Classification automatique des produits

place de marché

Août 2023

CONTEXTE

aujourd'hui

- catégorisation des articles peu fiable : manuelle, par le vendeur
- volume des articles petit

pour demain

nécessité d'automatiser la catégorisation

- améliorer l'expérience utilisateur (vendeur & acheteur)
- pouvoir passer à l'échelle

OBJECTIFS

- Déterminer la faisabilité d'un regroupement automatique
- Mettre en œuvre une classification supervisée à partir des images
- Tester la collecte de données de produits via une API

Le jeu de données

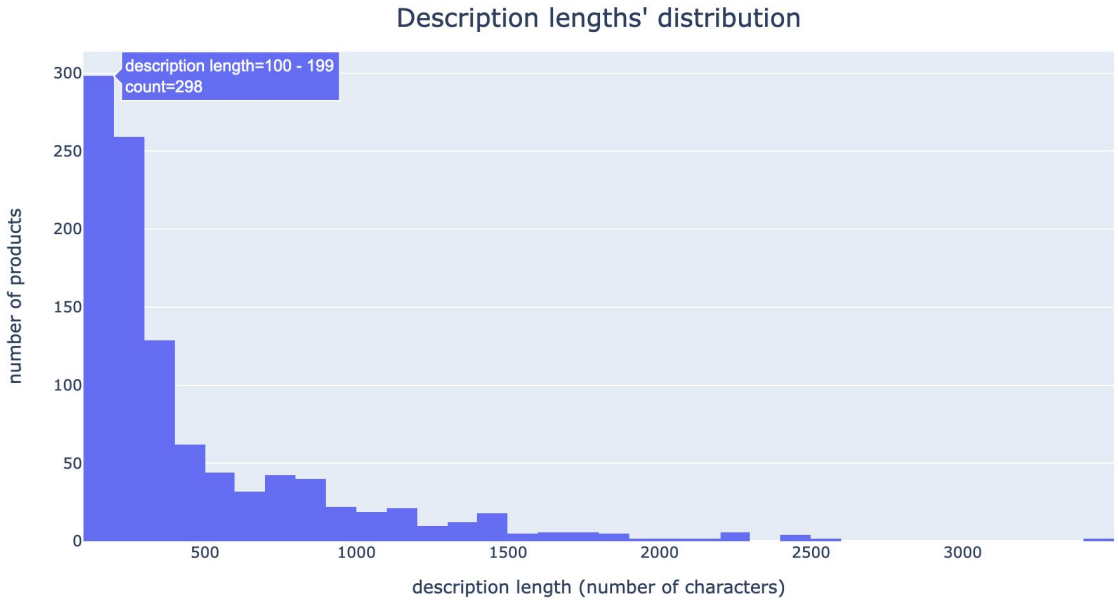
échantillon d'images des 7 catégories : Baby Care, Beauty and Personal Care, Computers, Home Decor & Festive Needs, Home Furnishing, Kitchen & Dining, Watches

- ➔ 1 table, 1050 produits, image & description
- ➔ 1er niveau de catégories (/3) : 7 catégories
  chacune de 150 produits
- ➔ 0 doublons, 0 valeurs manquantes
  sur les identifiants, les images et les descriptions

Le jeu de données

"Key Features of CoffeeBean Regular Fit Baby Girl's Pink Trousers Occasion: Casual Suitable For:Western Wear Color: Pink Fit: Regular Fit Closure:Button Fabric:Cotton,CoffeeBean Regular Fit Baby Girl's Pink Trousers Price: Rs. 599 Kids Girls Printed Trouser with curved pocket at front side, it is very good in quality your baby girl will look smart and cute in this trouser.,Specifications of CoffeeBean Regular Fit Baby Girl's Pink Trousers General Details Pattern Floral Print Occasion Casual Ideal For Baby Girl's Alteration Required No Color Pink Trouser Details Closure Button Number of Contents in Sales Package Pack of 1 Fabric Cotton Type Chinos Fit Regular Fit Belt Loops Yes Fly Zipper In the Box 1 Trouser Additional Details Style Code 240KGT_PRINTED Fabric Care Machine Wash in Lukewarm Water"

exemple de description de produit



➔ 53 % des descriptions < 300 caractères

➔ jusqu'à 3490 caractères

1.

Étude de faisabilité

Étude de faisabilité

méthode retenue : après divers traitements des données (descriptions puis images), visualisation en 2D de leur répartition

→ *les produits se regroupent-ils conformément aux catégories ?*

➔ méthodes d'embedding testées pour les descriptions :

- → BoW (term-frequency & TF-IDF)
- → Doc2Vec
- → BERT
- → USE

➔ méthodes d'extraction de features testées pour les images :

- → SIFT + bag of images
- → CNN

Étude de faisabilité > descriptions

➔ fonctions de traitement pré-embedding :

→ avant Term-Frequency, TF-IDF et Doc2Vec :

- mise en minuscules
- tokenisation
- suppression des stopwords
- suppression (ou non) des tokens rares / très courts / non-anglais
- stemming ou lemmatization

→ avant BERT :

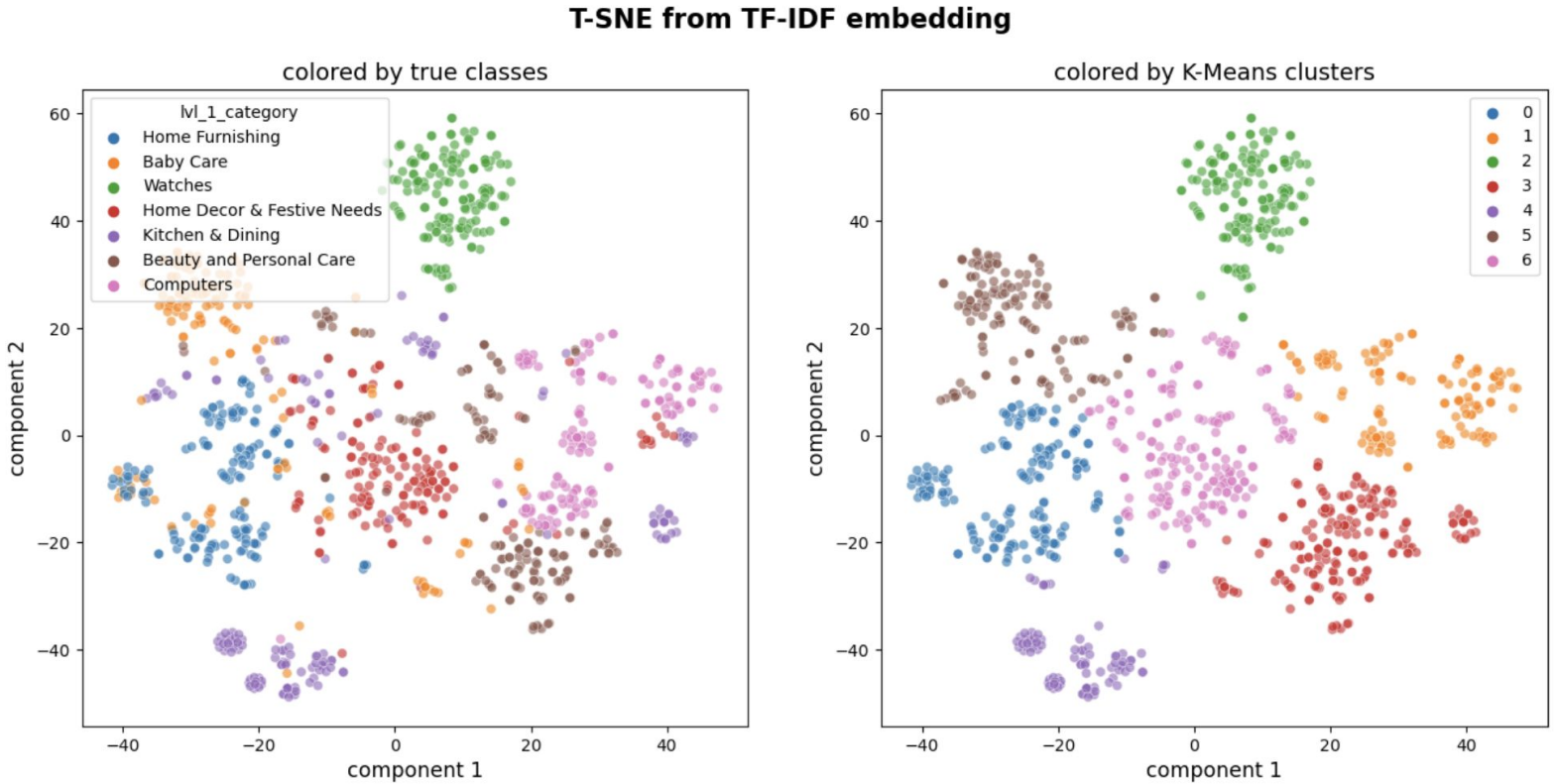
- tokenisation (BertTokenizer)

➔ méthodes d'embedding (TF, TF-IDF, Doc2Vec, BERT, USE)

➔ réduction en 2 dimensions avec T-SNE

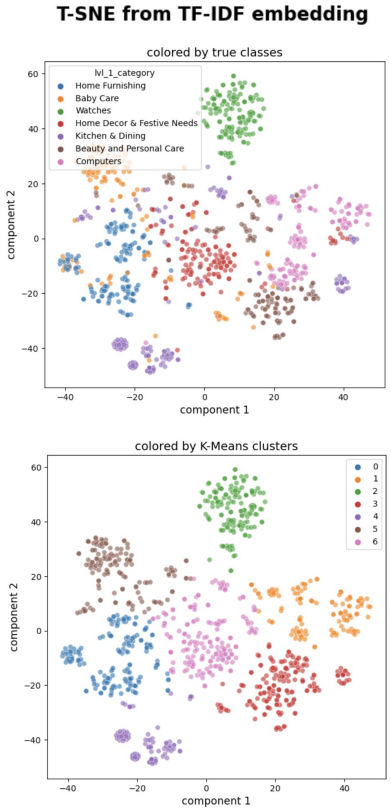
➔ visualisation

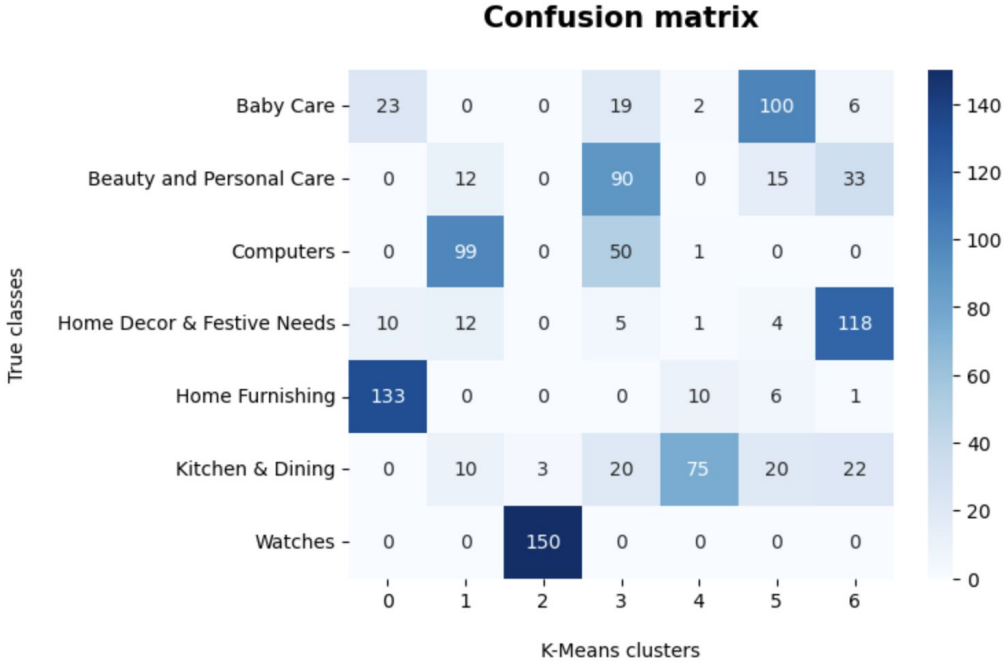
ARI = 0.51


T-SNE from TF-IDF embedding
colored by true classes
lvl_1_category
Home Furnishing
Baby Care
Watches
Home Decor & Festive Needs
Kitchen & Dining
Beauty and Personal Care
Computers
component 2
component 1
colored by K-Means clusters
0
1
2
3
4
5
6
component 2
component 1

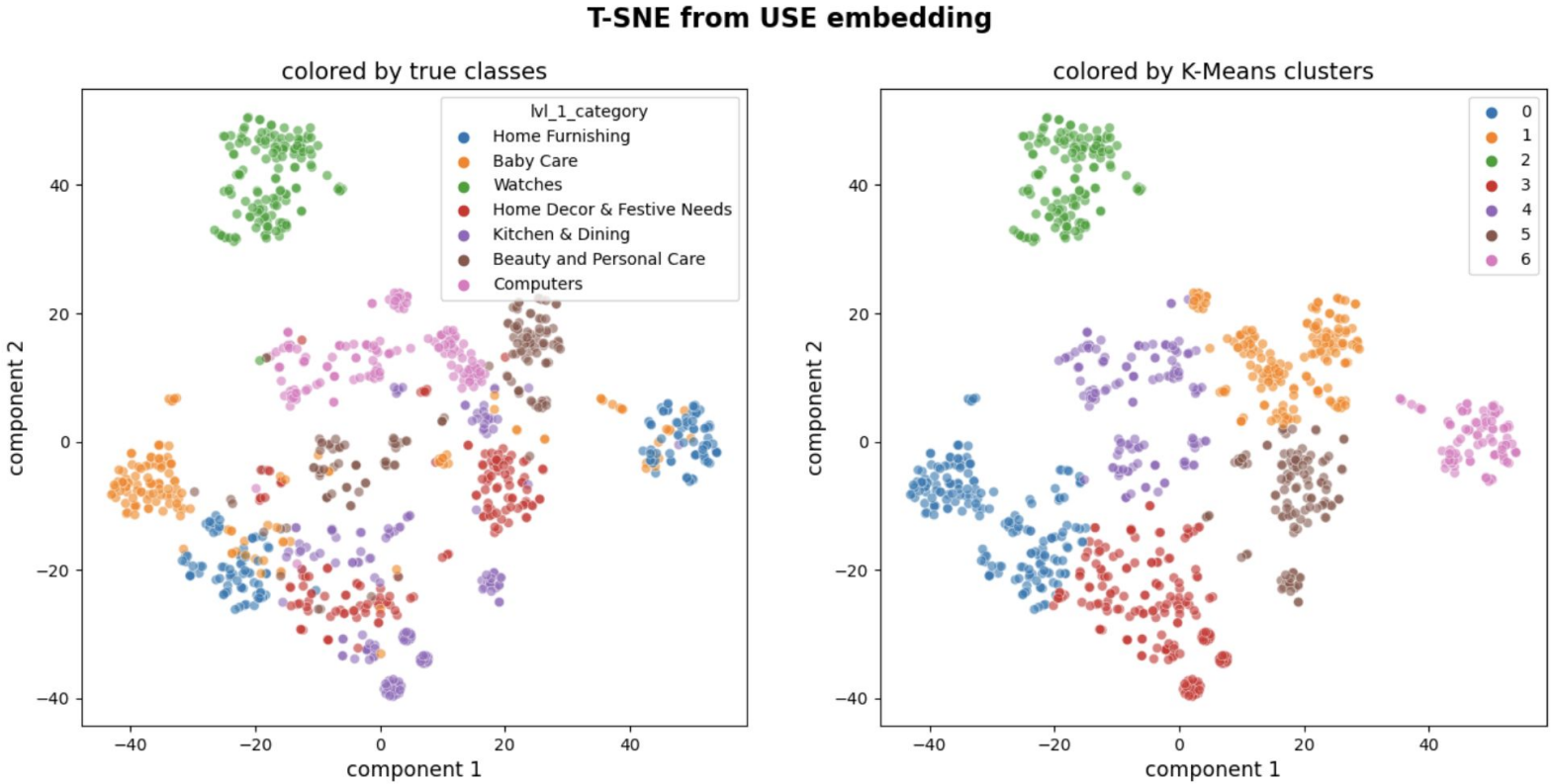
Étude de faisabilité > descriptions

ARI = 0.51



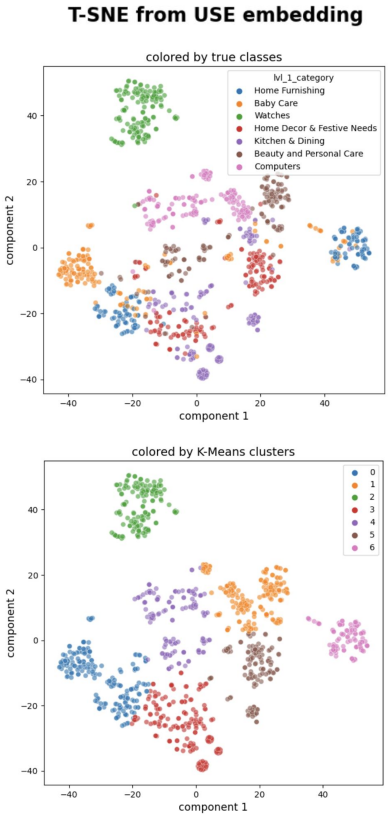


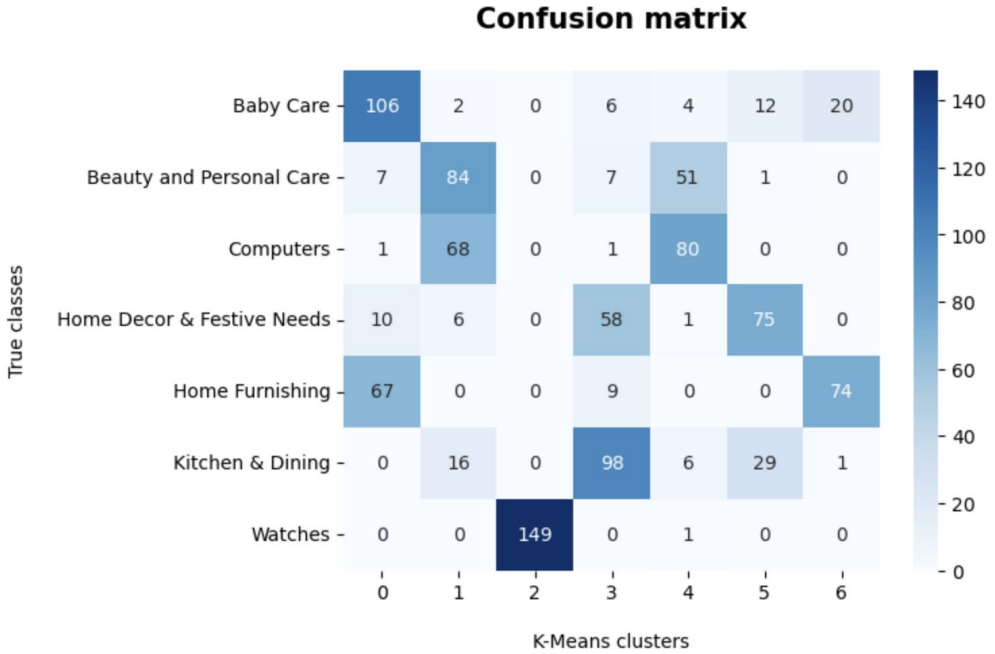
ARI = 0.44


T-SNE from USE embedding
colored by true classes
lvl_1_category
Home Furnishing
Baby Care
Watches
Home Decor & Festive Needs
Kitchen & Dining
Beauty and Personal Care
Computers
component 1
component 2
colored by K-Means clusters
0
1
2
3
4
5
6
component 1
component 2

Étude de faisabilité > descriptions

ARI = 0.44



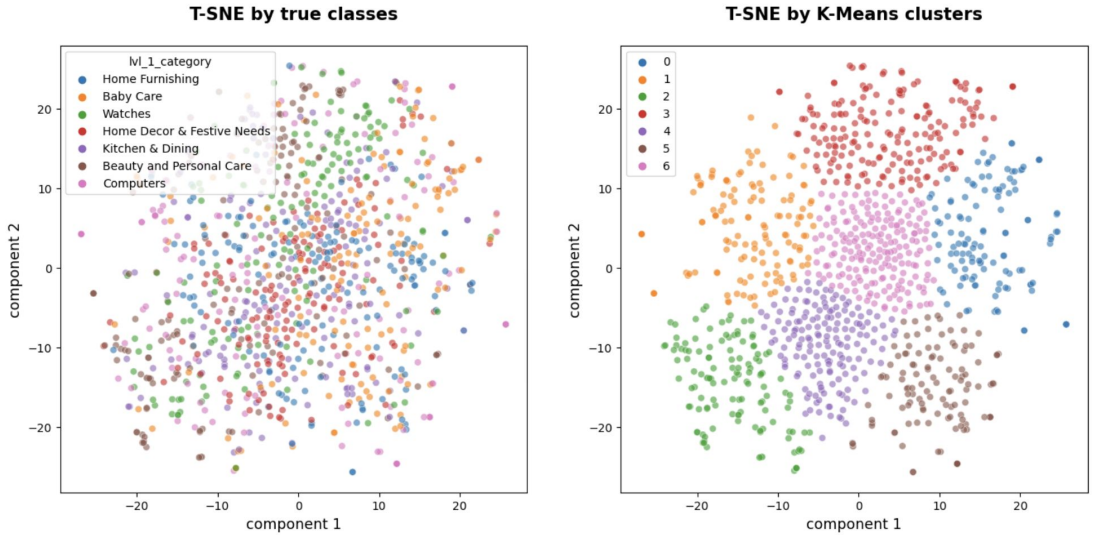


Étude de faisabilité > images

1ère approche, avec SIFT :

1. détection des keypoints et calcul des descriptors avec SIFT
2. calcul des visual words (centroïdes des classes distinguées par K-Means sur les descriptors concaténés)
3. pour chaque image, histogramme des visual words
4. PCA (conserve 99 % de la variance) puis T-SNE > réduction en 2 dimensions

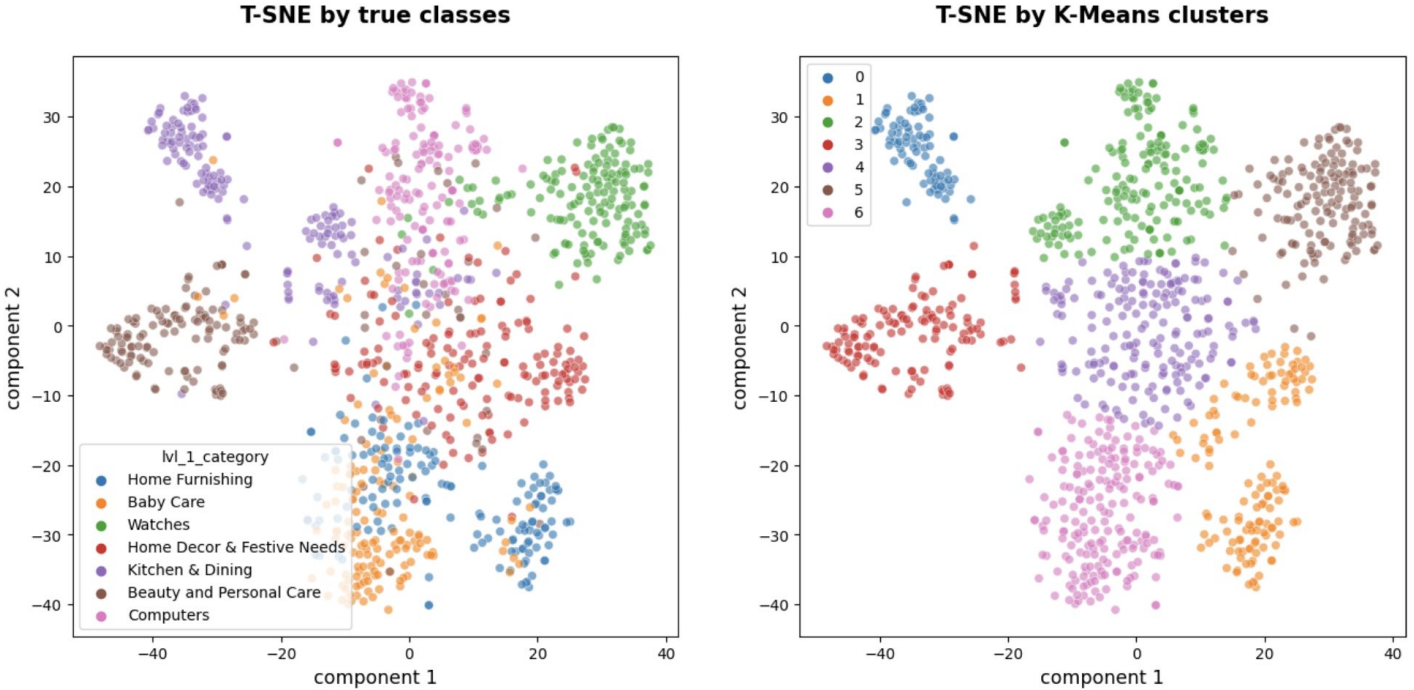
ARI = 0.05



Étude de faisabilité > images

ci-dessous : features extraites par VGG16

ARI = 0.45

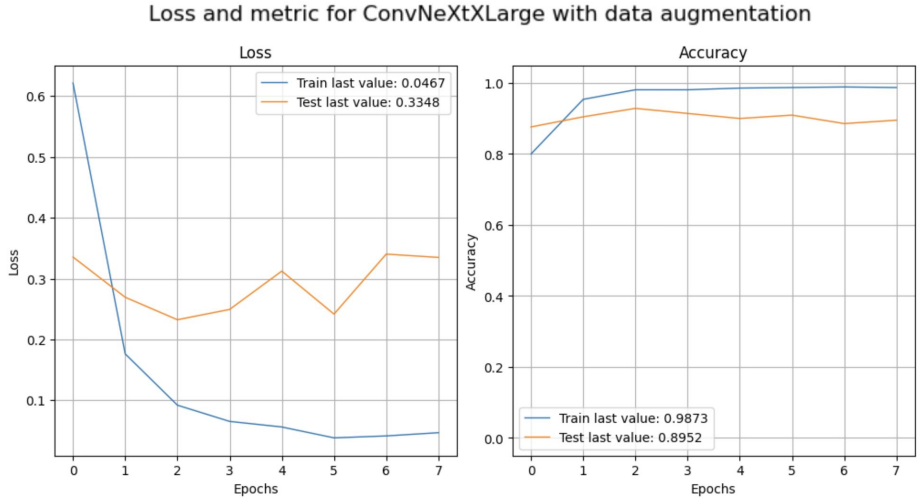


2.

Classification supervisée

Classification supervisée

- 3 modèles testés : VGG16 (2014), EfficientNetV2M (2021), ConvNeXtXLarge (2022)
  - *sur jeu de validation puis jeu de test*
  - *fonction de perte : categorical cross-entropy*
  - *métriques : accuracy, time (fit & predict)*
- utilisation des poids des réseaux pré-entraînés sur ImageNet
- chaque modèle testé sans, puis avec data augmentation



Classification supervisée > data augmentation

	base_model	indicator	without_data_augmentation	with_data_augmentation
0	VGG16	train_loss	0.2675	0.3402
1	VGG16	val_loss	0.9520	0.5741
2	VGG16	train_accuracy	0.9333	0.9032
3	VGG16	val_accuracy	0.8190	0.8190
4	VGG16	test_accuracy	0.7524	0.7571
5	VGG16	fit_time	50.0000	49.0000
6	EfficientNetV2M	train_loss	0.0879	0.1640
7	EfficientNetV2M	val_loss	0.3991	0.3738
8	EfficientNetV2M	train_accuracy	0.9794	0.9571
9	EfficientNetV2M	val_accuracy	0.8762	0.8762
10	EfficientNetV2M	test_accuracy	0.8667	0.8714
11	EfficientNetV2M	fit_time	34.0000	33.0000
12	ConvNeXtXLarge	train_loss	0.1439	0.0919
13	ConvNeXtXLarge	val_loss	0.2558	0.2324
14	ConvNeXtXLarge	train_accuracy	0.9603	0.9810
15	ConvNeXtXLarge	val_accuracy	0.9190	0.9286
16	ConvNeXtXLarge	test_accuracy	0.9143	0.9190
17	ConvNeXtXLarge	fit_time	545.0000	1582.0000

comparaison des résultats avec et sans augmentation des données

➔ objectif de l'augmentation : réduire l'overfitting

➔ chaque modèle testé sans, puis avec data augmentation

➔ transformations effectuées : inversions, rotations, zooms

➔ résultats :

- accuracy sur val & test sets : idem ou légère amélioration
- val loss : légère amélioration
- train accuracy : idem ou légère détérioration
- train loss : VGG16 et EfficientNetV2M : détérioration
  ConvNeXtXLarge : amélioration

Classification supervisée > résultats

	base_model	data_augmentation	train_loss	val_loss	train_accuracy	val_accuracy	test_accuracy	fit_time
0	VGG16	False	0.2675	0.9520	0.9333	0.8190	0.7524	50
1	VGG16	True	0.3402	0.5741	0.9032	0.8190	0.7571	49
2	EfficientNetV2M	False	0.0879	0.3991	0.9794	0.8762	0.8667	34
3	EfficientNetV2M	True	0.1640	0.3738	0.9571	0.8762	0.8714	33
4	ConvNeXtXLarge	False	0.1439	0.2558	0.9603	0.9190	0.9143	545
5	ConvNeXtXLarge	True	0.0919	0.2324	0.9810	0.9286	0.9190	1582

synthèse des résultats

➔ EfficientNetV2M + rapide et donne de meilleurs résultats que VGG16

➔ ConvNeXtXLarge :

- + lent
  - → *fit time 11 à 16 fois plus long (sans data augmentation)*
  - → *+ 500 sec*
- mais de meilleurs résultats sur tous les indicateurs
  - → *améliorés par l'augmentation des données*
  - → *test accuracy : + 21,3 % /VGG16, + 5,4 % /EfficientNetV2M*
  - → *val accuracy : + 13,3 % /VGG16, + 6 % /EfficientNetV2M*

3.

Collecte des données via l'API

Collecte des données — Edanam Food and Grocery Database API


```python
# download the data from the API
url = "https://edamam-food-and-grocery-database.p.rapidapi.com/api/food-database/v2/parser"

querystring = {"ingr":"champagne"}

headers = {
    "X-RapidAPI-Key": "722434ac2bmsh98aec1f033708a5p1afa15jsnd51bbdb6176b", # user personal key for the API
    "X-RapidAPI-Host": "edamam-food-and-grocery-database.p.rapidapi.com"
}

response = requests.get(url, headers=headers, params=querystring)
response = response.json()
```

code utilisé pour le test de collecte de données

- ➔ spécifier sa recherche de produits
  ex : indiquer « champagne » pour les produits dont les données contiennent ce mot
- ➔ un compte sur le site de l'API est requis (nécessité de renseigner son identifiant)


```
{'text': 'champagne',
 'parsed': [{'food': {'foodId': 'food_a656mk2a5dmqb2adiamu6beihduu',
    'uri': 'http://www.edamam.com/ontologies/edamam.owl#Food_table_white_wine',
    'label': 'Champagne',
    'knownAs': 'dry white wine',
    'nutrients': {'ENERC_KCAL': 82.0,
     'PROCNT': 0.07,
     'FAT': 0.0,
     'CHOCDF': 2.6,
     'FIBTG': 0.0},
```


aperçu des données collectées

➔ filtrer les données récupérées selon son besoin

	foodId	label	category	foodContentsLabel	image
0	food_a656mk2a5dmqb2adiamu6beihduu	Champagne	Generic foods	None	https://www.edamam.com/food-img/a71/a718cf3c52...
1	food_b753ithamdb8psbt0w2k9aquo06c	Champagne Vinaigrette, Champagne	Packaged foods	OLIVE OIL; BALSAMIC VINEGAR; CHAMPAGNE VINEGAR...	None
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...	None
4	food_an4jjueaucpus2a3u1ni8auhe7q9	Champagne Vinaigrette, Champagne	Packaged foods	WATER; CANOLA AND SOYBEAN OIL; WHITE WINE (CON...	None
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;...	None
7	food_byap67hab6evc3a0f9w1oag3s0qf	Champagne Sorbet	Generic meals	Sugar; Lemon juice; brandy; Champagne; Peach	None
8	food_am5egz6aq3fpjlaf8xpkdbc2asis	Champagne Truffles	Generic meals	butter; cocoa; sweetened condensed milk; vanil...	None
9	food_bcz8rhiajk1fuva0vkfmeakbouc0	Champagne Vinaigrette	Generic meals	champagne vinegar; olive oil; Dijon mustard; s...	None

dataframe contenant uniquement les données recherchées pour le test

Conclusion

- une catégorisation automatique est possible
  aussi bien à partir des textes que des images

- meilleurs résultats obtenus avec ConvNeXt
  accuracy = 0.98, 0.93, 0.92 sur train, val, test datasets

- test favorable de la collecte de données via la Edanam Food and Grocery Database API