FORMATION DATA SCIENTIST

PROJET 6: CLASSIFIEZ AUTOMATIQUEMENT DES BIENS DE CONSOMMATION

Presented By : Kourouma Sekouba

SOMMAIRE

1. **MISSION**

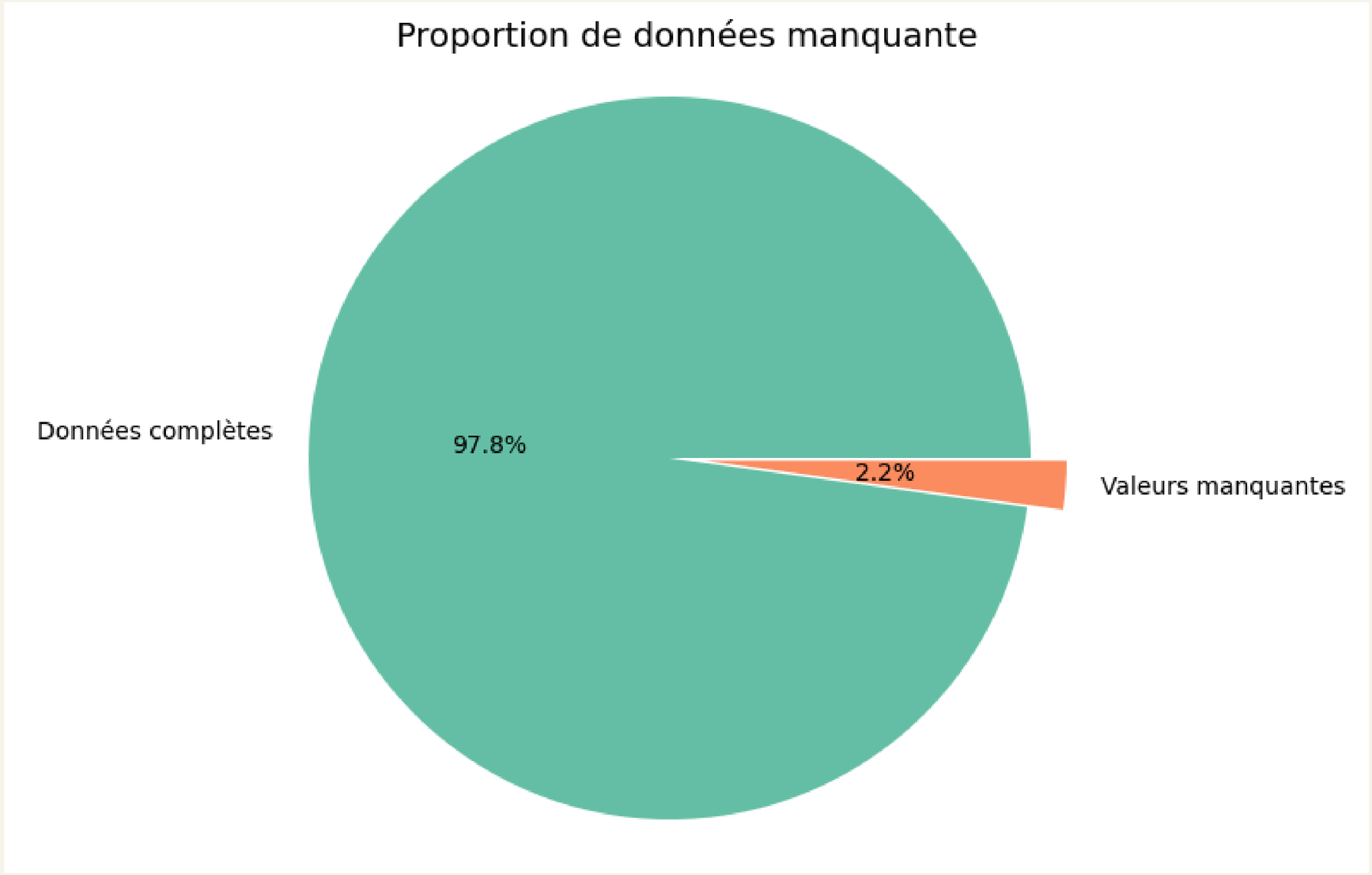
Les vendeurs proposent des articles à des acheteurs en postant une photo et une description.
L'objectif est de simplifier l'expérience utilisateur pour les vendeurs et les acheteurs en rendant plus fiable et efficace le processus de catégorisation des articles, afin de faciliter l'expansion de la place de marché

- Comment pouvons-nous automatiser la catégorisation des articles d'une place de marché en utilisant les données disponibles (images et descriptions textuelles) pour garantir une expérience fluide pour les utilisateurs et assurer une catégorisation cohérente ?

- Quels modèles et techniques d'extraction de fonctionnalités textuelles et visuelles seraient les plus adaptés pour garantir une classification précise et évolutive des articles en catégories ?

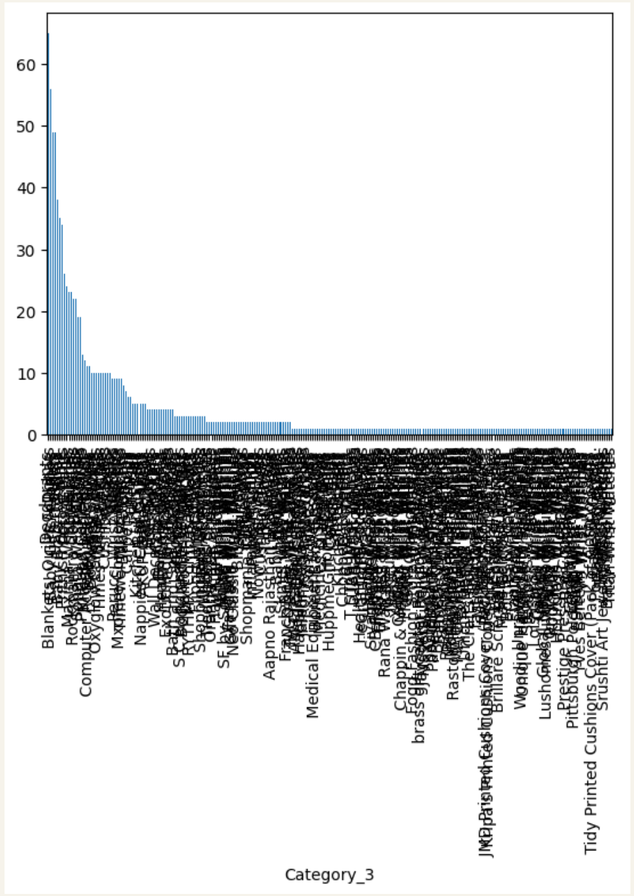
2. PRESENTATION DU JEU DE DONNÉE

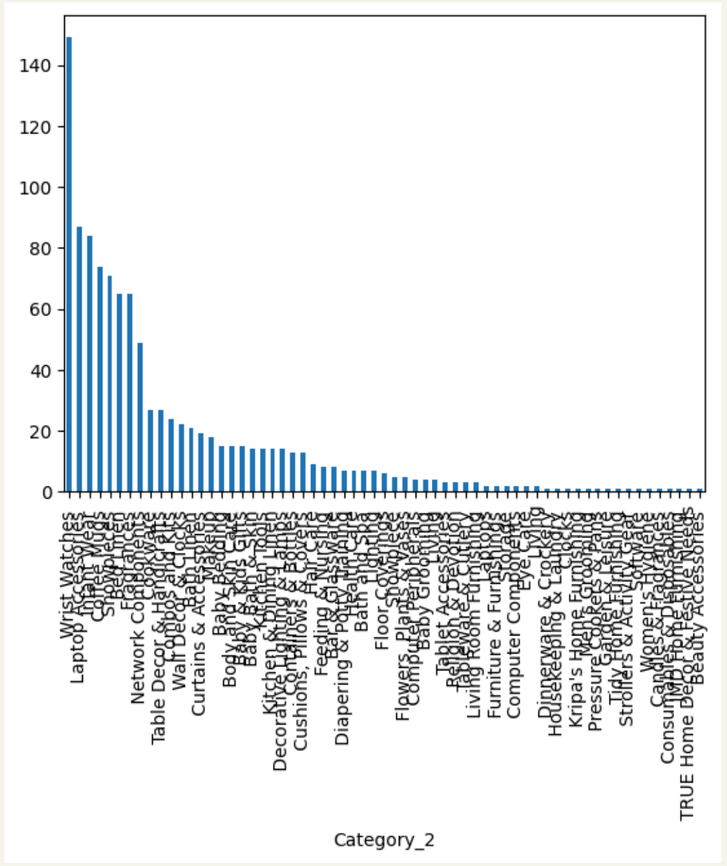
- La taille: (1050, 15)

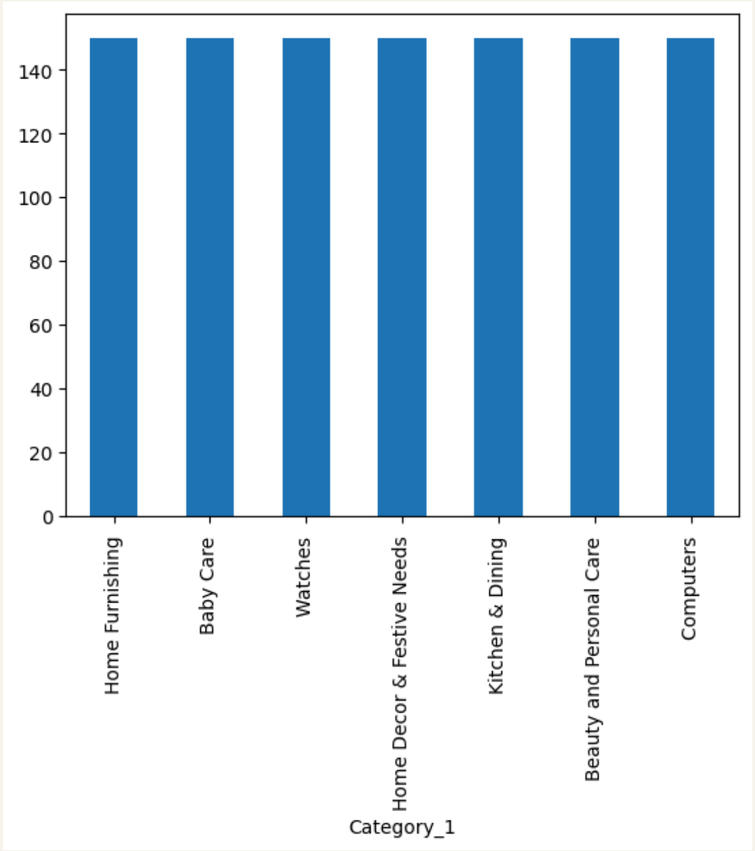


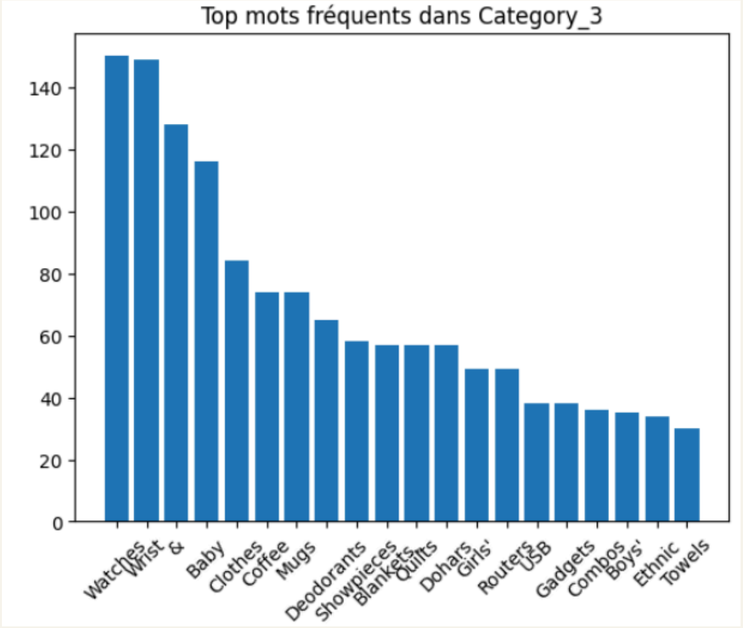
3. ANALYSE EXPLORATOIRE

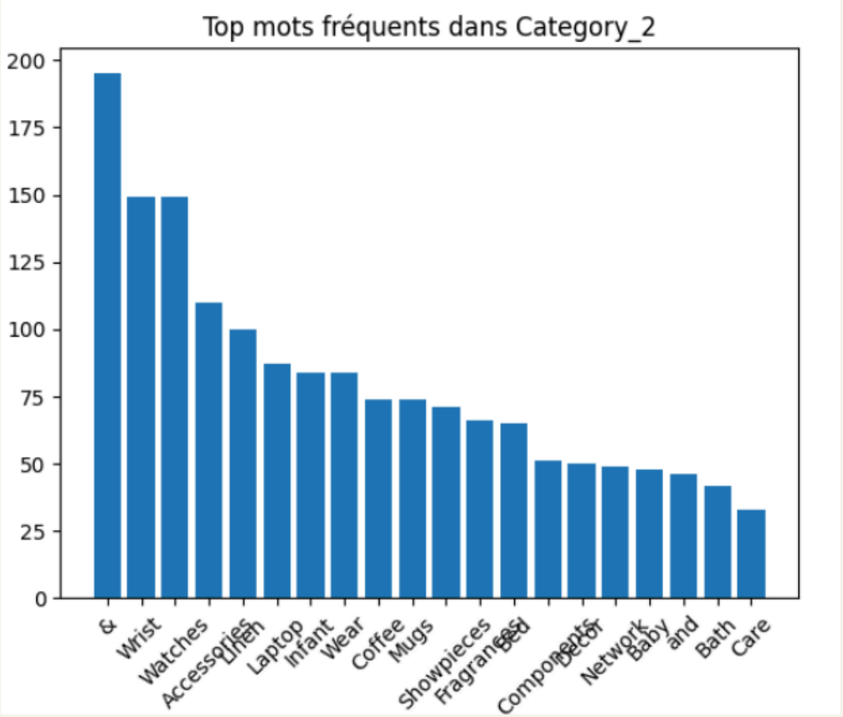
Category_1 présente une répartition plus équilibrée, contrairement aux catégories Category_2et Category_3

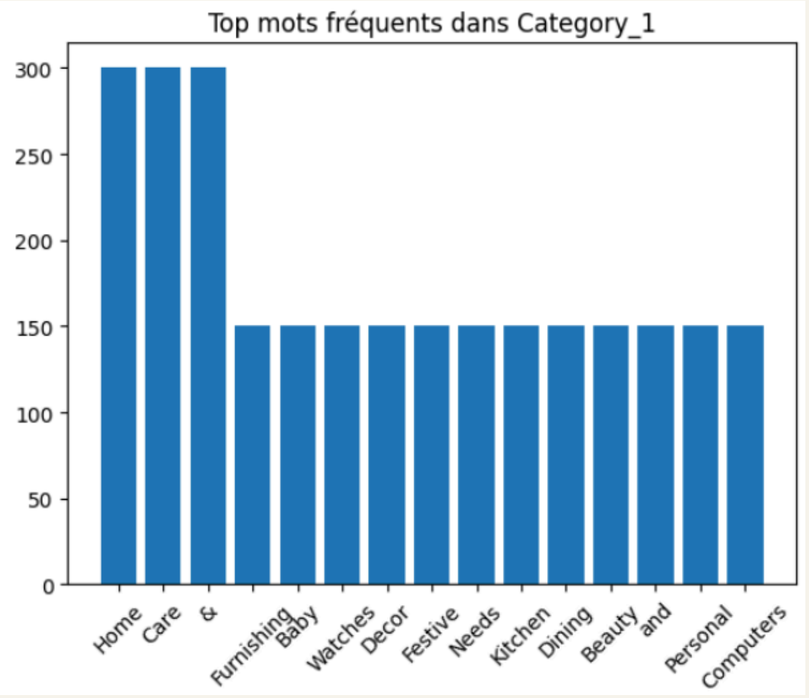




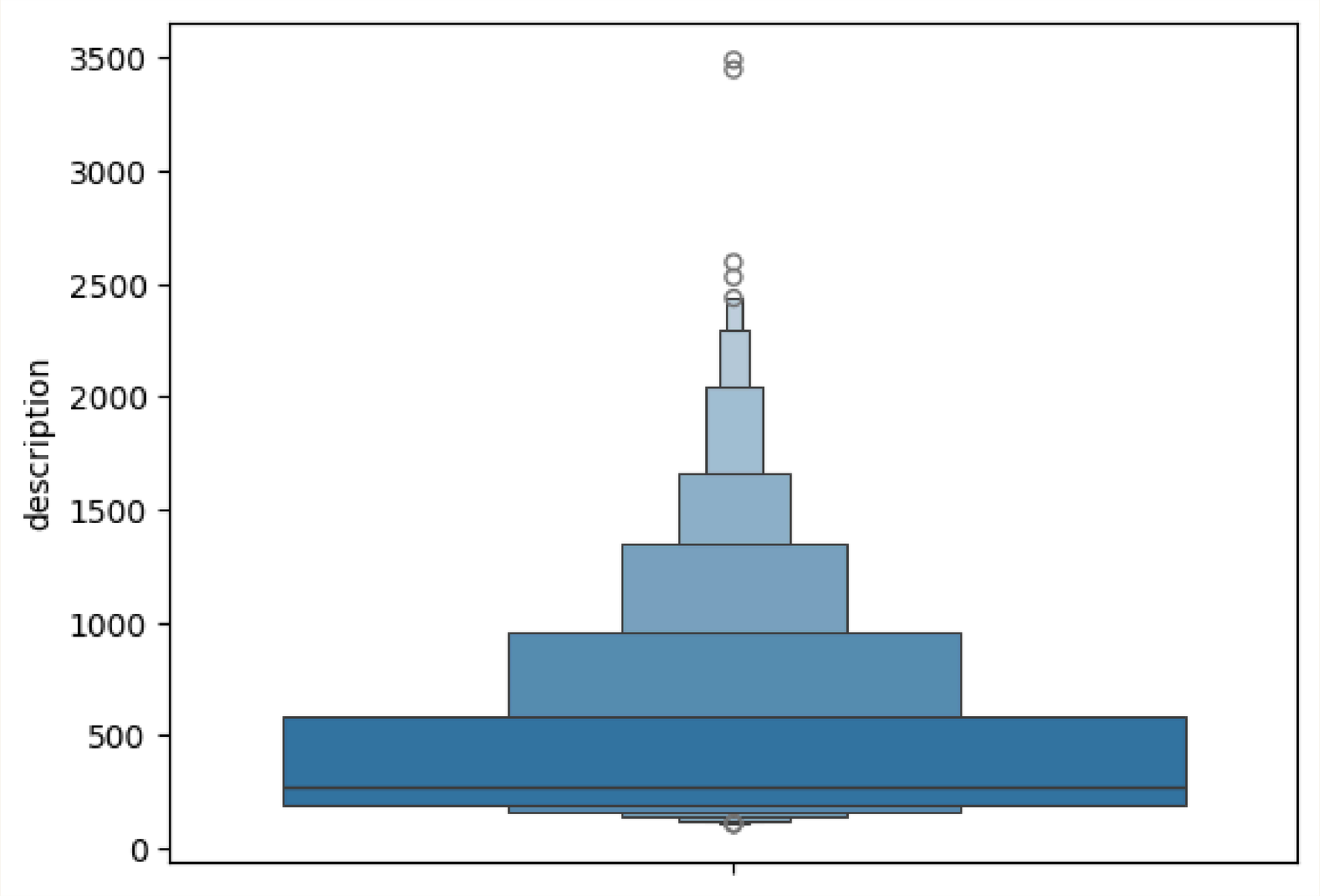



Top mots fréquents dans Category_3
0
20
40
60
80
100
120
140
Watches
Wrist
&
Baby
Clothes
Coffee
Mugs
Deodorants
Showpieces
Blankets
Quilts
Dohars
Girls'
Routers
USB
Gadgets
Combos
Boys'
Ethnic
Towels


Top mots fréquents dans Category_2
0
25
50
75
100
125
150
175
200
&
Wrist
Watches
Accessories
Linen
Laptop
Infant
Wear
Coffee
Mugs
Showpieces
Fragrances
Bed
Components
Decor
Network
Baby
and
Bath
Care


Top mots fréquents dans Category_1
0
50
100
150
200
250
300
Home
Care
&
Furnishing
Baby
Watches
Decor
Festive
Needs
Kitchen
Dining
Beauty
and
Personal
Computers

Le nombre de caractère des mots



4. ETUDE DE FAISABILITE DE CLASSIFICATION (TEXTE)

4.1 Méthodes NLP basique

4.1a Prétraitement des textes

- Tokenisation
- stemming
- lemmatization

4.1b méthodes basiques d'encodage de texte et réduction de dimension

- Bag of Words
- TfIdf

CountVectorizer , TF-IDF , Universal Sentence Encoder , Word2Vec , et BERT – sont des méthodes courantes en NLP pour représenter des mots, phrases ou documents sous une forme numérique compréhensible pour les modèles d'apprentissage automatique.

4.2 Méthodes NLP avancées

- Word2Vec
- BERT
- Universal Sentence Encoder

4.1 Méthodes NLP basique

4.1a Prétraitement des textes

- Tokenisation

processus de division d'un texte en unités plus petites, appelées tokens. (split, regex, nltk et spaCy)

- *Désaccentuation*
- *mettre en minuscule*
- *Suppression des stop Word*

Simplifier nos tokens

- stemming

est une technique qui réduit les mots à leur racine (ou stem), en supprimant généralement les suffixes et prefixe.

- lemmatization

Retrouver la racine du mot appelé lemme . l'infinitif , ou singulier etc.

	longueur_texte
count	1050.000000
mean	473.820952
std	457.910422
min	109.000000
25%	192.000000
50%	278.000000
75%	588.250000
max	3490.000000

dtype: float64

	long_apres_traitement
count	1050.000000
mean	324.009524
std	300.156251
min	61.000000
25%	131.250000
50%	200.000000
75%	423.000000
max	2330.000000

dtype: float64

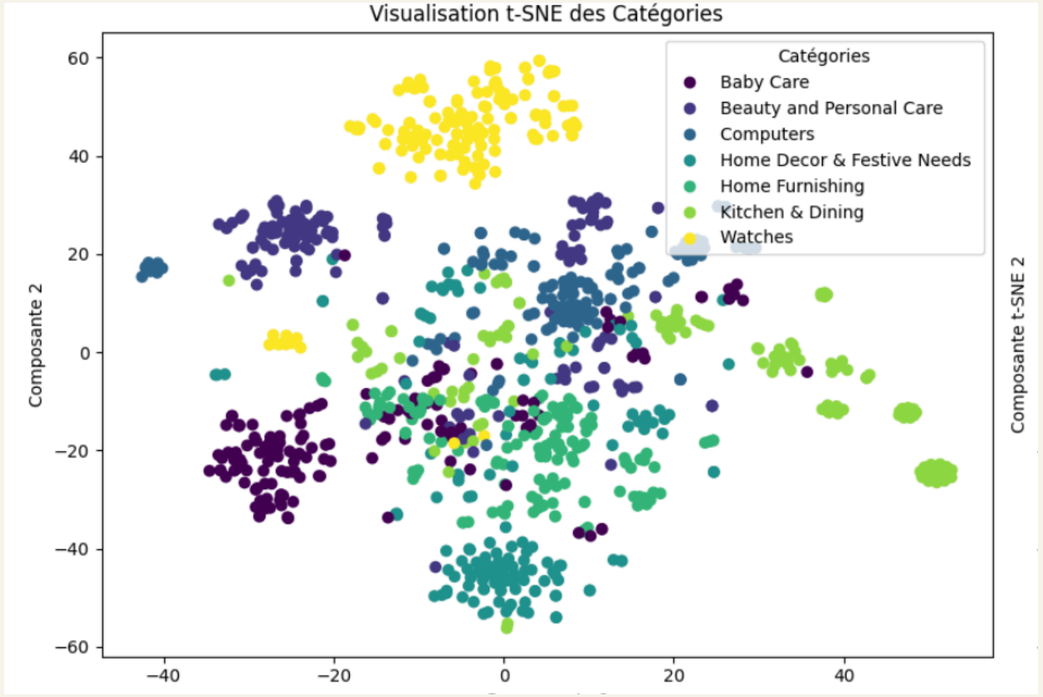
4.1b méthodes basiques d'encodage de texte et réduction de dimension

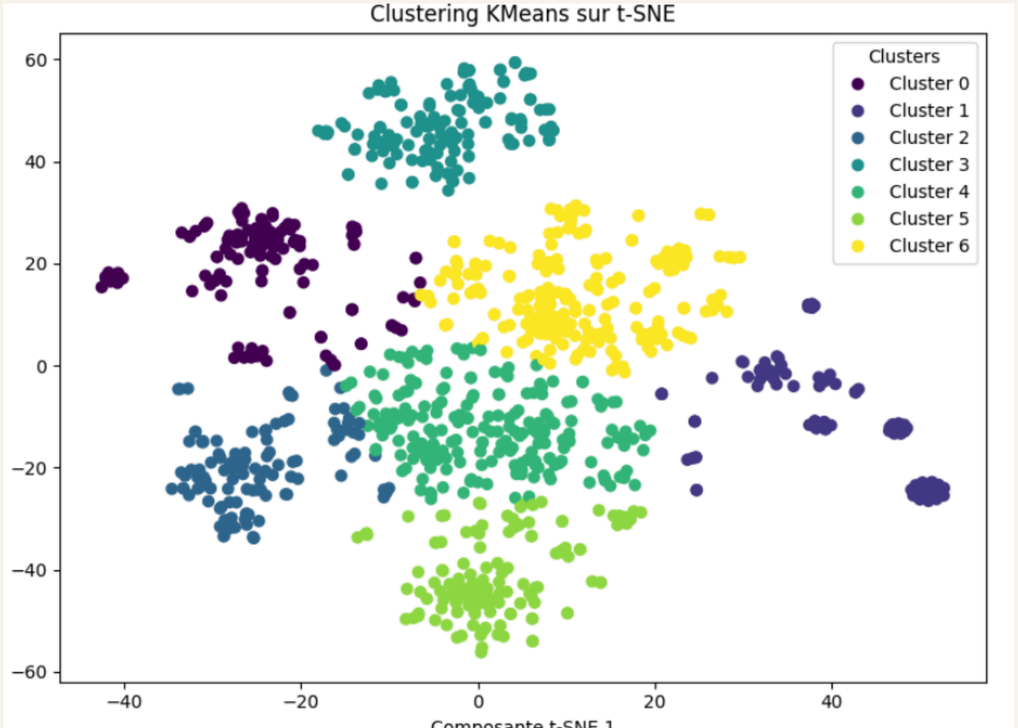
- countvectorizer

est une méthode simple pour transformer des textes en vecteurs. Il compte le nombre d'occurrences de chaque mot dans un document, créant un vecteur pour chaque document en fonction de la fréquence brute des mots.


```
Anciennes dimensions: (1050, 4553)
Nouvelles dimensions après PCA: (1050, 576)
```

Adjusted Rand Index (ARI): 0.39





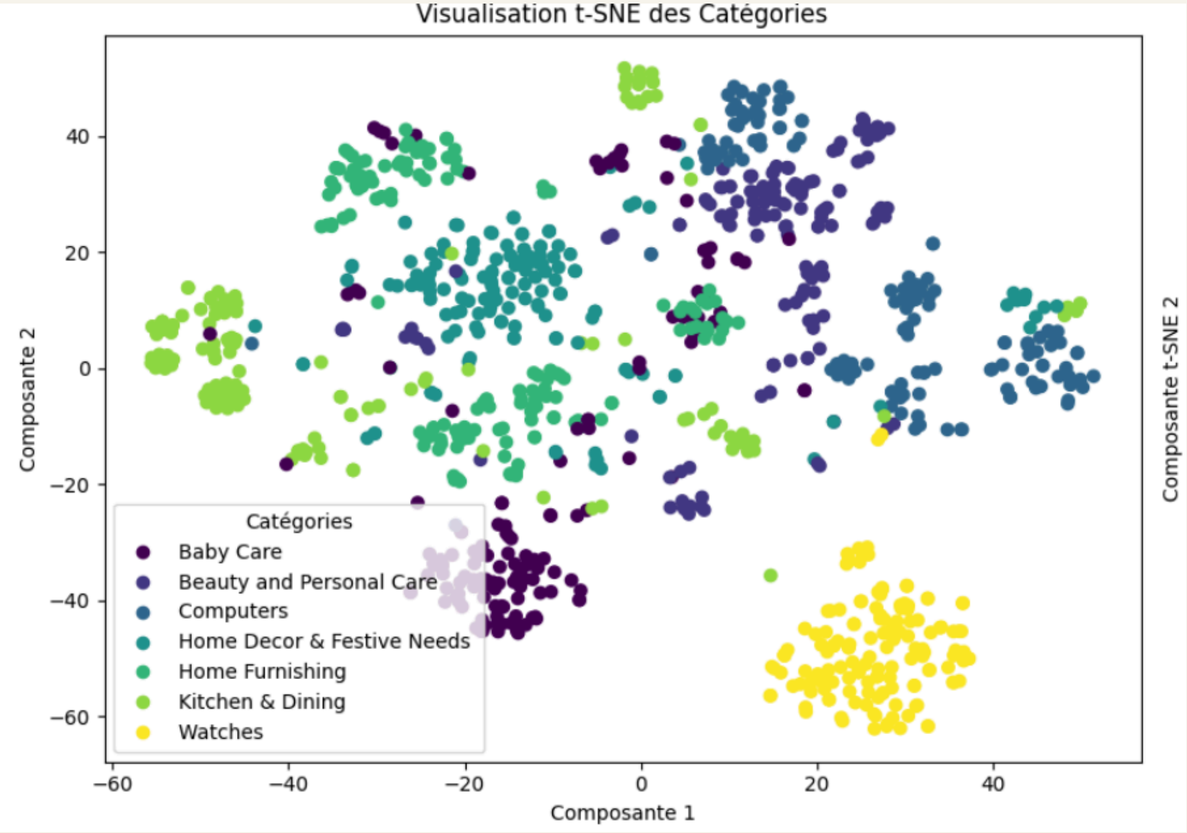
- ## TF-IDF VECTORIZER (TERM FREQUENCY INVERSE DOCUMENT FREQUENCY)

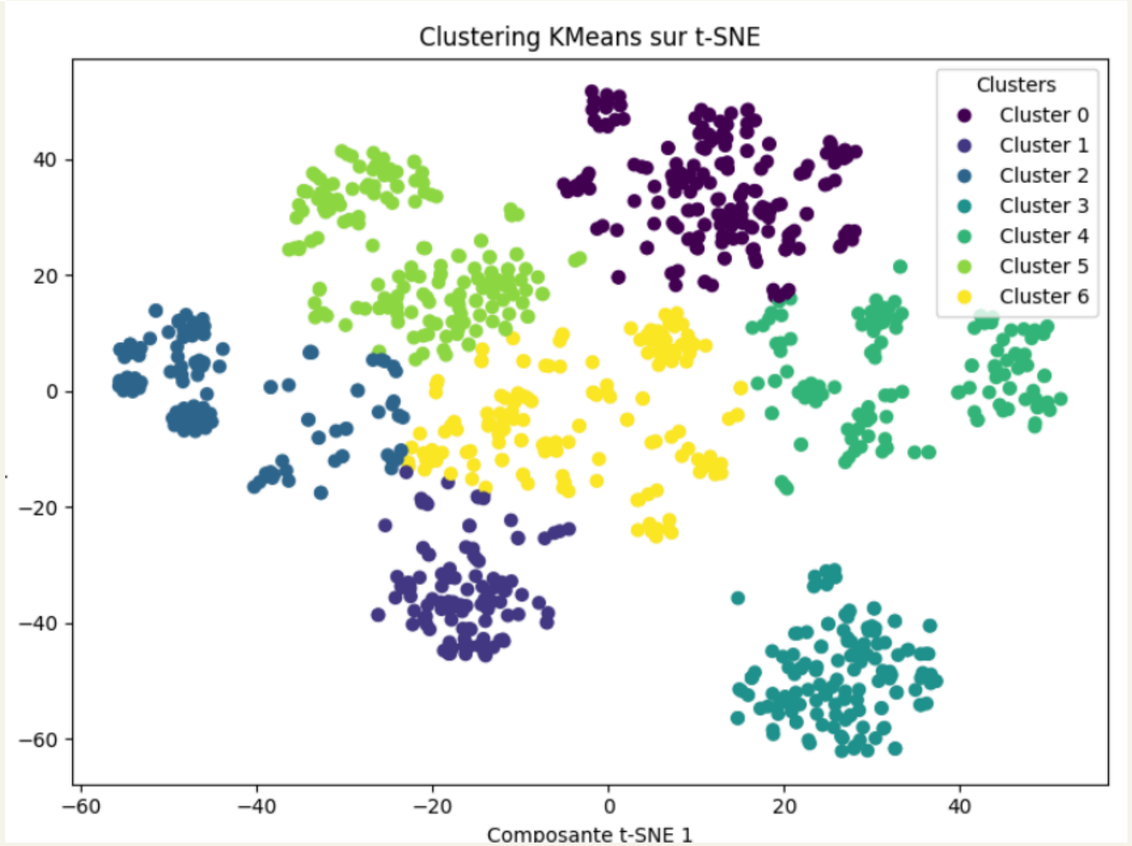
Est une méthode d'encodage qui pondère l'importance d'un mots dans le document par rapport à l'ensemble du corpus.

$$\text{poids} = \text{fréquence du terme} \times \text{indicateur similarité}$$

```
Anciennes dimensions: (1050, 4553)
Nouvelles dimensions après PCA: (1050, 822)
```

Adjusted Rand Index (ARI): 0.44



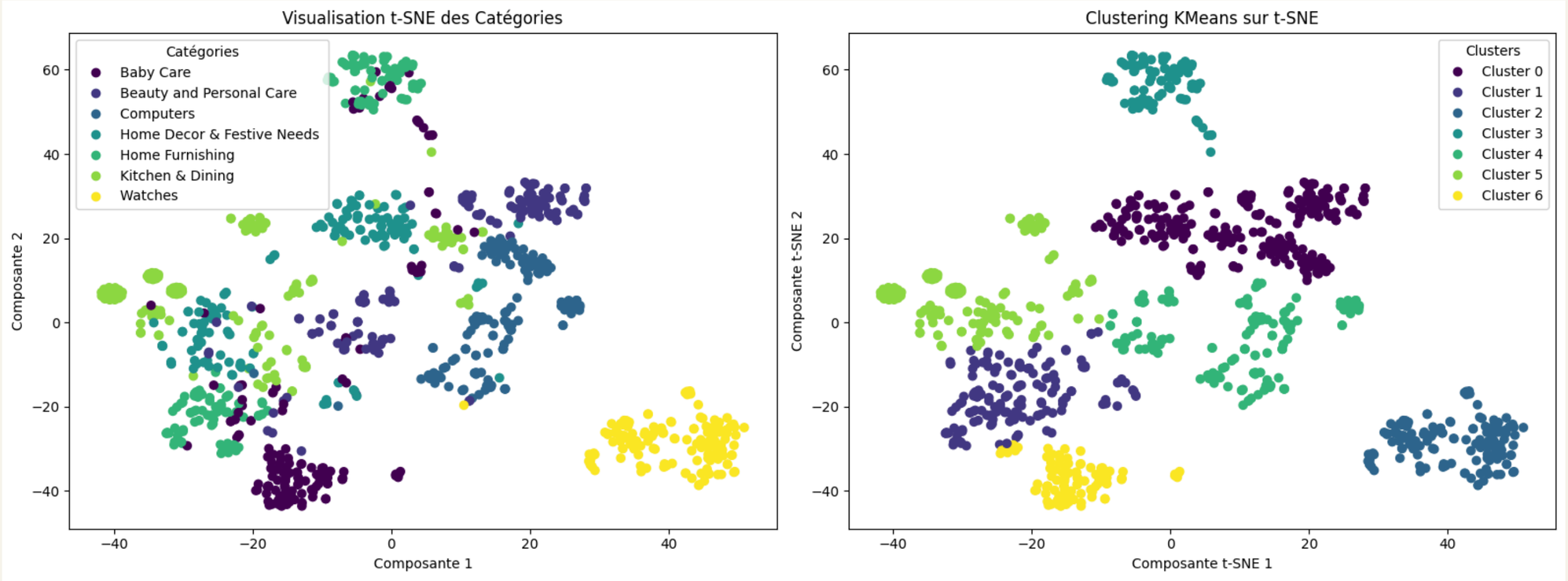


4.2 Méthodes NLP avancées

- Universal Sentence Encoder

est un modèle de Google qui représente des phrases ou des textes plus longs en vecteurs, tout en tenant compte des relations sémantiques entre les mots.

Adjusted Rand Index (ARI): 0.40

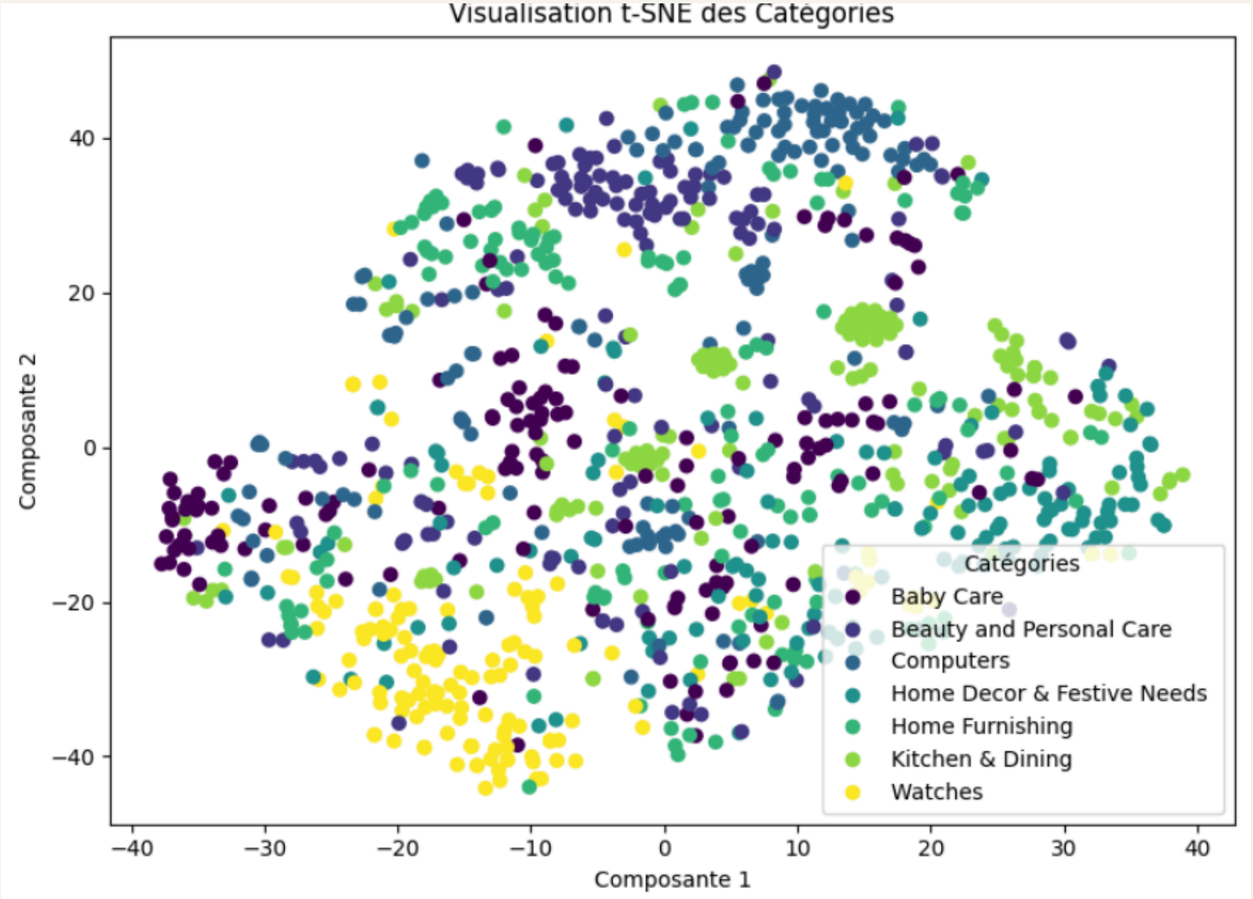


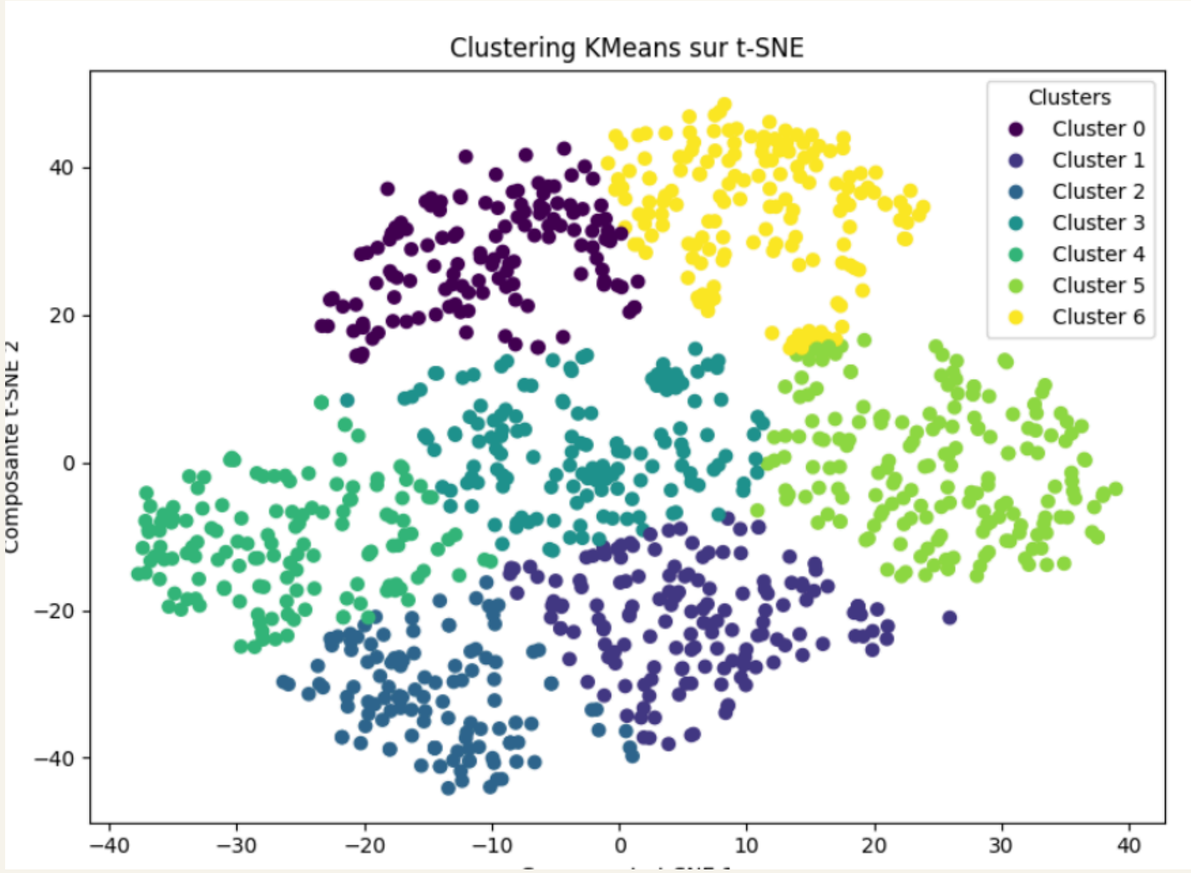
- Word2Vec

T.Mikolov, (Google), 2013

Propose d'apprendre les représentations vectorielles des mots présents dans un corpus en utilisant des réseaux de neurones.

Adjusted Rand Index (ARI): 0.14

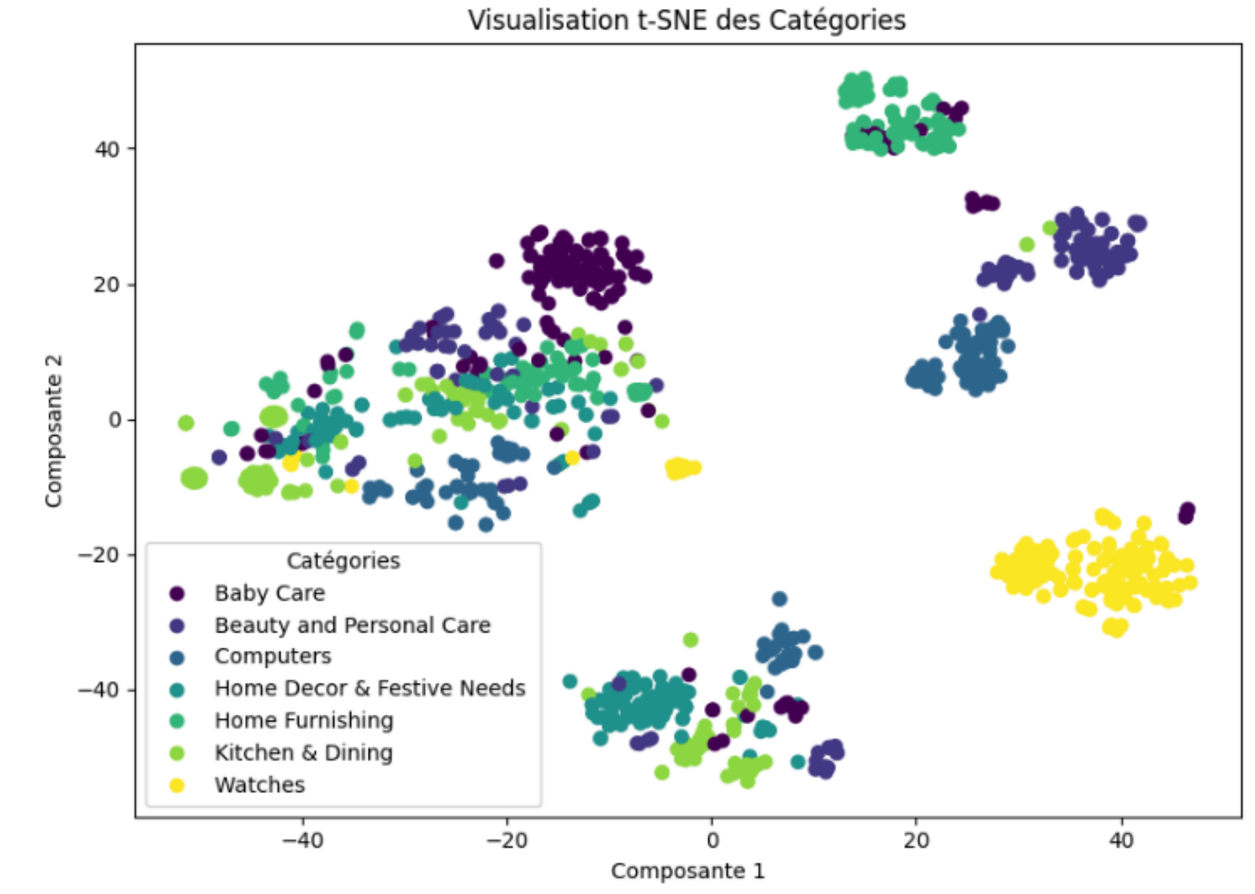


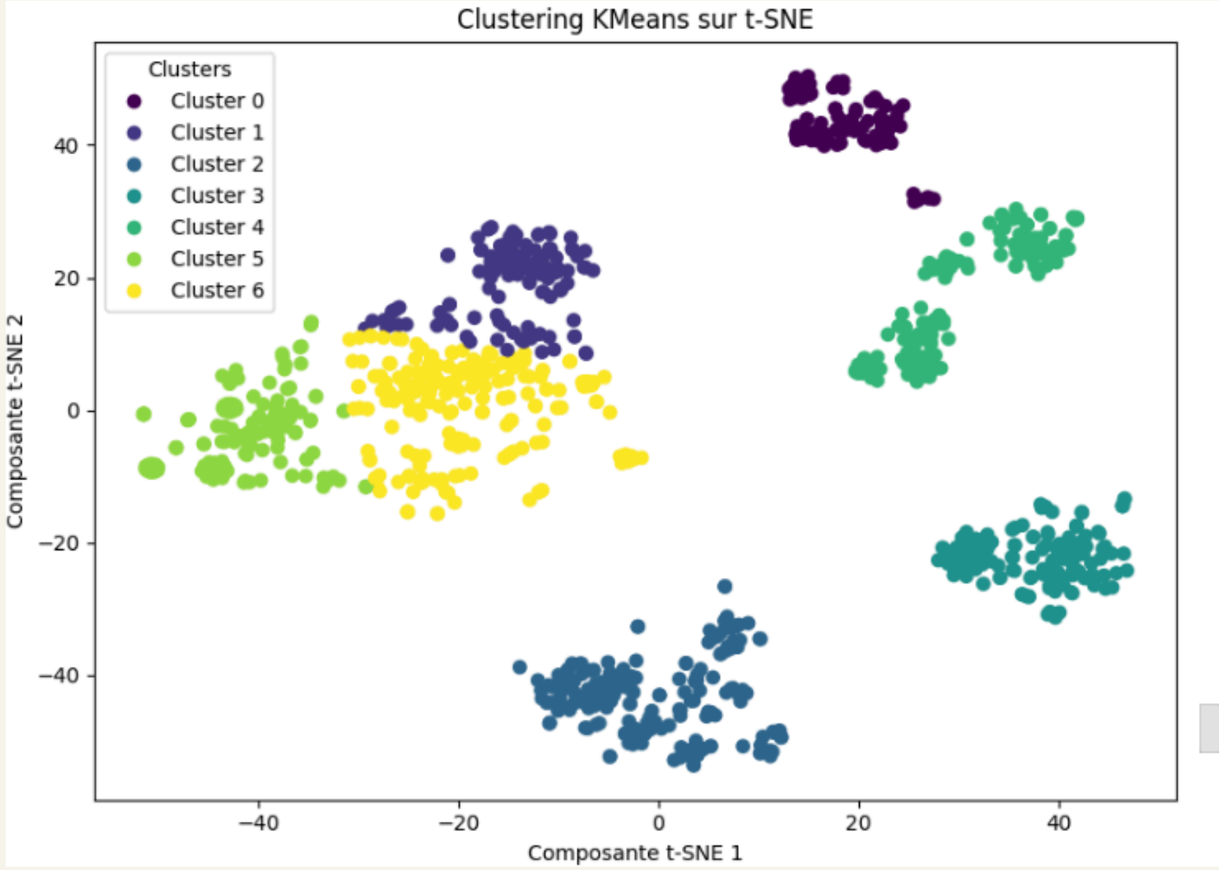


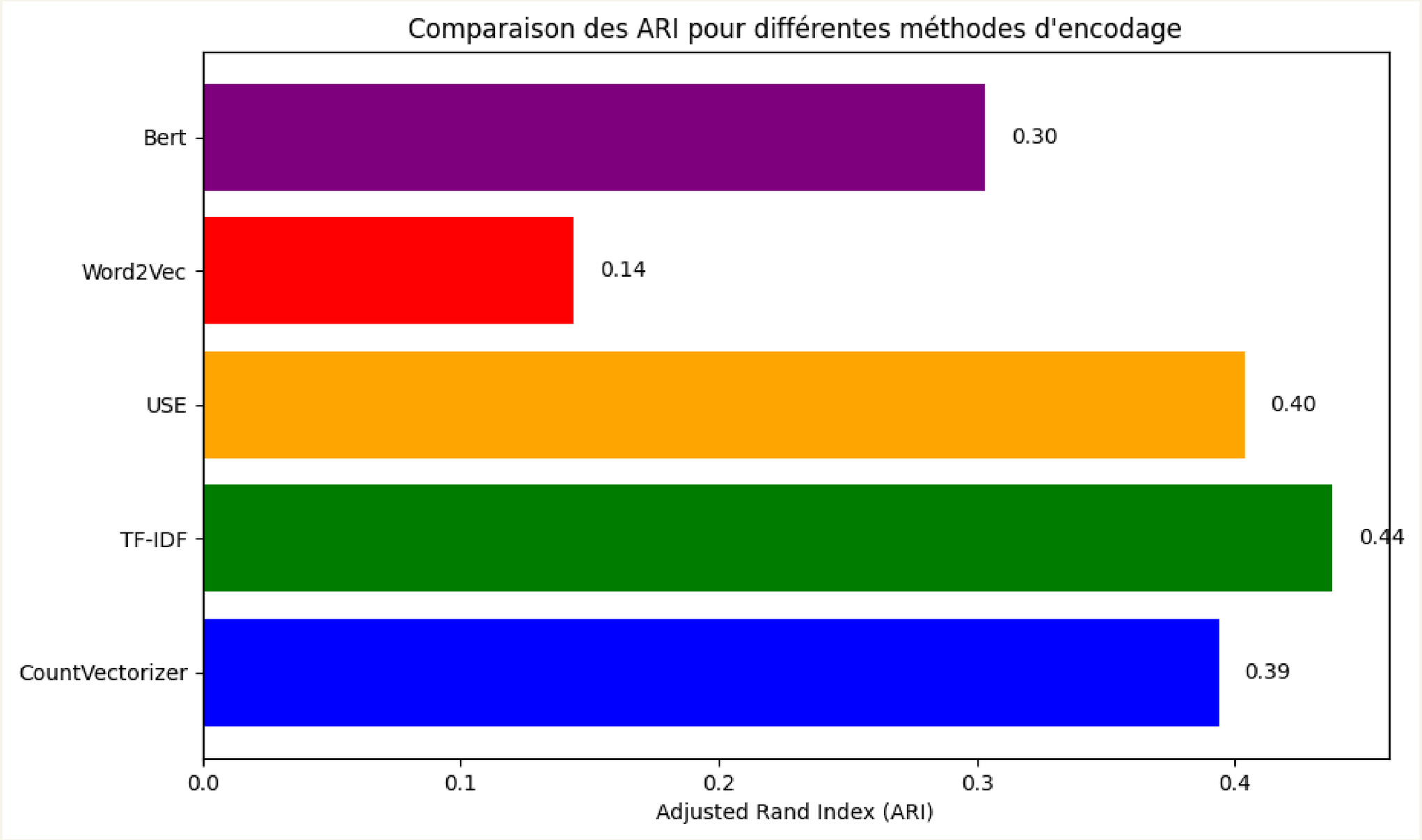
BERT

BERT est un modèle de traitement de langage naturel basé sur les transformateurs , développé par Google. Il capture les significations contextuelles des mots de manière bidirectionnelle, ce qui signifie qu'il prend en compte le contexte des mots avant et après chaque mot cible dans une phrase.

Adjusted Rand Index (ARI): 0.30



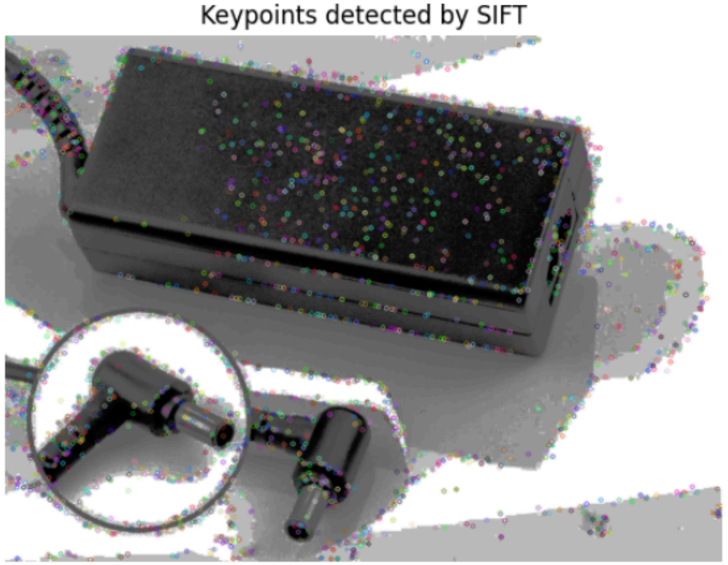



Comparaison des ARI pour différentes méthodes d'encodage
Bert
Word2Vec
USE
TF-IDF
CountVectorizer
0.30
0.14
0.40
0.44
0.39
0.0
0.1
0.2
0.3
0.4
Adjusted Rand Index (ARI)

5. ETUDE DE FAISABILITE DE CLASSIFICATION (IMAGE)

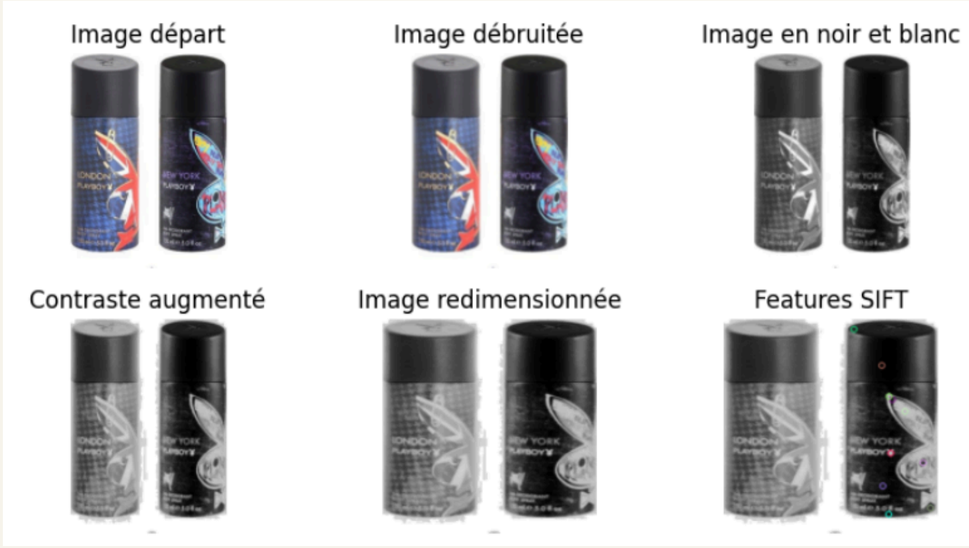
- Algorithme SIFT (Scale-Invariant Feature Transform)

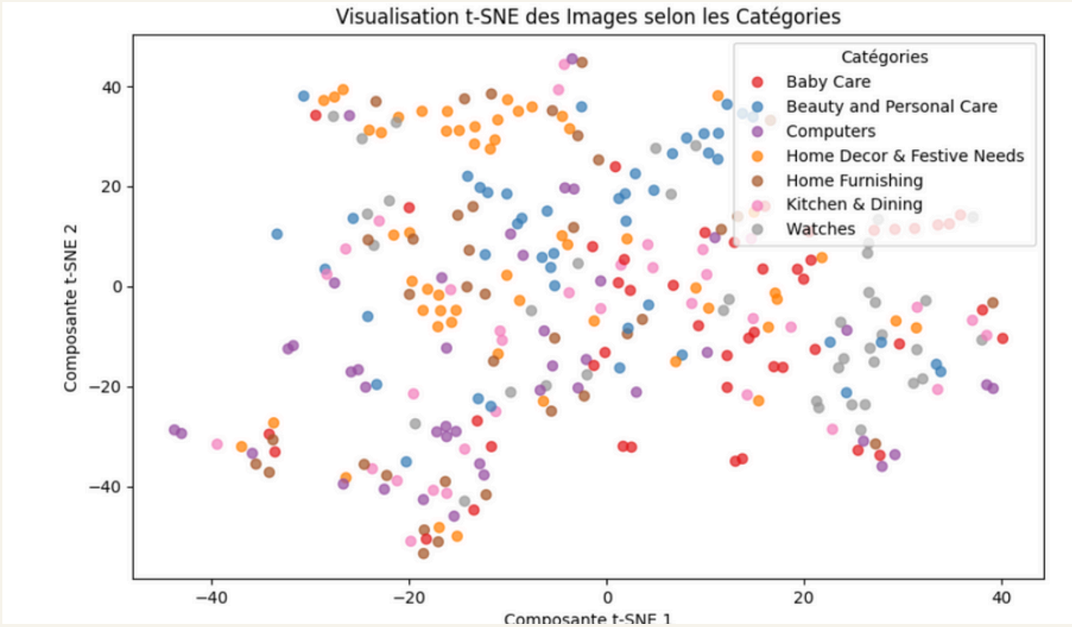
Chaque image contient des descripteurs,
qui sont des vecteurs de 128 éléments.




```
Descripteurs :  (2358, 128)

[[  0.   0.   0. ...  39.   3.   1.]
 [  1.   0.   0. ...   0.   2.   8.]
 [ 74.  10.   1. ...   0.   0.   0.]
 ...
 [  0.   0.  56. ...   0.  41. 129.]
 [ 15. 137.   0. ...   3.   0.   2.]
 [133. 111.   4. ...   0.   0.   0.]]
```

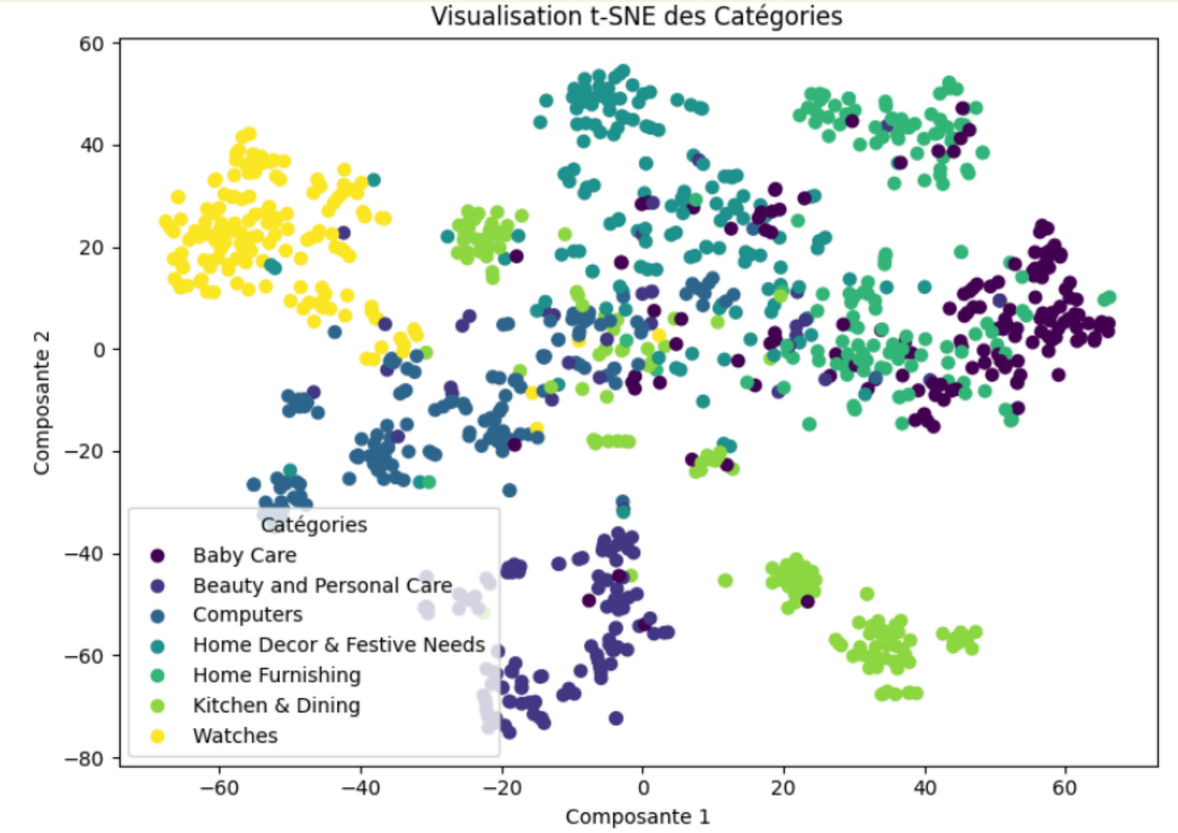


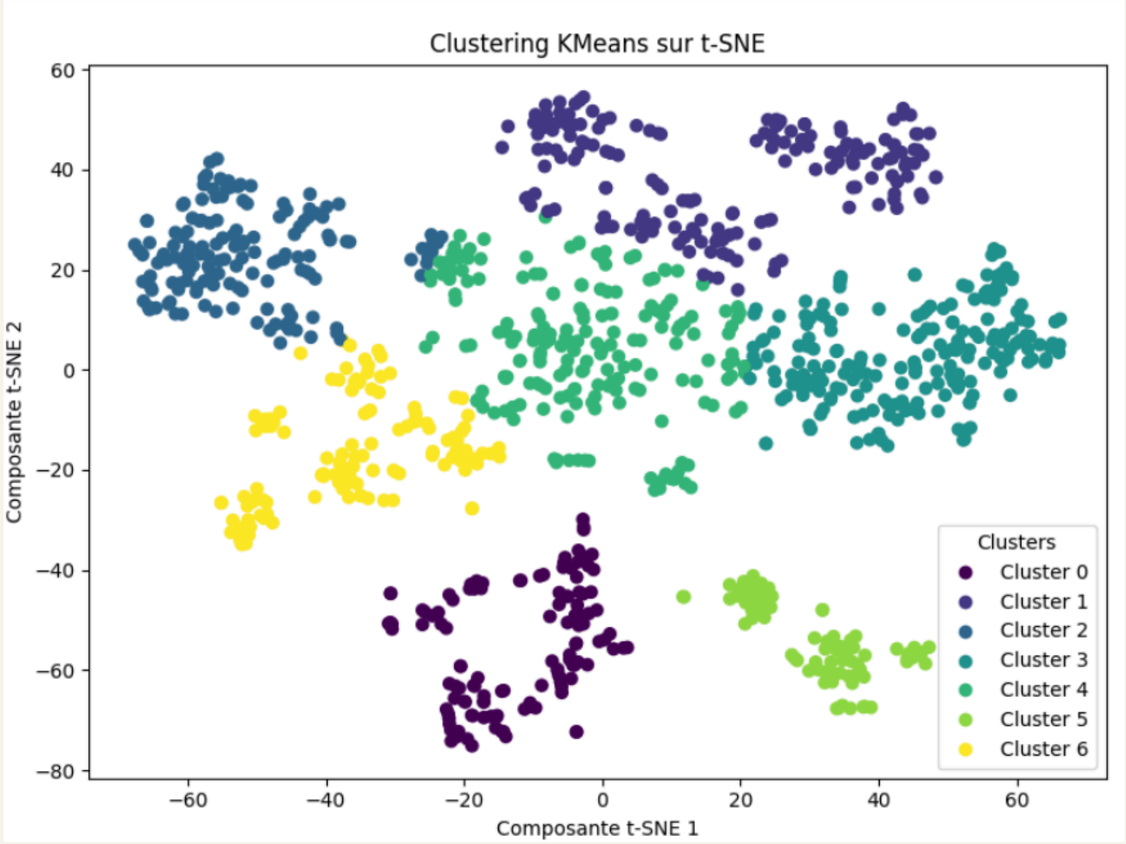


- # Algorithme de type CNN Transfer Learning

Le transfer learning, ou apprentissage par transfert, est une technique en apprentissage automatique qui utilise des modèles déjà entraînés sur des tâches similaires pour résoudre de nouveaux problèmes.

Adjusted Rand Index (ARI): 0.45





6. METHODE DE CLASSIFICATION

- ALGORITHME VGG16
  - TESTER AVEC LA DATA AUGMENTATION
    - TESTER EN MODIFIANT UN PARAMÈTRE COMME LE LEARNING RATE OU LE DROPOUT

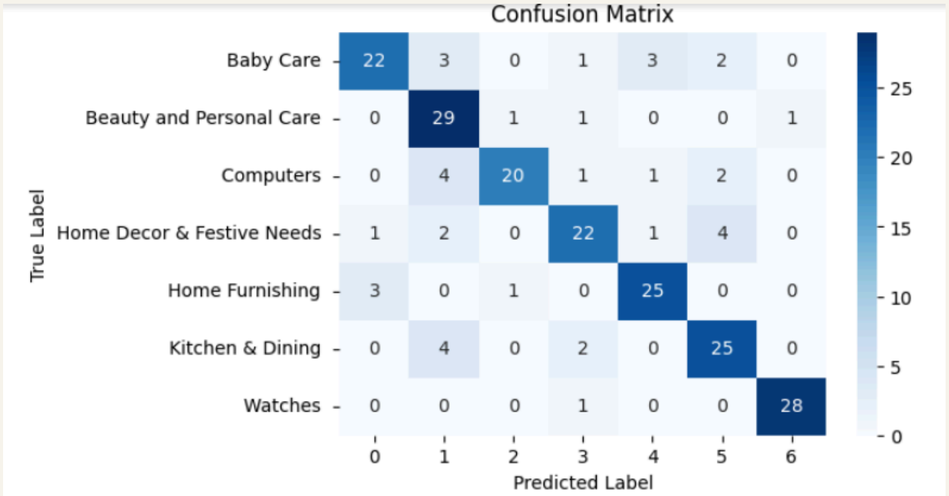
ALGORITHME EFFICIENTNET

• ALGORITHME VGG16

VGG16 EST UN MODÈLE DE RÉSEAU DE NEURONES CONVOLUTIFS (CNN) DÉVELOPPÉ PAR L'ÉQUIPE DE RECHERCHE VISUAL GEOMETRY GROUP (VGG) DE L'UNIVERSITÉ D'OXFORD.

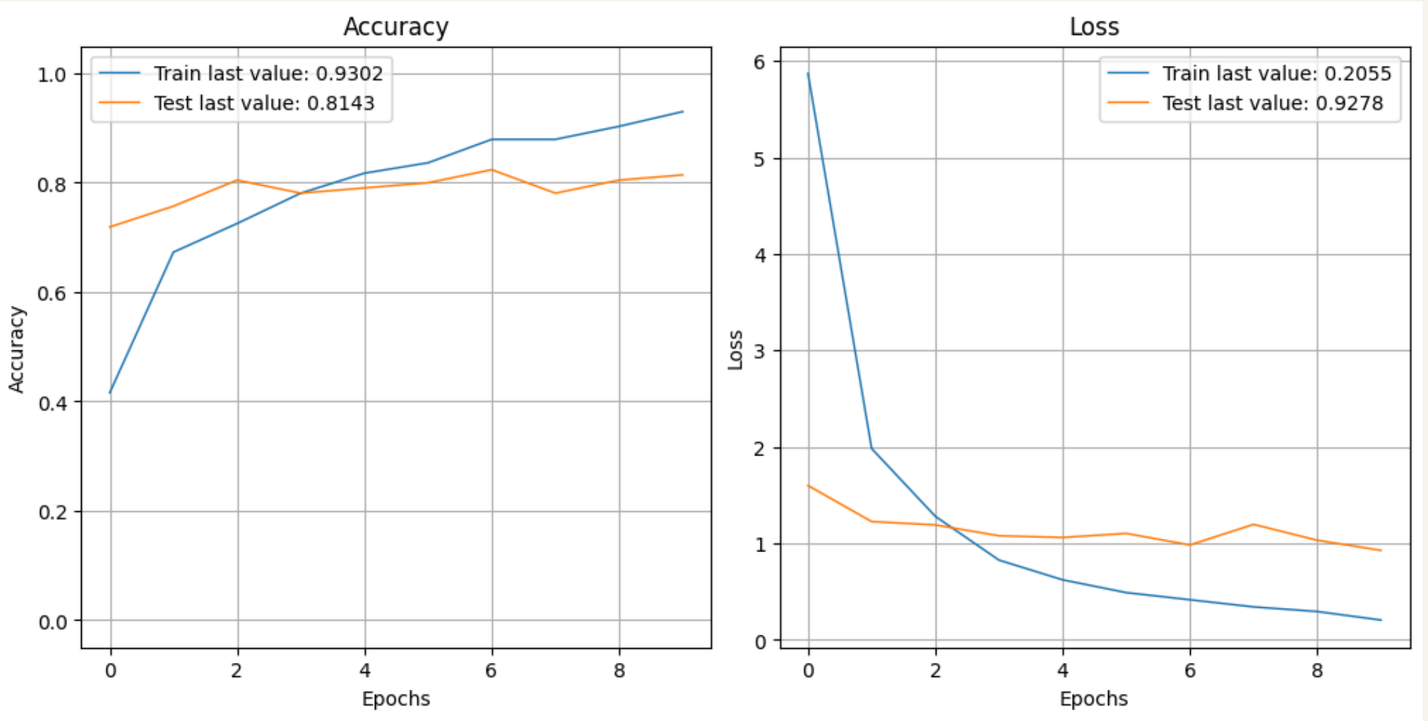
- VGG16 possède 16 couches de poids
- pré-entraîné sur le dataset ImageNet

```
Validation Accuracy :  0.8143
Test Accuracy       :  0.8333
```

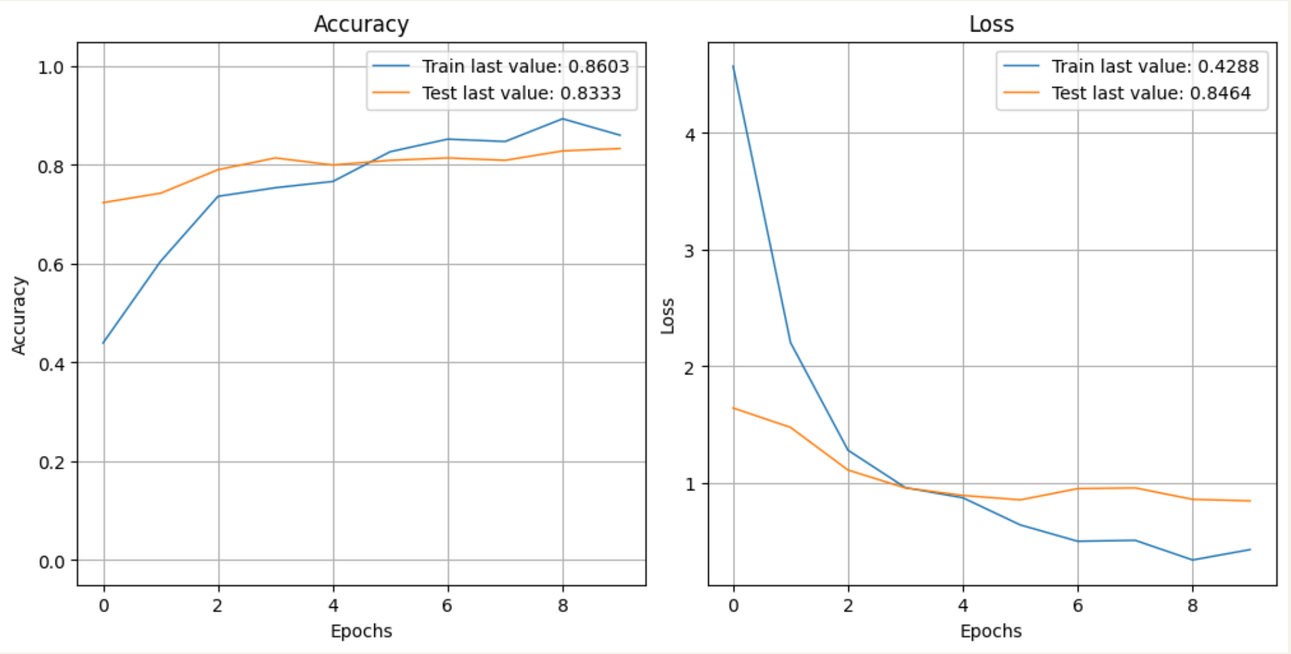


Classification Report:

	precision	recall	f1-score	support
Baby Care	0.85	0.71	0.77	31
Beauty and Personal Care	0.69	0.91	0.78	32
Computers	0.91	0.71	0.80	28
Home Decor & Festive Needs	0.79	0.73	0.76	30
Home Furnishing	0.83	0.86	0.85	29
Kitchen & Dining	0.76	0.81	0.78	31
Watches	0.97	0.97	0.97	29
accuracy			0.81	210
macro avg	0.83	0.81	0.82	210
weighted avg	0.82	0.81	0.81	210

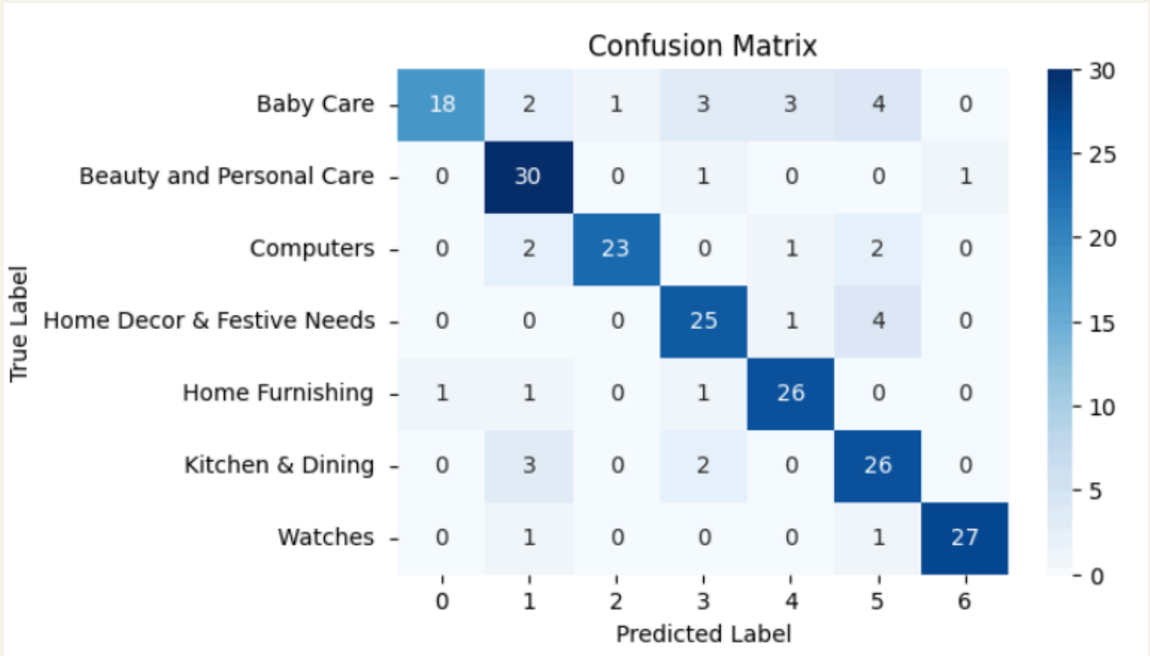


• TESTER AVEC LA DATA AUGMENTATION



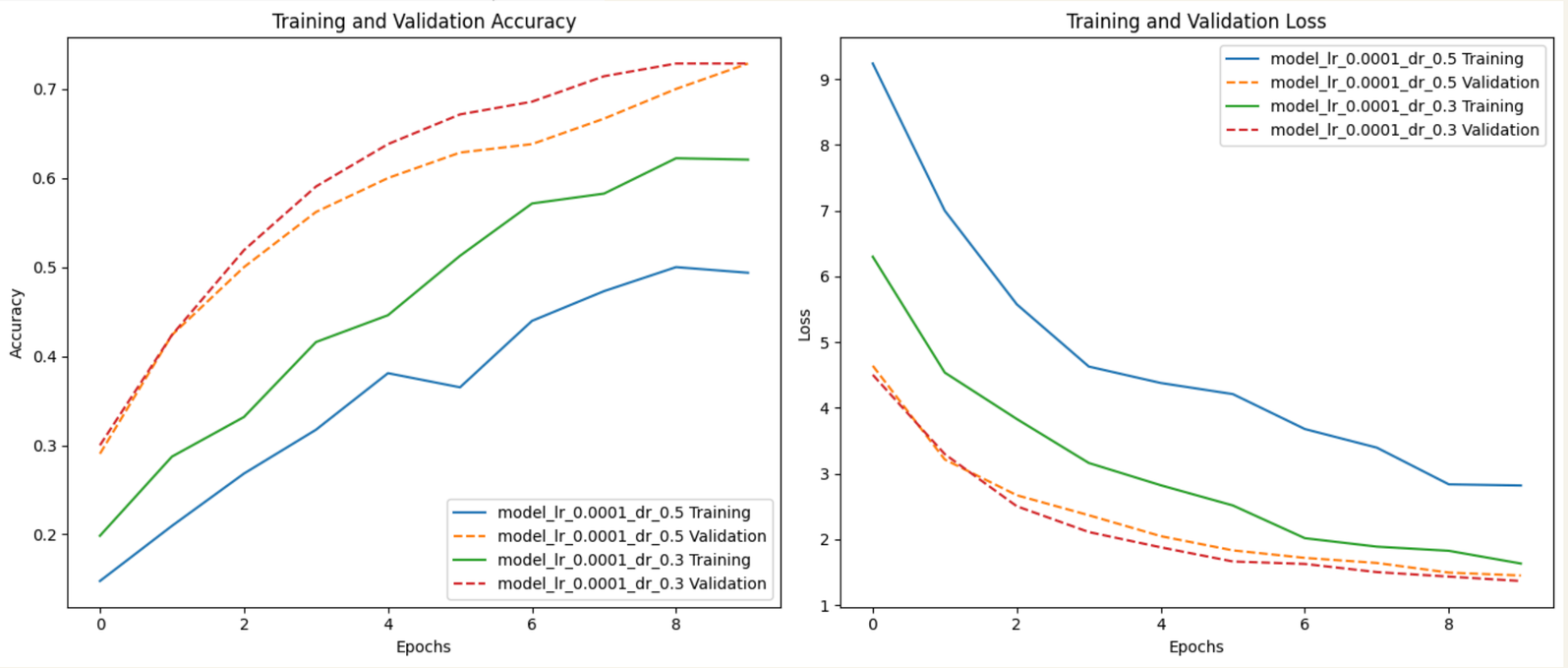

```
] # Création du générateur de Data Augmentation pour l'entraînement
  train_datagen = ImageDataGenerator(
      rotation_range=20,          # rotation aléatoire des images entre -20 et 20 degrés
      width_shift_range=0.2,      # décalage horizontal aléatoire jusqu'à 20%
      height_shift_range=0.2,     # décalage vertical aléatoire jusqu'à 20%
      shear_range=0.15,           # cisaillement aléatoire des images
      zoom_range=0.15,            # zoom avant/arrière aléatoire
      horizontal_flip=True,       # inversion horizontale aléatoire
      fill_mode='nearest'         # mode de remplissage pour les pixels hors cadre
  )
  train_generator = train_datagen.flow(X_train, y_train, batch_size=32)
```

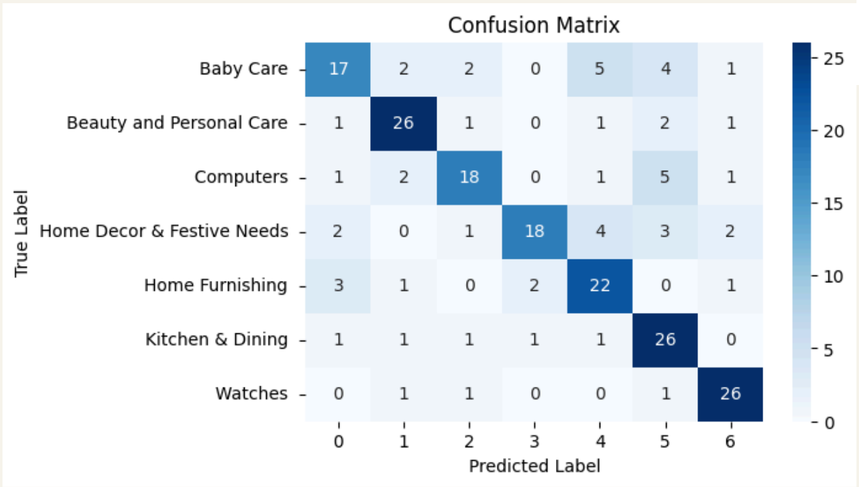
```
Validation Accuracy: 0.8333333134651184
Test Accuracy: 0.8380952477455139
```



• TESTER EN MODIFIANT UN PARAMÈTRE COMME LE LEARNING RATE OU LE DROPOUT

```
models = {
    'model_lr_0.0001_dr_0.5': create_model_fct(learning_rate=0.0001, dropout_rate=0.5),
    'model_lr_0.0001_dr_0.3': create_model_fct(learning_rate=0.0001, dropout_rate=0.3)
}
```

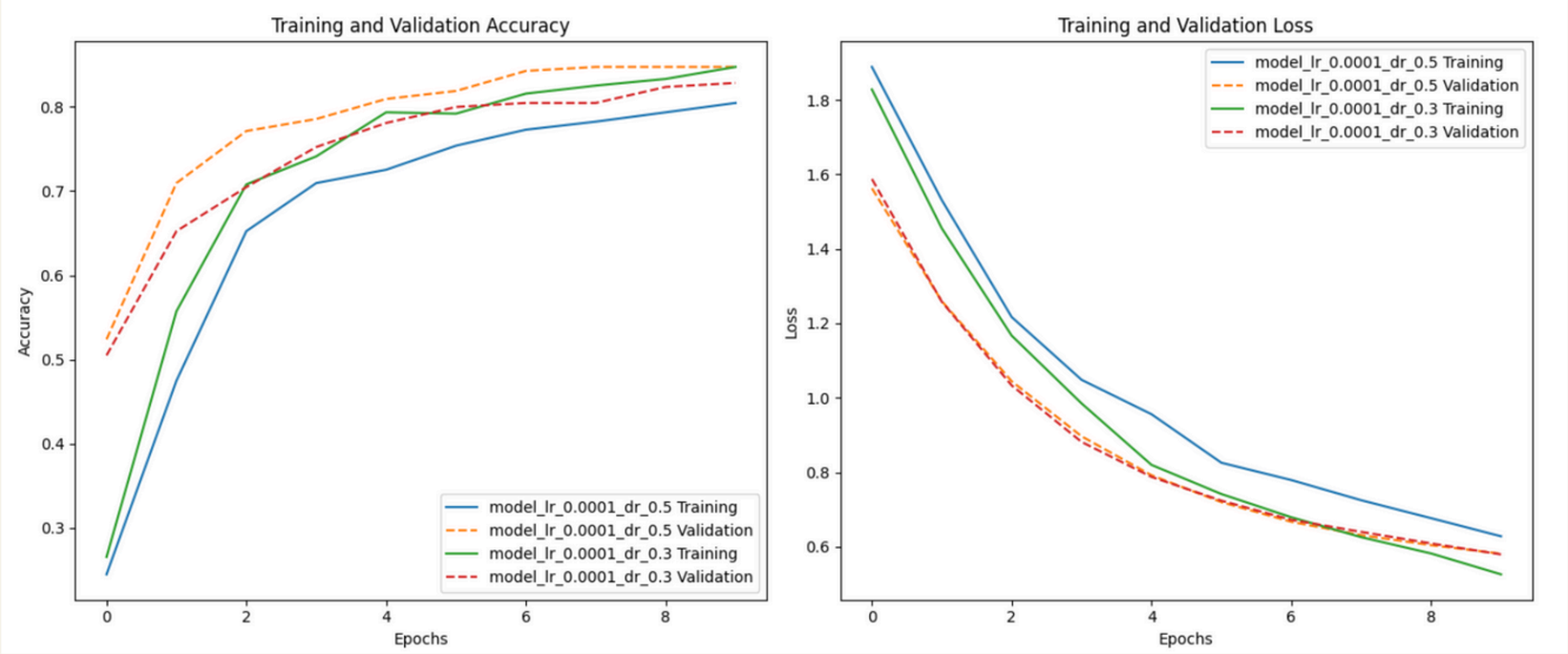


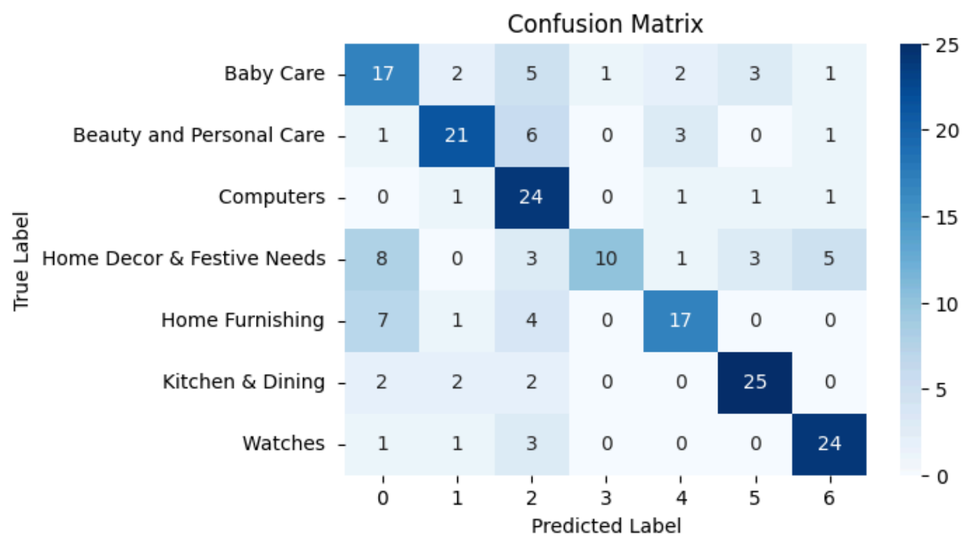



```
Evaluating model_lr_0.0001_dr_0.5 on the test set...
Test Accuracy for model_lr_0.0001_dr_0.5: 0.6619

Evaluating model_lr_0.0001_dr_0.3 on the test set...
Test Accuracy for model_lr_0.0001_dr_0.3: 0.6905
```

ALGORITHME EFFICIENTNET






```
Evaluating model_lr_0.0001_dr_0.5 on the test set...
Test Accuracy for model_lr_0.0001_dr_0.5: 0.6667

Evaluating model_lr_0.0001_dr_0.3 on the test set...
Test Accuracy for model_lr_0.0001_dr_0.3: 0.6952
```

APERÇU CHRONOLOGIQUE DE QUELQUES CNN

Xception

```
Evaluating model_xception_lr_0.0001_dr_0.5 on the test set...
Test Accuracy for model_xception_lr_0.0001_dr_0.5: 0.3286

Evaluating model_xception_lr_0.0001_dr_0.3 on the test set...
Test Accuracy for model_xception_lr_0.0001_dr_0.3: 0.3190
```

Resnet50

```
Evaluating model_resnet50_lr_0.0001_dr_0.5 on the test set...
Test Accuracy for model_resnet50_lr_0.0001_dr_0.5: 0.8476

Evaluating model_resnet50_lr_0.0001_dr_0.3 on the test set...
Test Accuracy for model_resnet50_lr_0.0001_dr_0.3: 0.8381
```

Modèle	Année	Paramètres	Performance	Architecture et Innovations
LeNet-5	1998	60 000	~99% sur MNIST	7 couches, premières convolutions et pooling, conçu pour la reconnaissance de chiffres manuscrits
AlexNet	2012	60 millions	16,4% erreur (top-5) sur ImageNet	8 couches (5 conv + 3 FC), activation ReLU, utilisation de GPU et dropout
VGGNet	2014	138-144 millions	7,3% erreur (top-5) sur ImageNet (VGG-19)	Convolutions 3x3 empilées, très profond mais coûteux en calcul
GoogLeNet (Inception v1)	2014	4 millions	6,7% erreur (top-5) sur ImageNet	Modules Inception (1x1, 3x3, 5x5), optimisation des paramètres avec convolutions 1x1
ResNet	2015	18 à 60 millions	3,6% erreur (top-5) sur ImageNet (ResNet-152)	Connexions résiduelles, très profond (jusqu'à 152 couches), permet un entraînement efficace des réseaux profonds
Inception v3	2015	23 millions	3,5% erreur (top-5) sur ImageNet	Modules Inception améliorés, factorisation des convolutions pour réduire les calculs
Xception	2017	23 millions	Similaire à Inception v3	Convolutions séparables en profondeur, inspiré de l'Inception pour plus d'efficacité
EfficientNet	2019	5 à 66 millions	84,4% précision (EfficientNet-B7)	Technique de "scaling" (ajustement largeur, profondeur, résolution), très efficace pour un bon rapport taille/performance
Vision Transformer (ViT)	2020	12 à 630 millions	88,5% précision (ViT-Large)	Transformateurs appliqués aux images, divisées en patches, offrant des performances élevées dans la classification d'images

7. CONCLUSION

- Efficacité des techniques employées
- Avantages de l'approche choisie
- Perspectives d'amélioration
- Impact sur l'expérience utilisateur