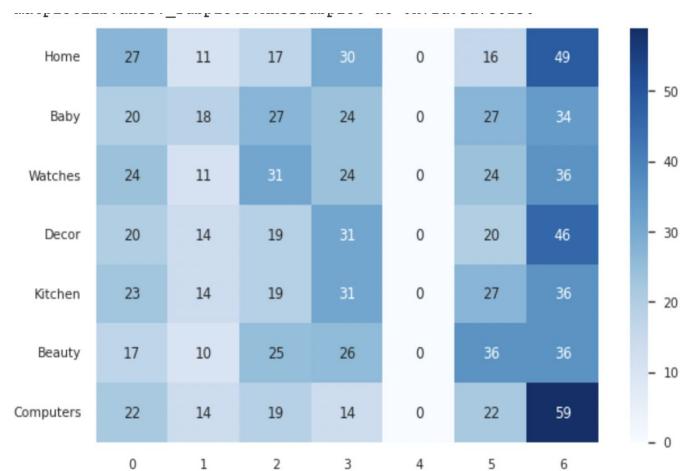
	tsne1	tsne2	class	cluster
0	60.044685	15.318463	Home	3
1	34.907322	-22.742502	Baby	0
2	37.115364	-68.522888	Baby	0
3	-28.270407	26.936005	Home	1
4	-27.400681	23.170702	Home	1
...
1045	32.896133	-90.189346	Baby	5
1046	-34.275818	36.261627	Baby	1
1047	3.202540	-82.891602	Baby	5
1048	-39.094593	-61.672245	Baby	4
1049	-30.648365	67.502457	Baby	1

1050 rows x 4 columns

Affichage des images selon clusters et calcul ARI de similarité catégories images /cluster



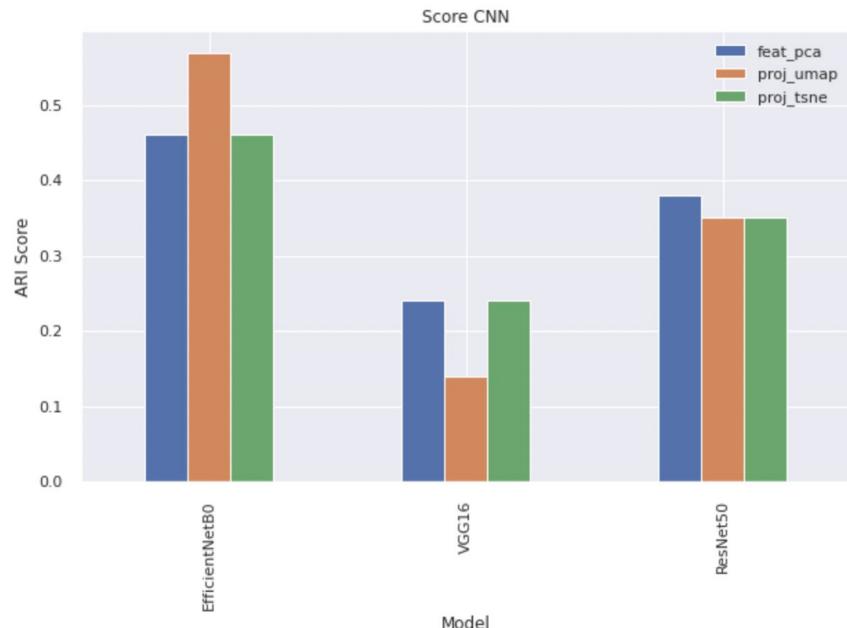
Analyse par classes /La matrice de confusion



Transfer learning CNN

- ❖ EfficientNetB0
- ❖ VGG-16
- ❖ ResNet50

	feat_pca	proj_umap	proj_tsne
EfficientNetB0	0.46	0.57	0.46
VGG16	0.24	0.14	0.24
ResNet50	0.38	0.35	0.35

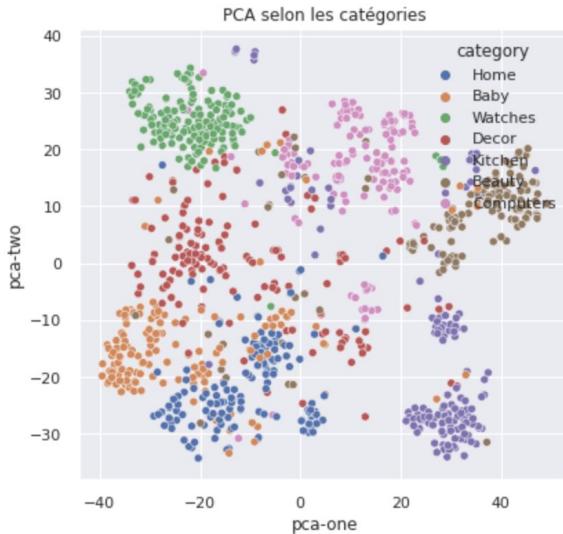


CNN tsne cluster et matrix confusion

Analyse des clusters réelles

ARI score mean : 0.586466981438084

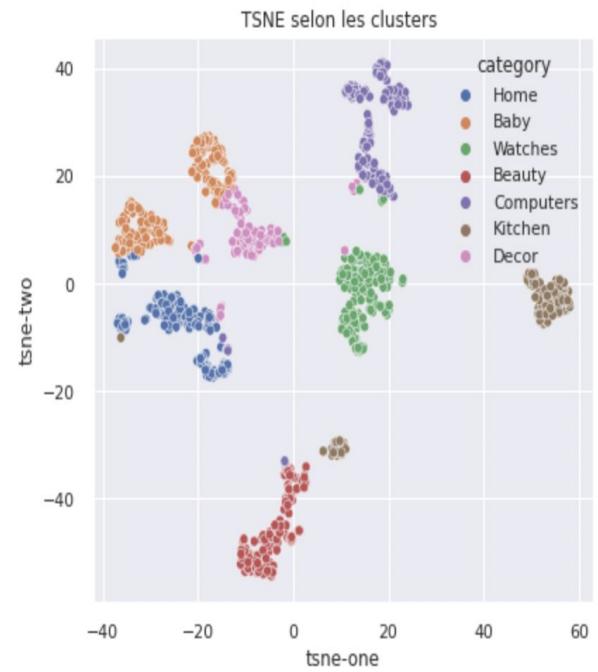
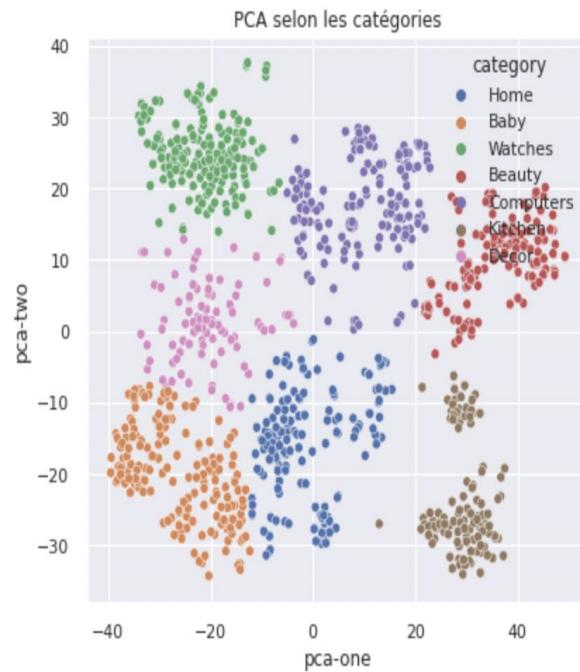
Best parameters: {'algorithm': 'auto', 'init': 'random', 'max_iter': 50, 'n_init': 5, 'random_state': 20}



CNN tsne cluster prédition transform

Metrics classification report

	precision	recall	f1-score	support
0	0.52	0.56	0.54	150
1	0.84	0.96	0.89	150
2	0.80	0.55	0.65	150
3	0.61	0.72	0.66	150
4	0.82	0.81	0.82	150
5	0.76	0.85	0.80	150
6	0.92	0.73	0.81	150
accuracy			0.74	1050
macro avg	0.75	0.74	0.74	1050
weighted avg	0.75	0.74	0.74	1050



Conclusion

La partie textuelle :

- Traitement du texte : La lemmatisation est le meilleur choix.
- Extraction des features : TF-IDF et USE ont donné des résultats acceptables (53% et 47%)
- Réduction de dimension : Les résultats du clustering sont présentés sous la forme d'une représentation en deux dimensions, qui a illustré le fait que les caractéristiques extraites permettent de regrouper des produits de même catégorie.
- Le TSNE améliore nettement l'ARI score.

La partie visuelle :

- Le Transfer Learning CNN: EfficientNetB0 a donné le meilleur score (46%)

Partitionnement des articles

- Présence de clusters bien distincts et caractérisés
- Bonne correspondance des clusters aux catégories « vraies »

Améliorations

- ❑ La combinaison des données textuelles et visuelles
- ❑ Améliorer les descriptions des produits en utilisant des mots clés
- ❑ Vérifier les étiquettes avant d'envisager d'utiliser l'apprentissage supervisé(erreurs entre les catégories home decor et Home Furnishing)

Questions



Bonus

La classification de texte (traitement du langage naturel) a été effectuée à l'aide de:

- Bag of Words (BoW) : comptage de mots et vectorisation TF-IDF, avec n-grammes
- Modélisation de sujets à l'aide de l'allocation Dirichlet latente
- Incorporation de mots à l'aide de modèles pré-formés Word2Vec
- Incorporation de mots utilisant l'apprentissage en profondeur (skigrammes contextuels dans les réseaux de neurones LSTM: BERT, transformateurs HuggingFace, Universal Sentence Encoder)
- Incorporation de mots Keras (supervisée): scission entraînement-test, démontrant le surajustement des données d'entraînement.

Image classification (Computer Vision) was performed:

- image feature extraction : bag of visual features (SIFT, ORB) ; Visual feature vectors
- supervised training on simple Convolution Neural Networks (CNN)
- semi-supervised on VGG16 pretrained (ImageNet - 1000 features)
- unsupervised (on VGG16 features minus 2 layers)
- supervised transfer learning, with fine-tuning
- regularization through the use of image augmentation and dropout layers

Word Embedding (Word2Vec) L'incorporation de mots à l'aide de word2vec est basée sur des skipgrams: les mots trouvés à proximité les uns des autres ont tendance à être étroitement liés et auront donc des vecteurs de caractéristiques similaires

Incorporation de phrases à l'aide de BERT the Universal Sentence Encoder (USE)

Les modèles BERT (Bidirectional Encoder Representations from Transformers) et USE ont été testés en fournissant des descriptions non lemmatisées aux modèles pré-entraînés.

Bien qu'il s'agisse de modèles d'apprentissage en profondeur et qu'il faille du temps pour traiter les mots intégrés, les résultats étaient moins impressionnantes que les modèles de texte plus simples.

Cela peut être dû au fait que les descriptions de produits ne sont généralement pas des phrases, mais sont souvent générées à partir de paires clé-valeur de caractéristiques de produit. En utilisant les skipgrams, les clés telles que {couleur, longueur, largeur, hauteur, quantité,...} peuvent ajouter du bruit plutôt que du contexte. En revanche, ces mots ont peu de poids dans la vectorisation TF-IDF.

CNN : composé de 2 couches de convolution (avec maxpooling), une couche de décrochage pour la régularisation, une couche d'aplatissement et 2 couches denses a été utilisé pour tester rapidement les pipelines de prétraitement et évaluer l'effet de la régularisation (~ 1 million de paramètres, temps d'entraînement de quelques secondes).

Pour ce problème particulier, de meilleurs résultats ont été obtenus à partir de modèles d'apprentissage en profondeur CNN, pré-entraînés sur des millions d'images

VGG16 : Le modèle pré-entraîné VGG-16 (poids ImageNet) a été utilisé pour détecter la probabilité que chaque image appartienne à une catégorie donnée.

Transfer Learning(classification supervisée) : les couches denses ont été supprimées et remplacées par une couche d'aplatissement et de nouvelles couches denses, ainsi qu'une dernière fonction softmax permettant de choisir entre les 7 catégories.

les couches de convolution ont été conservées et leurs poids pré-formés ont été gelés pour éviter de perdre les caractéristiques d'image pré-formées un réglage fin a été appliqué en ajustant uniquement les poids dans les nouvelles couches denses et softmax, tout en gelant les poids pré-formés dans la couche convective l'entropie croisée catégorielle a été utilisée comme fonction de perte l'algorithme d'optimisation d'Adam a été utilisé pour une optimisation rapide (une extension de la descente de gradient stochastique)

Le modèle ne demande qu'un prétraitement spécifique qui consiste à soustraire la valeur RGB moyenne, calculée sur l'ensemble d'apprentissage, de chaque pixel.

Durant l'apprentissage du modèle, l'input de la première couche de convolution est une image RGB de taille 224 x 224. Pour toutes les couches de convolution, le noyau de convolution est de taille 3x3: la plus petite dimension pour capturer les notions de haut, bas, gauche/droite et centre. C'était une spécificité du modèle au moment de sa publication. Jusqu'à VGG16 beaucoup de modèles s'orientaient vers des noyaux de convolution de plus grande dimension (de taille 11 ou bien de taille 5 par exemple). Rappelons que ces couches ont pour but de filtrer l'image en ne gardant que des informations discriminantes comme des formes géométriques atypiques.

Ces couches de convolution s'accompagnent de couche de Max-Pooling, chacune de taille 2x2, pour réduire la taille des filtres au cours de l'apprentissage.

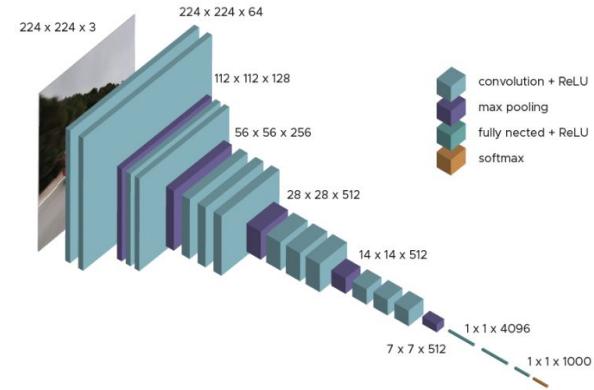
En sortie des couches de convolution et pooling, nous avons 3 couches de neurones Fully-Connected. Les deux premières sont composées de 4096 neurones et la dernière de 1000 neurones avec une fonction d'activation softmax pour déterminer la classe de l'image.

Comme vous avez pu le constater l'architecture est claire et simple à comprendre ce qui est aussi une force de ce modèle.



VGG16 Architecture

Architecture Algorithme VGG16



Structure Algorithme VGG16

Réduction de la dimensionnalité à l'aide de PCA et t-SNE

Les caractéristiques extraites (par exemple, les fréquences de mots) ont été réduites par l'analyse en composantes principales (ACP), en conservant 99% de la variance expliquée, avant d'appliquer l'incorporation de voisins stochastiques à distribution t (t-SNE) pour réduire à deux dimensions.

Nombre optimal de catégories (regroupement non supervisé à l'aide de Kmeans)

Le clustering K-means a été appliqué pour identifier les clusters, pour un nombre de clusters allant de 4 à 12

La classification automatique fonctionne mieux lorsque les catégories sont clairement séparées.

- coude de score de distorsion
- score de silhouette élevé
- faible score davies-bouldin

La classification non supervisée a produit les grappes les plus clairement séparées avec 7 catégories.

Évaluation du clustering semi-supervisé (k=7)

La performance de chaque modèle a été évaluée par la matrice de confusion multicatégorie, à partir de laquelle on peut calculer, pour chaque catégorie :

- précision
- rappeler
- précision

Ceux-ci peuvent être résumés dans le rapport de classification et visualisés dans un diagramme de Sankey

Indice Rand ajusté (ARI)

mesure de la similarité entre les catégories prévues et réelles

Dimension Reduction

PCA

- Composants qui explique la plus de variance de fréquences des mots
(réduction du bruit des mots non-commun entre produits de la même catégorie)

T-SNE

- Phrases avec similaire fréquences des mêmes mots

Bonus Evaluation

Précision

- Quelle portion du cluster prédit sont du vrai classe ?

$$\frac{TP}{TP + FP}$$

Recall

- Quelle portion du vrai classe sont présent dans le cluster prédit ?

$$\frac{TP}{TP + FN}$$

F1 Score

- accuracy « équilibré » :

$$2 * \frac{Precision * Recall}{Precision + Recall}$$

- Cluster prédit : TP + FP
- (Vrai classe : TP + FN)

