Classifier automatiquement des biens de consommation

Présentation de la mission et contexte

Contexte :

“Place de marché” est une future marketplace e-commerce anglophone où des vendeurs proposent des articles via une photo et une description.

Actuellement, l’attribution des catégories des articles est manuelle, peu fiable et difficile à scaler avec l’augmentation du volume.

Mission :

- Étudier la faisabilité d’un moteur de classification automatique d’articles basé sur le texte et l’image.
- Réaliser une classification supervisée d’images pour vérifier la capacité de distinguer automatiquement des catégories de produits.
- Enrichir les données via une collecte automatisée de produits complémentaires (ex : produits à base de champagne).

Objectifs :

- Faciliter la mise en ligne des produits pour les vendeurs.
- Améliorer la recherche et la navigation pour les acheteurs.
- Préparer la montée en charge de la plateforme.

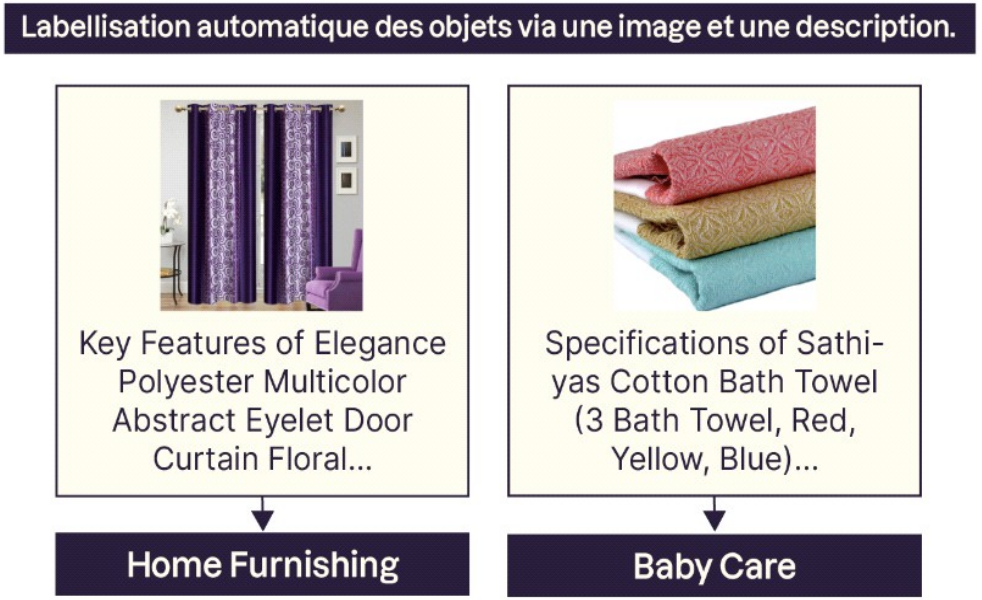
Plan de la présentation

- Présentation des données
- Étude de faisabilité pour la classification basée sur le texte
- Étude de faisabilité pour la classification basée sur les images
- Mise en œuvre d'une classification supervisée des images
- Collecte automatisée de nouveaux produits via l’API Open Food Facts
- Conclusion et perspectives

Présentation des données

Variables principales

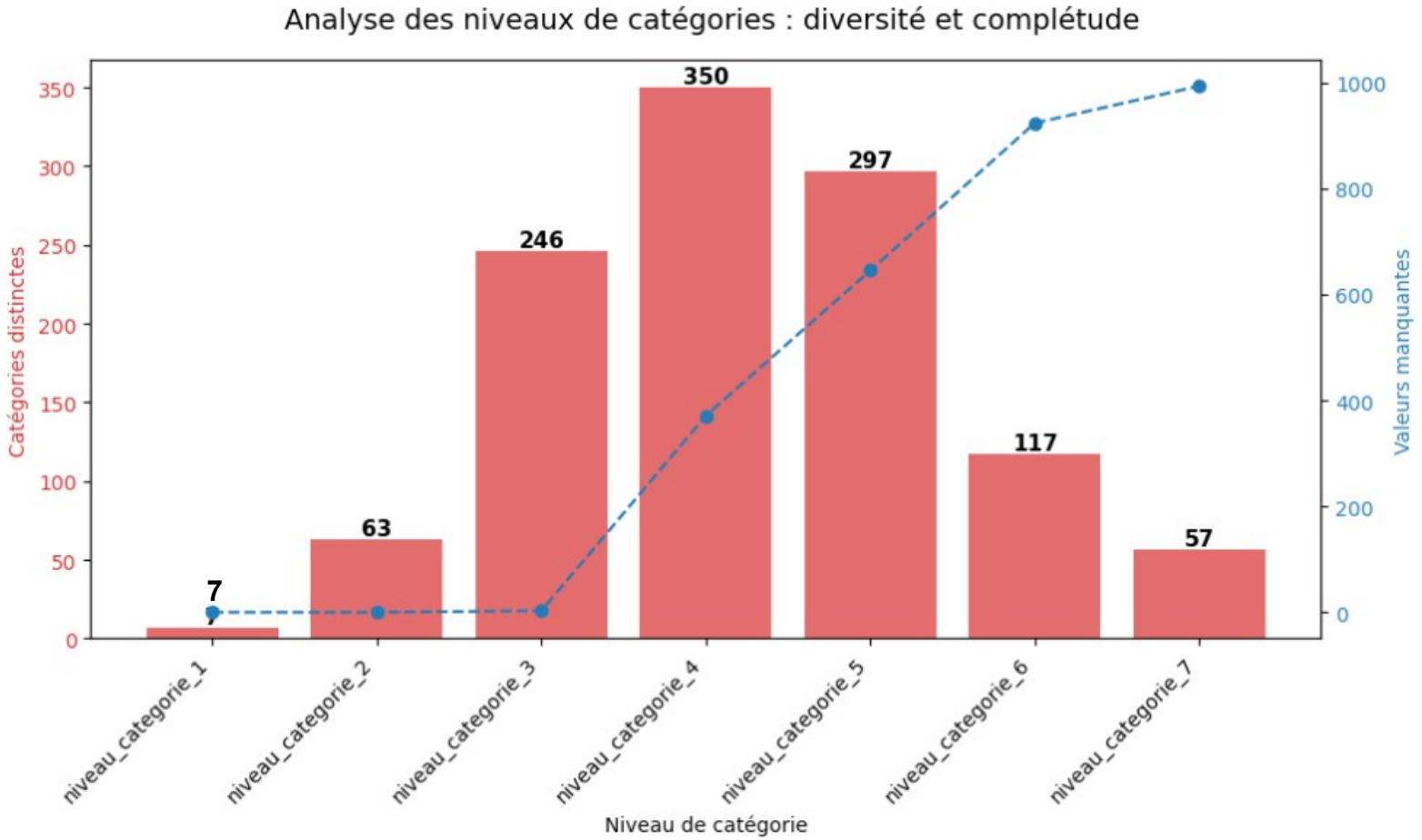
- product_name : nom du produit
- description : description textuelle du produit
- product_category_tree : catégorie attribuée (hiérarchique)
  (ex. : Home >> Linge >> Draps...)
- image : fichier image au format JPEG



Contenu du fichier

- 1050 produits issus du site Flipkart
- 15 colonnes décrivant chaque produit
- Textes en anglais
- Plusieurs niveaux de catégories
- Pas de doublons

Présentation des données : niveaux de catégories



- Le dataset contient 7 niveaux de catégories hiérarchiques.
- Le nombre de catégories distinctes augmente jusqu'au niveau 4, puis diminue.
- Les valeurs manquantes sont faibles aux premiers niveaux mais augmentent fortement à partir du niveau 4.

```
Répartition des produits par niveau_categorie_1 :
niveau_categorie_1
Home Furnishing                150
Baby Care                      150
Watches                        150
Home Decor & Festive Needs     150
Kitchen & Dining               150
Beauty and Personal Care       150
Computers                      150
```

=> Sélection du premier niveau

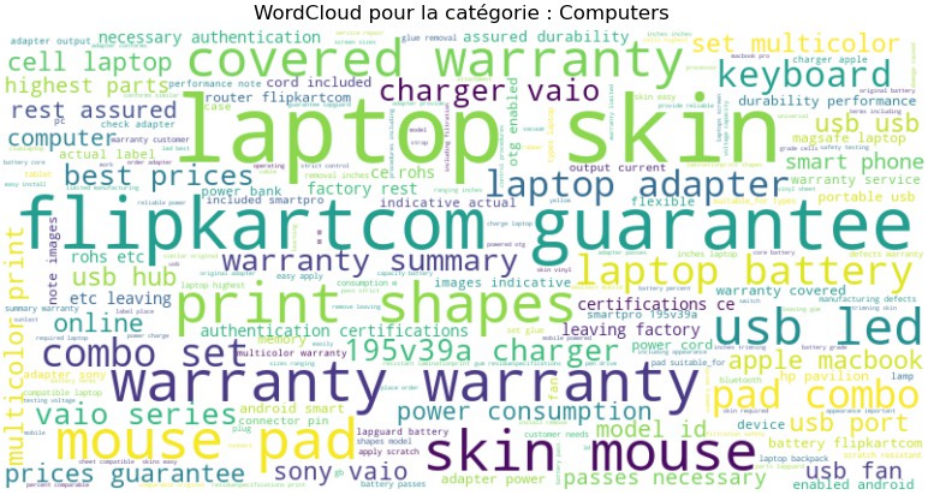
Étude de faisabilité pour la classification basée sur le texte

Prétraitement des données textuelles

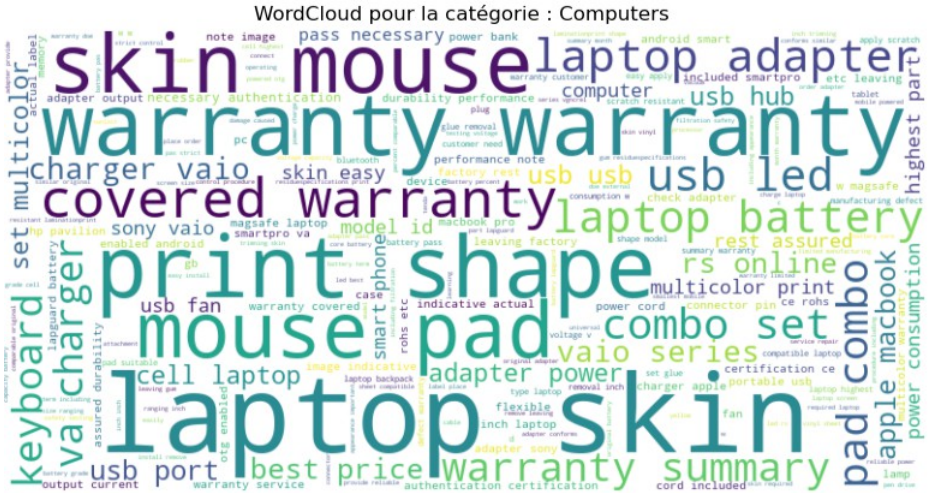
Objectif : Nettoyer et uniformiser les descriptions produits pour améliorer la qualité des représentations textuelles.

Transformations réalisées :

- Conversion en minuscules : uniformisation des textes
- Suppression des stopwords : mots fréquents sans valeur informative
- Nettoyage personnalisé : retrait de mots trop courants mais spécifiques au contexte (e.g., “product”, “buy”)
- Lemmatisation : réduction des mots à leur forme de base
- Suppression de la ponctuation et des chiffres : élimination du bruit inutile



Avant nettoyage



Après nettoyage

Méthodes d’encodage de texte pour la classification

Méthodes utilisées :

1) **Bag-of-Words (BoW)**

- Une approche simple pour convertir le texte en vecteurs de fréquence de mots.

2) **TF-IDF (Term Frequency - Inverse Document Frequency)**

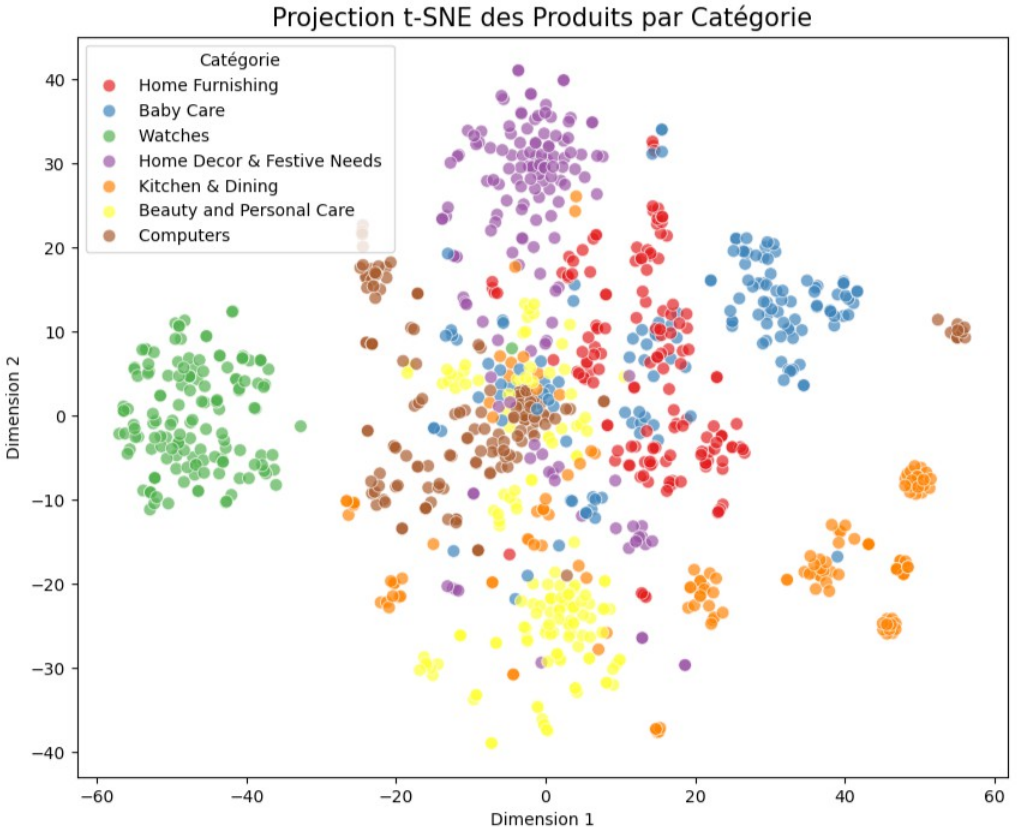
- Une méthode qui ajuste les fréquences de mots en fonction de leur importance dans l'ensemble du dataset.

3) Embeddings de texte :

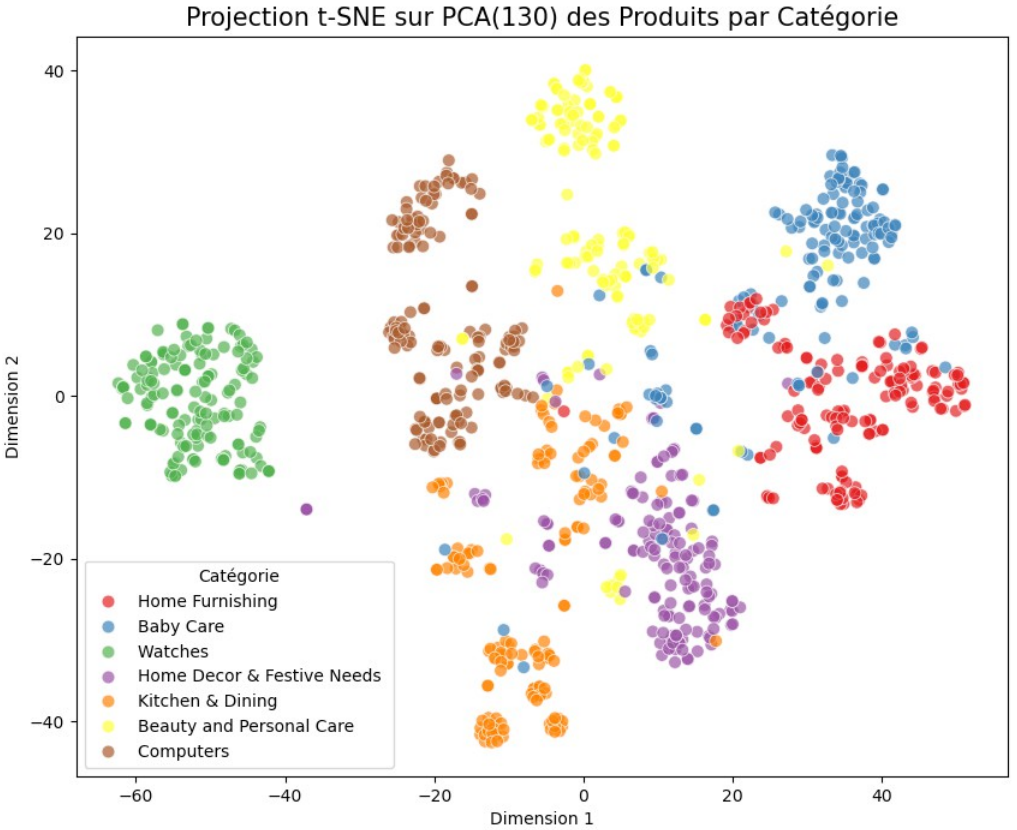
- **Word2Vec** : Un modèle qui représente les mots sous forme de vecteurs, en tenant compte de leur contexte.
- **BERT** : Un modèle pré-entraîné plus complexe qui comprend mieux le contexte de chaque mot dans une phrase.
- **USE (Universal Sentence Encoder)** : Un autre modèle pour encoder des phrases complètes en vecteurs.

Comparaison des méthodes d'encodage texte

Métriques	BoW	TF-IDF	Word2Vec	BERT	USE	USE + ACP
Homogeneity Score	0.324	0.503	0.469	0.372	0.577	0.690
Completeness Score	0.609	0.589	0.519	0.39	0.618	0.713
V-measure	0.423	0.543	0.493	0.381	0.597	0.701
Adjusted Rand Index	0.149	0.29	0.313	0.263	0.457	0.623
Silhouette Score	0.089	0.049	0.372	0.111	0.093	0.102



BoW



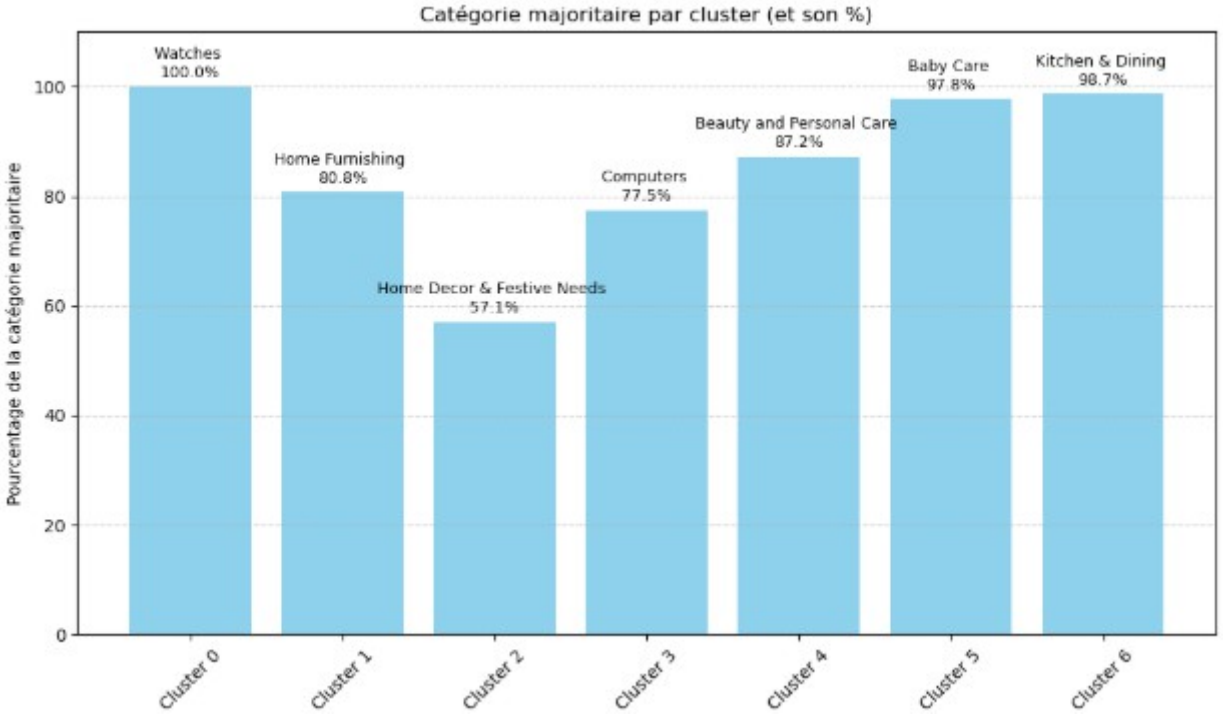
USE + ACP

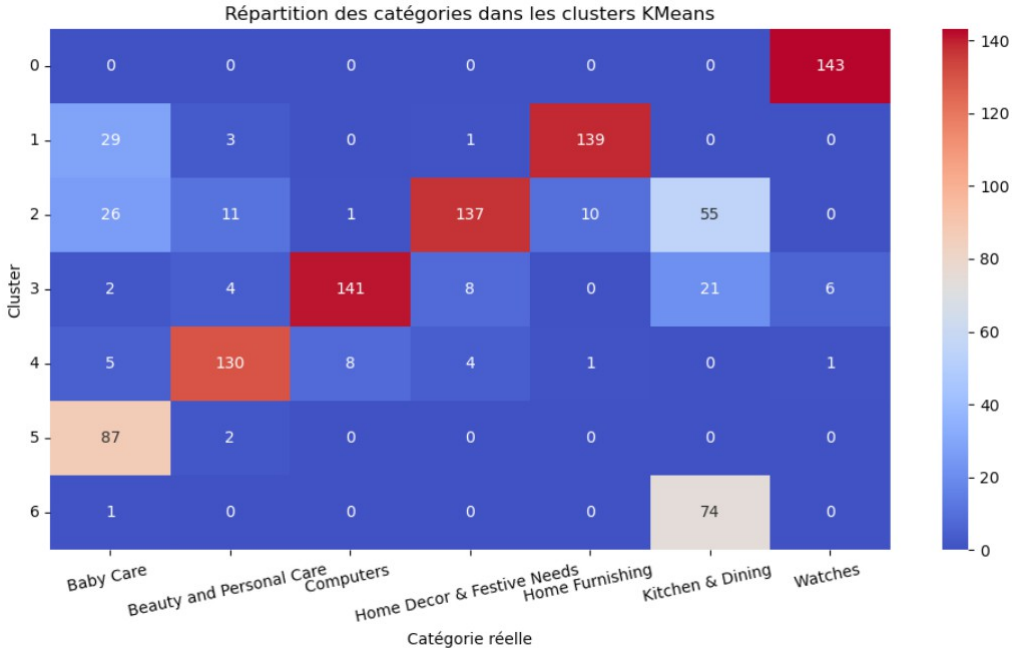
Zoom sur la meilleure méthode d'encodage (USE + PCA)

Optimisation du clustering : meilleurs réglages

- Réduction du nombre de features avec PCA à 130 composantes principales
- K-Means : 7 clusters (un par catégorie), initialisation k-means++, 10 essais, 50 itérations max

```
Homogeneity Score: 0.690
Completeness Score: 0.713
V-Measure: 0.701
Adjusted Rand Index: 0.623
Silhouette Score: 0.103
```





Clusters bien alignés avec certaines catégories :

- Watches : 143/150 dans un seul cluster
- Computers : 141/150 dans un cluster
- Home Furnishing : 139/150 dans un cluster
- Home Decor & Festive Needs : 137/150 dans un cluster

Catégories plus diffuses :

- Beauty and Personal Care et Baby Care sont réparties sur plusieurs clusters
- Kitchen & Dining est aussi dispersée

Étude de faisabilité pour la classification basée sur les images

Méthodes d'encodage d'images pour la classification

Méthodes classiques (SIFT + Bag of Visual Words)

- Préparation des images : passage en niveaux de gris, normalisation de l'image, redimensionnement, réduction du bruit et ajustement du contraste/luminosité .
- Détection des zones visuellement importantes.
- Construction d'un "vocabulaire visuel" pour représenter les images.
- Peu sensible à la position ou l'orientation des objets.

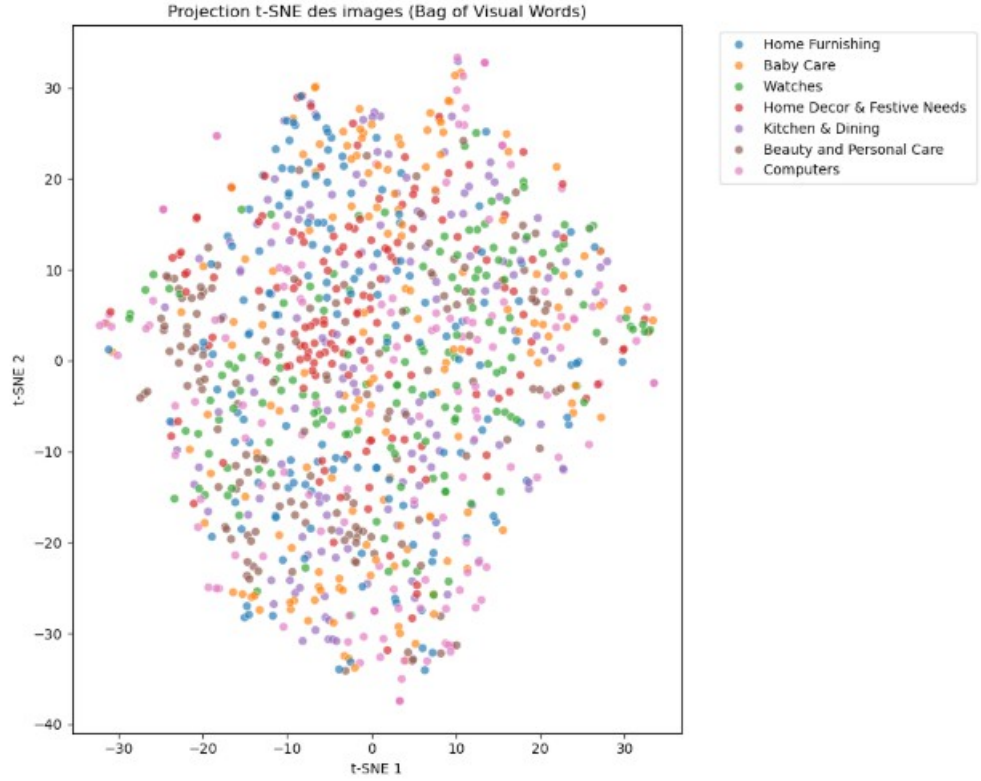
Deep learning (modèles pré-entraînés)

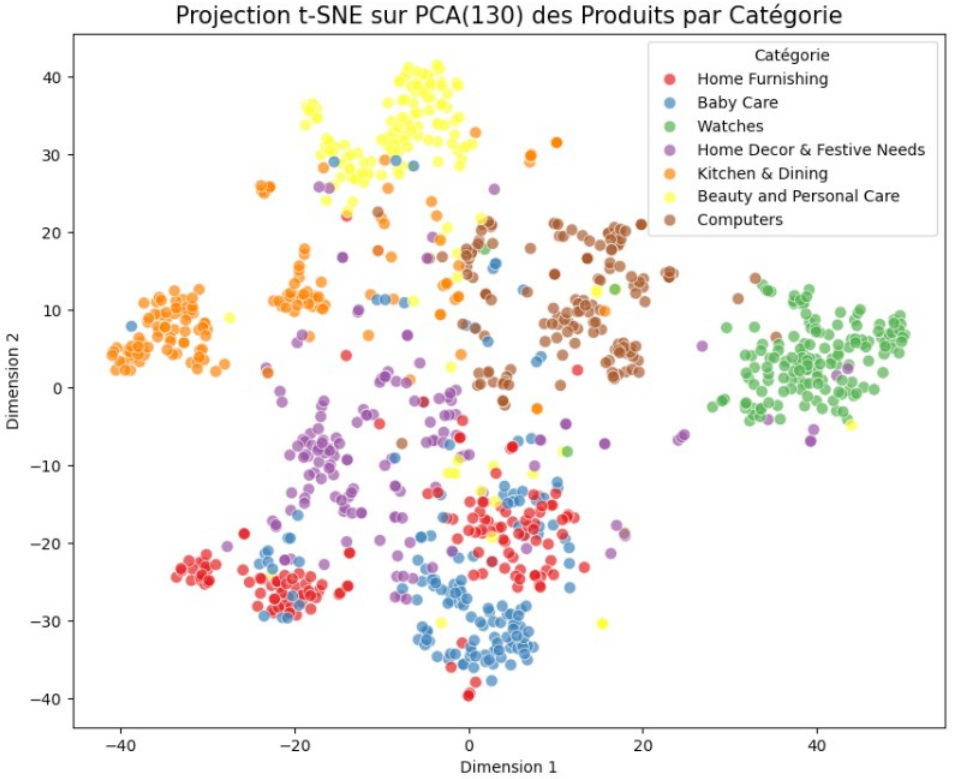
- Nécessitent un redimensionnement et une normalisation spécifiques.
- Transforment automatiquement chaque image en vecteur de caractéristiques.
- VGG16 : simple et efficace.
- ResNet50 : plus profond, capte des détails complexes.
- EfficientNetB0 : compact, efficace sur petits jeux de données.

Ces représentations sont ensuite utilisées pour regrouper automatiquement les images (clustering) et observer la séparation entre catégories.

Comparaison des méthodes d’encodage d'images

Méthode	Homogeneity	Completeness	V-measure	Adjusted Rand Index	Silhouette Score
SIFT + BoVW	0.041	0.047	0.044	0.011	0.042
VGG16 (features)	0.496	0.518	0.507	0.417	0.053
ResNet50 (features)	0.552	0.581	0.566	0.455	0.029
EfficientNetB0 (features)	0.633	0.645	0.639	0.585	0.067
EfficientNetB0 (features) opti	0.635	0.648	0.641	0.589	0.110



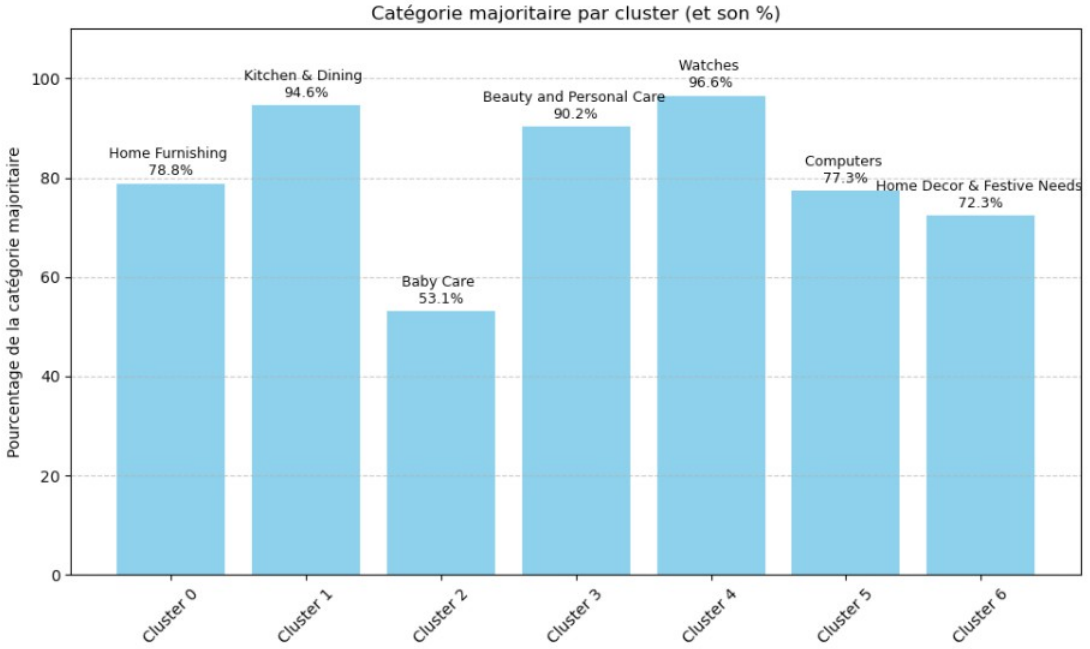


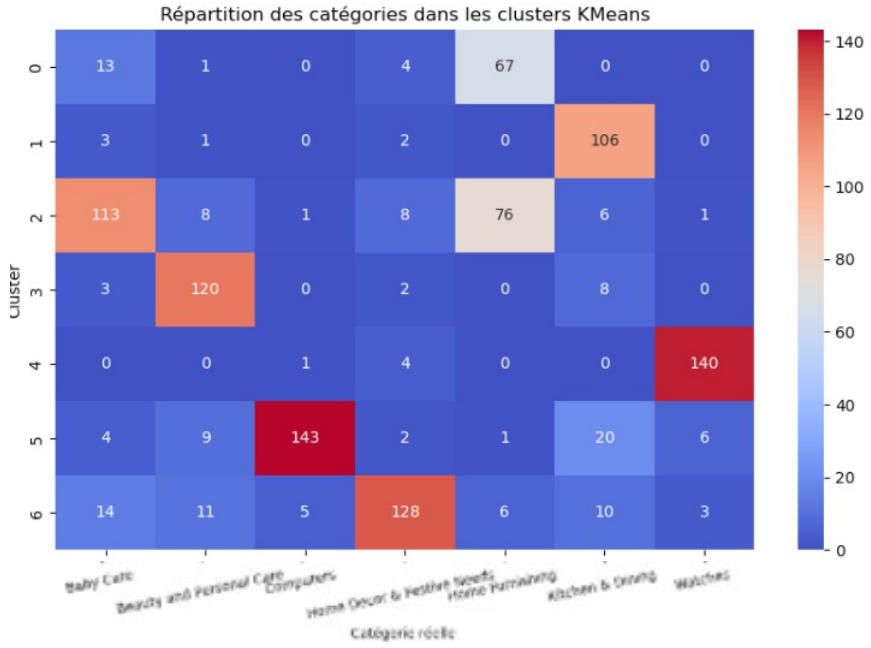
Zoom sur la meilleure méthode d'encodage EfficientNetB0

Optimisation du clustering : meilleurs réglages

- Réduction du nombre de features avec PCA à 100 composantes principales
- K-Means : 7 clusters (un par catégorie), initialisation k-means++, 10 essais, 50 itérations max

```
Homogeneity Score: 0.635
Completeness Score: 0.648
V-Measure: 0.642
Adjusted Rand Index: 0.589
Silhouette Score: 0.110
```





Clusters bien alignés avec certaines catégories :

- Watches : 140/150 dans un seul cluster
- Computers : 143/150 dans un cluster
- Beauty and Personal Care : 120/150 dans un cluster
- Home Decor & Festive Needs : 128/150 dans un cluster

Catégories plus diffuses :

- Home furnishing et Baby Care sont réparties sur plusieurs clusters
- Kitchen & Dining est aussi dispersée

Images mal classées : exemples

Cat. réelle: Baby Care
Cluster: 6
Cat. majoritaire: Home Decor & Festive Needs

Cat. réelle: Home Furnishing
Cluster: 2
Cat. majoritaire: Baby Care

Cat. réelle: Kitchen & Dining
Cluster: 5
Cat. majoritaire: Computers

Cat. réelle: Home Furnishing
Cluster: 2
Cat. majoritaire: Baby Care

Cat. réelle: Home Furnishing
Cluster: 2
Cat. majoritaire: Baby Care

Cat. réelle: Computers
Cluster: 6
Cat. majoritaire: Home Decor & Festive Needs

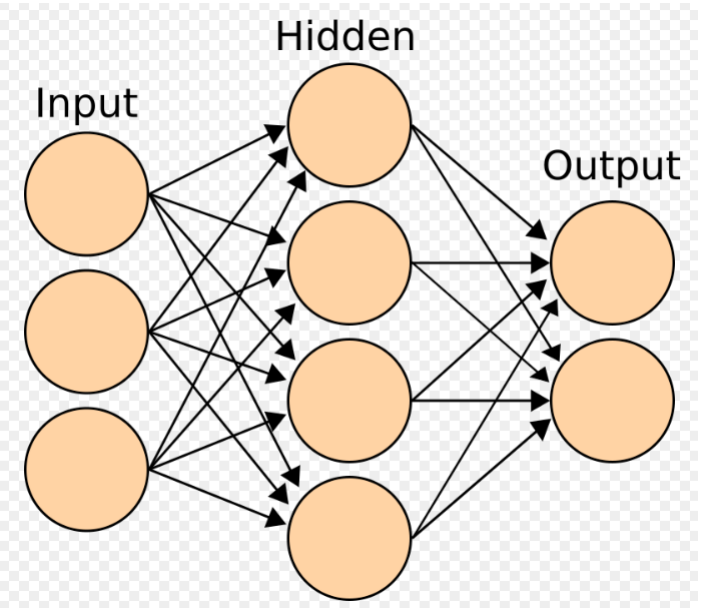
Cat. réelle: Home Decor & Festive Needs
Cluster: 2
Cat. majoritaire: Baby Care

Cat. réelle: Beauty and Personal Care
Cluster: 2
Cat. majoritaire: Baby Care

Mise en œuvre d'une classification supervisée des images

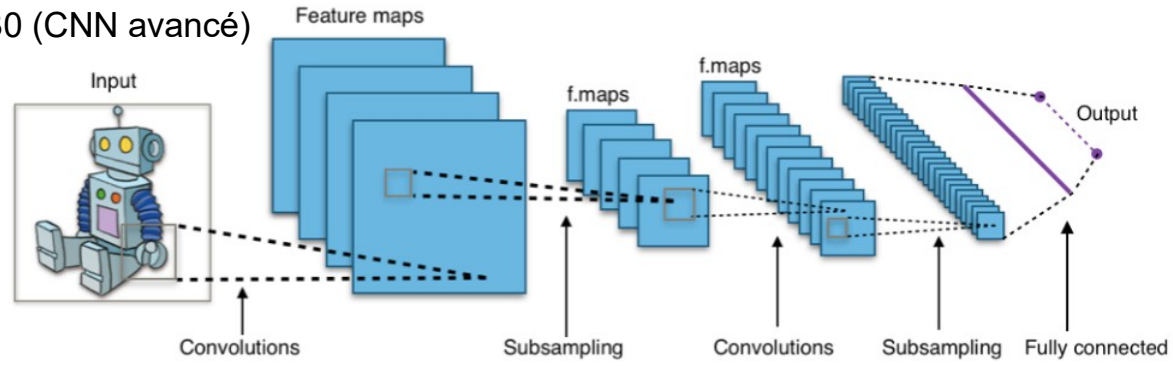
Présentation rapide des modèles testés

Modèle	Type	Pourquoi ce choix ?	Limites principales
MLP minimal	Perceptron multicouche	• Simple à comprendre et à entraîner • Sert de référence (baseline)	• Performances limitées sur des images complexes • Ne capture pas bien les structures spatiales
CNN simple	Réseau convolutif	• Capte la structure spatiale des images • Apprend des motifs visuels simples	• Moins performant que les CNN avancés • Structure manuelle
EfficientNetB0	CNN avancé pré-entraîné	• Excellente extraction des caractéristiques visuelles • Bonnes performances en clustering	• Plus coûteux en calcul • Nécessite plus de données et de temps pour l'entraînement

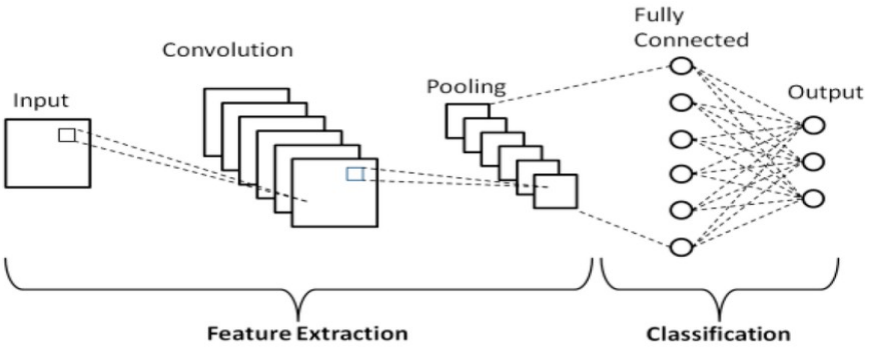


MLP minimal (réseau simple)

EfficientNetB0 (CNN avancé)



CNN simple



Préparation des données

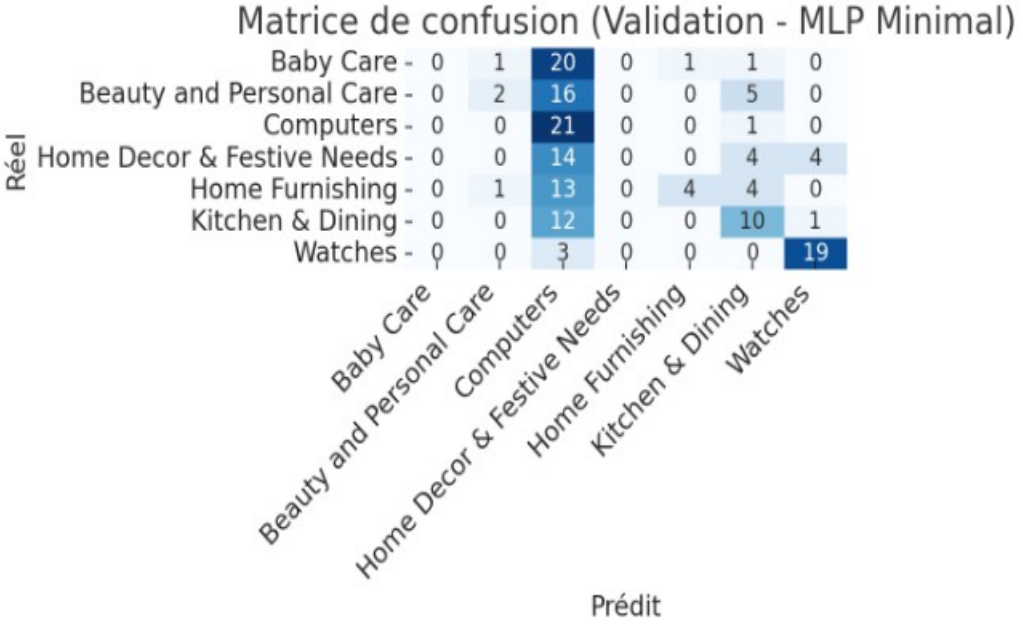
Séparation des données :

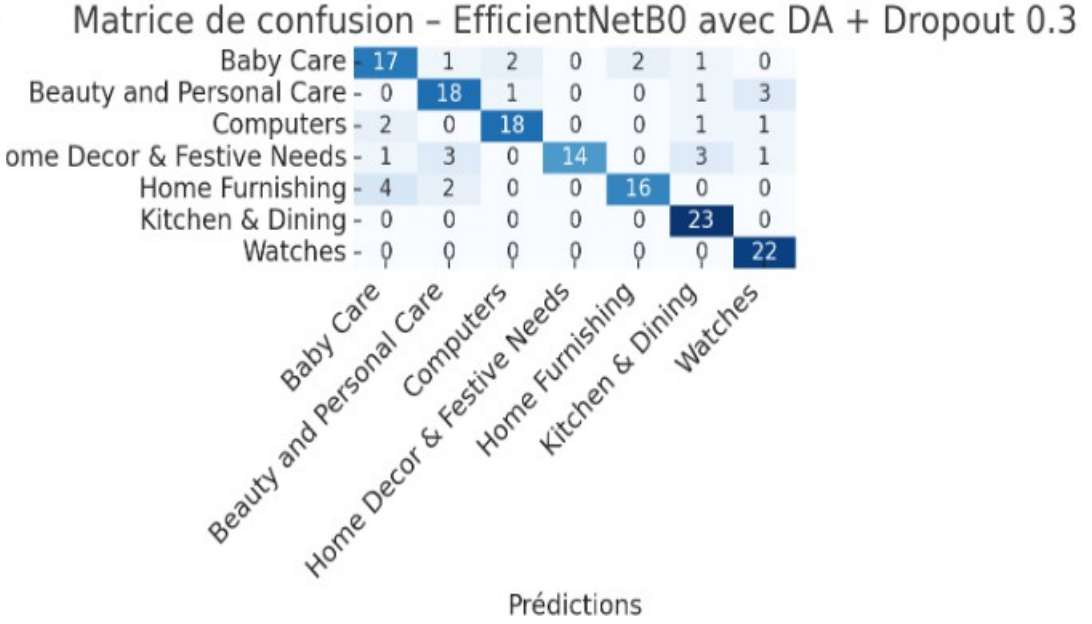
- Répartition stratifiée des images en trois ensembles :
- Entraînement : 70 %
- Validation : 15 %
- Test : 15 %
- Cela garantit que chaque catégorie est bien représentée dans chaque ensemble.

Étape	MLP minimal	CNN simple	EfficientNetB0
Chargement	Depuis le chemin de l'image	Depuis le chemin de l'image	Depuis le chemin de l'image
Couleurs	Couleur (RGB)	Couleur (RGB)	Couleur (RGB)
Redimensionnement	224 x 224 pixels	224 x 224 pixels	224 x 224 pixels
Normalisation	Pixels entre 0 et 1 (img / 255.0)	Pixels entre 0 et 1 (img / 255.0)	Pixels en float32 (pas de division, valeurs inchangées)
Format final	Tableau NumPy (n_images, 224, 224, 3)	Tableau NumPy (n_images, 224, 224, 3)	Tableau NumPy (224, 224, 3) (une image à la fois, passage direct au modèle)
Prétraitement modèle	Aucun spécifique (images directement utilisées)	Aucun spécifique (images directement utilisées)	Prétraitement intégré via la couche Rescaling d'EfficientNetB0

Comparaison globale des performances

	Entrainement	Validation	Test	Remarques
Accuracy (MLP minimal)	0,47	0,36	0,37	Baseline
Accuracy (CNN simple)	0,66	0,48	0,51	Première amélioration par rapport au MLP, exploite la structure des images
Accuracy(EfficientNetB0 sans DA)	0,99	0,88	0,88	Surapprentissage
Accuracy (EfficientNetB0 avec DA)	0,88	0,79	0,80	Moins de surapprentissage, plus robuste
Précision (EfficientNetB0 avec DA + Dropout 0,3)	**0,89**	**0,82**	0,81	Meilleur compromis





Architecture et initialisation du modèle EfficientNetB0

Architecture du modèle

- EfficientNetB0 pré-entraîné (ImageNet) utilisé comme extracteur de caractéristiques
- **Poids gelés** pour conserver ce qu'il a appris
- Couches ajoutées :
  - **GlobalAveragePooling2D** : réduit la dimension des features extraites.
  - **Dense**(256, activation='relu') : apprend les combinaisons pertinentes pour notre tâche.
  - **Dropout**(0.5 ou 0.3) : régularise l'apprentissage, limite le surapprentissage.
  - **Dense**(num_classes, activation='softmax') : sortie multi-classes adaptée à notre problème.

Stratégie d'entraînement

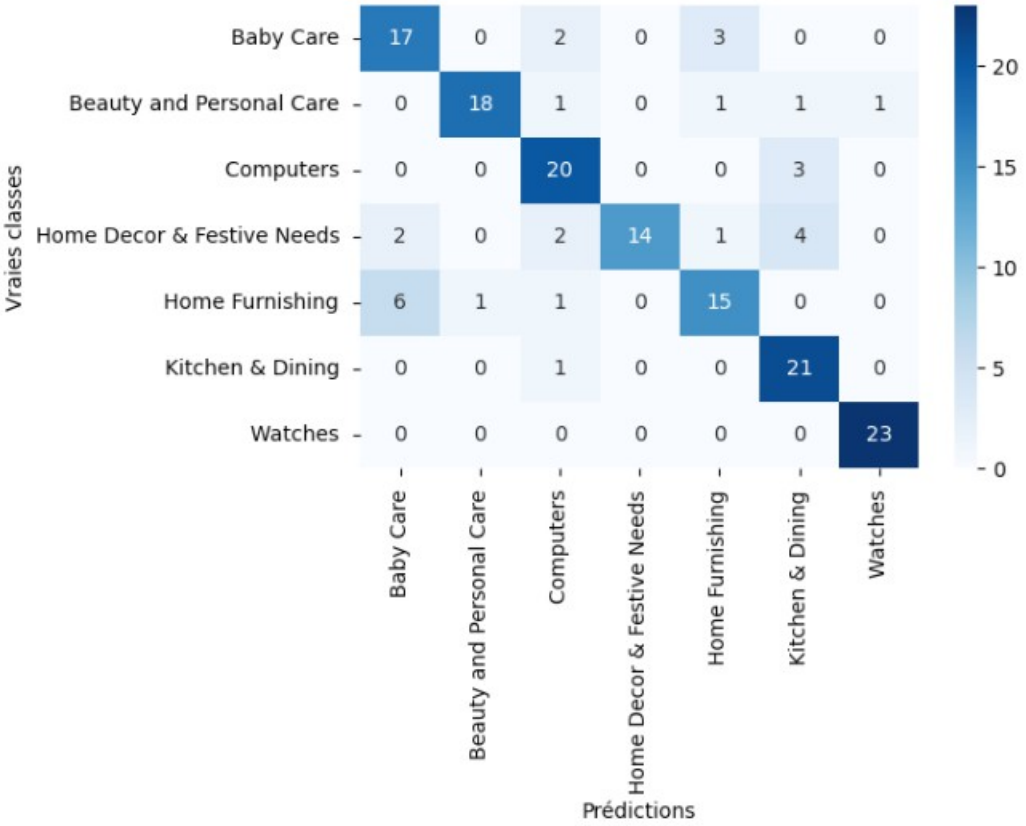
- Callbacks utilisés :
  - **ModelCheckpoint** (sauvegarde du meilleur modèle sur val_loss)
  - **EarlyStopping** (arrêt précoce si stagnation de val_loss)

Variante	Data Augmentation	Dropout
V1	❌	0.5
V2	✅	0.5
V3	✅	0.3

Analyse du modèle EfficientNetB0 (Data Augmentation + Dropout 0.3)

	Accuracy	Loss
Entrainement	0,89	0,32
Validation	0,82	0,77
Test	0,81	0,56

Écart modéré entre entraînement et test : surapprentissage limité, bonne généralisation.



	precision	recall	f1-score	support
Baby Care	0.68	0.77	0.72	22
Beauty and Personal Care	0.95	0.82	0.88	22
Computers	0.74	0.87	0.80	23
Home Decor & Festive Needs	1.00	0.61	0.76	23
Home Furnishing	0.75	0.65	0.70	23
Kitchen & Dining	0.72	0.95	0.82	22
Watches	0.96	1.00	0.98	23
accuracy			0.81	158
macro avg	0.83	0.81	0.81	158
weighted avg	0.83	0.81	0.81	158

Bonnes performances (F1 élevé):

- Watches (F1 = 0,98)
- Beauty and Personal Care (0,88)
- Kitchen & Dining (F1 = 0,82)
- Computers (F1 = 0,80)

Performances plus faibles (F1 bas) :

- Home Decor & Festive Needs (0,76)
- Home Furnishing (0,70)
- Baby Care (0,72)

Catégories les plus confondues :

- Home Furnishing ↔ Baby Care
- Home Decor & Festive Needs ↔ Kitchen & Dining
- Computers ↔ Kitchen & Dining

Collecte automatisée de nouveaux produits via l’API Open Food Facts

Collecte automatisée de nouveaux produits via l'API Open Food Facts

Objectif : Élargir la gamme à l'épicerie fine en collectant des produits à base de champagne via une API ouverte.

Démarche :

- Requête API ciblée sur le mot-clé "champagne" (50 produits initialement récupérés).
- Filtrage :
  - Seuls les produits dont la liste des ingrédients contient "champagne" sont retenus.
  - Exclusion des produits contenant "Ardenne" ou "fine champagne" dans le nom, la catégorie ou les ingrédients
- Extraction des champs clés :
  - ID unique (foodId)
  - Nom du produit
  - Catégorie
  - Liste des ingrédients
  - URL de l'image
- Limitation à 10 produits validés et export en CSV.

Limites de la démarche :

- Présence de faux positifs : présence de produits régionaux (ex : "Miel de Champagne").
- Impossibilité de distinguer automatiquement entre "champagne" boisson et "Champagne" région.
- Taux d'erreur observé : environ 1/10 produits collectés sont des faux positifs dans l'échantillon testé.
- Limite inhérente à la donnée : une validation manuelle serait nécessaire pour une sélection 100 % fiable.

Respect RGPD : Collecte limitée aux données nécessaires, issues d'une base publique, sans données personnelles sensibles.

Conclusion & Perspectives

Bilan du projet

- Performances variables selon les catégories : texte et image sont complémentaires.
- Certaines classes ambiguës restent difficiles à distinguer.
- Collecte automatique possible (ex : produits à base de champagne), mais nécessite un contrôle qualité pour fiabiliser les données.

Pistes d'amélioration :

- Modèle multimodal texte + image
- Sous-modèles/règles pour les catégories ambiguës
- Enrichissement des données
- Modèles avancés (Vision Transformers, CLIP…)
- Fusion intelligente des prédictions