Classifier automatiquement des biens de consommation

23/02/2024

Introduction

- “Place du marché”: objectif, lancer une marketplace e-commerce

- Contexte:
  - Attribution de la catégorie d'un article: manuellement par les vendeurs
  - Petit volume de produits
  - Souhait d’**automatiser** l‘attribution de la catégorie

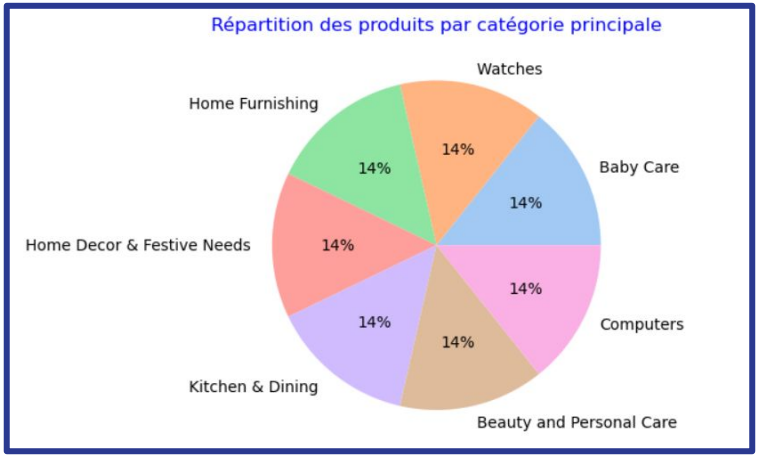
- Démarche globale:
  - Etudier la **faisabilité** de classification textes et images
  - **Classification supervisée** des images
  - Elargissement gamme de produits -> Tester une API -> produits base de champagne

Description du jeu de données

- 1050 produits
  - **Textes**:Descriptions, catégorie …
  - **Images**

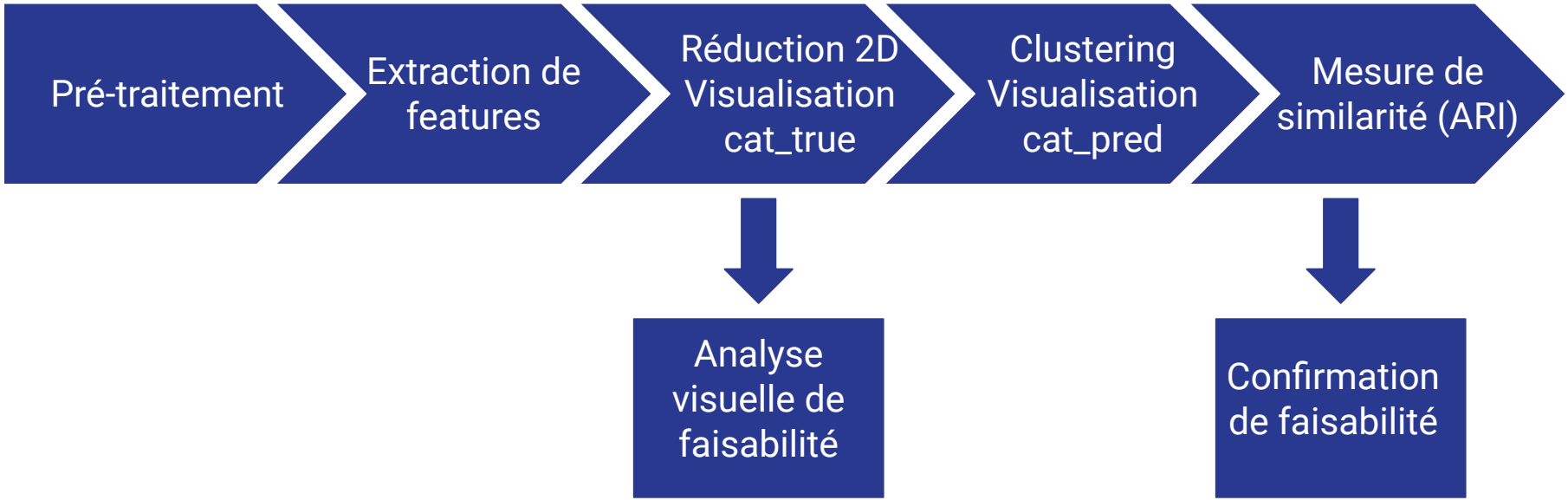
- Nettoyage:
  - Pas de valeurs manquantes
  - Pas de valeurs aberrantes
  - Pas de doublons

product_name	product_category_tree
Elegance Polyester Multicolor Abstract Eyelet ...	["Home Furnishing >> Curtains & Accessories >>...
Sathiyas Cotton Bath Towel	["Baby Care >> Baby Bath & Skin >> Baby Bath T...
Eurospa Cotton Terry Face Towel Set	["Baby Care >> Baby Bath & Skin >> Baby Bath T...



Partie 1: Faisabilité classification texte

Démarche de faisabilité commune texte/images



Pré-traitement texte

- Mise en minuscules
- Tokenisation => fonction word_tokenize (divise un document en mots)
- Suppression des stopwords => fonction stopwords.words('english')
- Stemming => classe PorterStemmer() -> réduction des mots à leur racine (suffixe tronqué)
- Lemmatisation => classe WordNetLemmatizer -> réduction des mots à leur racine avec dictionnaire

```
1 texte['preprocessed_product_description'][0]
```

```
'elegance polyester multicolor abstract eyelet door curtain key feature elegance polyester multicolor abstract eyelet door curt
ain floral curtain elegance polyester multicolor abstract eyelet door curtain height pack price curtain enhances look curtain m
ade high quality polyester feature eyelet style stitch metal make room environment romantic curtain wrinkle anti shrinkage eleg
ant home bright modernistic appeal design surreal attention sure steal heart contemporary eyelet valance curtain slide smoothly
```

Extraction de features texte - Approche BOW

- **Comptage simple**
  - Sortie: vecteur dimension 1988

	aapno	ability	able	abode	absorbency	absorbent	abstract
0	0	0	0	0	0	0	5
1	0	0	0	0	0	0	0

`sklearn.feature_extraction.text`.CountVectorizer

- **Tf-idf**
  - Méthode de pondération
  - Sortie: vecteur dimension 1988

	aapno	ability	able	abode	absorbency	absorbent	abstract
0	0	0	0	0	0	0	0.200871
1	0	0	0	0	0	0	0

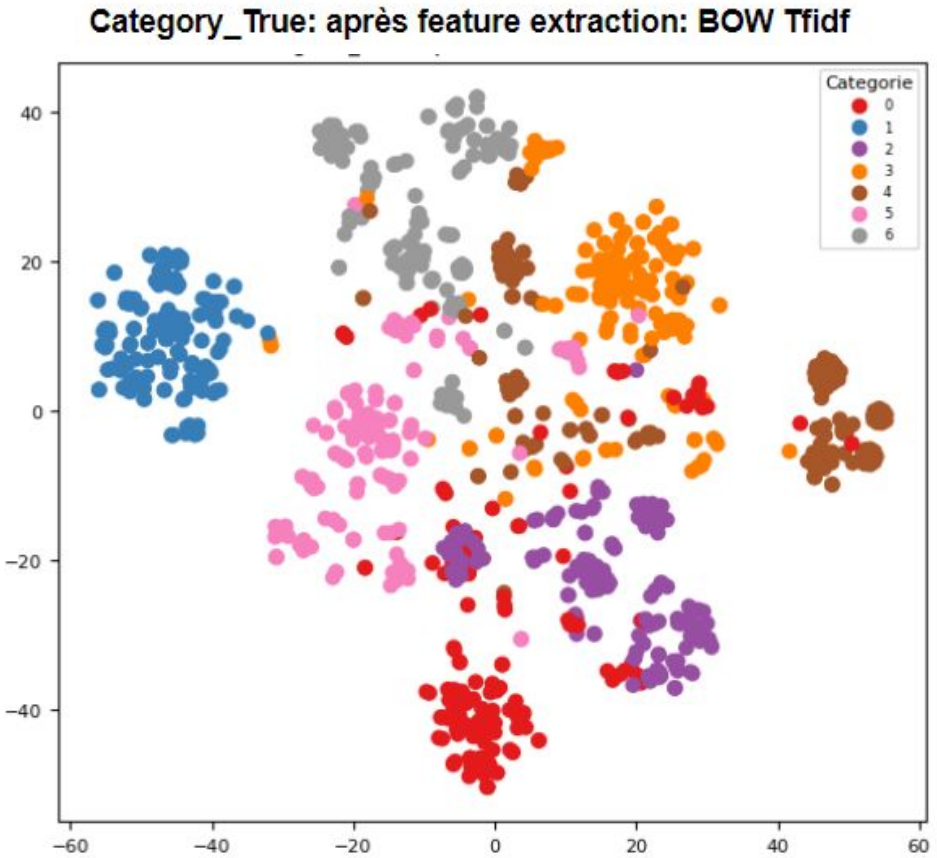
`sklearn.feature_extraction.text`.TfidfVectorizer

Extraction de features texte - Word Embedding

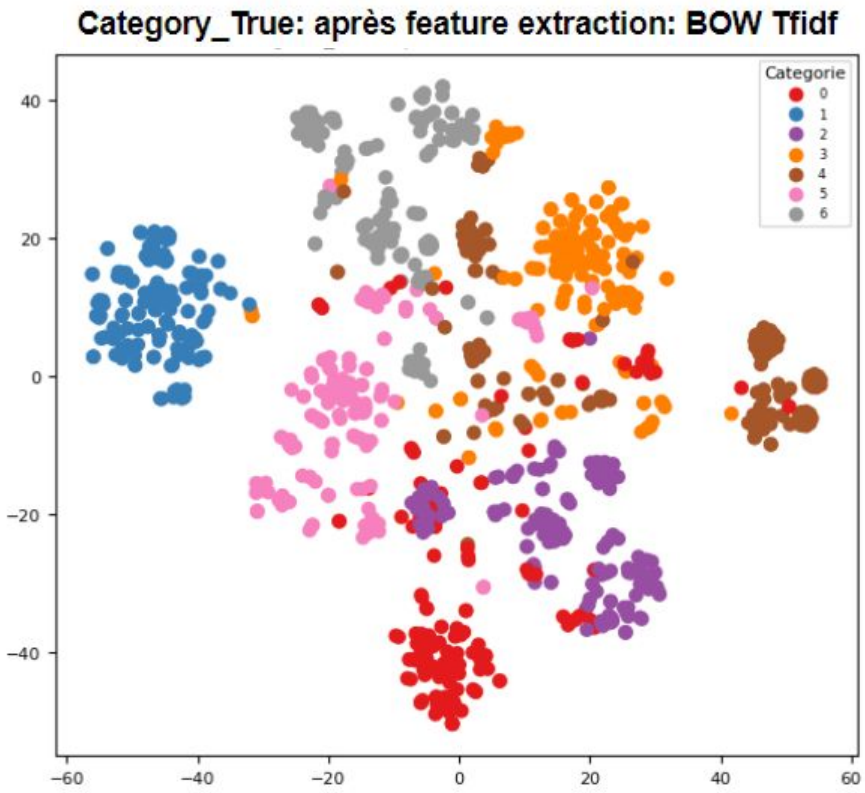
- **Word2Vec**
  - Représentation de mots dans un espace qui rapproche les mots similaires
  - **Modèle à entraîner**
  - Entrée: description produits
  - Sortie: vecteur de dimension 100
- **BERT**
  - Google AI (2018)
  - Pré-entrainé avec corpus Wikipedia
  - Bibliothèque Transformers dev par HuggingFace
  - Entrée: description produits
  - Sortie: vecteur dimension 768
- **USE**
  - Google - USE4 (2020)
  - Modèle déjà entraîné
  - Sortie: vecteur dimension 512

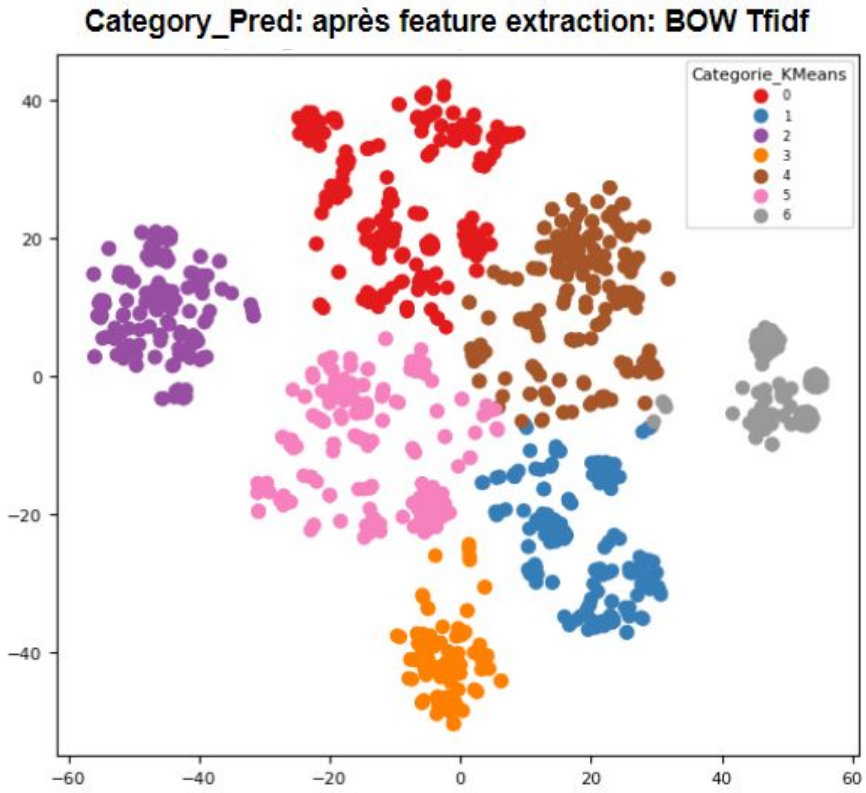
Réduction 2D et visualisation des produits

- Réduction 2D: TSNE
- Visualisation des produits (category_true)
- Analyse graphique: **séparation faisable** (pour toutes les méthodes d'extraction de features)



Mesure de similarité





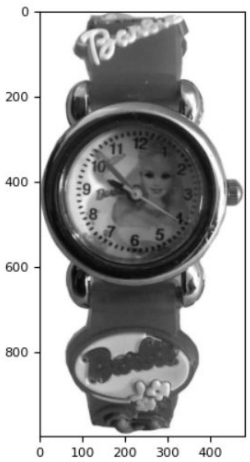
ARI : 0.5615100707190486

La mesure de similarité confirme la faisabilité de la classification de produits à partir du texte

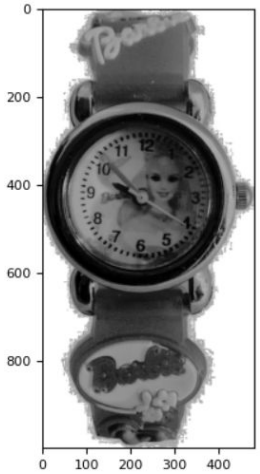
Partie 2: Faisabilité classification images

Pré-traitement images

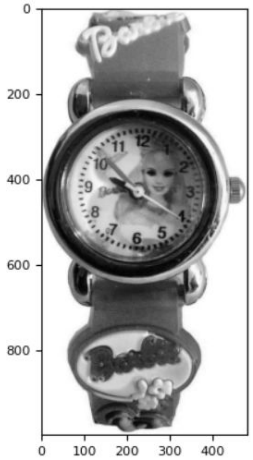
Passage en niveau de gris



cv2.equalizeHist



cv2.createCLAHE



- Passage en niveau de gris: nécessaire pour **SIFT**
- Egalisation: méthode **CLAHE** meilleure

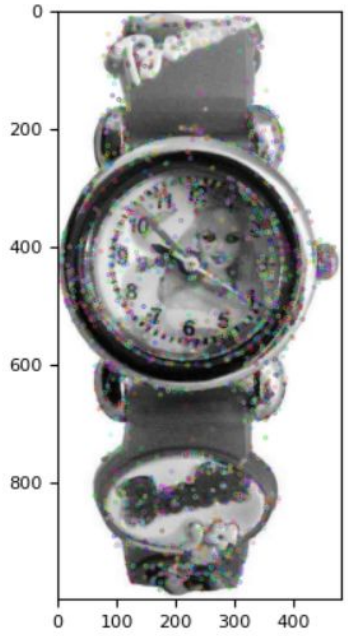
Extraction de features images avec SIFT

Principes:

- Extraire des features (ou points d'intérêt) de l'image et de calculer leurs descripteurs
- Un descripteur SIFT est composé de 128 valeurs entières

Etapes:

- Extraction des descripteurs de chaque image -> Nombre de descripteurs : (517351, 128)
- Création des clusters de descripteurs (719 clusters)
- Création des histogrammes/features des images -> (1050, 719)




```
Descripteurs :  (2188, 128)
```

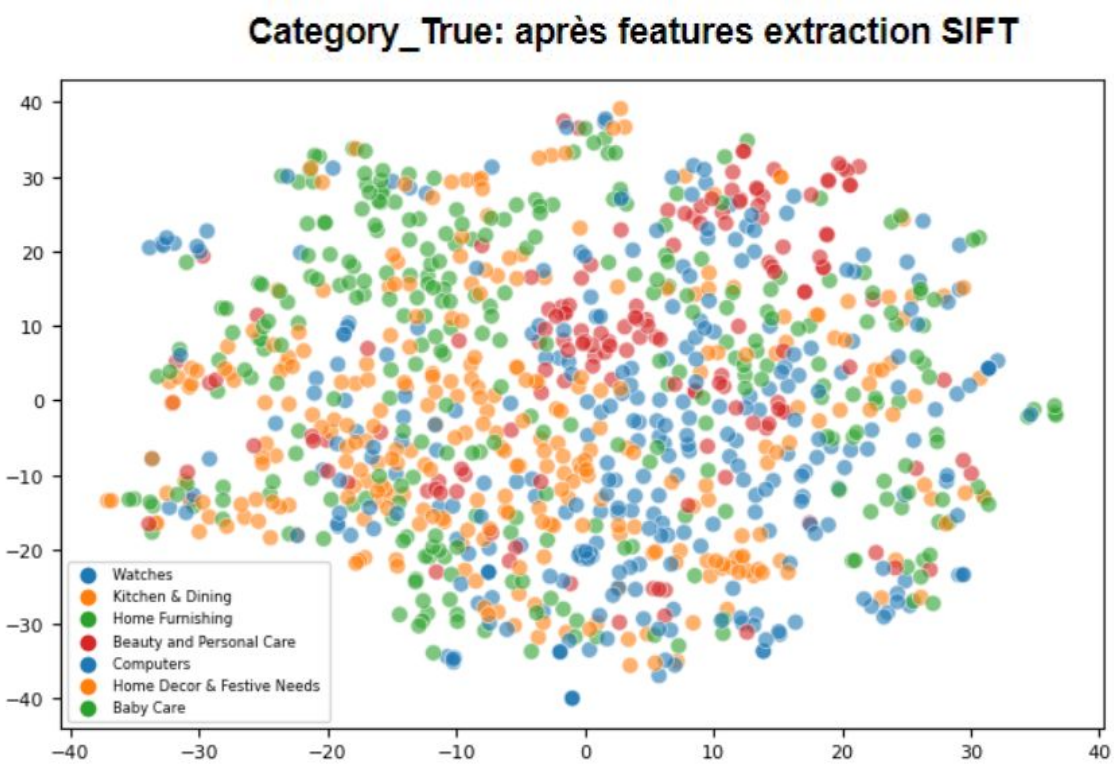
Réduction 2D et visualisation des produits

- Réduction 2D:
  - (1050, 719) -> ACP -> (1050, 502)
  - TSNE -> (1050, 2)

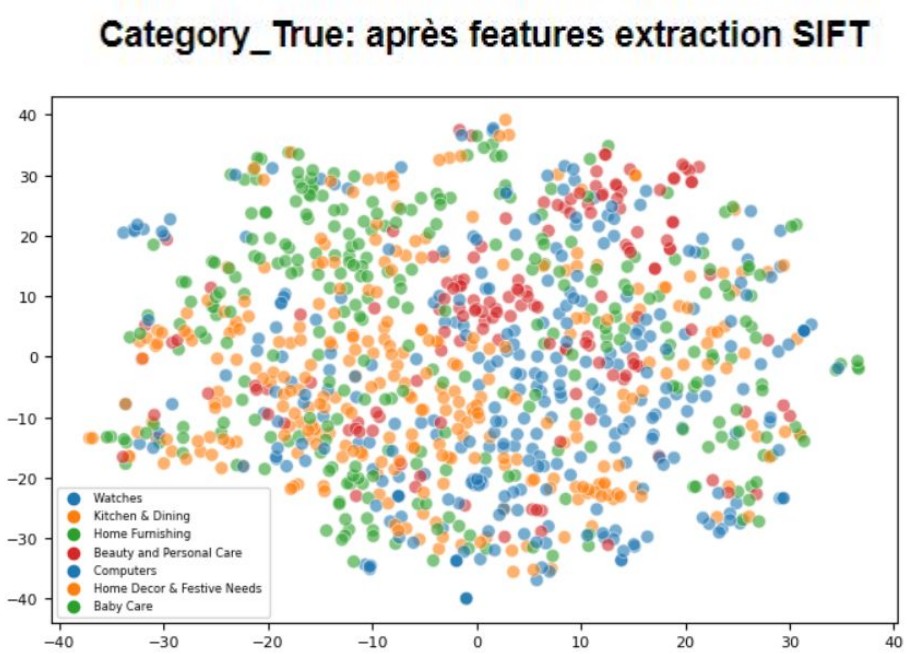
- Visualisation des produits (category_true)

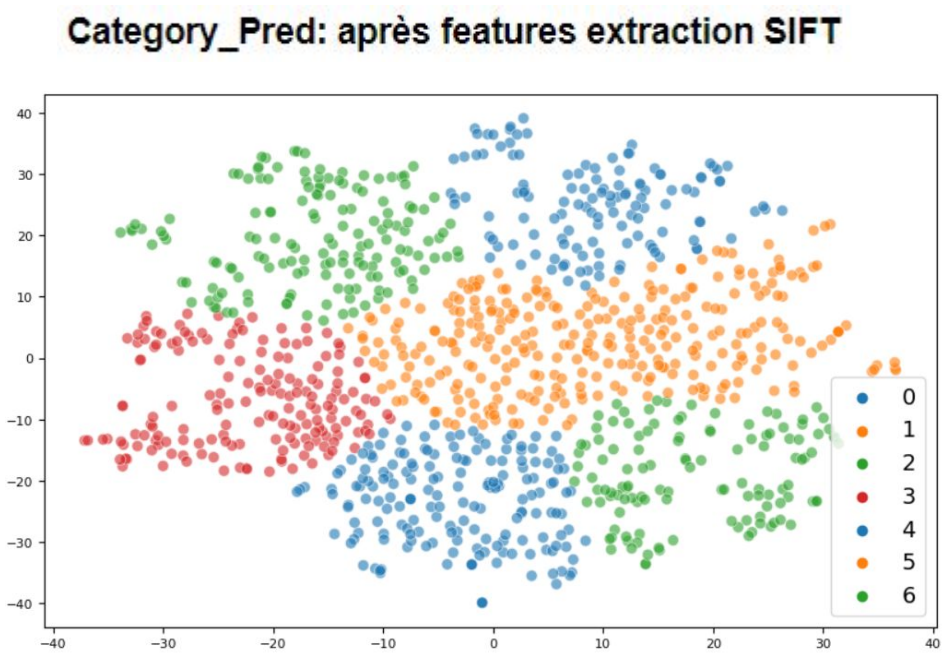
- Analyse graphique:

  pas possible de séparer les images selon leur vraie classe



Mesure de similarité

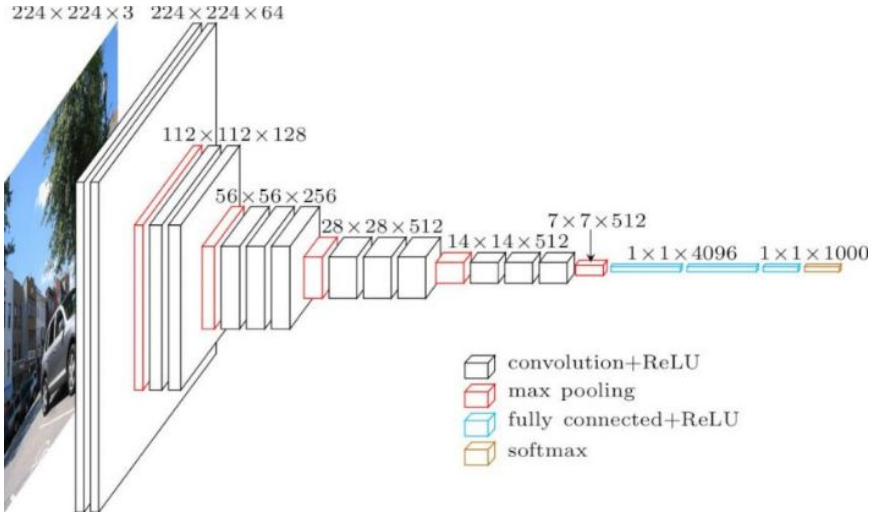





```
ARI :  0.044
```

Le score ARI très faible de 0.04 confirme l'**impossibilité de séparer les images** selon leurs vraies classes après features extraction SIFT

Extraction de features images avec VGG16



Etapes:

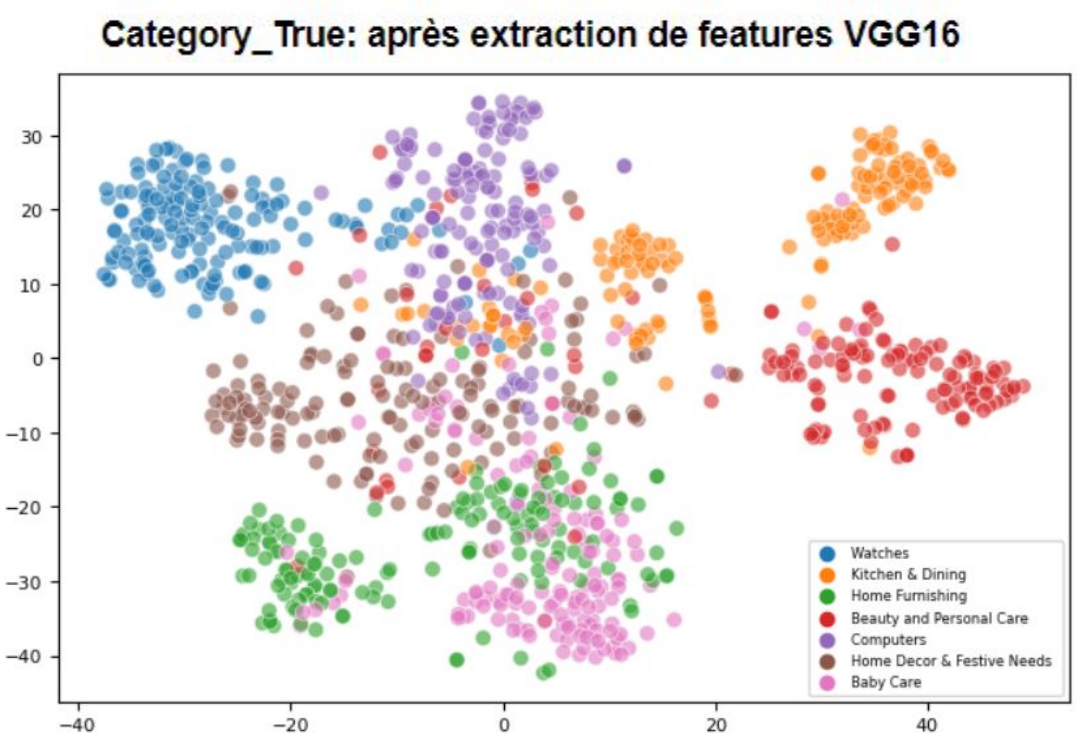
- Preprocessing Keras -> images 224x224x3
- Création du modèle VGG16 à partir du modèle de base
  - On choisit la sortie de l'avant dernière couche (4096)
- Extraction des features des images
  - Prédiction à partir du modèle VGG16 de base pré-entraîné
  - Sortie: (1050,4096)

Réduction 2D et visualisation des produits

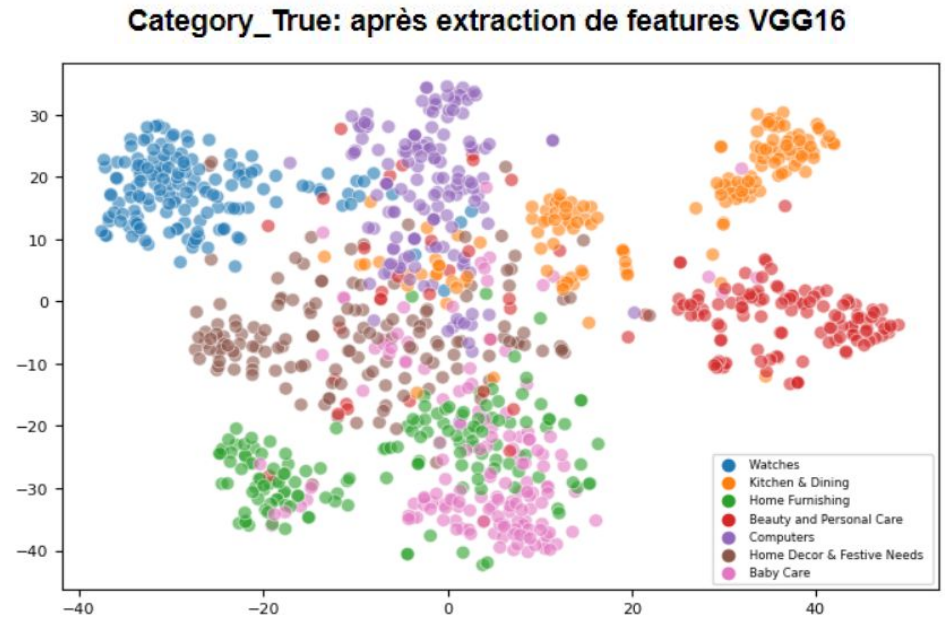
- Réduction 2D:
  - (1050, 4096) -> ACP -> (1050, 803)
  - TSNE -> (1050, 2)

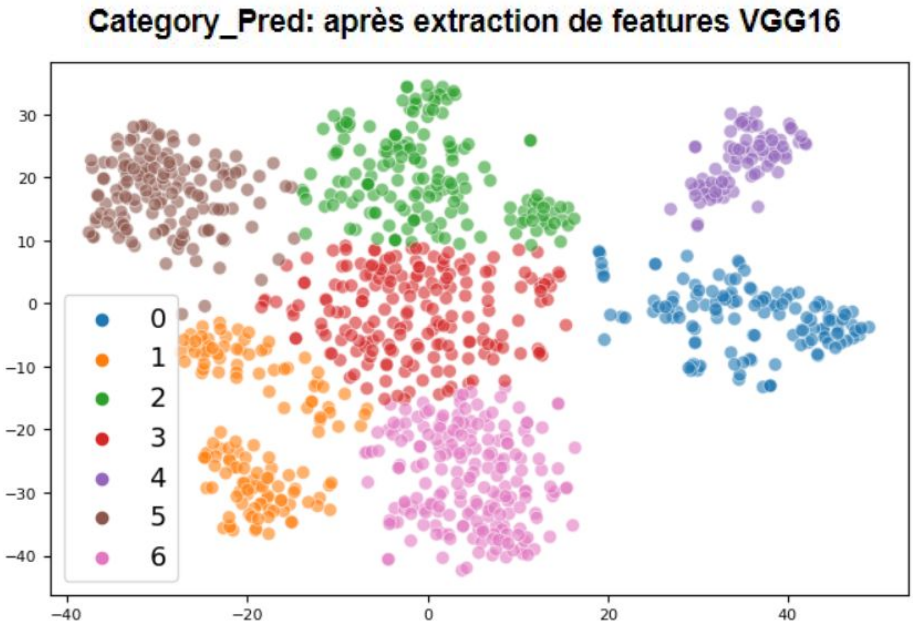
- Visualisation des produits (category_true)

- Analyse graphique: **possible de séparer les images** selon leur vraie classe



Mesure de similarité





ARI : 0.449

Le score ARI=0.449 confirme la **possibilité de séparer les images** selon leurs vraies classes après **features extraction VGG16**

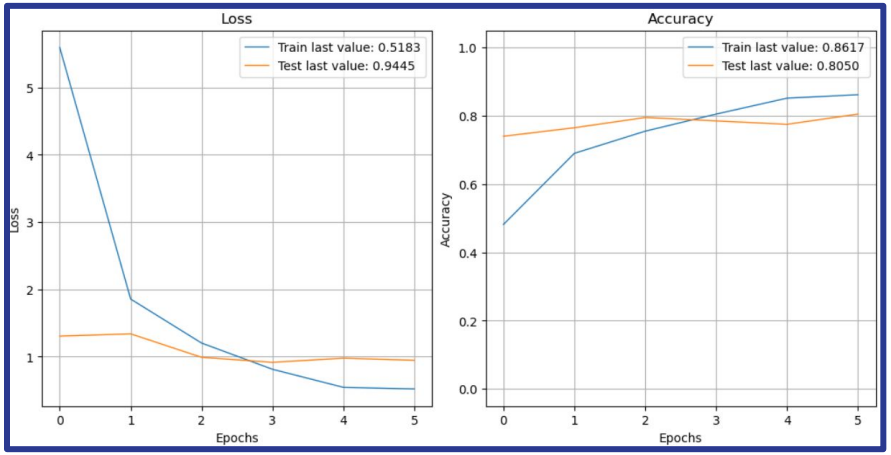
Partie 3: Classification supervisée

Démarche

- Faisabilité par méthode non supervisée (temps) -> Classification supervisée
- CNN VGG16 - Keras
- Transfert learning (ImageNet)
- 3 Stratégies possibles
  - 1: Fine-tuning total
  - **2: Extraction de features => Stratégie choisie**
  - 3: Fine-tuning partiel

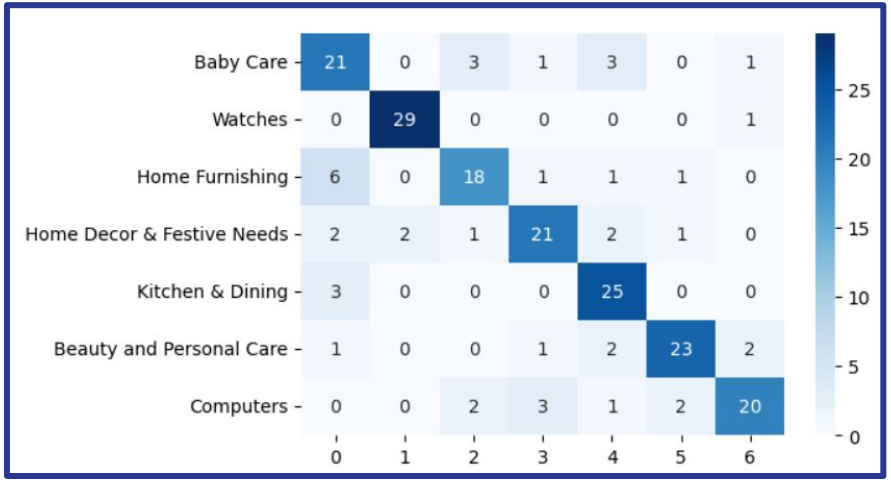
- 2 modèles testés
  - VGG16 sans data augmentation
  - VGG16 avec data augmentation
    - Essais pour optimiser les hyperparamètres (Flip, zoom, ...)

Modèle 1: VGG16 sans data augmentation



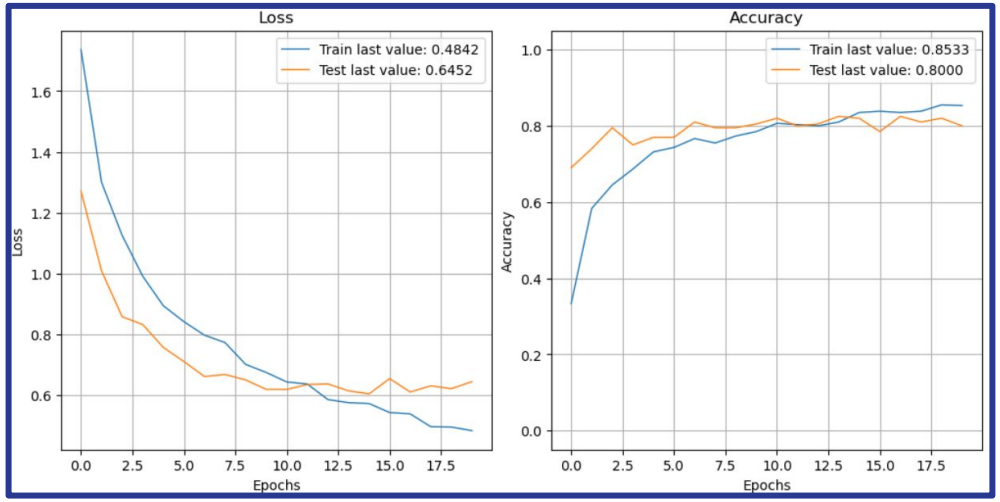
- Epoch optimal (min loss):

Validation Accuracy : 0.78
Test Accuracy : 0.72



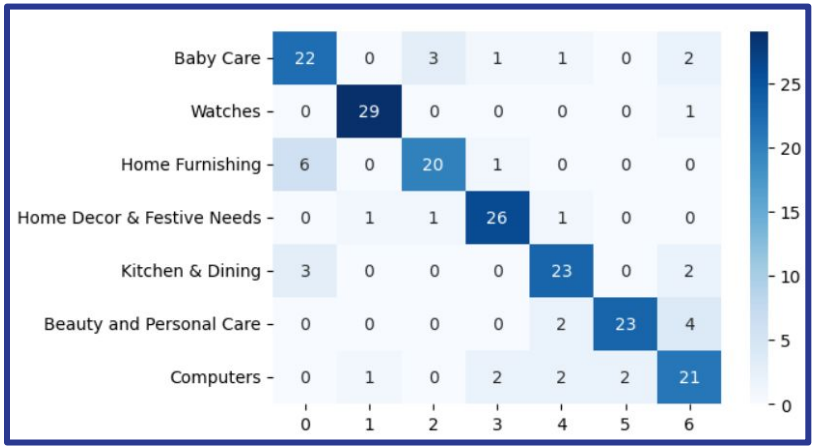
	precision	recall	f1-score	support
0	0.64	0.72	0.68	29
1	0.94	0.97	0.95	30
2	0.75	0.67	0.71	27
3	0.78	0.72	0.75	29
4	0.74	0.89	0.81	28
5	0.85	0.79	0.82	29
6	0.83	0.71	0.77	28
accuracy			0.79	200
macro avg	0.79	0.78	0.78	200
weighted avg	0.79	0.79	0.78	200

Modèle 2: VGG16 avec data augmentation



- Epoch optimal (min loss):

Validation Accuracy : 0.82
Test Accuracy : 0.77



	precision	recall	f1-score	support
0	0.71	0.76	0.73	29
1	0.94	0.97	0.95	30
2	0.83	0.74	0.78	27
3	0.87	0.90	0.88	29
4	0.79	0.82	0.81	28
5	0.92	0.79	0.85	29
6	0.70	0.75	0.72	28
accuracy			0.82	200
macro avg	0.82	0.82	0.82	200
weighted avg	0.82	0.82	0.82	200

Comparaison des performances

Modèle	Hyperparamètres	Validation Accuracy	Test Accuracy	Temps de train
VGG16 sans data augmentation	optimizer='rmsprop'	0.78	0.72	269 sec
VGG16 avec data augmentation	optimizer='rmsprop' RandomFlip=H RandomRot=0.1 RandomZoom=0.1 Rescaling=[-1,1] Dropout=0.5 Early Stop - Patience=5	**0.82**	**0.77**	973 sec

Partie 4: Présentation du test de l’API

Test de l'API "Edamam Food and Grocery Database"

- Clé d'accès personnelle à l'API: header 'X-RapidAPI-Key'
- Requête GET /api/food-database/v2/parser
- Conversion réponse JSON (dictionnaire)
- Traitement de la réponse
  - Dictionnaire -> liste de liste (10 produits)
  - Transformation liste en dataframe
  - Export dans fichier csv
- Prise en compte des normes RGPD (sécurité, minimisation)

	foodId	label	category	foodContentsLabel	image
0	food_a656mk2a5dmqb2adiamu6beihduu	Champagne	Generic foods		https://www.edamam.com/food-img/a71/a718cf3c52...
1	food_b753ithamdb8psbt0w2k9aquo06c	Champagne Vinaigrette, Champagne	Packaged foods	OLIVE OIL; BALSAMIC VINEGAR; CHAMPAGNE VINEGAR...	
2	food_b3dyababjo54xobm6r8jzbghjgqe	Champagne Vinaigrette, Champagne	Packaged foods	INGREDIENTS: WATER; CANOLA OIL; CHAMPAGNE VINE...	https://www.edamam.com/food-img/d88/d88b64d973...
3	food_a9e0ghsamvoc45bwa2ybsa3gken9	Champagne Vinaigrette, Champagne	Packaged foods	CANOLA AND SOYBEAN OIL; WHITE WINE (CONTAINS S...	
4	food_an4jjueaucpus2a3u1ni8auhe7q9	Champagne Vinaigrette, Champagne	Packaged foods	WATER; CANOLA AND SOYBEAN OIL; WHITE WINE (CON...	
5	food_bmu5dmkazwuvpaa5prh1daa8jxs0	Champagne Dressing, Champagne	Packaged foods	SOYBEAN OIL; WHITE WINE (PRESERVED WITH SULFIT...	https://www.edamam.com/food-img/ab2/ab2459fc2a...
6	food_alpl44taoyv11ra0lic1qa8xculi	Champagne Buttercream	Generic meals	sugar; butter; shortening; vanilla; champagne;...	
7	food_byap67hab6evc3a0f9w1oag3s0qf	Champagne Sorbet	Generic meals	Sugar; Lemon juice; brandy; Champagne; Peach	
8	food_am5egz6aq3fpjlaf8xpkdbc2asis	Champagne Truffles	Generic meals	butter; cocoa; sweetened condensed milk; vanil...	
9	food_bcz8rhiajk1fuva0vkfmeakbouc0	Champagne Vinaigrette	Generic meals	champagne vinegar; olive oil; Dijon mustard; s...	

Conclusion

- Classification texte faisable: meilleur score BOW-Tfidf > USE
- Classification images: pas faisable avec SIFT, faisable avec VGG16
- Classification images:
  - transfert learning
  - impact data augmentation
- Axes d'amélioration: optimisation hyperparamètres data augmentation