PROJET 6

« CLASSIFIEZ AUTOMATIQUEMENT DES BIENS DE CONSOMMATION »

Sommaire

1. Présentation
2. Pretraitement et clusterisation
    * Données textuelles
    * Données Visuelles
    * Modélisations effectuées
3. Données complémentaires : Edamam API
4. Conclusion

L’entreprise **Marketplace** souhaite lancer un marché de e-commerce.
Sur le marché, les vendeurs proposent des produits aux acheteurs en publiant une photo et une description. Actuellement, l’attribution des catégories de produits est effectuée manuellement par les vendeurs, elle n’est donc pas fiable. De plus, le volume d’articles est actuellement très faible. Pour rendre l'expérience utilisateur des vendeurs (facilitant la mise en ligne de nouveaux produits) et des acheteurs (facilitant la recherche de produits) aussi fluide que possible, ainsi qu'à des fins de mise à l'échelle, il devient nécessaire d'automatiser cette tâche.

Moyen : automatisation de l’attribution des catégories aux articles.

Objectif : améliorer l’expérience utilisateur et fiabiliser la catégorisation

But du projet : étudier la faisabilité de cette catégorisation :

- **Extraction de données depuis une API**
- **Analyse et prétraitement du jeu de données** : visuelles / textuelles
- **Clustering**

Jeu de données

- **1050 articles**
- **15 colonnes par article**:

1. Identifiant : id, nom, catégorie de produit, marque, description
2. Prix/prix soldé
3. Note du produit,
4. Image
5. etc.

categories = ['Home Furnishing',
'Baby Care',
'Watches',
'Home Decor & Festive Needs',
'Kitchen & Dining',
'Beauty and Personal Care',
'Computers']

Exemples d'articles:

Baby Care

Beauty and Personal Care

Computers

Home Decor & Festive Needs

Home Furnishing

Kitchen & Dining

Watches

Données textuelles : prétraitement

Nous utiliserons à la fois le nom du produit, le brand et la description textuelle pour maximiser le caractère informatif de nos textes: `df['combined'] = df['product_name'] + ' ' + df['brand'] + ' ' + df['description']`

Passer en minuscules → Tokenisation → Suppression des stopwords → Supprimer les mots qui n'ont pas de sens ou qui ne font pas partie de la langue analysée → Enlever la ponctuation et les chiffres → Supprimer plusieurs espaces → Lemmatisation /Stemmer

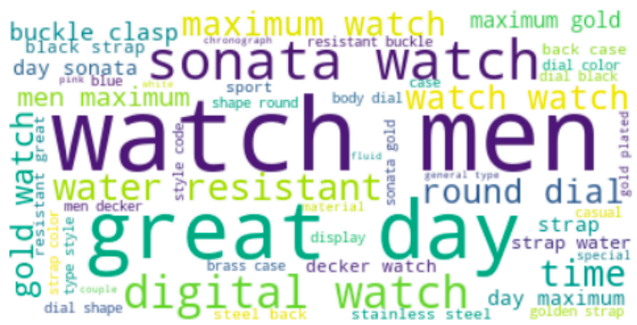
```
Buy Epresent Mfan 1 Fan USB USB Fan for Rs.219 online. Epresent Mfan 1 Fan USB USB Fan at best prices with FREE sh
ipping & cash on delivery. Only Genuine Products. 30 Day Replacement Guarantee.

======== LOWERCASE ========
buy epresent mfan 1 fan usb usb fan for rs.219 online. epresent mfan 1 fan usb usb fan at best prices with free sh
ipping & cash on delivery. only genuine products. 30 day replacement guarantee.

======== TOKENIZER ========
['buy', 'epresent', 'mfan', '1', 'fan', 'usb', 'usb', 'fan', 'for', 'rs.219', 'online', '.', 'epresent', 'mfan', '
1', 'fan', 'usb', 'usb', 'fan', 'at', 'best', 'prices', 'with', 'free', 'shipping', '&', 'cash', 'on', 'delivery',
'.', 'only', 'genuine', 'products', '.', '30', 'day', 'replacement', 'guarantee', '.']

======== STOPWORDS ========
['buy', 'epresent', 'mfan', '1', 'fan', 'usb', 'usb', 'fan', 'rs.219', 'online', 'epresent', 'mfan', '1', 'fan', '
usb', 'usb', 'fan', 'prices', 'free', 'shipping', 'cash', 'delivery', 'genuine', 'products', '30', 'day', 'replace
ment', 'guarantee']

======== LEMMATISATION ========
['buy', 'epresent', 'mfan', '1', 'fan', 'usb', 'usb', 'fan', 'rs.219', 'online', 'epresent', 'mfan', '1', 'fan', '
usb', 'usb', 'fan', 'price', 'free', 'shipping', 'cash', 'delivery', 'genuine', 'product', '30', 'day', 'replaceme
nt', 'guarantee']
```

WordCloud

Home Furnishing

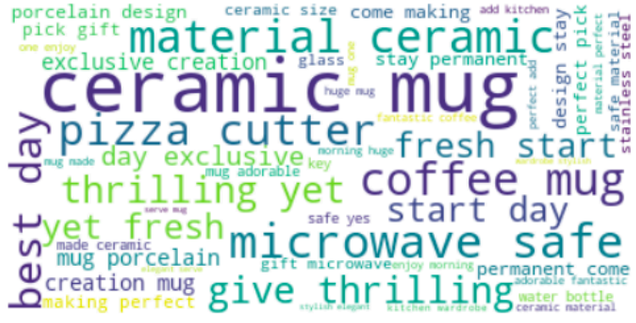
Baby Care

Watches

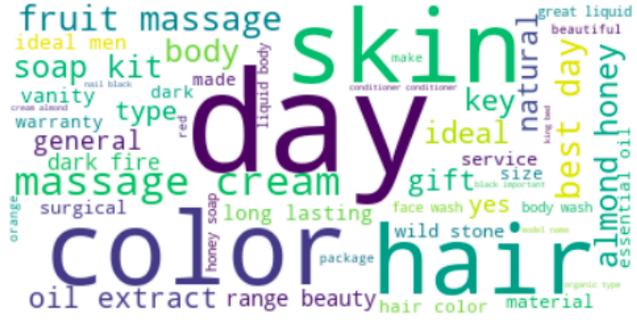


Home Decor & Festive Needs

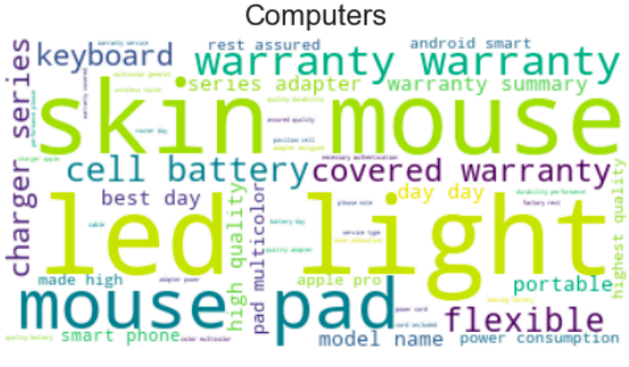
Kitchen & Dining



Beauty and Personal Care



Computers



Bag-of-Words

On se retrouve avec une matrice bow_df très clairsemée de 1050 lignes (produits) x 1454 colonnes (mots).

```
Bag-of-Words Representation:
   able  abode  absorbency  absorbent  abstract  accent  access  accessory  \
0     0      0           0          0         5       0       0          0   
1     0      0           0          0         0       0       0          0   
2     0      0           1          0         0       0       0          0   
3     0      0           0          0         0       0       0          0   
4     0      0           0          0         0       0       0          0   

   accident  accidental  according  across  act  actual  adapter  adaptor  \
0         0           0          0       0    0       0        0        0   
1         0           0          0       0    0       0        0        0   
2         0           0          0       0    0       0        0        0   
3         0           0          0       0    0       0        0        0   
4         0           0          0       0    0       0        0        0   

   addition  additional  adhesive  adjustable  admired  adorable  adorn  \
0         0           0         0           0        0         0      0   
1         0           0         0           0        0         0      0   
2         0           0         0           0        0         0      0   
3         0           0         0           0        0         0      0   
4         0           0         0           0        0         0      0   

   advice  aero  affect  affordable  age  air  alarm  allow  alloy  alluring  \
0       0     0       0           0    0    0      0      0      0         0   
1       0     0       0           0    0    0      0      0      0         0   
2       0     0       0           0    0    0      0      0      0         0   
3       0     0       0           0    0    0      0      0      0         0   
4       0     0       0           0    0    0      0      0      0         0   

   almond  along  also  alternative  aluminium  always  amazed  amazing  \
0       0      0     0            0          0       0       0        0   
1       0      0     0            0          0       0       0        0   
2       0      0     0            0          0       0       0        0   
3       0      0     0            0          0       0       0        0   
4       0      0     0            0          0       0       0        0   

   amount  android  animal  anna  another  ant  anti  antibacterial  antique  
0       1        0       0     0        0    1     1              0        0  
1       0        0       0     0        0    0     0              0        0  
2       0        0       0     0        0    0     0              0        0  
3       0        0       0     0        0    0     0              0        0  
4       0        0       0     0        0    0     0              0        0  
```

PCA Explained Variance (first 350 components): 97.21%

Kmeans (après PCA)

Fit time: 0.081s
Inertia: 69394
Silhouette score: 0.075
Davies-Bouldin score: 1.553

Adjusted Rand Index: 0.04761764968680273
Normalized Mutual Information: 0.24419275739201599

Non concluante

Cluster 0

Baby Care	1
Beauty and Personal Care	1
Home Decor & Festive Needs	102

Cluster 1

Baby Care	148
Beauty and Personal Care	147
Computers	133
Home Decor & Festive Needs	48
Home Furnishing	150
Kitchen & Dining	117
Watches	150

Cluster 2

Computers	7

Cluster 3

Kitchen & Dining	11

Cluster 4

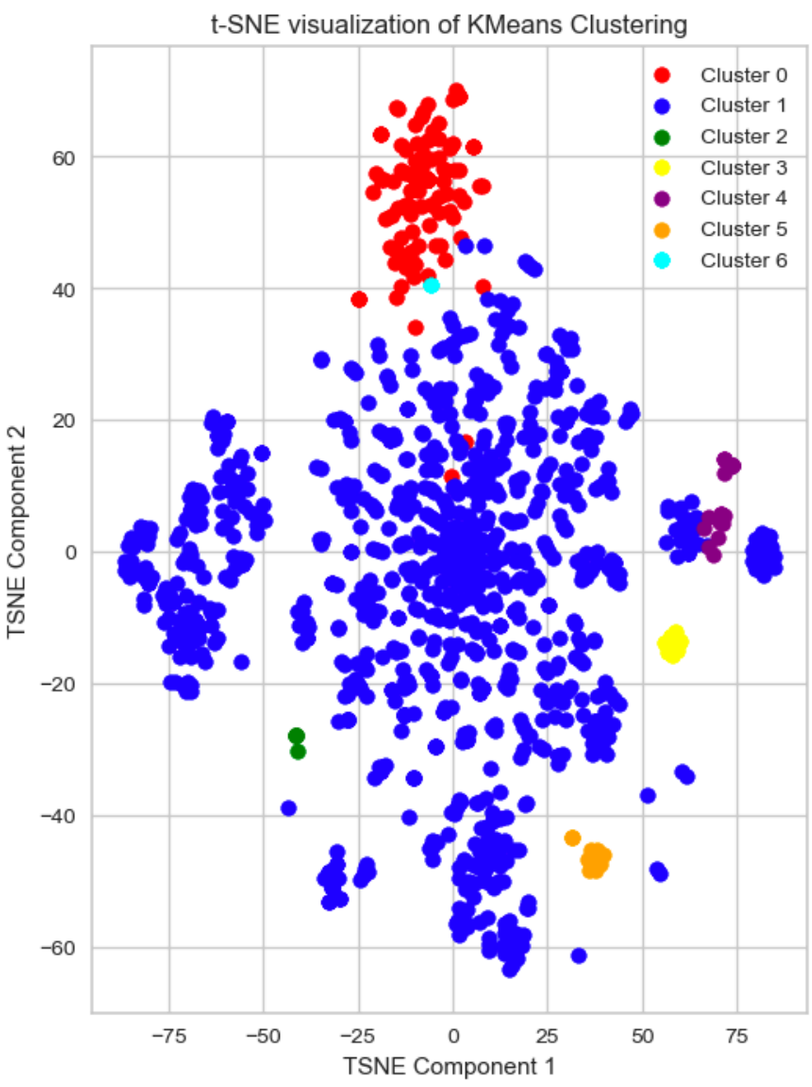
Bay Care	1
Kitchen & Dining	21

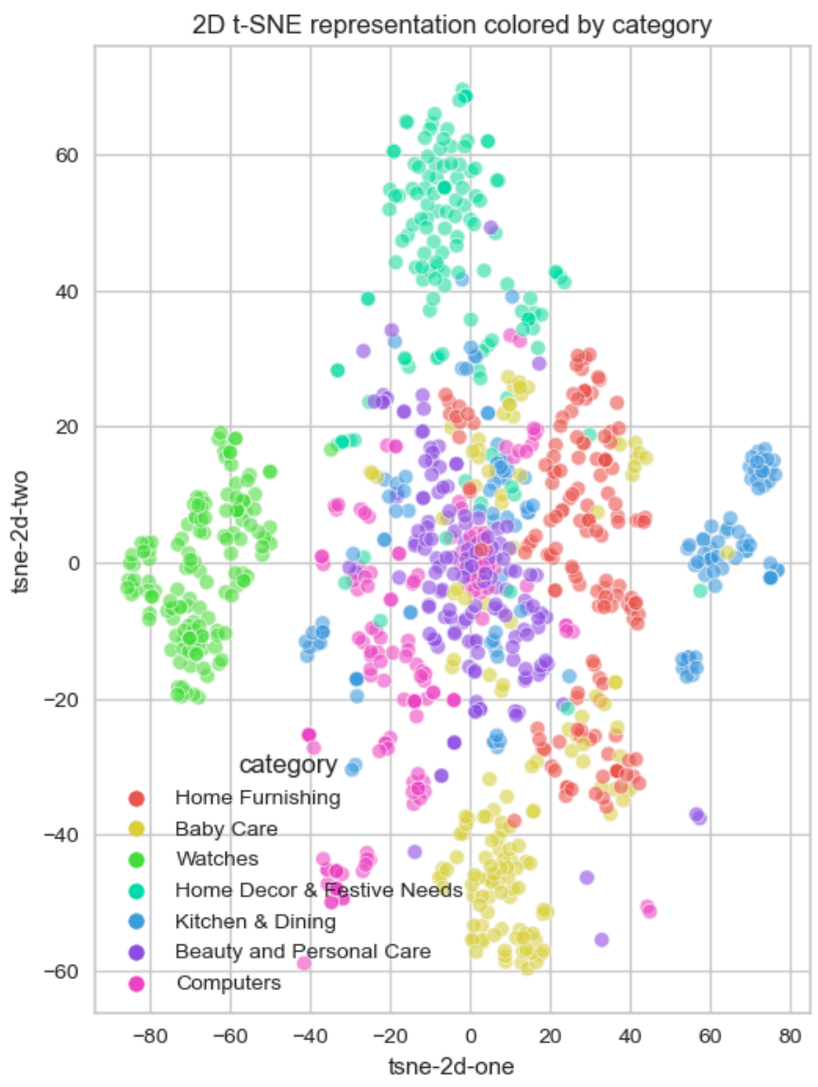
Cluster 5

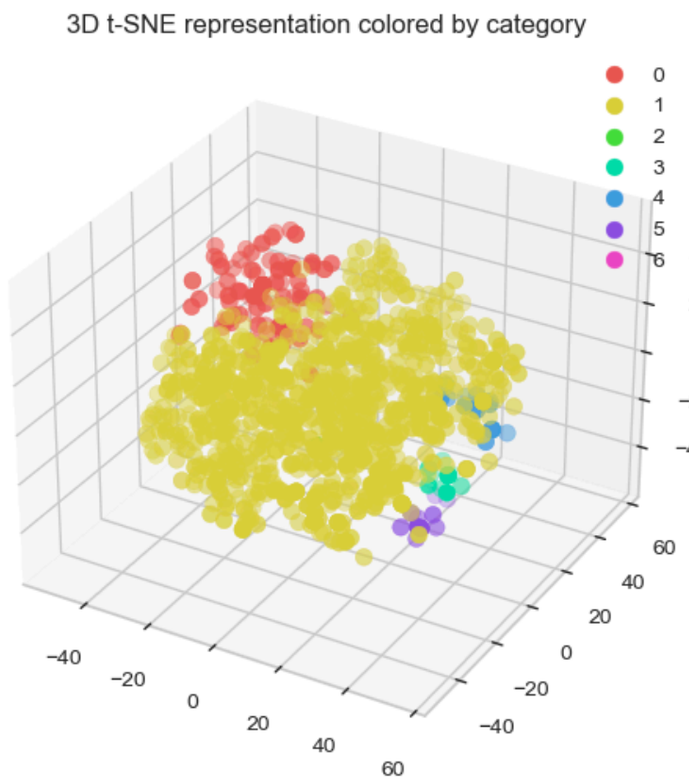
Beauty and Personal Care	2
Computers	10

Cluster 6

Kitchen & Dining	1







TF-IDF

On se retrouve avec une matrice tfidf_df de 1050 lignes (produits) x 2791 colonnes (mots).

TF-IDF Representation:

	ability	able	abode	abroad	absolute	absorbency	absorbent	absorber
0	0.0	0.0	0.0	0.0	0.0	0.000000	0.0	0.0
1	0.0	0.0	0.0	0.0	0.0	0.000000	0.0	0.0
2	0.0	0.0	0.0	0.0	0.0	0.076492	0.0	0.0
3	0.0	0.0	0.0	0.0	0.0	0.000000	0.0	0.0
4	0.0	0.0	0.0	0.0	0.0	0.000000	0.0	0.0

	absorbing	abstract	accent	access	accessory	accident	accidental
0	0.0	0.222428	0.0	0.0	0.0	0.0	0.0
1	0.0	0.000000	0.0	0.0	0.0	0.0	0.0
2	0.0	0.000000	0.0	0.0	0.0	0.0	0.0
3	0.0	0.000000	0.0	0.0	0.0	0.0	0.0
4	0.0	0.000000	0.0	0.0	0.0	0.0	0.0

	accommodate	according	aching	acid	acne	across	acrylic	act	active
0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
2	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
3	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
4	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0

	actu	actual	adapter	adaptor	added	addiction	addition	additional
0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
2	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
3	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
4	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0

	additionally	adhesive	adjust	adjustable	admiration	admired	adorable
0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
1	0.0	0.0	0.0	0.0	0.0	0.0	0.0
2	0.0	0.0	0.0	0.0	0.0	0.0	0.0
3	0.0	0.0	0.0	0.0	0.0	0.0	0.0
4	0.0	0.0	0.0	0.0	0.0	0.0	0.0

	adorn	advance	advice	advisable	aero	affect	affordable	afternoon
0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
2	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
3	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
4	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0

```
PCA Explained Variance (first 600 components): 98.53%
```

Kmeans (après PCA)

```
Fit time: 0.124s
Inertia: 854
Silhouette score: 0.064
Davies-Bouldin score: 3.420


Adjusted Rand Index: 0.1934770911423072
Normalized Mutual Information: 0.5068622066365257
```

Dans l'ensemble, ces indicateurs indiquent que notre modèle KMeans n'est peut-être pas le meilleur pour cet ensemble de données particulier. On peut essayer d'autres méthodes de clustering, modifier les méthodes de réduction de dimensionnalité ou ajuster les hyperparamètres de votre modèle actuel.

Cluster 0

Baby Care	1
Kitchen & Dining	73

Cluster 2

Baby Care	83
Home Furnishing	1

Cluster 4

Watches	106

Cluster 6

Beauty and Personal Care
Home Decor & Festive Needs
Kitchen & Dining

Cluster 1

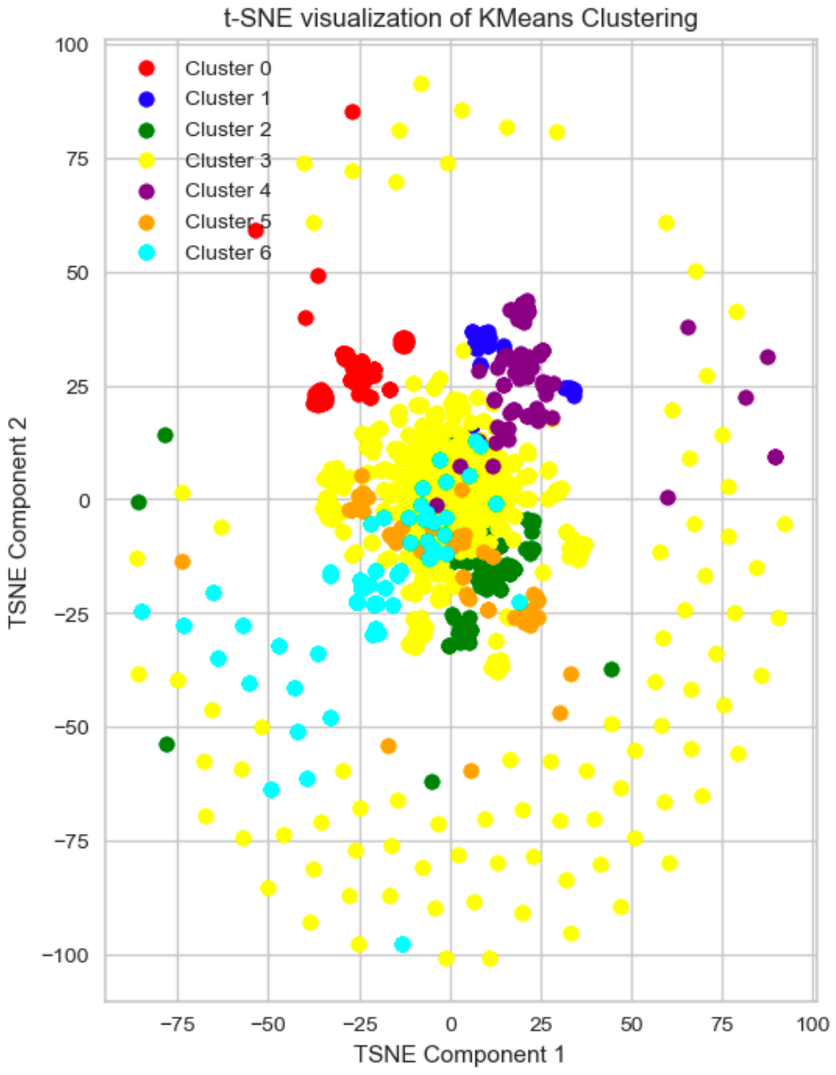
Watches	34

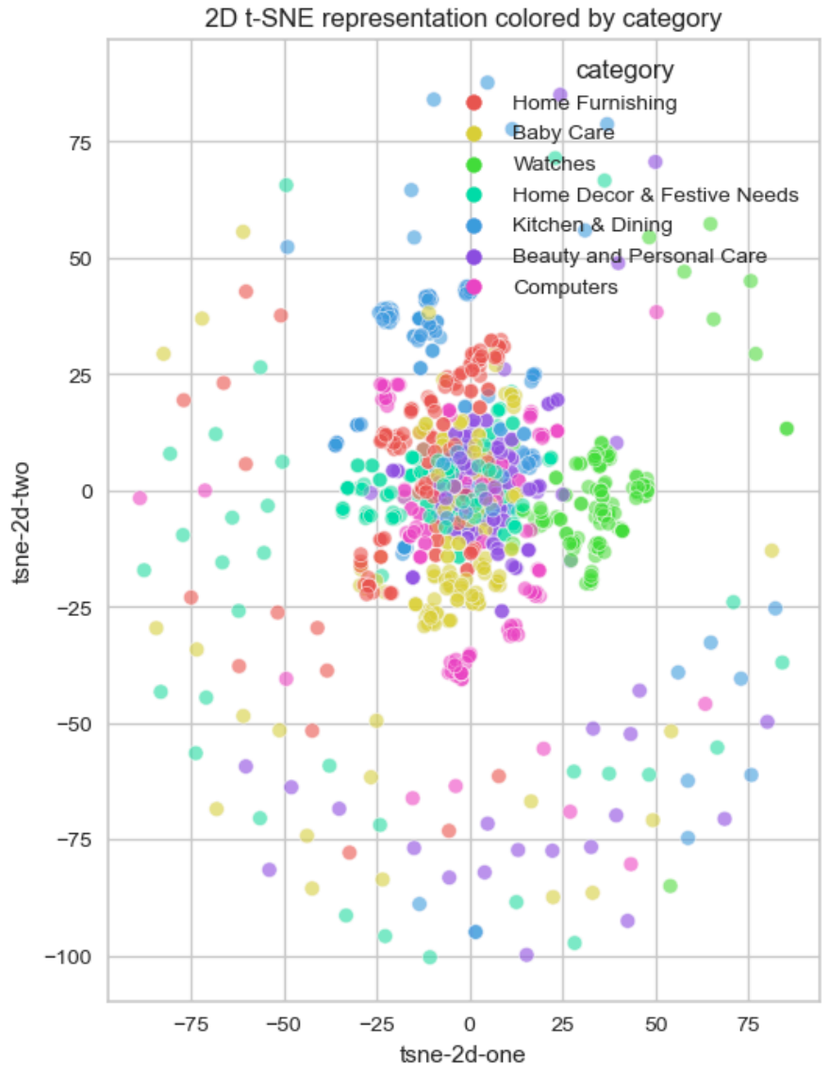
Cluster 3

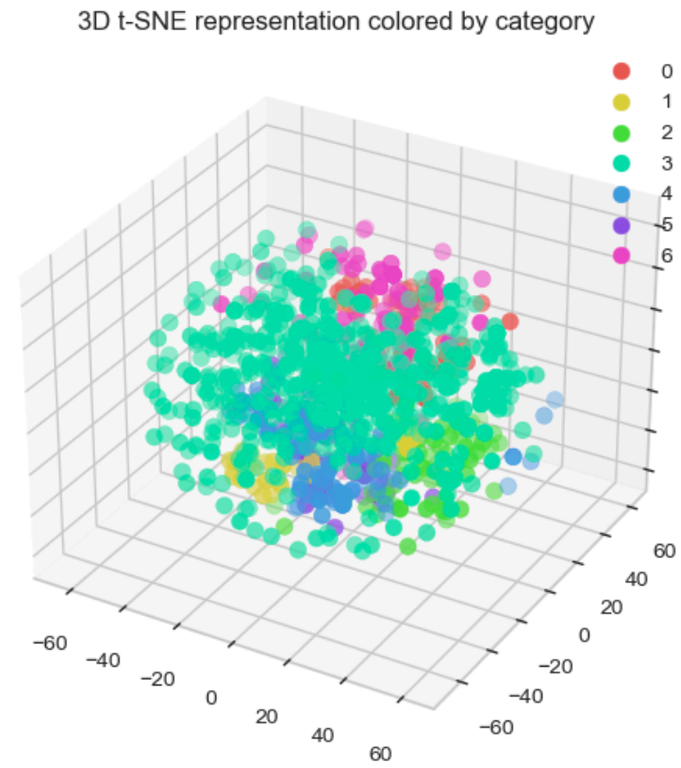
Baby Care	48
Beauty and Personal Care	149
Computers	150
Home Decor & Festive Needs	76
Home Furnishing	101
Kitchen & Dining	71
Watches	10

Cluster 5

Baby Care	18
Home Furnishing	48







Doc2Vec

On obtient la représentation vectorielle pour chaque document

Doc2Vec Representation:

	D2V_0	D2V_1	D2V_2	D2V_3	D2V_4	D2V_5	D2V_6
0	0.249412	0.021126	-1.034192	0.272835	0.425736	1.288181	-0.011335
1	-0.248269	-0.439130	-1.144321	-1.138456	0.913568	0.257461	-0.436136
2	-0.185453	0.602077	1.117174	-0.636189	0.417453	1.082409	0.270928
3	-0.903399	-0.201273	-0.970941	0.159663	-0.777194	-1.266511	0.872513
4	-0.374385	0.124883	-0.339655	0.182571	-0.866697	-0.682106	0.735051

	D2V_7	D2V_8	D2V_9	D2V_10	D2V_11	D2V_12	D2V_13
0	0.379542	-0.151936	0.487163	0.864310	0.451261	-0.251389	-0.342030
1	0.601524	-0.285708	0.978074	0.526136	0.475692	-0.145069	-0.607013
2	-0.240034	-1.293422	0.155666	-0.487968	0.867903	0.039795	0.287300
3	-0.837307	-0.681290	-0.541291	-0.914236	0.499954	-1.251088	0.112999
4	0.109006	-0.328889	-0.346200	-0.819216	-0.756549	-0.589941	0.367478

	D2V_14	D2V_15	D2V_16	D2V_17	D2V_18	D2V_19
0	0.064112	-0.918481	-1.003679	-0.566968	-1.045051	-0.778152
1	-0.129646	-0.979626	1.053031	3.407783	0.241224	-2.012348
2	-0.168066	0.034767	0.466797	1.123726	0.705669	-1.313975
3	0.735340	-0.209953	0.813313	1.140287	1.108747	-1.779100
4	0.090930	0.094024	1.642792	1.834044	1.128798	-1.070935

PCA Explained Variance (first 18 components): 99.23%

Kmeans (après PCA)

Fit time: 0.088s
Inertia: 7360
Silhouette score: 0.092
Davies-Bouldin score: 2.341

Adjusted Rand Index: 0.10450703389318301
Normalized Mutual Information: 0.15546051055316618

Non concluante

Cluster 0

Baby Care	23
Beauty and Personal Care	61
Computers	41
Home Decor & Festive Needs	29
Home Furnishing	41
Kitchen & Dining	65
Watches	5

Cluster 1

Baby Care	45
Beauty and Personal Care	16
Computers	32
Home Decor & Festive Needs	11
Home Furnishing	33
Kitchen & Dining	17
Watches	1

Cluster 2

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

Cluster 3

Baby Care	46
Beauty and Personal Care	19
Computers	8
Home Decor & Festive Needs	11
Home Furnishing	38
Kitchen & Dining	23
Watches	9

Cluster 4

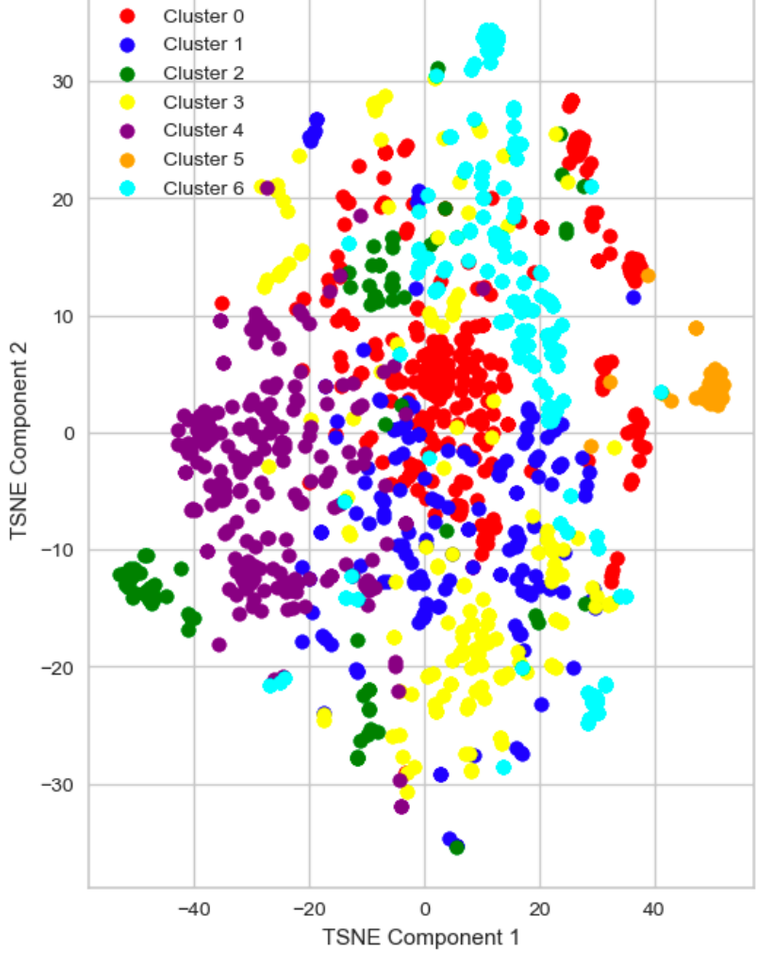
Baby Care	10
Beauty and Personal Care	6
Computers	9
Home Decor & Festive Needs	64
Home Furnishing	18
Kitchen & Dining	11
Watches	107

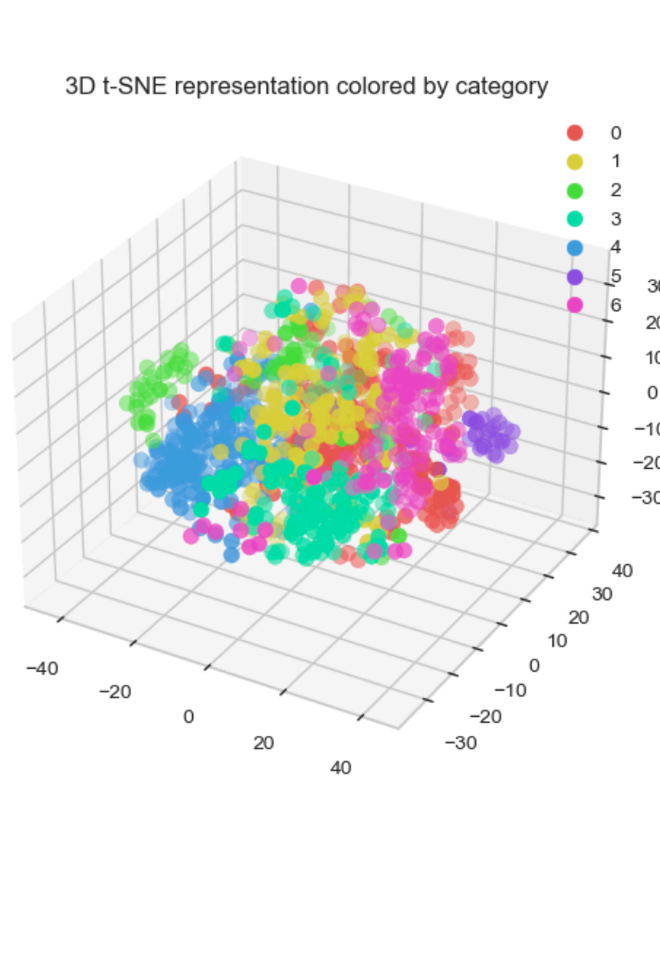
Cluster 5

Baby Care	
Beauty and Personal Care	
Home Furnishing	
Kitchen & Dining	

Cluster 6

Baby Care	20
Beauty and Personal Care	30
Computers	41
Home Decor & Festive Needs	26
Home Furnishing	12
Kitchen & Dining	9
Watches	4





BERT

BERT Representation

	0	1	2	3	4	5	6
0	-0.100548	-0.058811	0.488872	0.015085	0.159717	0.143835	0.031892
1	0.111775	-0.093875	0.260308	0.266598	-0.027499	0.046607	0.058420
2	0.031208	-0.056335	0.234802	0.122480	0.074681	0.188588	-0.033550
3	0.088796	-0.103713	0.462404	0.075606	-0.003312	0.019137	0.044282
4	0.144286	-0.164465	0.576499	0.128164	0.067908	-0.062060	-0.070824

	7	8	9	10	11	12	13
0	-0.079651	-0.135779	-0.257277	0.018432	0.018066	0.072655	0.173041
1	-0.302133	-0.175565	-0.208237	0.138224	0.002969	-0.009781	0.408124
2	-0.099784	-0.151226	-0.244095	-0.050439	-0.039499	0.184147	0.210672
3	-0.110452	-0.130985	0.021645	-0.100494	-0.010305	0.038457	0.171441
4	0.002517	-0.030763	-0.181761	-0.046646	-0.103654	0.052404	0.257768

	14	15	16	17	18	19	20
0	-0.191024	0.167529	-0.154038	0.165250	0.038152	0.319895	0.124213
1	-0.089733	0.061185	-0.280488	0.211016	-0.049639	0.388542	-0.036240
2	-0.201612	0.008652	-0.199808	0.145450	-0.041116	0.373776	0.054285
3	-0.022912	0.224996	-0.142579	-0.016853	-0.036720	0.158054	0.216320

PCA Explained Variance (first 300 components): 98.73%

Kmeans (après PCA)

Fit time: 0.122s
Inertia: 19854
Silhouette score: 0.092
Davies-Bouldin score: 2.483

Adjusted Rand Index: 0.17055205913492422
Normalized Mutual Information: 0.29258515622239906

Non concluante

Cluster 0

_____

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

Cluster 2

_____

Baby Care	3
Beauty and Personal Care	7
Computers	60
Home Decor & Festive Needs	5
Home Furnishing	4
Kitchen & Dining	8
Watches	69

Cluster 3

_____

Beauty and Personal Care	13
Computers	23
Home Furnishing	1

Cluster 5

_____

Beauty and Personal Care	2
Computers	1
Watches	69

Cluster 1

_____

Baby Care	105
Beauty and Personal Care	48
Computers	31
Home Decor & Festive Needs	30
Home Furnishing	51
Kitchen & Dining	32
Watches	2

Cluster 4

_____

Baby Care	24
Beauty and Personal Care	58
Computers	19
Home Decor & Festive Needs	79
Home Furnishing	7
Kitchen & Dining	50
Watches	5

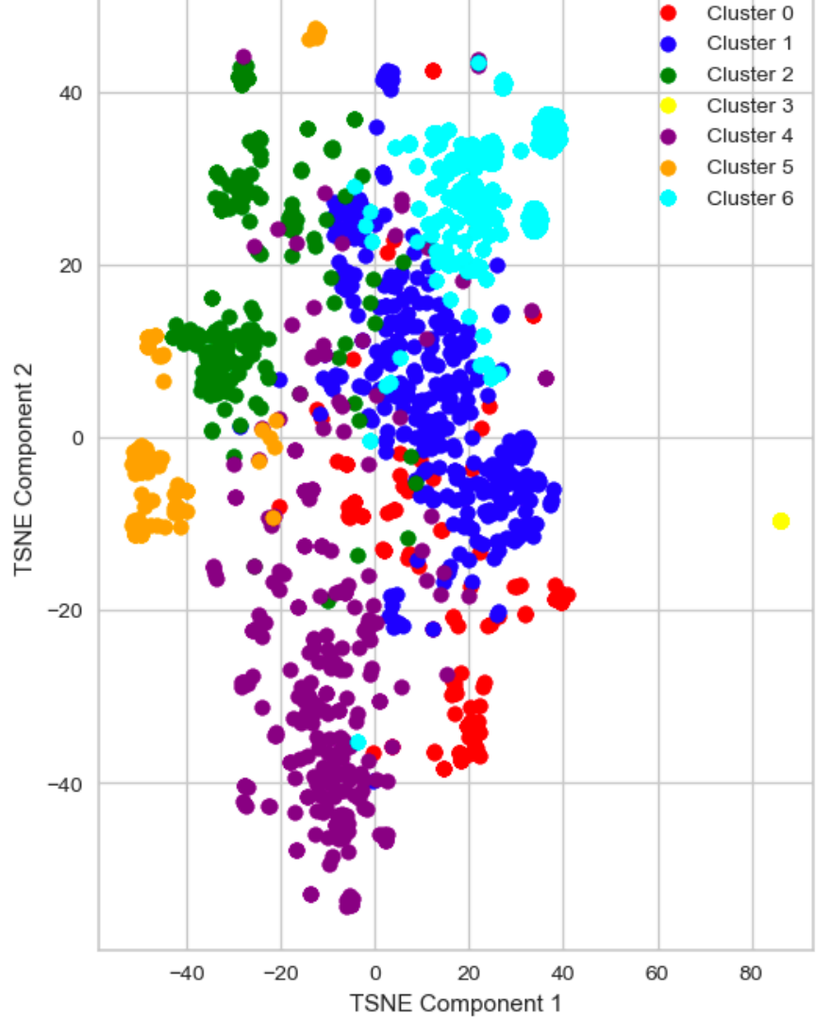
Cluster 6

_____

Baby Care	7
Beauty and Personal Care	9
Computers	16
Home Decor & Festive Needs	35
Home Furnishing	14
Kitchen & Dining	58
Watches	5

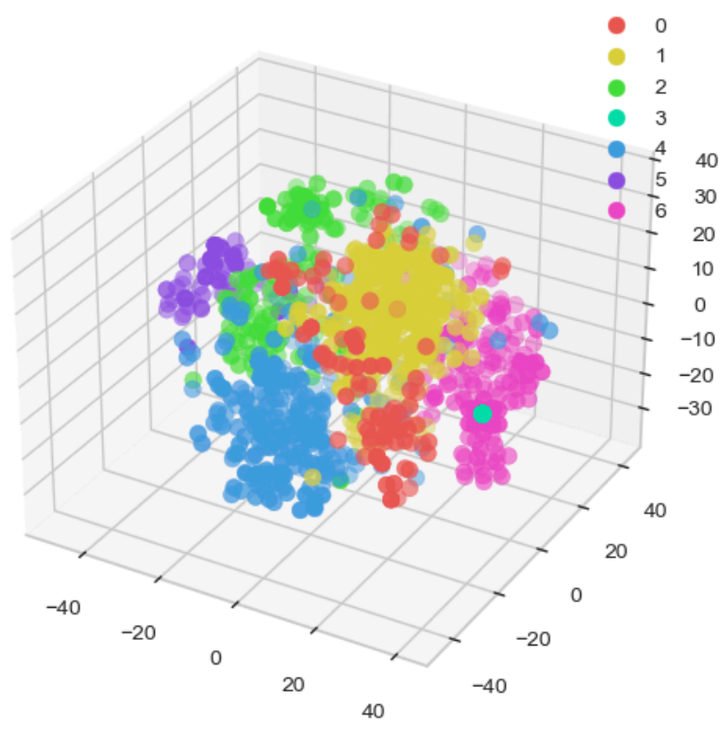
---

t-SNE visualization of KMeans Clustering



2D t-SNE representation colored by category

3D t-SNE representation colored by category



USE (Universal Sentence Encoder)

Universal Sentence Encoder Representation

```
                                                  USE_0
0  tf.Tensor(-0.054714132, shape=(), dtype=float32)  \
1  tf.Tensor(-0.014209846, shape=(), dtype=float32)
2   tf.Tensor(-0.05418499, shape=(), dtype=float32)
3  tf.Tensor(-0.055911593, shape=(), dtype=float32)
4  tf.Tensor(-0.054030288, shape=(), dtype=float32)

                                                  USE_1
0  tf.Tensor(-0.049381454, shape=(), dtype=float32)  \
1   tf.Tensor(0.023907378, shape=(), dtype=float32)
2   tf.Tensor(0.024871513, shape=(), dtype=float32)
3  tf.Tensor(-0.056374468, shape=(), dtype=float32)
4   tf.Tensor(-0.05410431, shape=(), dtype=float32)

                                                  USE_2
0   tf.Tensor(0.008858256, shape=(), dtype=float32)  \
1  tf.Tensor(-0.026434414, shape=(), dtype=float32)
2  tf.Tensor(-0.039922446, shape=(), dtype=float32)
3    tf.Tensor(0.04707284, shape=(), dtype=float32)
4   tf.Tensor(0.032971125, shape=(), dtype=float32)
```

PCA Explained Variance (first 300 components): 98.28%

Kmeans (après PCA)

Fit time: 0.092s
Inertia: 738
Silhouette score: 0.108
Davies-Bouldin score: 3.291

Adjusted Rand Index: 0.3129676163725078
Normalized Mutual Information: 0.4738498279728325

Cluster 0

Watches	147

Cluster 1

Baby Care	17
Beauty and Personal Care	53
Computers	64
Home Decor & Festive Needs	83
Home Furnishing	8
Kitchen & Dining	38
Watches	3

Cluster 2

Beauty and Personal Care	13
Computers	23
Home Furnishing	1
Kitchen & Dining	6

Cluster 3

Baby Care	1
Kitchen & Dining	74

Cluster 4

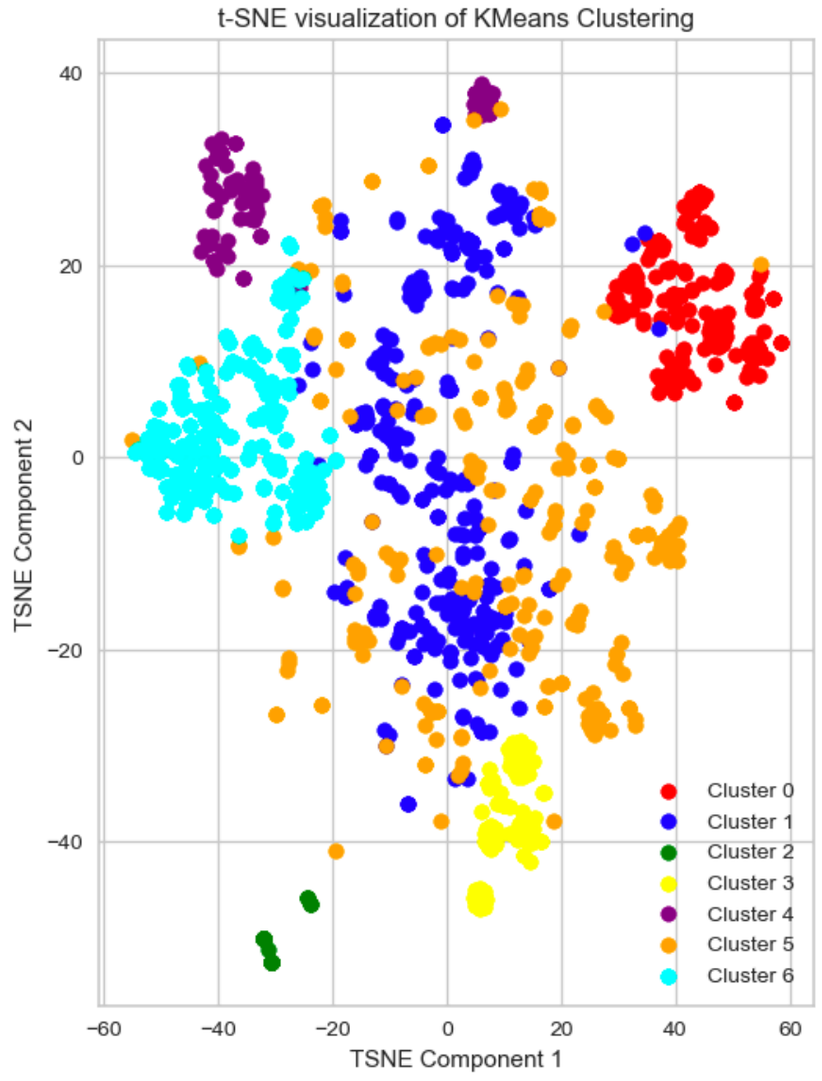
Baby Care	5
Computers	16
Home Furnishing	51

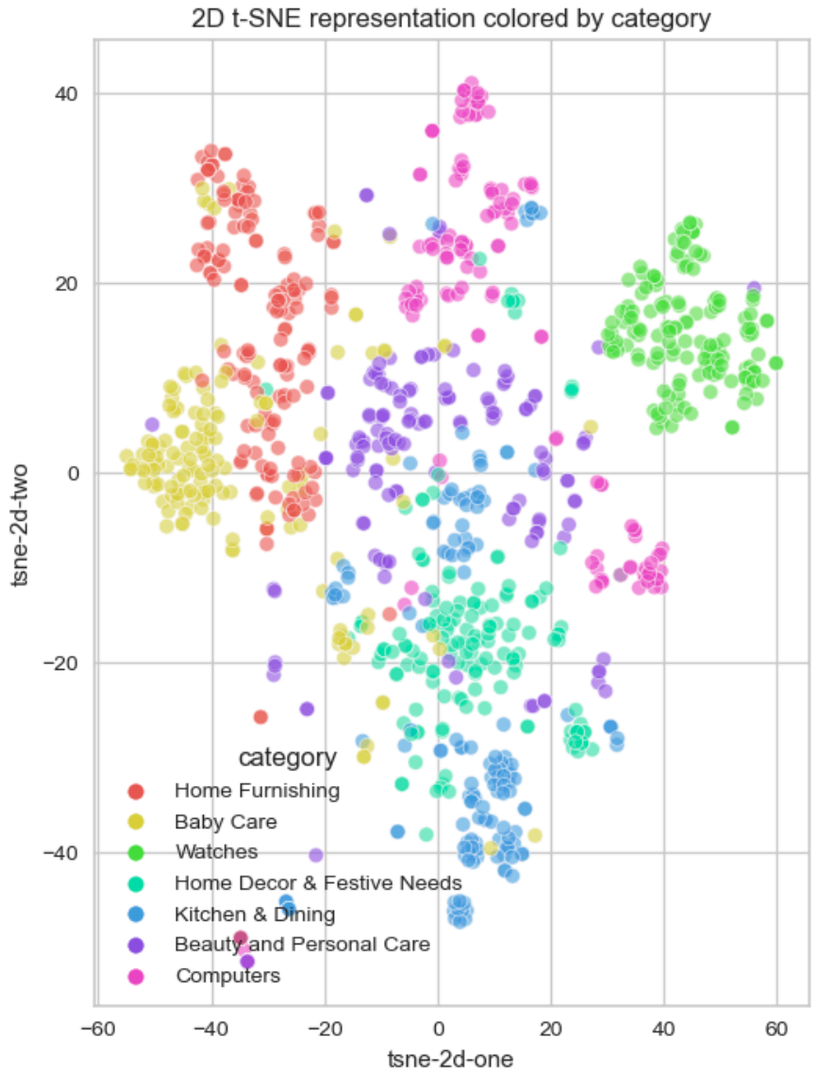
Cluster 5

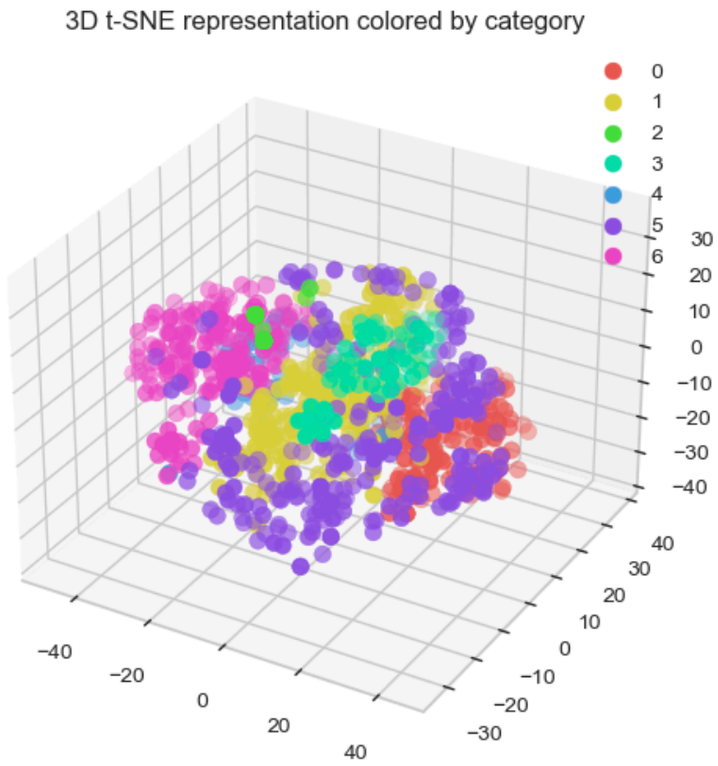
Baby Care	20
Beauty and Personal Care	80
Computers	47
Home Decor & Festive Needs	66
Home Furnishing	15
Kitchen & Dining	32

Cluster 6

Baby Care	107
Beauty and Personal Care	4
Home Decor & Festive Needs	1
Home Furnishing	75







Données visuelles : extraction de features

Pré-traitement

Amélioration du contraste

Image floue

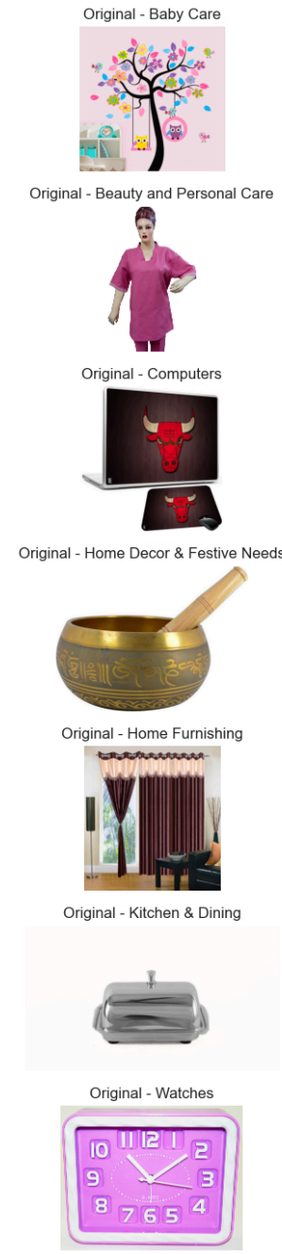
Réglage de la luminosité

Amélioration du contraste

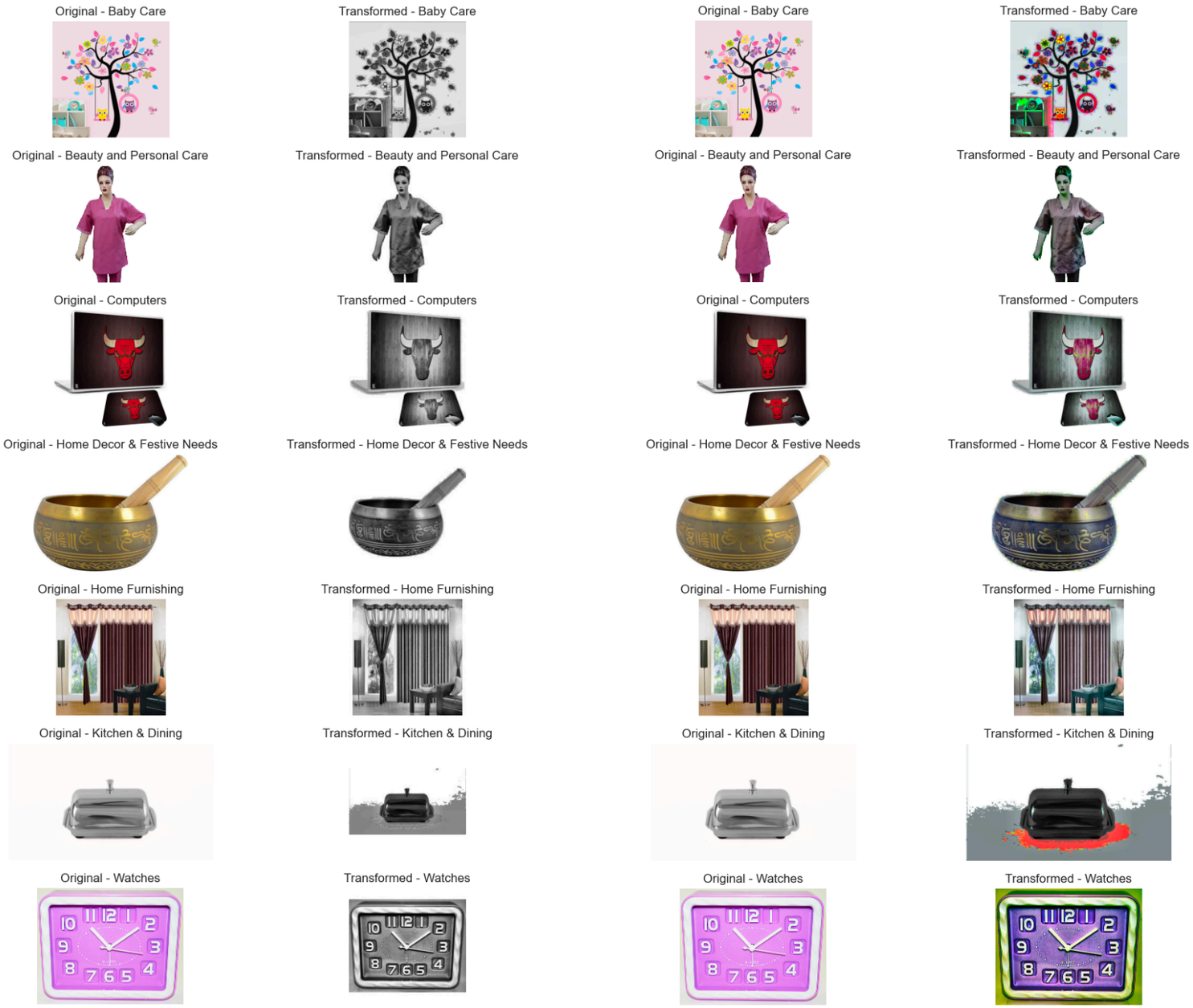
Augmentation de la saturation des couleurs

Redimensionner l'image

PILLOW



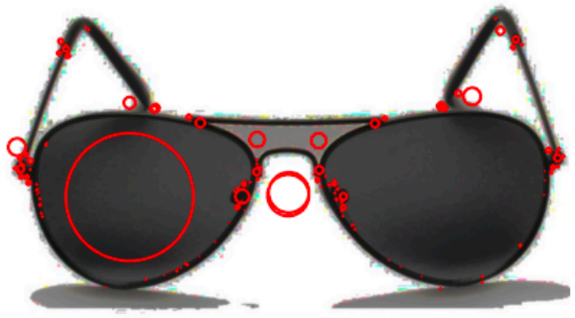
OpenCV (cv2)

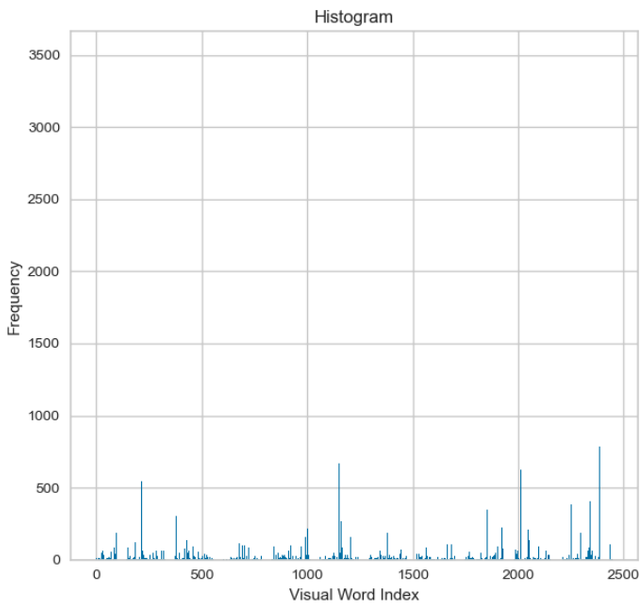


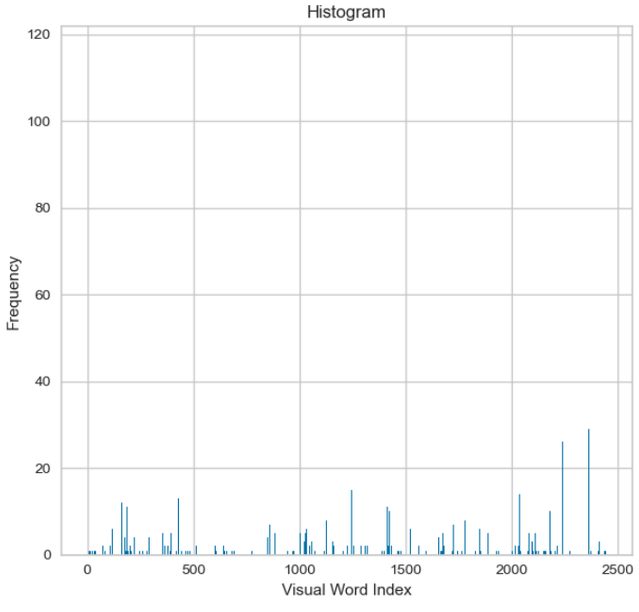
SIFT

Pour chaque image, applique l'algorithme SIFT pour détecter les descripteurs.

- Récupérer les descripteurs de chaque image par un algorithme de type SIFT;
- Clusteriser l'ensemble de tous les descripteurs;
- Associer les descripteurs de chaque image aux centres obtenus par clustering;
- Construction des histogrammes.







Données visuelles : catégorisation

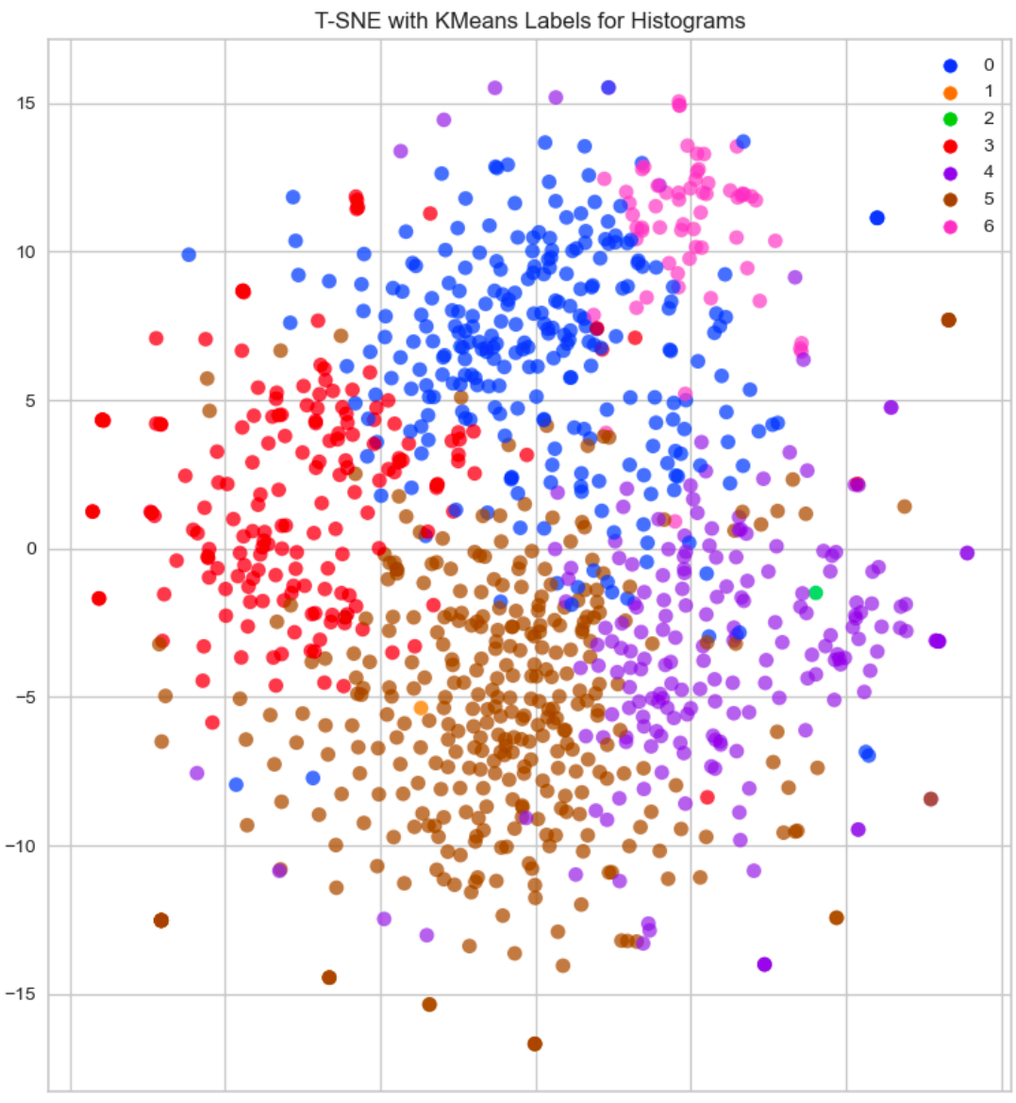
Normaliser les histogrammes

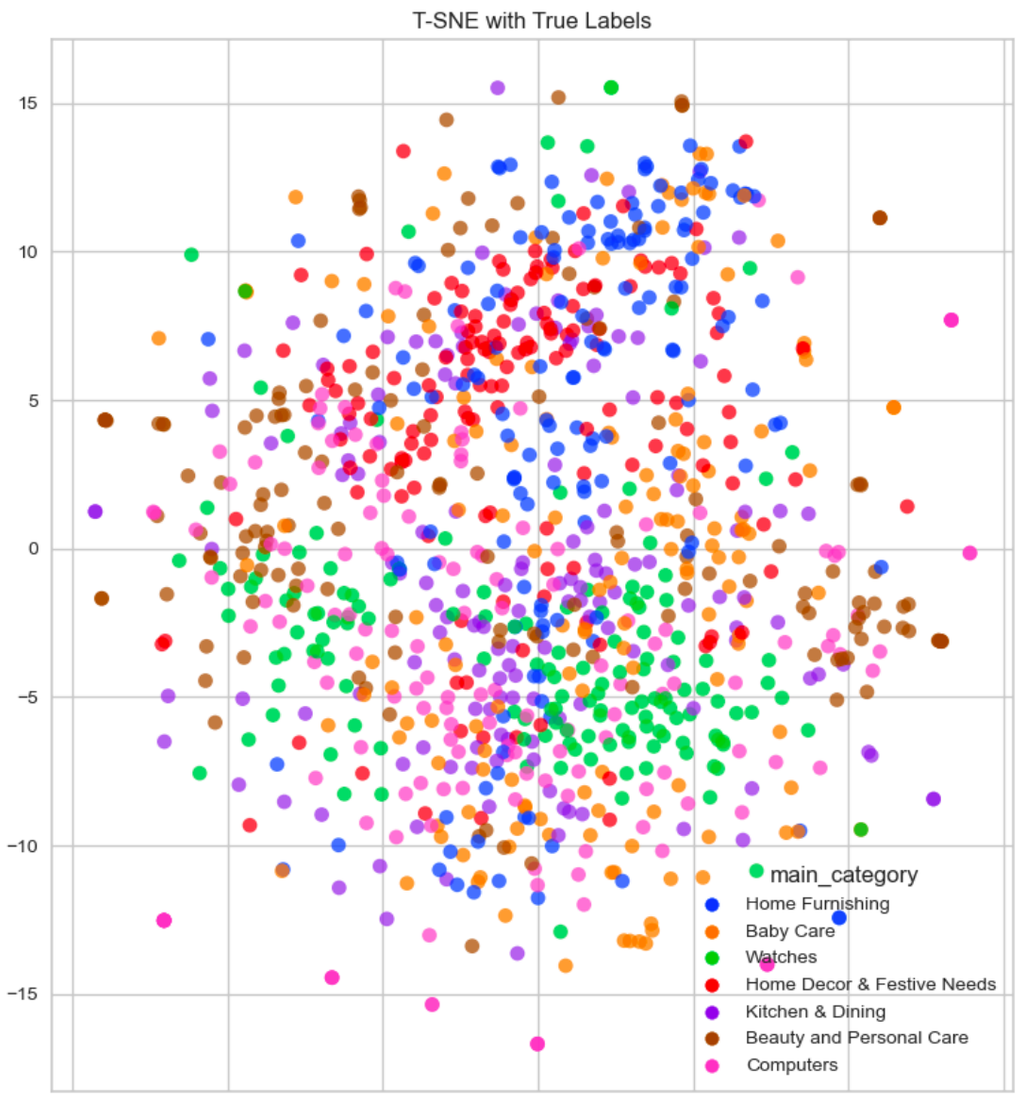
Standardiser les données scaled_histograms: (1050, 2446)

PCA **(n_components = 611)** reduced_histograms:(1050, 611)

```
PCA Explained Variance  (first 611 components): 95%
```

KMeans (Obtention d'étiquettes de cluster pour les histogrammes)



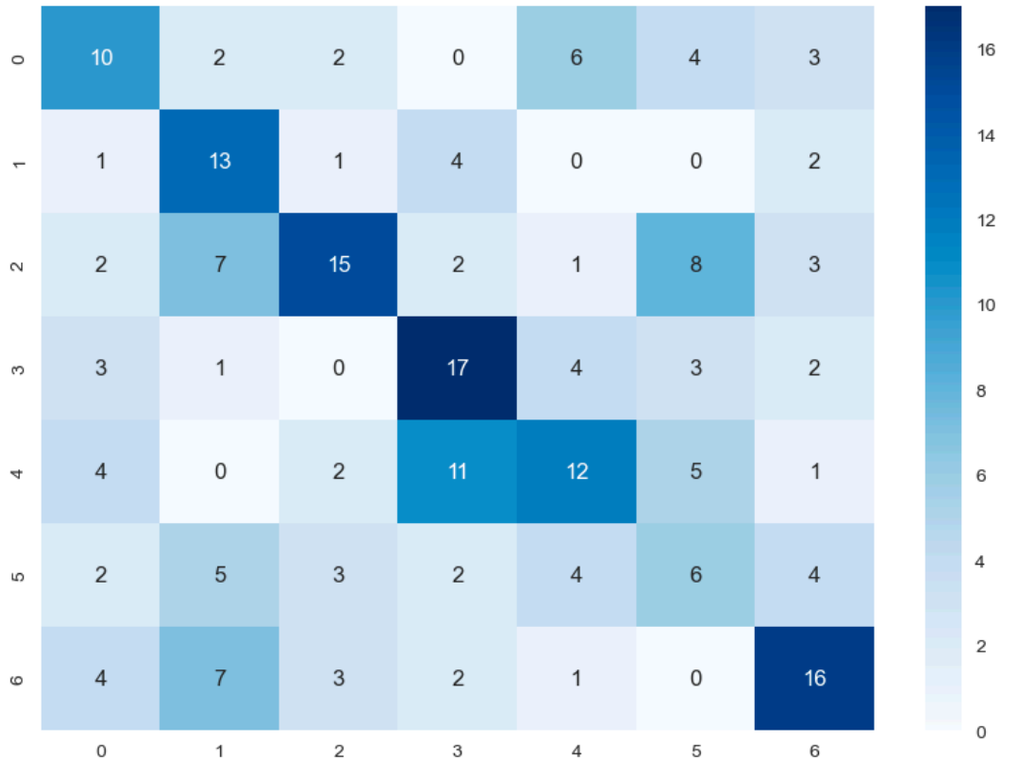


Données visuelles : classification

RandomForestClassifier

Accuracy: 0.4238095238095238

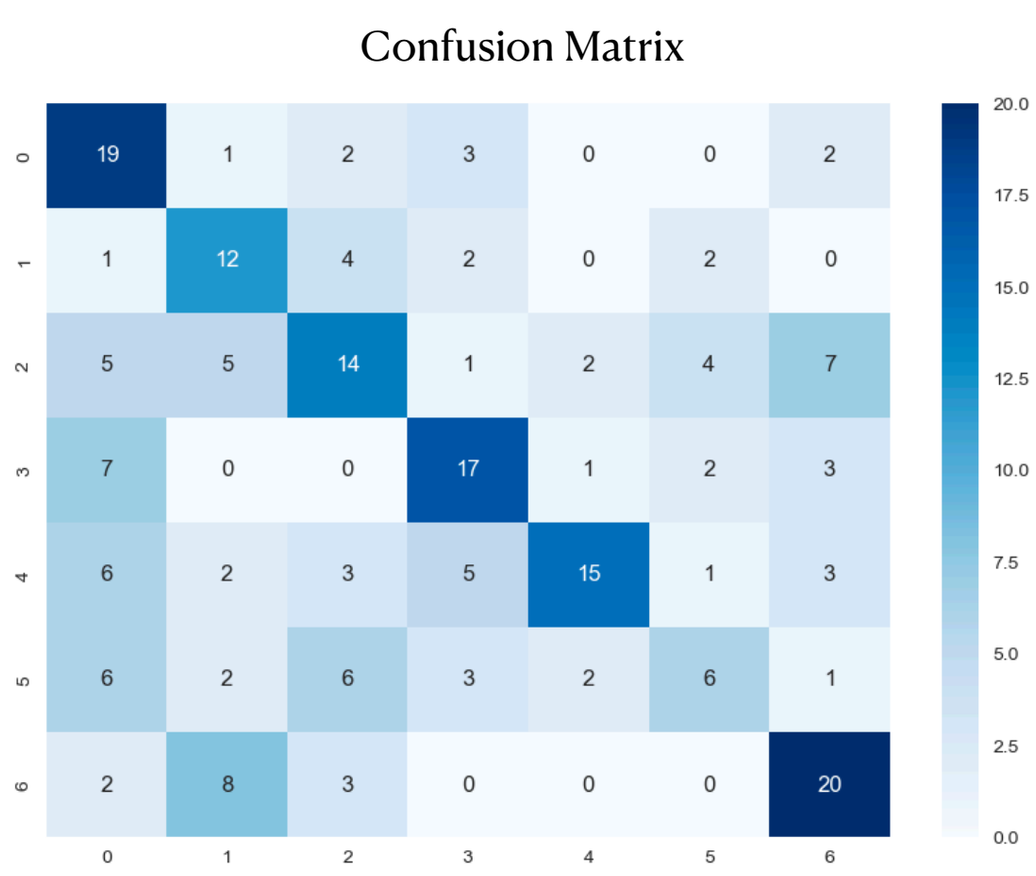
	precision	recall	f1-score	support
0	0.38	0.37	0.38	27
1	0.37	0.62	0.46	21
2	0.58	0.39	0.47	38
3	0.45	0.57	0.50	30
4	0.43	0.34	0.38	35
5	0.23	0.23	0.23	26
6	0.52	0.48	0.50	33
accuracy			0.42	210
macro avg	0.42	0.43	0.42	210
weighted avg	0.44	0.42	0.42	210



SVM

Accuracy: 0.49047619047619045

	precision	recall	f1-score	support
0	0.41	0.70	0.52	27
1	0.40	0.57	0.47	21
2	0.44	0.37	0.40	38
3	0.55	0.57	0.56	30
4	0.75	0.43	0.55	35
5	0.40	0.23	0.29	26
6	0.56	0.61	0.58	33
accuracy			0.49	210
macro avg	0.50	0.50	0.48	210
weighted avg	0.51	0.49	0.48	210



Transfert Learning

Convolutional Neural Networks, CNN (VGG16)

- Chargement du modèle VGG16 pré-entraîné sans couches supérieures:

  base_model = VGG16(weights='imagenet', include_top=False)

- Extraction de fonctionnalités pour notre ensemble d'images par une fonction **extract_features_from_image(img_path)** pour extraire des caractéristiques d'une image en utilisant VGG16

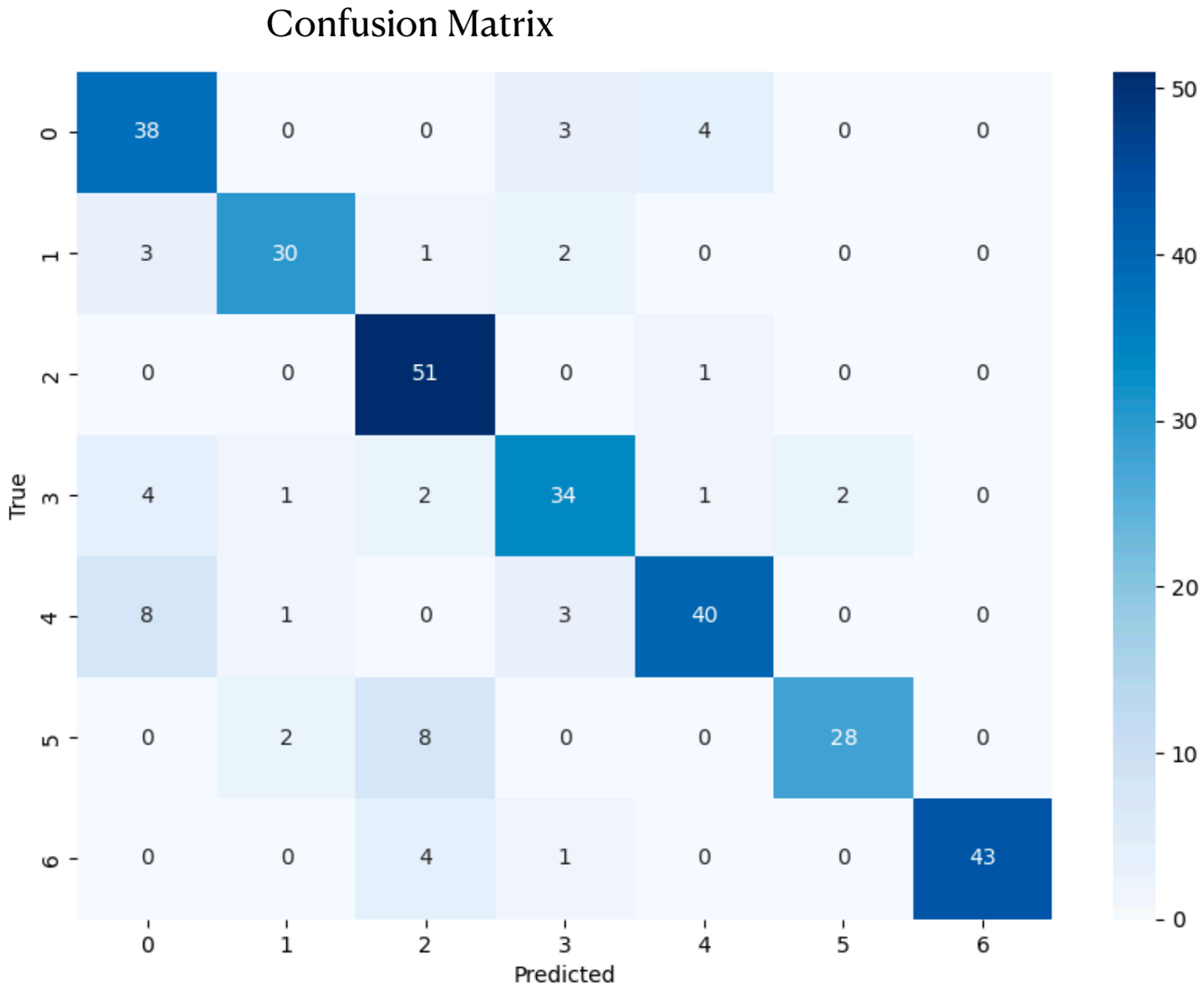
- La classification d'images à l'aide de SVM

```
labels = data['main_category'].tolist()
```

```
%%time
X_train_cnn, X_test_cnn, y_train_cnn, y_test_cnn = train_test_split(features_list, labels, test_size=0.3, random_st

svm_cnn = SVC(kernel='linear', C=1)

svm_cnn.fit(X_train_cnn, y_train_cnn)

y_pred_cnn = svm_cnn.predict(X_test_cnn)

accuracy_cnn = accuracy_score(y_test_cnn, y_pred_cnn)
print("Accuracy CVM:", accuracy_cnn)
```

```
Accuracy CVM: 0.8380952380952381
CPU times: user 54.8 s, sys: 515 ms, total: 55.3 s
Wall time: 9.49 s
```

Le modèle classe correctement environ 83,8 % des échantillons dans l'ensemble de test. Le temps réel (le temps de l'horloge murale) qui s'est écoulé lors de l'exécution de la cellule. Cela fait 9.21 secondes. On peut conclure que l'utilisation des fonctionnalités obtenues à partir de VGG16 pour la classification d'images à l'aide de SVM a donné une précision assez élevée sur l'ensemble de test



	precision	recall	f1-score	support
Baby Care	0.72	0.84	0.78	45
Beauty and Personal Care	0.88	0.83	0.86	36
Computers	0.77	0.98	0.86	52
Home Decor & Festive Needs	0.79	0.77	0.78	44
Home Furnishing	0.87	0.77	0.82	52
Kitchen & Dining	0.93	0.74	0.82	38
Watches	1.00	0.90	0.95	48
accuracy			0.84	315
macro avg	0.85	0.83	0.84	315
weighted avg	0.85	0.84	0.84	315

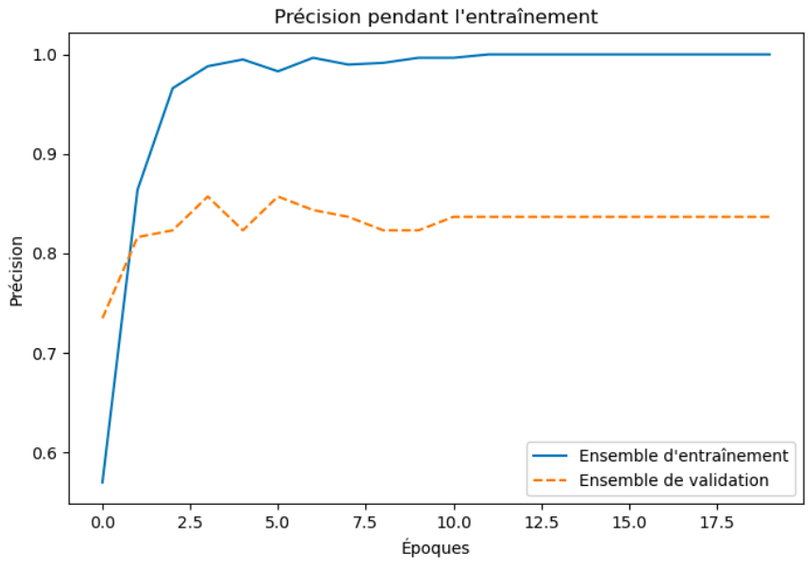
Deep Neural Network
(Fully Connected Neural Network / Dense Neural Network)

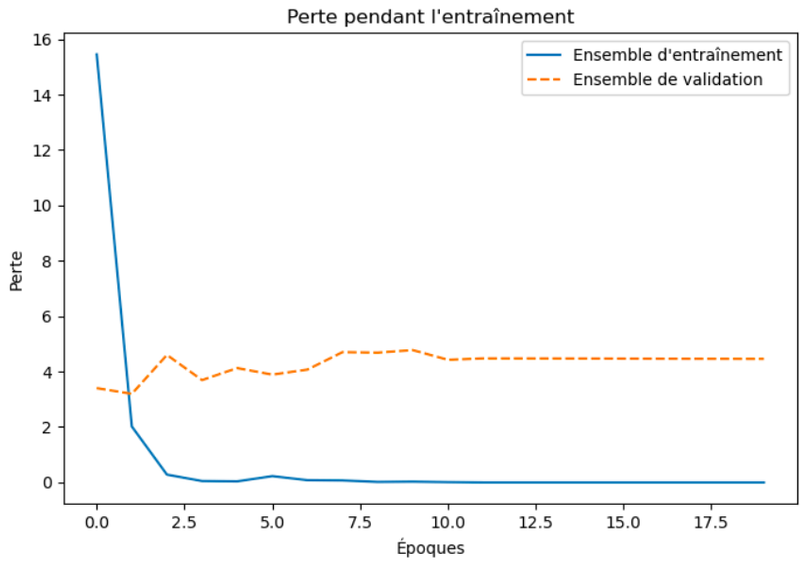
```
# Définir la couche d'entrée
input_layer = Input(shape=(X_array.shape[1],))

# Couches supplémentaires
x = Dense(416, activation='relu')(input_layer)
x = Dense(96, activation='relu')(x)
output_layer = Dense(len(np.unique(y_encoded)), activation='softmax')(x)

# Créer un modèle
model = Model(inputs=input_layer, outputs=output_layer)

# Compilation du modèle
model.compile(optimizer='adam', loss='categorical_crossentropy', metrics=['accuracy'])
```

```
%%time
history_img = model.fit(X_train, y_train, epochs=20, batch_size=32, validation_split=0.2, verbose=1)
```

```
poch 7/20
9/19 [==============================] - 3s 135ms/step - loss: 0.0816 - accuracy: 0.9966 - val_loss: 4.0728 - val_acc
racy: 0.8435
poch 8/20
9/19 [==============================] - 3s 133ms/step - loss: 0.0723 - accuracy: 0.9898 - val_loss: 4.7032 - val_acc
racy: 0.8367
poch 9/20
9/19 [==============================] - 3s 134ms/step - loss: 0.0186 - accuracy: 0.9915 - val_loss: 4.6845 - val_acc
racy: 0.8231
poch 10/20
9/19 [==============================] - 3s 133ms/step - loss: 0.0280 - accuracy: 0.9966 - val_loss: 4.7771 - val_acc
racy: 0.8231
poch 11/20
9/19 [==============================] - 3s 135ms/step - loss: 0.0105 - accuracy: 0.9966 - val_loss: 4.4268 - val_acc
racy: 0.8367
poch 12/20
9/19 [==============================] - 3s 133ms/step - loss: 5.2363e-05 - accuracy: 1.0000 - val_loss: 4.4769 - val
_accuracy: 0.8367
poch 13/20
9/19 [==============================] - 3s 135ms/step - loss: 2.1634e-05 - accuracy: 1.0000 - val_loss: 4.4762 - val
_accuracy: 0.8367
poch 14/20
9/19 [==============================] - 3s 139ms/step - loss: 8.2227e-06 - accuracy: 1.0000 - val_loss: 4.4740 - val
_accuracy: 0.8367
poch 15/20
9/19 [==============================] - 3s 135ms/step - loss: 5.8711e-06 - accuracy: 1.0000 - val_loss: 4.4722 - val
_accuracy: 0.8367
poch 16/20
9/19 [==============================] - 3s 133ms/step - loss: 4.7361e-06 - accuracy: 1.0000 - val_loss: 4.4689 - val
_accuracy: 0.8367
poch 17/20
9/19 [==============================] - 3s 138ms/step - loss: 4.0575e-06 - accuracy: 1.0000 - val_loss: 4.4659 - val
_accuracy: 0.8367
poch 18/20
9/19 [==============================] - 3s 135ms/step - loss: 3.5973e-06 - accuracy: 1.0000 - val_loss: 4.4637 - val
_accuracy: 0.8367
poch 19/20
9/19 [==============================] - 3s 134ms/step - loss: 3.2026e-06 - accuracy: 1.0000 - val_loss: 4.4616 - val
_accuracy: 0.8367
poch 20/20
9/19 [==============================] - 3s 136ms/step - loss: 2.8641e-06 - accuracy: 1.0000 - val_loss: 4.4598 - val
_accuracy: 0.8367
PU times: user 4min 4s, sys: 1min 53s, total: 5min 57s
all time: 51.7 s
```

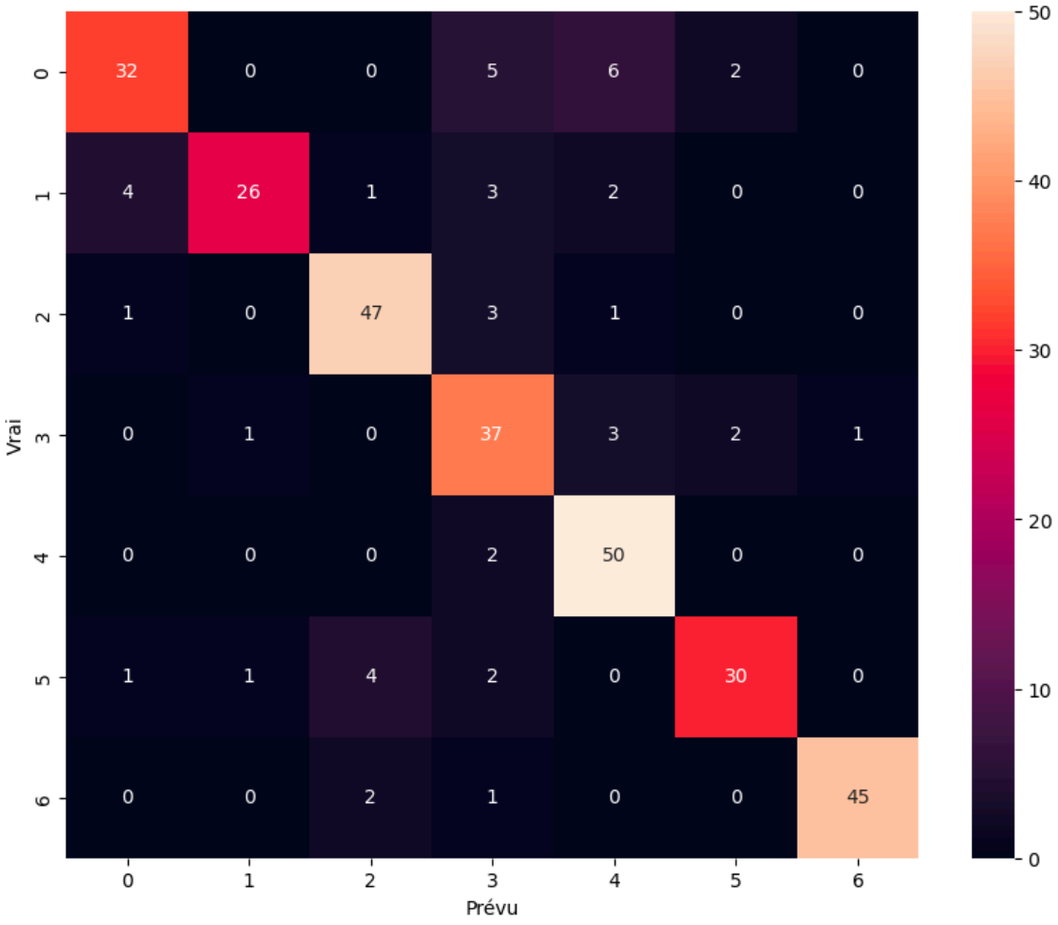
Visualisation des graphiques d'entraînement du modèle





	precision	recall	f1-score	support
0	0.84	0.71	0.77	45
1	0.93	0.72	0.81	36
2	0.87	0.90	0.89	52
3	0.70	0.84	0.76	44
4	0.81	0.96	0.88	52
5	0.88	0.79	0.83	38
6	0.98	0.94	0.96	48
accuracy			0.85	315
macro avg	0.86	0.84	0.84	315
weighted avg	0.86	0.85	0.85	315

Confusion Matrix



Choisissez des images aléatoires

```
Image: ./data/source/Flipkart/images/e4922f01eda047582cd72e9d1063ab7a.jpg
True Label: Home Decor & Festive Needs, Predicted Label: Home Decor & Festive Needs
---------
Image: ./data/source/Flipkart/images/07912328f580cf080d721e6466287896.jpg
True Label: Home Decor & Festive Needs, Predicted Label: Home Decor & Festive Needs
---------
Image: ./data/source/Flipkart/images/74e5a3f6edb34d7e593a0d1854b0b886.jpg
True Label: Kitchen & Dining, Predicted Label: Kitchen & Dining
---------
Image: ./data/source/Flipkart/images/a76bf8400b3dbcdbe5a5678f4a8ea0f6.jpg
True Label: Computers, Predicted Label: Computers
---------
Image: ./data/source/Flipkart/images/f309bdd259c5b46a560bc1620e641947.jpg
True Label: Computers, Predicted Label: Computers
---------
Image: ./data/source/Flipkart/images/caabe6014b914fe2874a9a8d7284f79b.jpg
True Label: Home Decor & Festive Needs, Predicted Label: Home Decor & Festive Needs
---------
Image: ./data/source/Flipkart/images/5fdb912462da9891e5b21c677ceb15e4.jpg
True Label: Home Decor & Festive Needs, Predicted Label: Home Decor & Festive Needs
---------
Image: ./data/source/Flipkart/images/3ccceaae844f34180708cb6cba3441bf.jpg
True Label: Computers, Predicted Label: Computers
---------
Image: ./data/source/Flipkart/images/13596c5cc53a74268613e5c0b7d46b60.jpg
True Label: Home Furnishing, Predicted Label: Home Furnishing
---------
Image: ./data/source/Flipkart/images/219b24362655097cb41bf06a0be8ee79.jpg
True Label: Kitchen & Dining, Predicted Label: Kitchen & Dining
---------
```

Le modèle n'a commis aucune erreur de prédiction sur un ensemble aléatoire d'images

Classification d'images par augmentation de données

```python
# Augmentation des données
datagen = ImageDataGenerator(
    rotation_range=40,
    width_shift_range=0.2,
    height_shift_range=0.2,
    shear_range=0.2,
    zoom_range=0.2,
    horizontal_flip=True,
    fill_mode='nearest'
)
```

*rotations aléatoires,
*décalages horizontaux ou verticaux,
*décalages aléatoires de la partie de l'image (en degrés),
*mise à l'échelle aléatoire,
*mode d'affichage aléatoire,
*remplissage des points en dehors de l'image d'entrée après rotation ou décalage

```
132/132 [==============================] - 551s 4s/step - loss: 1.3490 - accuracy: 0.4862 - val_loss: 2.5028 - val_ac
curacy: 0.1200
Epoch 3/10
132/132 [==============================] - 587s 4s/step - loss: 1.0157 - accuracy: 0.6055 - val_loss: 2.6887 - val_ac
curacy: 0.1400
Epoch 4/10
132/132 [==============================] - 572s 4s/step - loss: 0.7216 - accuracy: 0.7064 - val_loss: 3.3983 - val_ac
curacy: 0.1352
Epoch 5/10
132/132 [==============================] - 554s 4s/step - loss: 0.5582 - accuracy: 0.7738 - val_loss: 3.6368 - val_ac
curacy: 0.1457
Epoch 6/10
132/132 [==============================] - 575s 4s/step - loss: 0.4500 - accuracy: 0.8210 - val_loss: 3.8953 - val_ac
curacy: 0.1343
Epoch 7/10
132/132 [==============================] - 572s 4s/step - loss: 0.3677 - accuracy: 0.8550 - val_loss: 4.6056 - val_ac
curacy: 0.1600
Epoch 8/10
132/132 [==============================] - 567s 4s/step - loss: 0.3054 - accuracy: 0.8714 - val_loss: 5.1964 - val_ac
curacy: 0.1314
Epoch 9/10
132/132 [==============================] - 555s 4s/step - loss: 0.2738 - accuracy: 0.8919 - val_loss: 6.0394 - val_ac
curacy: 0.1248
Epoch 10/10
132/132 [==============================] - 557s 4s/step - loss: 0.2486 - accuracy: 0.8983 - val_loss: 6.2485 - val_ac
curacy: 0.1410
33/33 [==============================] - 107s 3s/step - loss: 6.2485 - accuracy: 0.1410
Loss: 6.248546123504639
Accuracy: 0.14095237851142883
33/33 [==============================] - 111s 3s/step
              precision    recall  f1-score   support

           0       0.16      0.08      0.11       210
           1       0.05      0.07      0.06        95
           2       0.16      0.09      0.12       180
           3       0.28      0.48      0.35       190
           4       0.39      0.04      0.07       315
           5       0.03      0.09      0.04        55
           6       0.00      0.00      0.00         5

    accuracy                           0.14      1050
   macro avg       0.15      0.12      0.11      1050
weighted avg       0.23      0.14      0.13      1050
```

Label: 2

Label: 1

Label: 1

L’augmentation des données ne conduit pas toujours à une amélioration des performances:**nature des données** ou un ensemble de données plus important peut nécessiter **davantage d'époques** d’apprentissage pour que le modèle s'adapte correctement aux nouvelles données

Conclusion

- Clustering non supervisé : résultat non concluant
- Alternative envisageable : apprentissage supervisé

Aller plus loin

- Penser à utiliser des API pour augmenter le volume de données.
- Concentrer sur des Stopwords spécifiques au commerce électronique.
- Utiliser des approches d'apprentissage par transfert learning pour les données textuelles et les images.
- Obtention d’un jeu de données labelisé :
  - Collecte de donnée sur le site ou obtention externe,
  - Déploiement dans un second temps.

Données complémentaires : Edamam API

Stratégie de base:

- **Des besoins :** d'informations sur les produits liés au champagne tels que

```
foodId                object
label                 object
category              object
foodContentsLabel     object
image                 object
dtype: object
```

- **Des ressources** : **l'API Edamam Food and Grocery Database**

url = "https://api.edamam.com/api/food-database/v2/parser"

- **Accès à l'API** : on créet **un compte** sur la plateforme Edamam, et obtenons **APP_ID** et **APP_KEY** pour accéder à l'API.

```
APP_ID = 'your_app_id'
APP_KEY = ,your_app_key'
```

4)<u>Requête API</u> :utilisons un point de terminaison spécifique pour extraire des informations relatives au champagne à l'aide **d'une requête avec le paramètre ingr.**

```
# Demande à API(appliquer un filtre "champagne" au requête API)
querystring = {
    "ingr": "champagne",
    "app_id": APP_ID,
    "app_key": APP_KEYfoodId
}
response = requests.request("GET", url, params=querystring)
data = response.json()
```

label	category
Champagne	Generic foods
Champagne Vinaigrette, Champagne	Packaged foods
Champagne Vinaigrette, Champagne	Packaged foods
Champagne Vinaigrette, Champagne	Packaged foods
Champagne Vinaigrette, Champagne	Packaged foods
Champagne Dressing, Champagne	Packaged foods
Champagne Buttercream	Generic meals
Champagne Sorbet	Generic meals
Champagne Truffles	Generic meals
Champagne Vinaigrette	Generic meals

5)<u>Filtrage des résultats</u> : concentrons-nous sur les 10 premiers résultats et extrayons les champs

6)**Enregistrons des données** au format CSV pour une analyse ou un traitement ultérieur.

Champagne

Champagne Dressing, Champagne

Champagne Simply Dressed Vinaigrette, Champagne

RGPD

Légalité : j'utilise une API pour collecter des données qui dispose des autorisations nécessaires pour accéder aux données.

Finalité de la restriction : Je collecte des données uniquement à des fins de recherche et d'analyse des produits « champagne ».

Minimisation des données : je collecte uniquement les données nécessaires à mon projet (foodId, label,category, foodContentsLabel, image).

Précision : j'utilise un processus automatisé pour collecter des données

qui garantissent l'exactitude des données.

Limites de stockage : je stocke les données dans un fichier CSV

qui peut être facilement supprimé ou modifié selon les besoins.

De plus, je ne stocke pas de données sur des individus spécifiques ni sur leurs données personnelles,

ce qui minimise le risque de violation de la confidentialité des données.

MERCI DE VOTRE ATTENTION