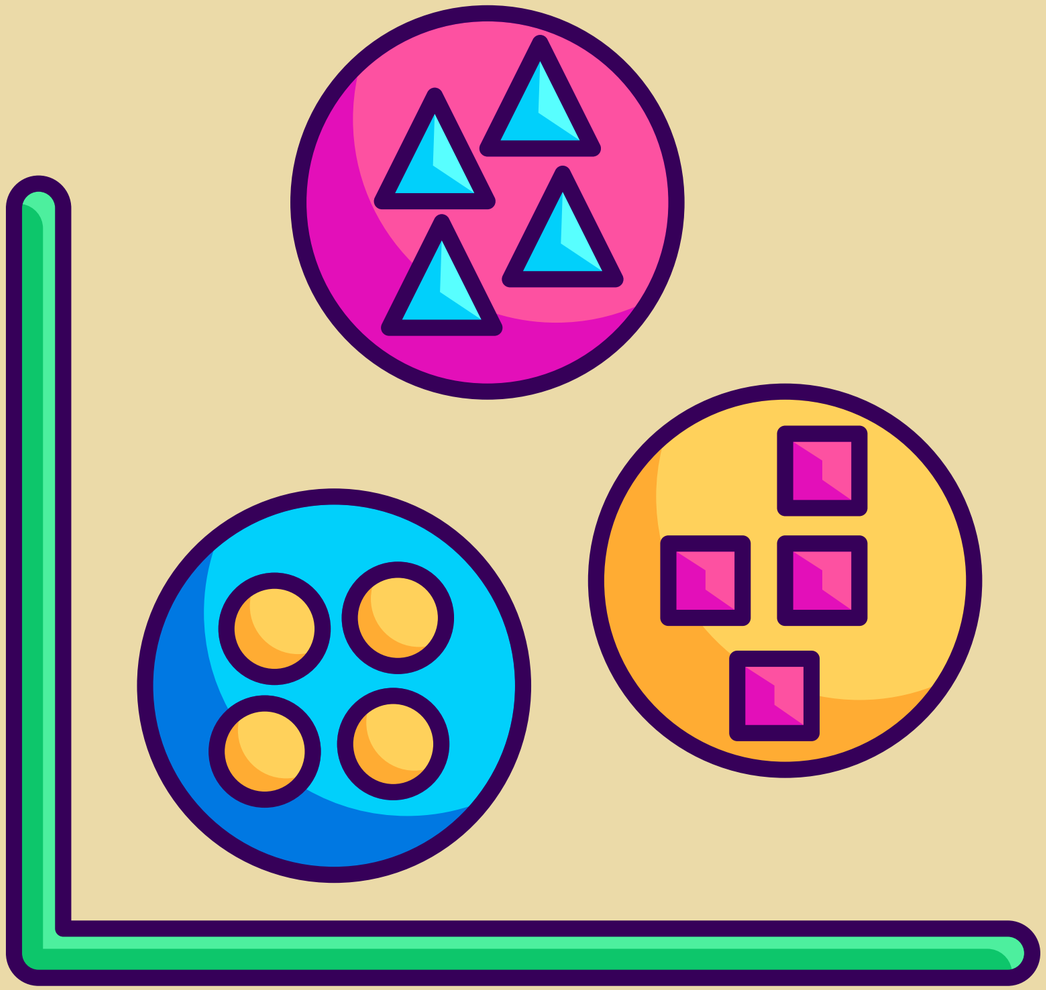


PROJET 6

Classifiez automatiquement des biens de consommation

Alexandre ROGUES - Parcours Data Scientist

e-commerce marketplace

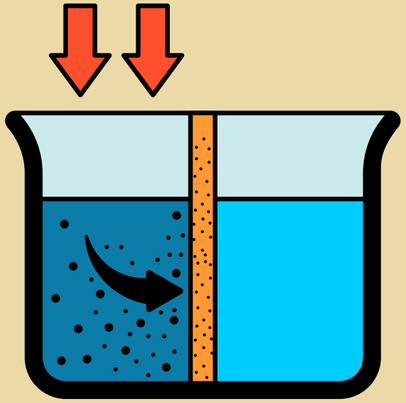
LABELLISATION AUTOMATIQUE

- UX Vendeurs : faciliter la mise en ligne
- UX Clients : faciliter la recherche

Perspective : passage à l’échelle

Partie 1 - A

PRÉTRAITEMENT



Pourquoi prétraiter ?

- Améliorer la qualité des données
- Optimiser les performances des modèles
- Faciliter l'extraction de features
- Réduire les coûts de calcul
- Permettre une meilleure interprétation

2 types de données traitées :

- Texte (description de l’objet)
- Image (illustration de l’objet)

Prétraitement des Données Textuelles

NETTOYAGE DES TEXTES

(suppression de la ponctuation, nombres, mise en minuscules, stopwords)

TOKENISATION
STEMMING
LEMMATISATION

Specifications of Brillare Science Dandruff Control Shampoo & Intenso Creme Combo (Set of) Combo Set Details Combo Set Contents 1 Dandruff Control Shampoo 150ml, 1 Dandruff Control Intenso Creme 125g Ideal For Women, Men Organic No General Traits Professional Care Yes

initial

specifications brillare science dandruff control shampoo intenso creme combo set combo set details combo set contents dandruff control shampoo ml dandruff control intenso creme g ideal women men organic general traits professional care yes

cleaned

specif brillar scienc dandruff control shampoo intenso creme combo set combo set detail combo set content dandruff control shampoo ml dandruff control intenso creme g ideal women men organ gener trait profession care ye

Stemmed

specification brillare science dandruff control shampoo intenso creme combo set combo set detail combo set content dandruff control shampoo ml dandruff control intenso creme g ideal woman men organic general trait professional care yes

lemmatised

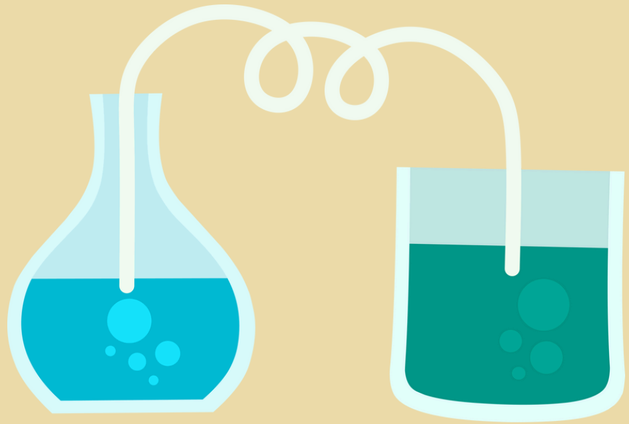
Prétraitement des Données Images

TRANSFORMATION EN NIVEAUX DE GRIS

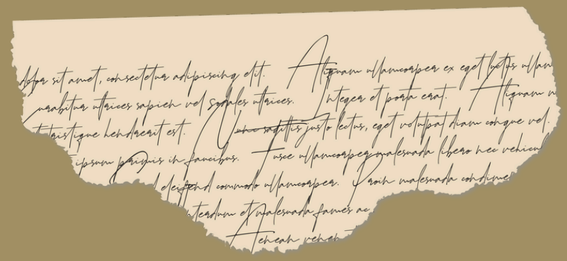
FILTRAGE DU BRUIT, AJUSTEMENT DU CONTRASTE

Partie 1 - B

EXTRACTION DE FEATURES



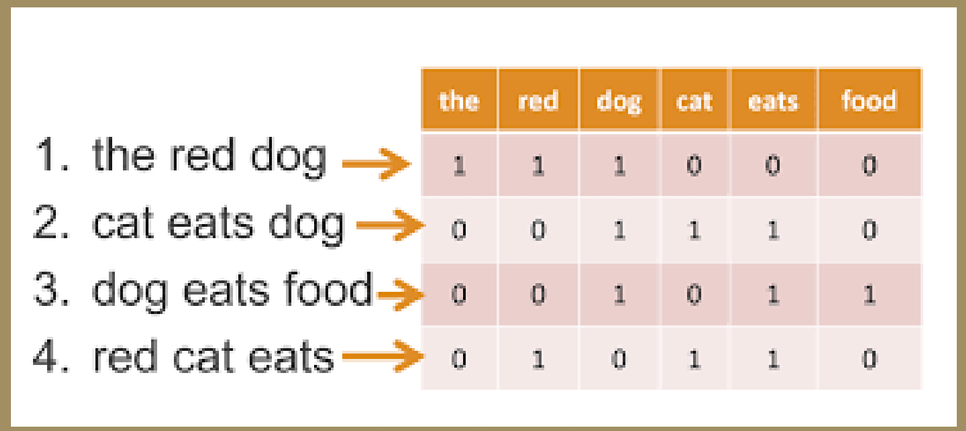
Transformer du texte et des images en données numériques utilisables par des modèles



Extraction de Features Textuelles

MÉTHODES BASIQUES :

- BAG OF WORDS
- TF-IDF



	the	red	dog	cat	eats	food
1. the red dog →	1	1	1	0	0	0
2. cat eats dog →	0	0	1	1	1	0
3. dog eats food →	0	0	1	0	1	1
4. red cat eats →	0	1	0	1	1	0

bag of words

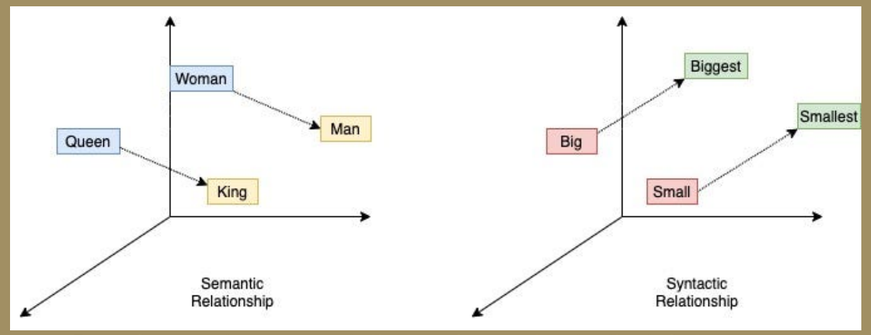
$$\mathrm{tfidf}(t, d, D) = \mathrm{tf}(t, d) \cdot \mathrm{idf}(t, D)$$

term frequency / inverse document frequency

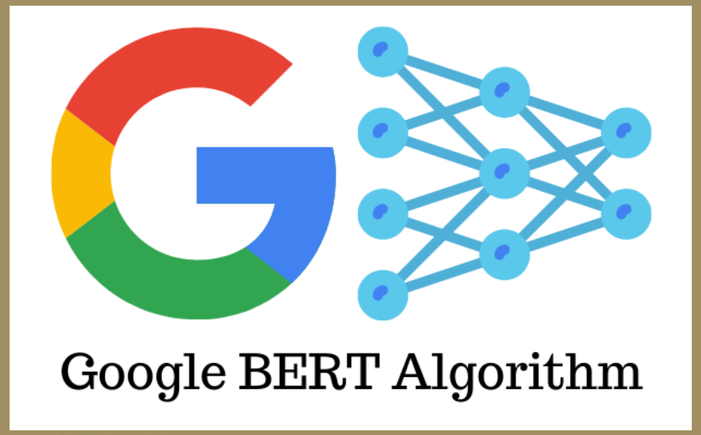
Extraction de Features Textuelles

MÉTHODES AVANCÉES :

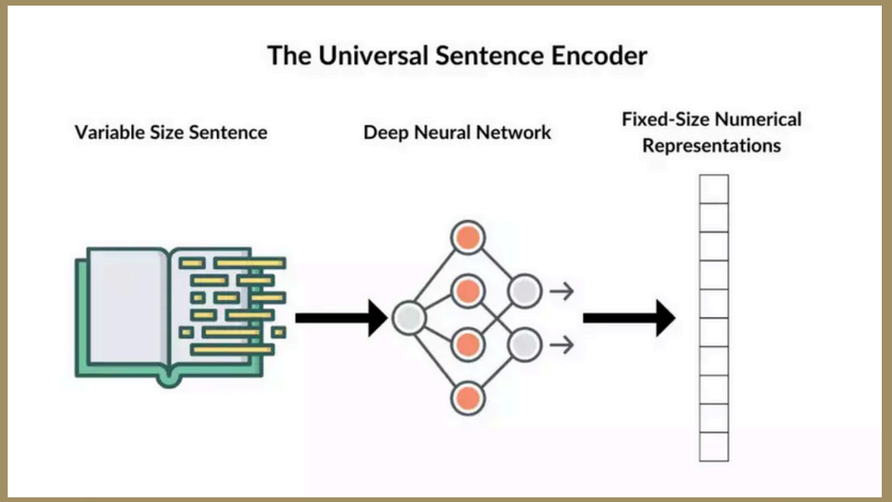
- WORD2VEC
- BERT
- UNIVERSAL SENTENCE ENCODER



Représentation vectorielle basée sur la similarité des mots



BERT : "Encodage contextuel bidirectionnel des phrases"



Encodage universel des phrases pour une compréhension sémantique globale

utilisent les descriptions complètes

Extraction de Features d'Images

ALGORITHME SIMPLE :

- SIFT


```
Descripteurs :  (3526, 128)

[[ 0.  0.  0. ...  0.  1.  5.]
 [ 0.  0.  0. ...  0.  0.  0.]
 [ 0.  0.  0. ...  0.  0.  4.]
 ...
 [ 2.  0.  0. ... 15. 81. 43.]
 [15. 32.  3. ...  0.  0. 40.]
 [22. 49.  0. ...  1. 14. 16.]]
```

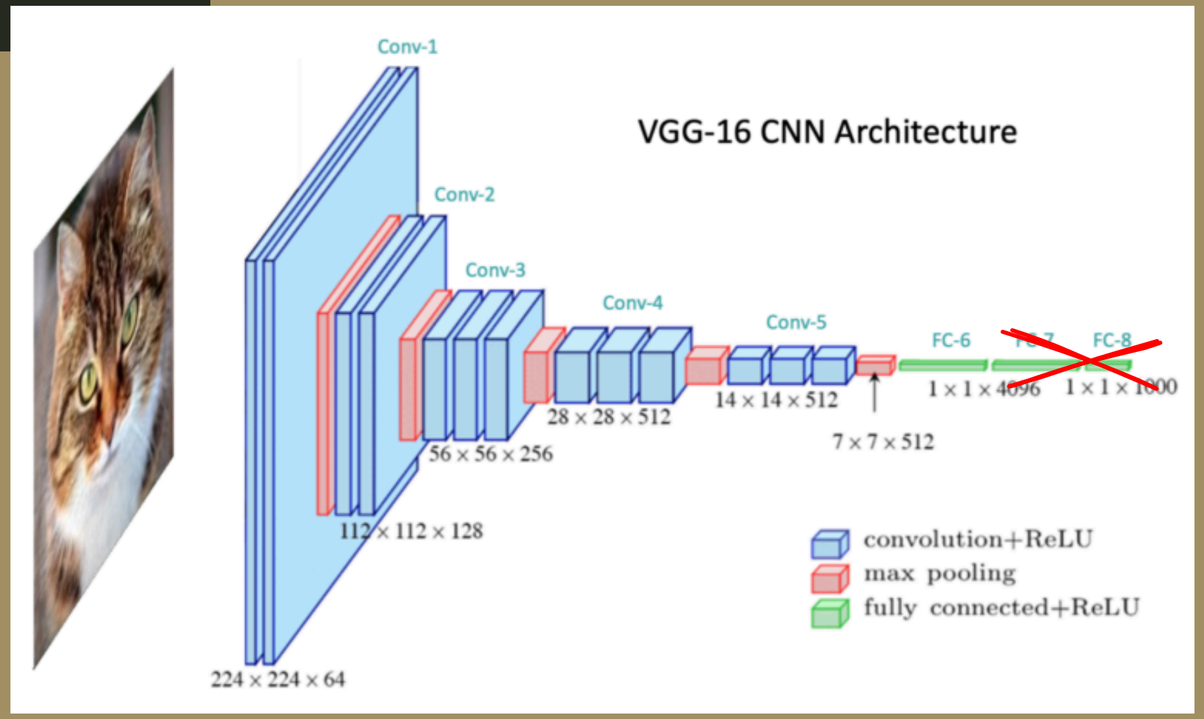
- SIFT (Scale-Invariant Feature Transform) détecte et décrit des caractéristiques locales dans les images.
- Les points d'intérêt SIFT capturent des informations clés comme les bords et les coins.
- Les descripteurs SIFT sont invariants aux transformations d'échelle, de rotation, et partiellement aux variations d'éclairage.
- Utilisé pour la reconnaissance d'objets, la reconstruction 3D, et la navigation robotique.

ALGORITHME AVANCÉ :

- RÉSEAUX DE NEURONES PRÉ-ENTRAÎNÉS (CNN)

Extraction de Features d'Images

- **Chargement du Modèle Pré-entrainé** : Utilisation de VGG16, un modèle puissant pré-entrainé sur le dataset ImageNet.
- **Modification du Modèle** : Création d'un nouveau modèle en omettant la dernière couche de classification pour obtenir des vecteurs de features.
- **Utilisation des Features** : Les vecteurs de features extraits sont utilisés pour le clustering

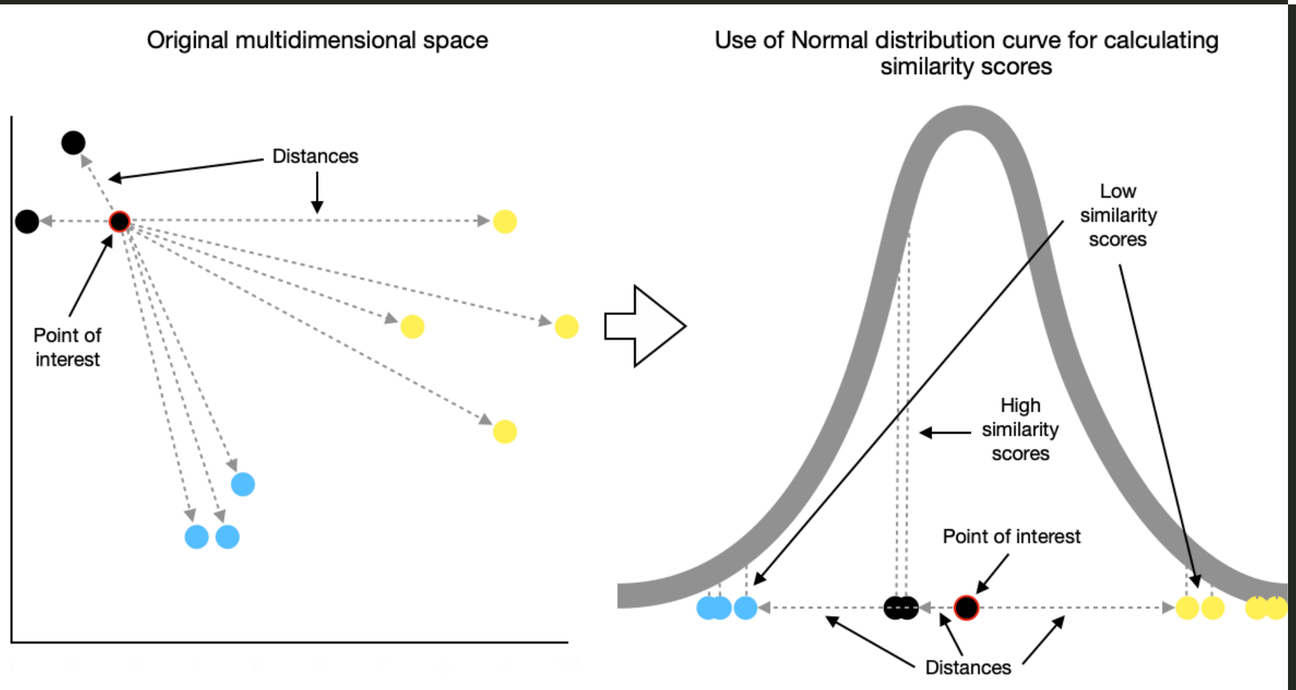


RÉDUCTION DIMENSION 1

PCA

99% de variance conservée

4096 =>803 (ex : image_advanced)

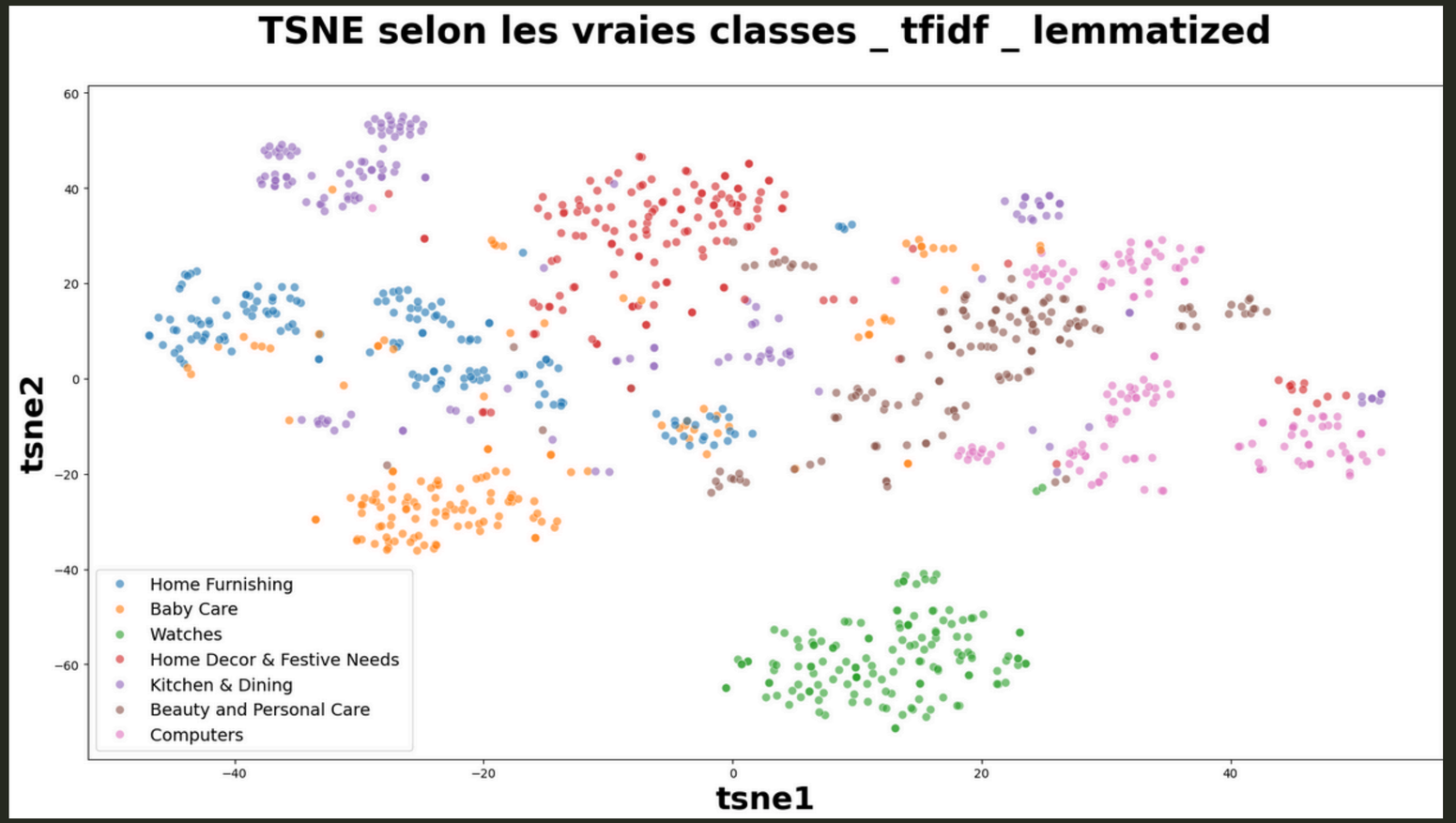


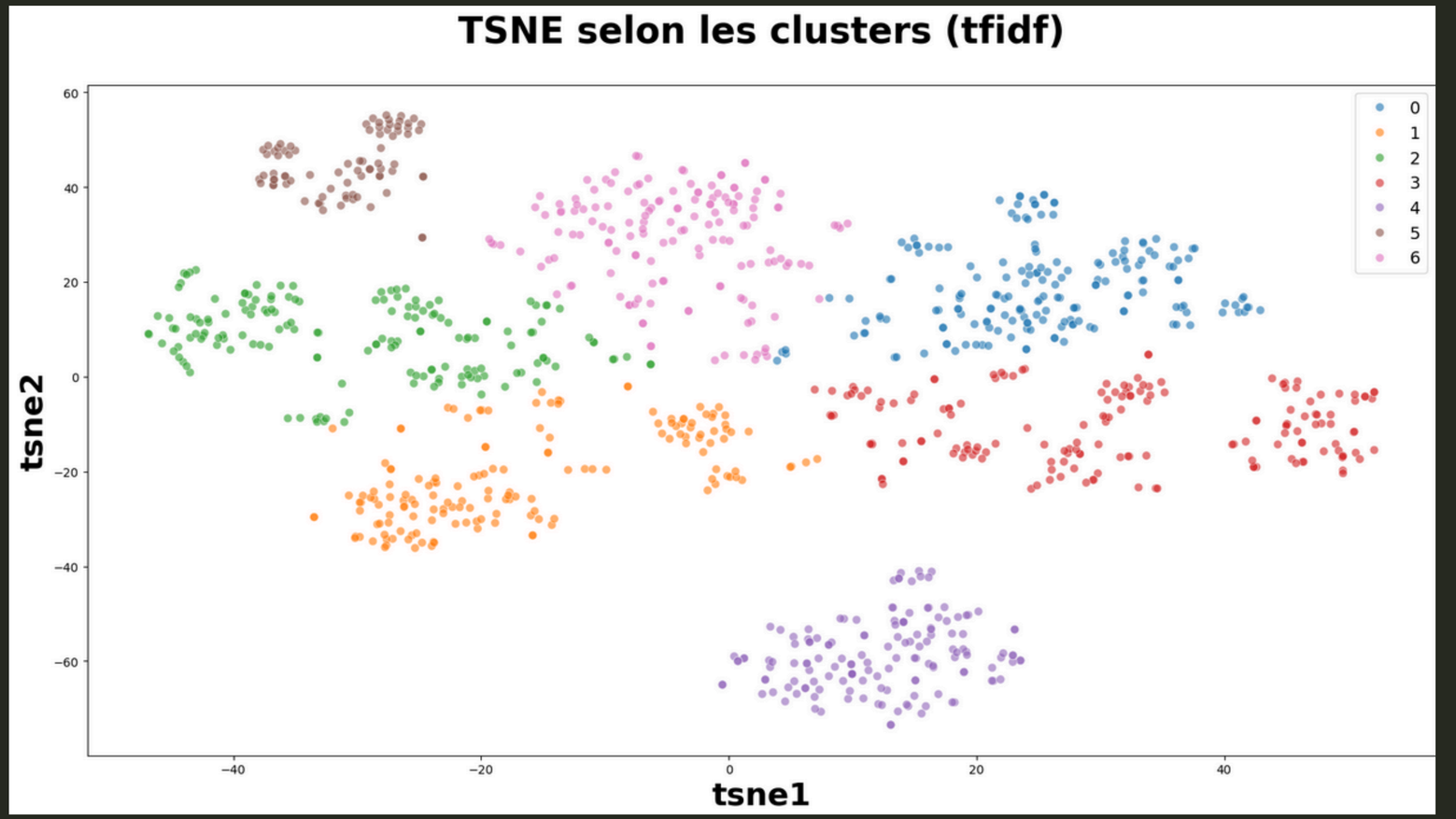
RÉDUCTION DIMENSION 2

tSNE

803=>2

```
temps de T-SNE :                    2.60 secondes
```


TSNE selon les vraies classes _ tfidf _ lemmatized
tsne2
tsne1
60
40
20
0
−20
−40
−60
−40
−20
0
20
40
Home Furnishing
Baby Care
Watches
Home Decor & Festive Needs
Kitchen & Dining
Beauty and Personal Care
Computers


TSNE selon les clusters (tfidf)
tsne2
tsne1
60
40
20
0
−20
−40
−60
−40
−20
0
20
40
0
1
2
3
4
5
6

ARI

Mesure statistique utilisée pour évaluer la similarité entre deux regroupements

- **2 individus dans le même cluster sont ils toujours dans le même cluster ?**
- **2 individus dans deux clusters différents sont ils toujours dans deux clusters différents**

	Prédit Positif	Prédit Négatif
Réel Positif	VP	FN
Réel Négatif	FP	VN

Exactitude (Accuracy) : $\frac{VP+VN}{VP+FP+VN+FN}$

Précision (Precision) : $\frac{VP}{VP+FP}$

Rappel (Recall) ou Sensibilité : $\frac{VP}{VP+FN}$

Score F1 : $2 \times \frac{\text{Précision} \times \text{Rappel}}{\text{Précision} + \text{Rappel}}$

```
Correspondance des clusters :  [4 5 3 0 2 6 1]
[[109   3   1  26   9   1   1]
 [  3 117   9  14   5   1   1]
 [  1   1 113  34   0   0   1]
 [  2   1   5  79  55   0   8]
 [ 71   0   0   9  70   0   0]
 [  0   9  33  29   0  79   0]
 [  0   0  13   2   0   0 135]]

              precision    recall  f1-score   support

           0       0.59      0.73      0.65       150
           1       0.89      0.78      0.83       150
           2       0.65      0.75      0.70       150
           3       0.41      0.53      0.46       150
           4       0.50      0.47      0.48       150
           5       0.98      0.53      0.68       150
           6       0.92      0.90      0.91       150

    accuracy                           0.67      1050
   macro avg       0.71      0.67      0.67      1050
weighted avg       0.71      0.67      0.67      1050
```

exemple VGG16

Matrice de confusion

methode	ARI	ACCURACY
BOW	0.40	0.67
TF-IDF	0.48	0.71
W2V	0.19	0.45
BERT	0.30	0.53
USE	0.43	0.64
SIFT	0.04	0.28
VGG16	0.45	0.67

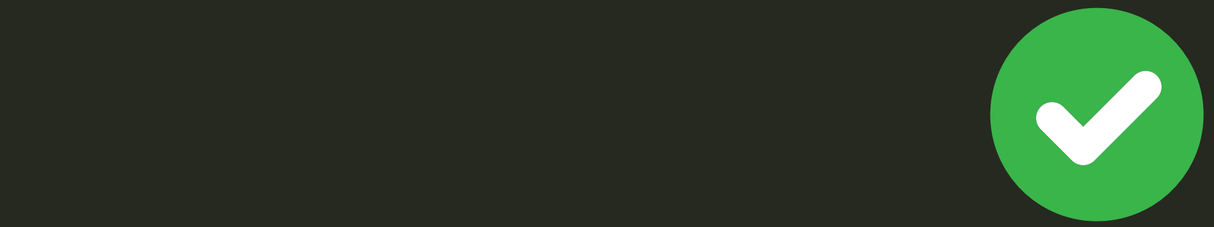
Résultats

FAISABILITÉ

Les résultats de l'étude de faisabilité montrent des scores ARI et d'Accuracy variables selon les techniques utilisées pour la classification d'objets à partir de descriptions et d'images.

Les méthodes textuelles affichent des performances robustes, suggérant une faisabilité prometteuse pour la classification.

Les autres résultats, plus faibles, indiquent que des méthodes plus sophistiquées ou combinées pourraient être nécessaires pour améliorer la précision.



Classification supervisée

PRINCIPE

Ensemble d'entraînement (train dataset) :

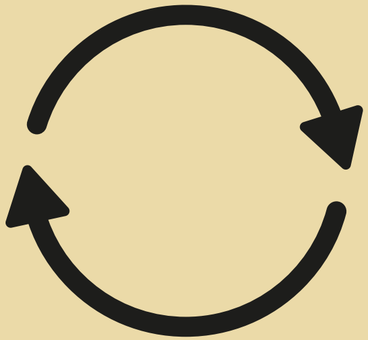
Objectif : Utilisé pour entraîner le modèle.
Contenu : Contient la majorité des données disponibles.
Fonctionnement : Le modèle apprend à partir de ces données en ajustant ses poids pour minimiser la fonction de perte.

Ensemble de validation (validation dataset) :

Objectif : Utilisé pour évaluer les performances du modèle pendant l'entraînement.
Contenu : Séparé de l'ensemble d'entraînement, mais aussi dérivé des données disponibles.
Fonctionnement : Permet de surveiller les performances du modèle à chaque époque pour ajuster les hyperparamètres et éviter le surapprentissage (overfitting). Les callbacks comme EarlyStopping et ModelCheckpoint utilisent cet ensemble pour leurs critères de surveillance.

Ensemble de test (test dataset) :

Objectif : Utilisé pour évaluer les performances finales du modèle.
Contenu : Complètement séparé des ensembles d'entraînement et de validation.
Fonctionnement : Fournit une évaluation impartiale des performances du modèle après l'entraînement, pour s'assurer qu'il généralise bien sur des données inédites.



Classification supervisée

1er modèle

Surveillance de la perte de validation (val_loss) :

Surveille la perte de validation pendant l'entraînement.

Utilise cette métrique pour évaluer les performances du modèle.

Callback ModelCheckpoint :

Chemin de sauvegarde : "./model1_best_weights.keras"

Critère de surveillance : 'val_loss'

Affichage des messages : verbose=1 (messages affichés lors de la sauvegarde des poids)

Sauvegarde conditionnelle : save_best_only=True (sauvegarde les poids uniquement si la perte de validation s'améliore)

Mode de surveillance : mode='min' (cherche à minimiser la perte de validation)

Callback EarlyStopping :

Critère de surveillance : 'val_loss'

Mode de surveillance : mode='min' (cherche à minimiser la perte de validation)

Affichage des messages : verbose=1 (messages affichés lorsque l'entraînement s'arrête)

Patience : patience=5 (arrête l'entraînement si la perte de validation ne s'améliore pas pendant 5 epochs)

Configuration de TensorFlow :

Utilisation du CPU uniquement : tf.config.set_visible_devices([], 'GPU') (désactive l'utilisation du GPU)

Ces éléments garantissent que le modèle s'entraîne efficacement, en sauvegardant les meilleurs poids et en arrêtant l'entraînement lorsque les améliorations cessent, tout en étant configuré pour utiliser le CPU.

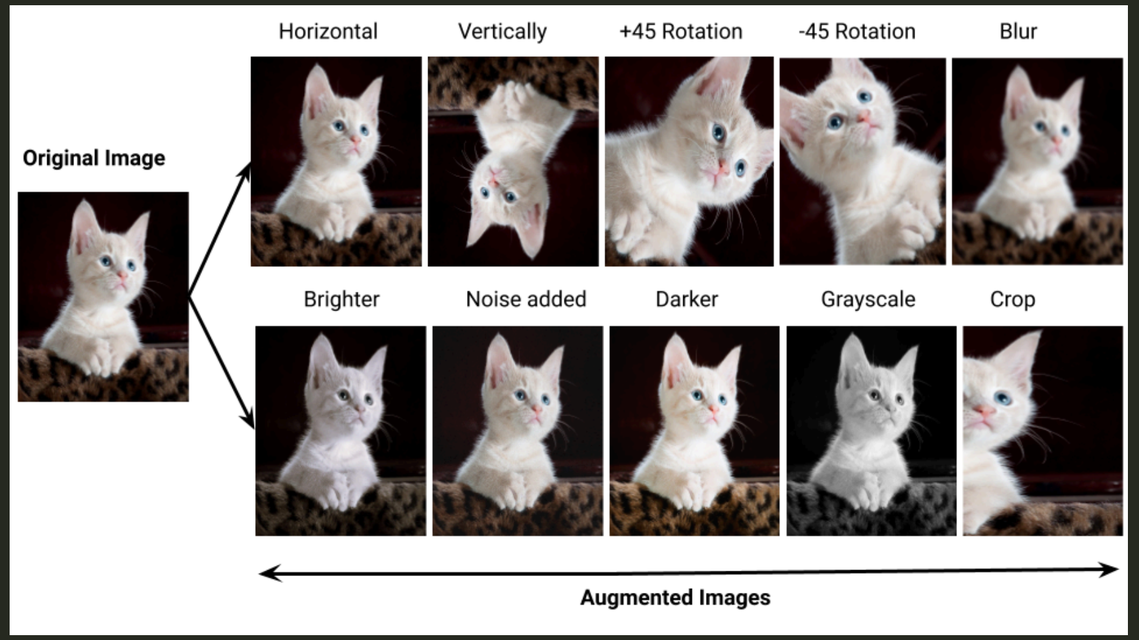
Classification supervisée

2nd modèle - Data augmentation

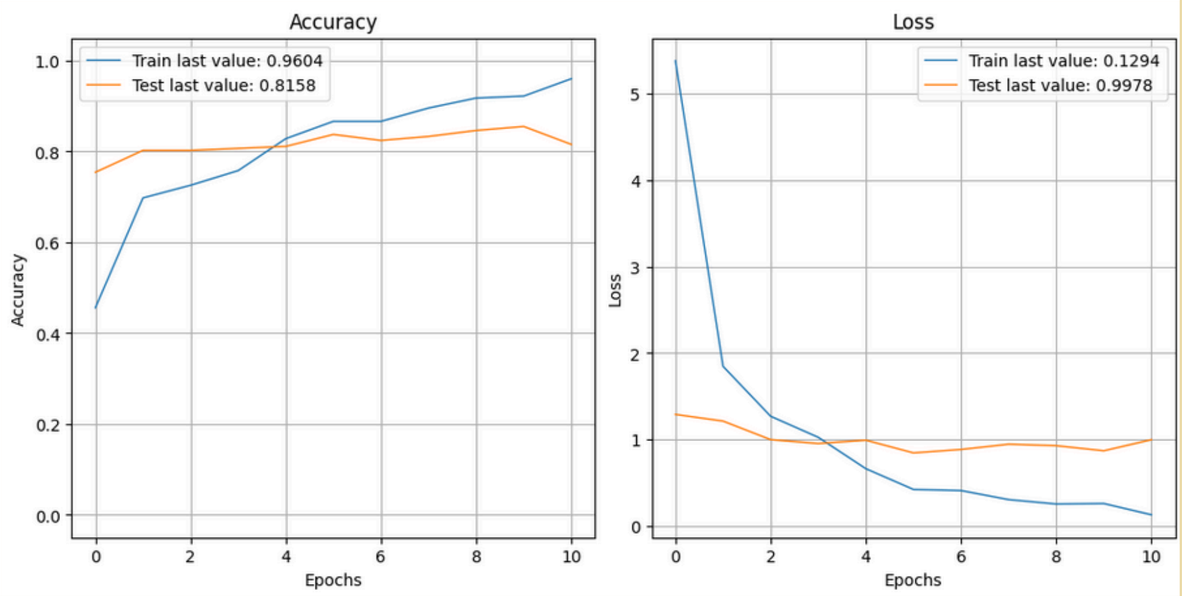
modèle identique +

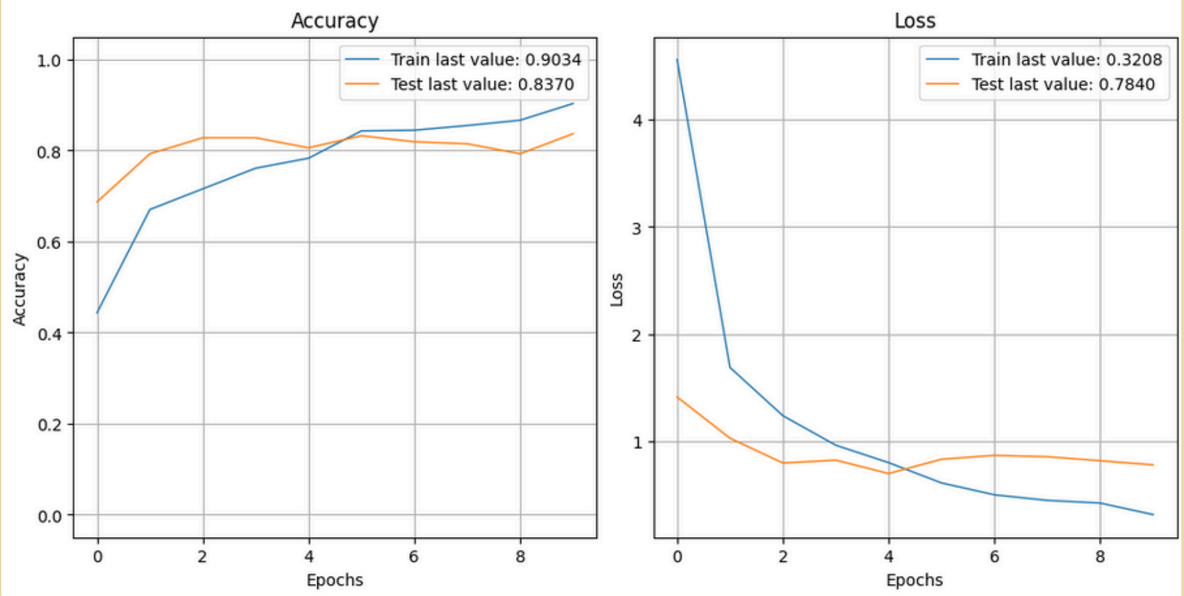
rotation_range=20,
width_shift_range=0.2,
height_shift_range=0.2,
horizontal_flip=True,

On transforme les images du dataset train mais pas du dataset test



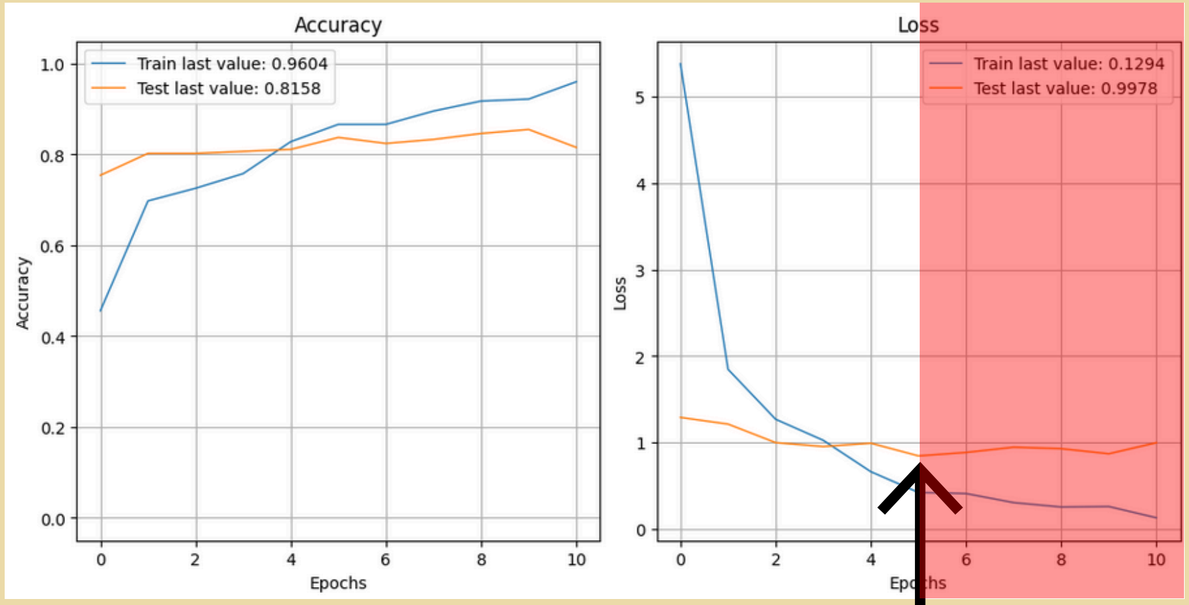
Résultats


Accuracy
Train last value: 0.9604
Test last value: 0.8158
Accuracy
Epochs
Loss
Train last value: 0.1294
Test last value: 0.9978
Loss
Epochs


Accuracy
Train last value: 0.9034
Test last value: 0.8370
Accuracy
Epochs
Loss
Train last value: 0.3208
Test last value: 0.7840
Loss
Epochs

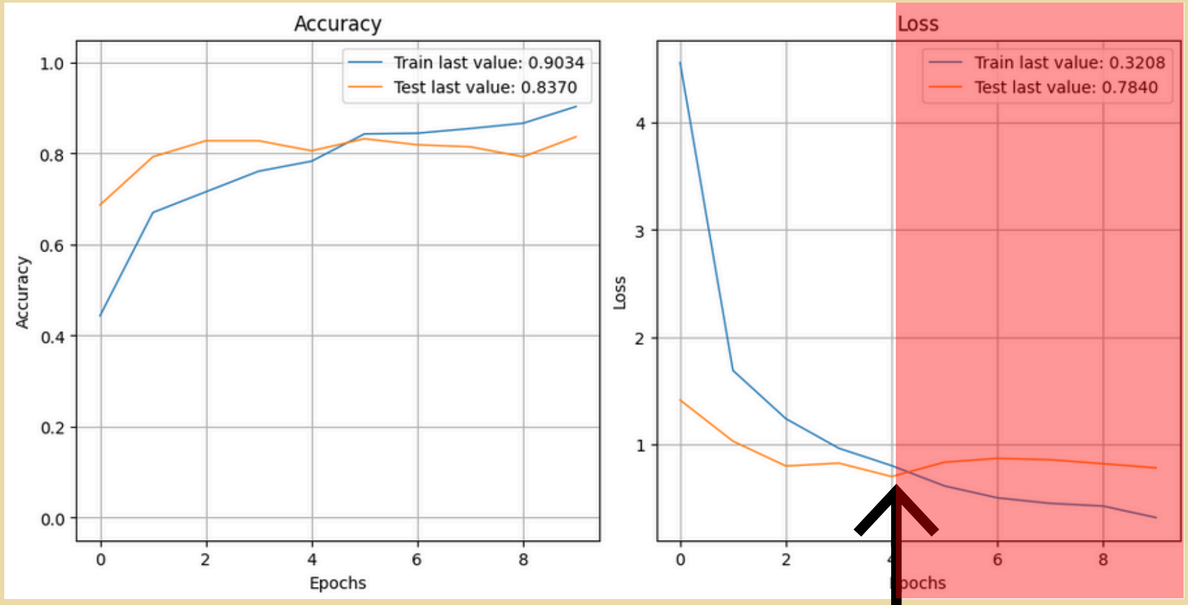
Résultats

modèle 1



es

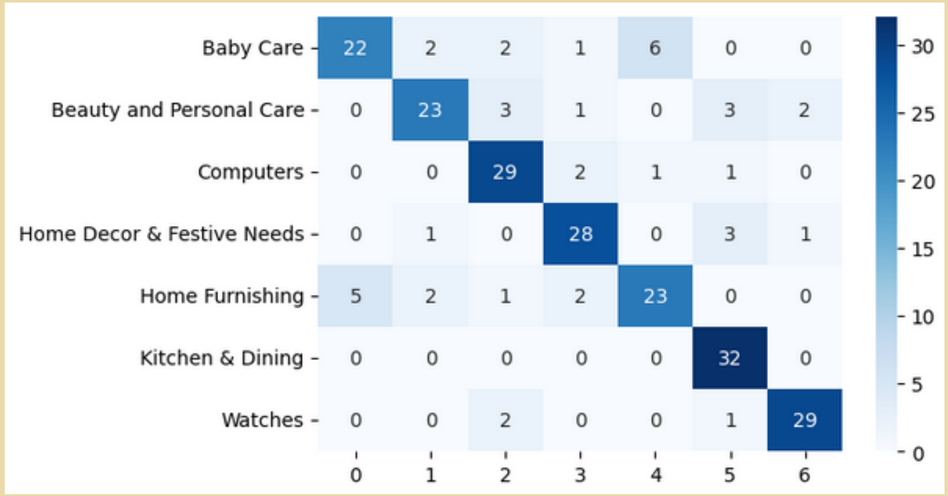
modèle 2 - data augmentation



es

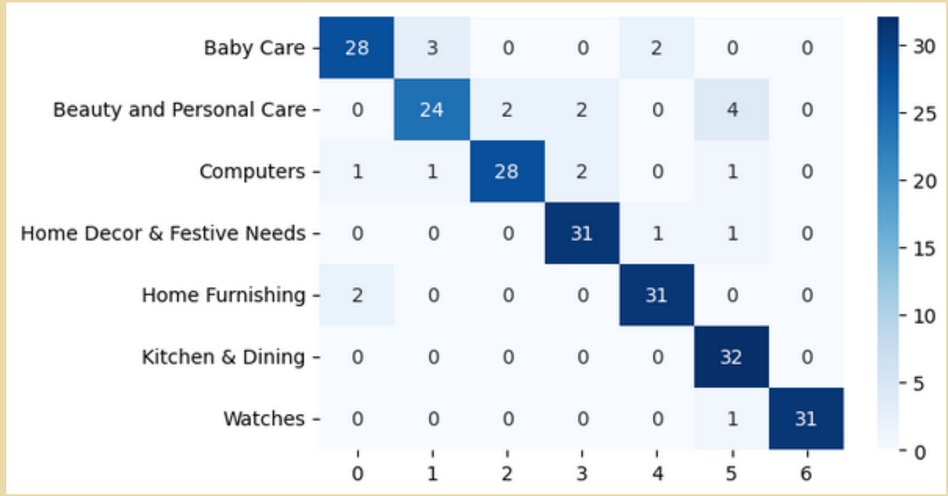
Résultats

modèle 1



	precision	recall	f1-score	support
0	0.81	0.67	0.73	33
1	0.82	0.72	0.77	32
2	0.78	0.88	0.83	33
3	0.82	0.85	0.84	33
4	0.77	0.70	0.73	33
5	0.80	1.00	0.89	32
6	0.91	0.91	0.91	32
accuracy			0.82	228
macro avg	0.82	0.82	0.81	228
weighted avg	0.82	0.82	0.81	228

modèle 2 - data augmentation



	precision	recall	f1-score	support
0	0.90	0.85	0.88	33
1	0.86	0.75	0.80	32
2	0.93	0.85	0.89	33
3	0.89	0.94	0.91	33
4	0.91	0.94	0.93	33
5	0.82	1.00	0.90	32
6	1.00	0.97	0.98	32
accuracy			0.90	228
macro avg	0.90	0.90	0.90	228
weighted avg	0.90	0.90	0.90	228

Ce script permet d'interroger l'API Edamam pour obtenir des informations sur des produits alimentaires spécifiques et de sauvegarder ces informations dans un fichier CSV.

API

- **Définition des informations d'authentification et de l'URL de l'API :**
  - api_key : Clé API pour authentification.
  - api_url : URL de l'API Edamam Food Database.
- **Définition des paramètres de la requête :**
  - ingr : Ingrédient recherché, ici 'champagne'.
- **Définition des headers pour l'API RapidAPI :**
  - X-RapidAPI-Key : Utilise la clé API.
  - X-RapidAPI-Host : Spécifie l'hôte de l'API.
- **Requête à l'API :**
  - Utilise requests.get pour envoyer la requête avec les headers et les paramètres définis.
- **Vérification du statut de la réponse :**
  - Si response.status_code == 200 (succès) :
    - Convertit la réponse JSON en dictionnaire Python.
    - Extrait les 10 premiers produits de la réponse.
- **Écriture des données dans un fichier CSV :**
  - Ouvre un fichier champagne_products.csv en mode écriture.
  - Écrit les en-têtes des colonnes : foodId, label, category, foodContentsLabel, image.
  - Pour chaque produit extrait :
    - Récupère les informations : foodId, label, category, foodContentsLabel, image.
    - Écrit les données dans le fichier CSV.
- **Gestion des erreurs :**
  - Si la requête échoue, affiche un message d'erreur avec le code de statut HTTP.
- **Message de confirmation :**
  - Indique que les données ont été sauvegardées dans le fichier CSV.

merci