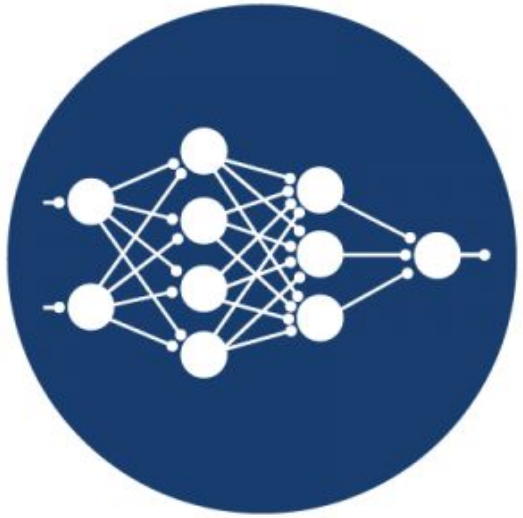


Projet 6 - Openclassrooms
Classifier automatiquement des biens de consommation

Dabidin Keshika
03.07.2023

OPENCLASSROOMS

Plan

1. Contexte de l'étude
2. Présentation des données
3. Prétraitement des données textuelles
4. Classification des données textuelles
5. Prétraitement des données d'images
6. Classification des données d'images
7. Classification supervisée des données
8. Récupération des données API
9. Conclusion

Introduction

Contexte :

"Place de marché" souhaite lancer une marketplace e-commerce.

Sur la place de marché, des vendeurs proposent des articles à des acheteurs en postant une photo et une description.

Pour l'instant, l'attribution de la catégorie d'un article est effectuée manuellement par les vendeurs et est donc peu fiable. De plus, le volume des articles est pour l'instant très petit.

Objectifs:

- Automatiser l'attribution des catégories afin de faciliter la mise en ligne des nouveaux produits et rendre l'expérience plus fluide.

- Linda, lead data scientist, demande donc d'étudier la faisabilité d'un moteur de classification des articles en différentes catégories, avec un niveau de précision suffisant.

Présentation des données

Résumé :

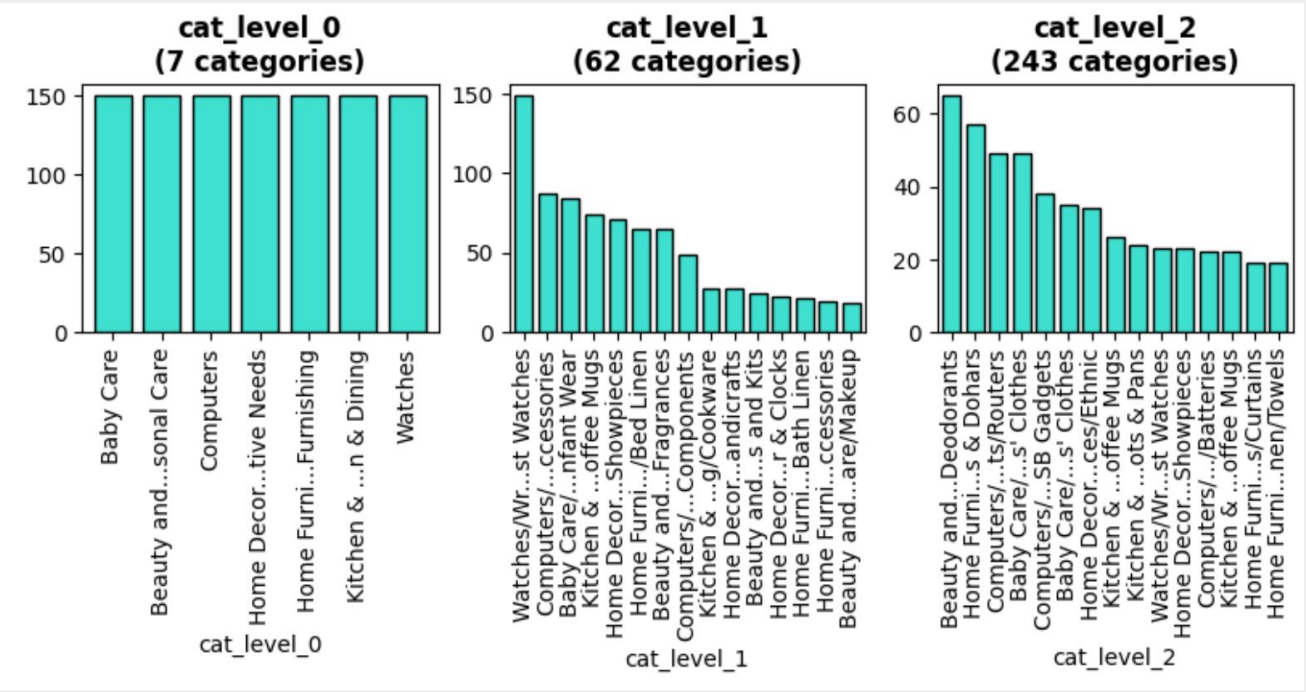
- 2 types de données : fichier texte et dossier contenant des images.

	Nombre lignes	Nombre colonnes	Taux remplissage moyen	Doublons
Fichier Texte	1050	14	97.7%	0

	Pourcentage Valeurs Manquantes	Nombre de valeurs manquantes
brand	32.19	338
retail_price	0.10	1
discounted_price	0.10	1
product_specifications	0.10	1

Remplacer les marques non connues par 'Unknown' ou des valeurs manquantes dans les colonnes numériques par la valeur médiane.

Présentation des données



Le seul niveau qui a un ensemble équilibré au niveau des éléments est le niveau 0, avec 7 catégories.

Conservation de ces 7 catégories uniquement.

Prétraitement des données textuelles

```
"Specifications of 612 League Baby Boy's Checkered Casual Shirt General Details Pattern Checkered Occasion Casual Ideal For Bab
y Boy's Shirt Details Sleeve Half Sleeve Number of Contents in Sales Package Pack of 1 Brand Fit Regular Fabric 100% COTTON Fit
Regular Additional Details Style Code BLS00S380001B Fabric Care ENZYME WASH"
```

- **Tokénisation** : découpe les mots d'une phrase et conserve uniquement les caractères alphanumériques

```
['Specifications',
 'of',
 'League',
```

- **Word filter et Stop words :** suppression des mots vides et des lettres simples, garder que certains tags et enlever les mots courants ne servant pas à classifier les produits.

```
['Specifications',
 'League',
 'Baby',
```

- **Minuscule** (.lower())

```
['specifications',
 'league',
 'baby',
```

Prétraitement des données textuelles

- **Lemmatisation** : trouver l'origine des mots ex. Studies --> Study

```
['specification',
 'league',
 'baby',
```

- **Stemming** : couper des suffixes pour réduire les mots ex. Studies --> Studi

```
['specif',
 'leagu',
 'babi',
```

Classification des données textuelles

Bag of Words (BoW)

Un bag of words ou BoW est une représentation de texte qui décrit l'occurrence de mots dans un document.

L'intuition est que les documents sont similaires s'ils ont un contenu similaire. De plus, à partir du contenu seul, nous pouvons apprendre quelque chose sur la signification du document.

Les étapes pour la création d'un BoW sont les suivantes :

- Faire un modèle de mots uniques et significatifs lié au jeu de données
- Créer des vecteurs de mots dans chaque document. Si le mot du modèle est présent dans le document on compte 1, s'il est absent on compte 0. On obtient ainsi un vecteur de mots pour chaque phrase. La longueur du vecteur document est égale au nombre de mots connus dans le document.

On peut compter les mots de deux façons notamment :

- Comptes : Compter le nombre de fois que chaque mot apparaît dans un document.
- Fréquences: Calculer la fréquence à laquelle chaque mot apparaît dans un document parmi tous les mots du document.

Classification des données textuelles

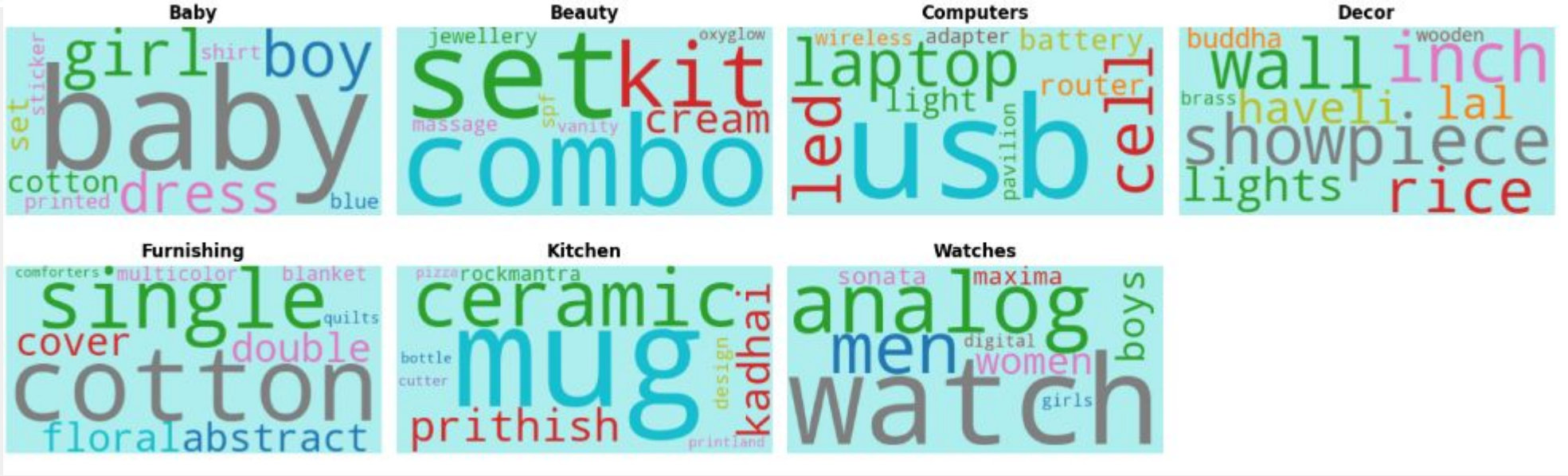
Bag of Words (BoW) - Exemple

```
'elegance polyester multicolor abstract eyelet door curtain'
```

Application BoW : CountVectorizer

	aapno	aapno rajasthan	aari	aari embroidered	aarika	aarika self	abkl	abkl grn	abkl pl	abstract	abstract abstract	abstract checkered	abstract colors	abstract cushions	abstract design	abstract double	abstract eyelet
0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	1

	abstract	abstract eyelet	curtain	door	door curtain	elegance	elegance polyester	eyelet	eyelet door	multicolor	multicolor abstract	polyester	polyester multicolor
0	1	1	1	1	1	1	1	1	1	1	1	1	1



Classification des données textuelles

TF - IDF (Term Frequency - Inverse Document Frequency)

Cette approche consiste à redimensionner la fréquence des mots en fonction de leur fréquence d'apparition dans tous les documents, de sorte que les scores des mots fréquents comme « le » qui sont également fréquents dans tous les documents soient pénalisés.

- Fréquence du terme : est une notation de la fréquence du mot dans le document actuel.
- Inverse Document Frequency : est une notation de la rareté du mot dans les documents.

Les scores sont une pondération où tous les mots ne sont pas aussi importants ou intéressants et ont pour effet de mettre en évidence des mots distincts (contenant des informations utiles) dans un document donné.

Application sur l'exemple :

	abstract	curtain	door	elegance	eyelet	multicolor	polyester
0	0.319227	0.360825	0.376344	0.503739	0.364408	0.33463	0.357408

Classification des données textuelles

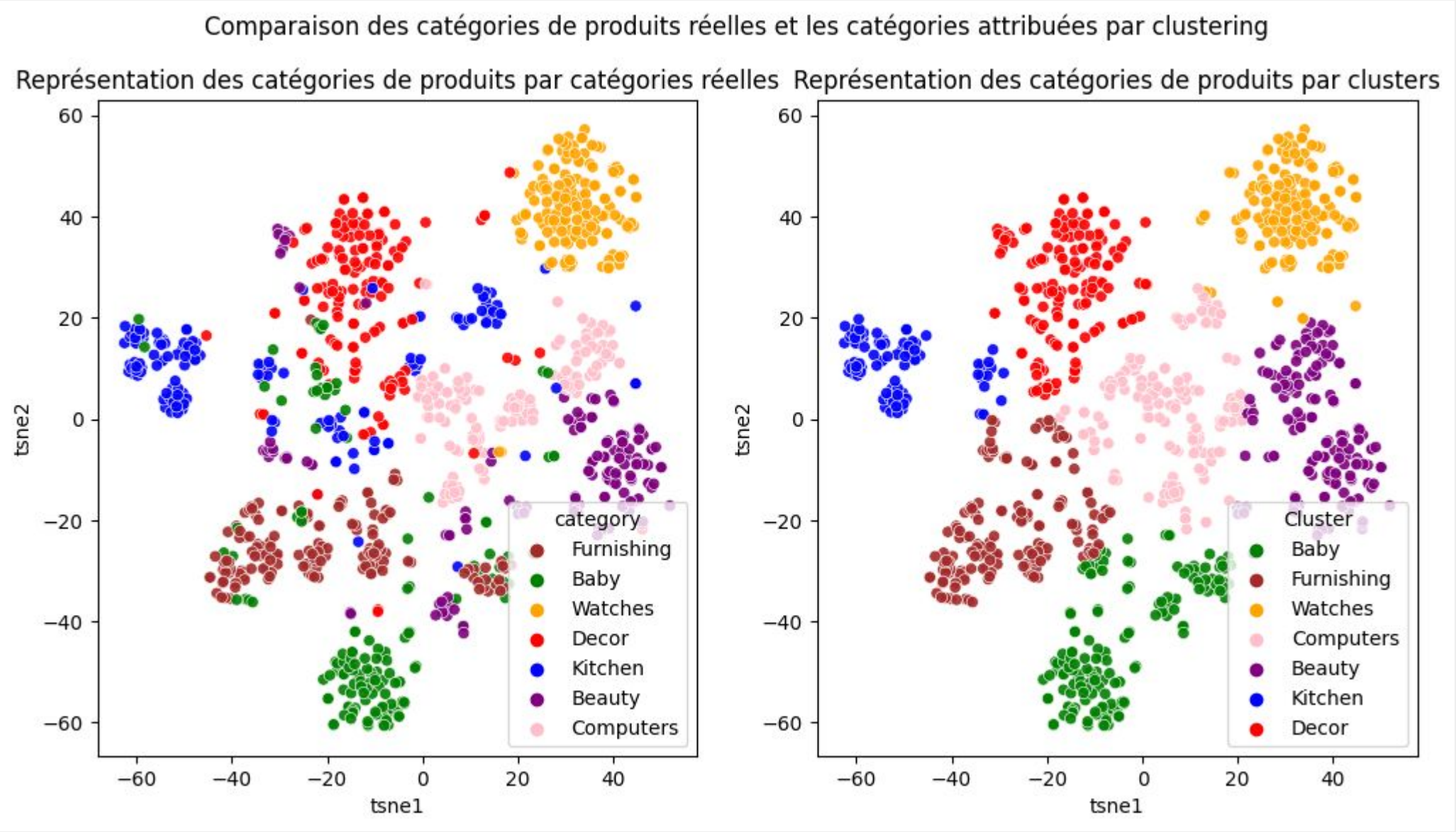
Plusieurs approches - BoW et TF-IDF avec le fit et transform uniquement sur « product_name » + « description »

Évaluation des modèles BoW et TF-IDF : Score ARI (Adjusted Rand Index)

	BoW	TF-IDF
ARI	0.4619	0.5054
Temps	10	9

	Furnishing (%)	Baby (%)	Watches (%)	Decor (%)	Kitchen (%)	Beauty (%)	Computers (%)
0	0.000000	0.000000	98.666667	2.666667	2.666667	0.000000	2.666667
1	0.000000	3.333333	0.000000	2.666667	56.666667	0.000000	0.000000
2	74.666667	8.666667	0.000000	0.666667	11.333333	9.333333	0.000000
3	0.000000	5.333333	0.000000	1.333333	2.666667	67.333333	35.333333
4	0.666667	9.333333	0.000000	78.666667	2.666667	7.333333	1.333333
5	2.666667	2.666667	1.333333	12.666667	23.333333	4.666667	60.666667
6	22.000000	70.666667	0.000000	1.333333	0.666667	11.333333	0.000000

Classification des données textuelles



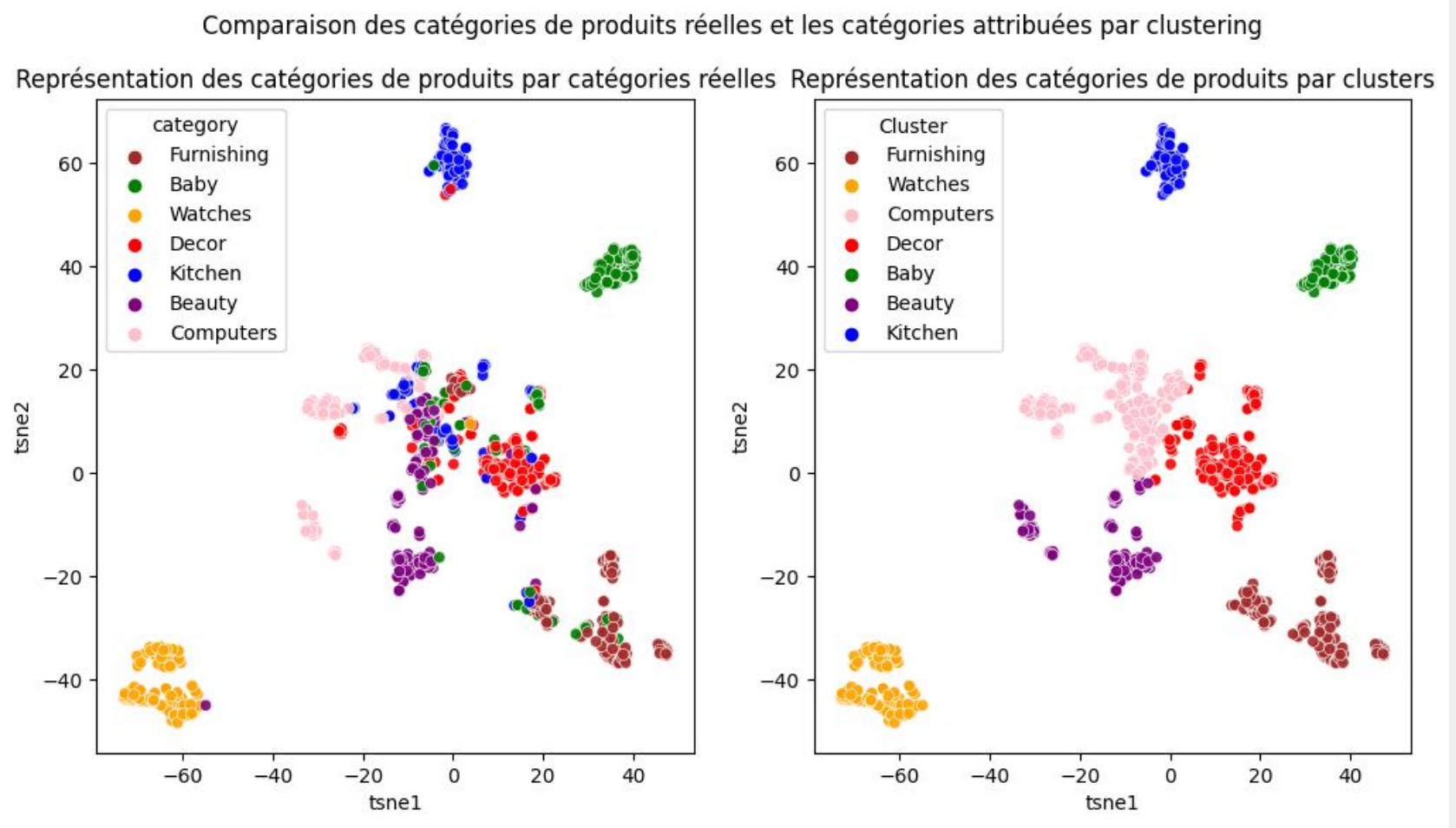
Classification des données textuelles

Word2vec (Word Embedding) avec noms des produits

Cette approche désigne un ensemble de méthode d'apprentissage visant à représenter les mots d'un texte par des vecteurs de nombres réels. Le word embedding est capable, en réduisant la dimension, de capturer le contexte, la similarité sémantique et syntaxique (genre, synonymes, ...) d'un mot.

ARI score : 0.4811

Temps : 8.0



Classification des données textuelles

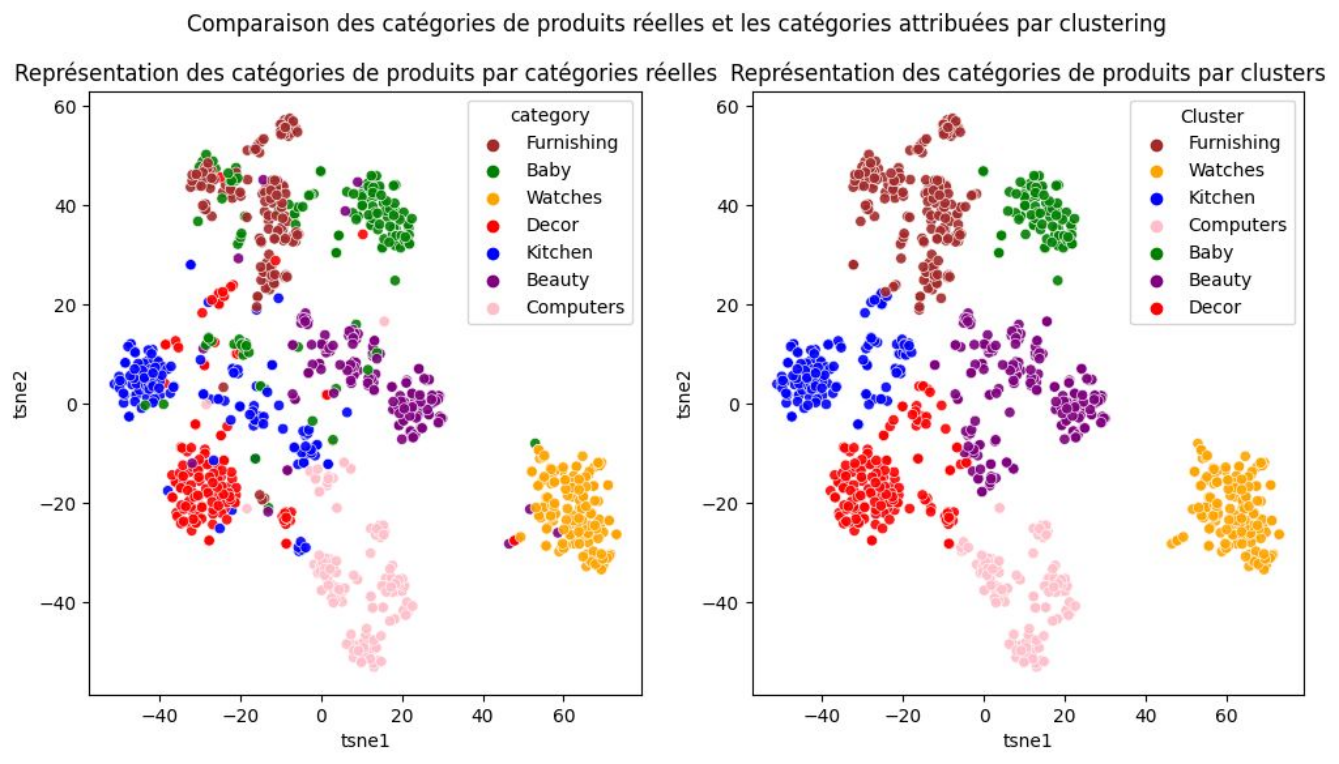
BERT - Modèle pré-entraîné (ARI score bas)

USE (Universal sentence Encoder)

Cette méthode encode le texte en vecteurs de grande dimension qui peuvent être utilisés pour la classification de texte, la similarité sémantique, le regroupement et d'autres tâches en langage naturel. L'encodeur de phrases universel pré-formé est disponible publiquement dans Tensorflow-hub.

ARI score : 0.6387

Temps : 8.0



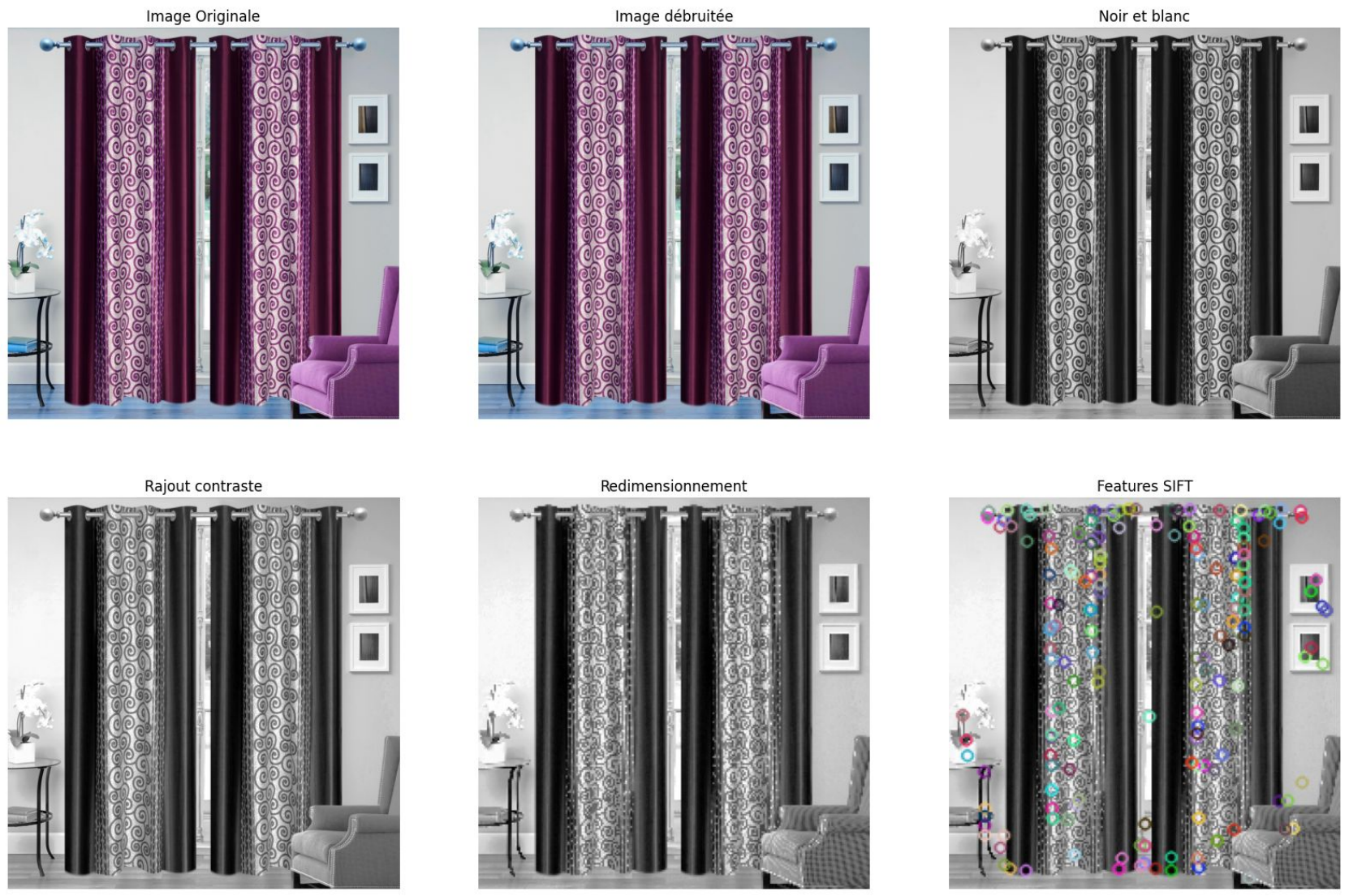
	Furnishing (%)	Baby (%)	Watches (%)	Decor (%)	Kitchen (%)	Beauty (%)	Computers (%)
0	2.666667	3.333333	0.0	76.666667	20.000000	2.666667	0.666667
1	0.000000	0.666667	100.0	2.000000	0.000000	2.000000	0.000000
2	0.000000	56.666667	0.0	0.666667	0.000000	1.333333	0.000000
3	0.000000	5.333333	0.0	1.333333	13.333333	92.000000	12.000000
4	0.666667	10.666667	0.0	16.000000	59.333333	0.666667	0.666667
5	0.000000	0.000000	0.0	0.000000	4.666667	0.000000	86.666667
6	96.666667	23.333333	0.0	3.333333	2.666667	1.333333	0.000000

Prétraitement des données d'images

SIFT/ORB - Pour chaque image:

- Création d'une liste de descripteurs par image ("sift_keypoints_by_img") qui sera utilisée pour réaliser les histogrammes par image
- Création d'une liste de descripteurs pour l'ensemble des images ("sift_keypoints_all") qui sera utilisé pour créer les clusters de descripteurs.
- Prédiction des numéros de cluster de chaque descripteur via la création d'un histogramme = comptage pour chaque numéro de cluster du nombre de descripteurs de l'image.
- Features d'une image = Histogramme d'une image = Comptage pour une image du nombre de descripteurs par cluster
- Réduction en dimension avec la PCA

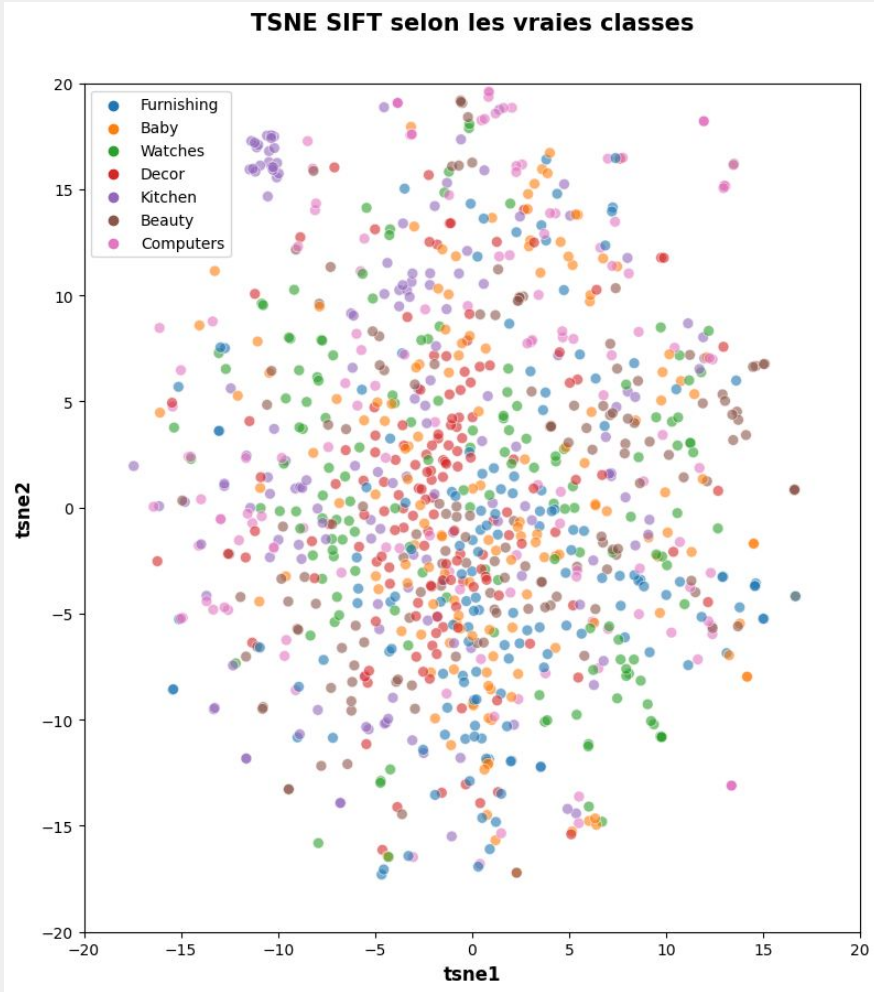
```
Dimensions dataset SIFT avant réduction PCA :  (1050, 590)
Dimensions dataset SIFT après réduction PCA :  (1050, 474)
Dimensions dataset  ORB avant réduction PCA :  (1050, 619)
Dimensions dataset  ORB après réduction PCA :  (1050, 510)
```

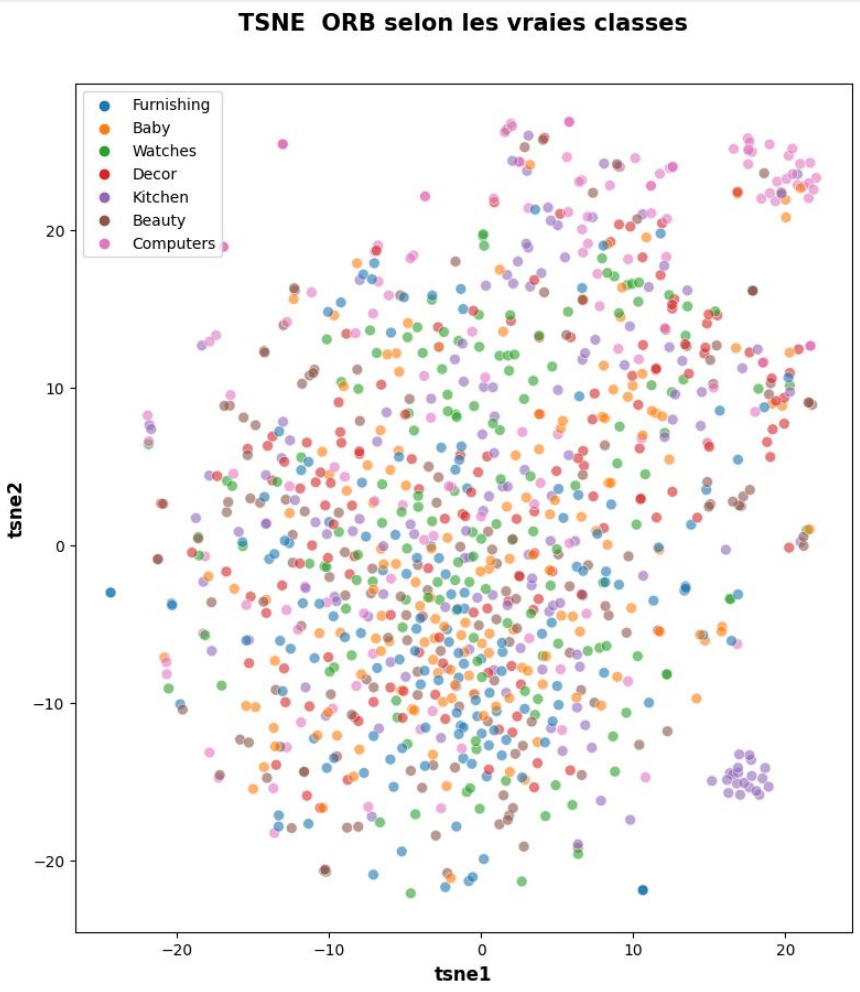


Classification des données d'images

SIFT/ORB -

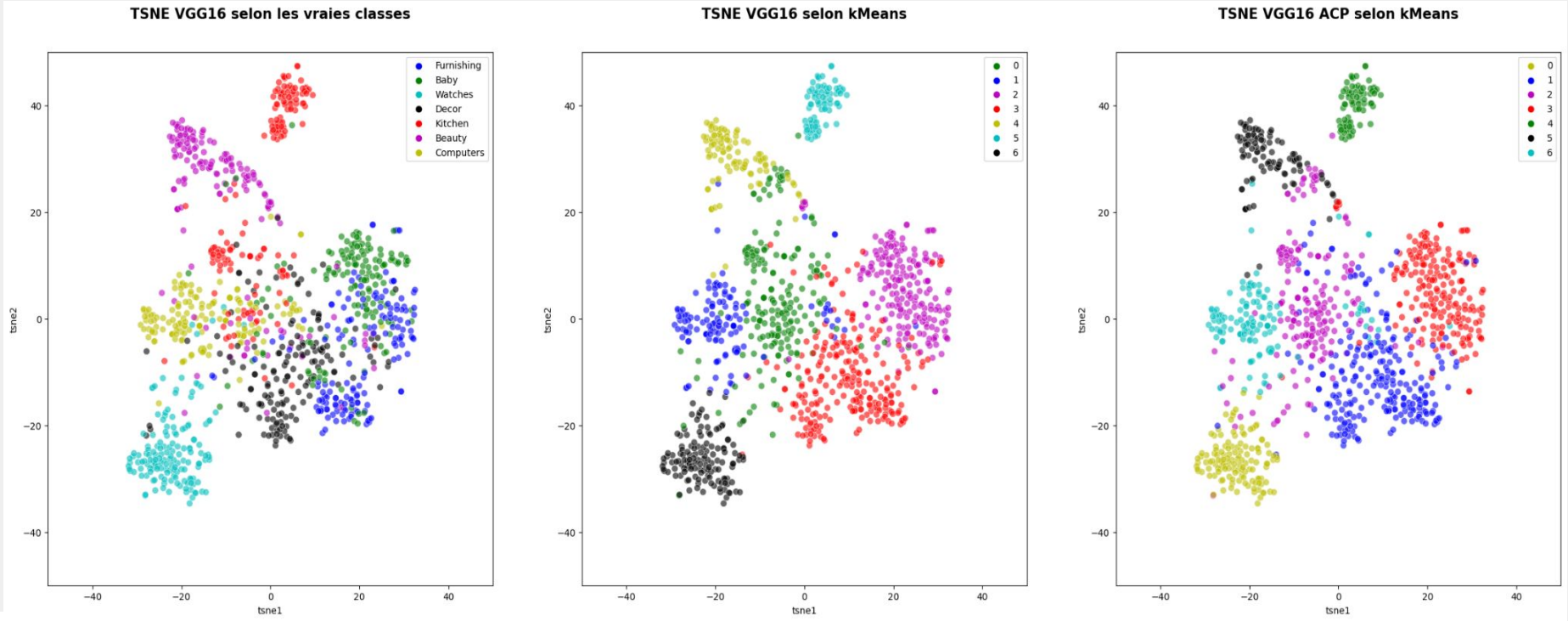
```
ARI BOW: 0.015
ARI TFIDF: 0.024
```

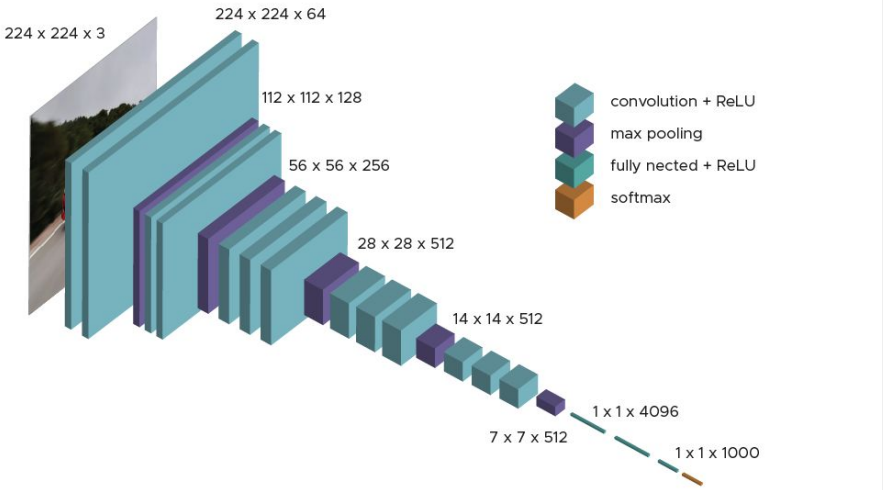




Classification des données d'images

Modèle pré-entraîné - VGG 16

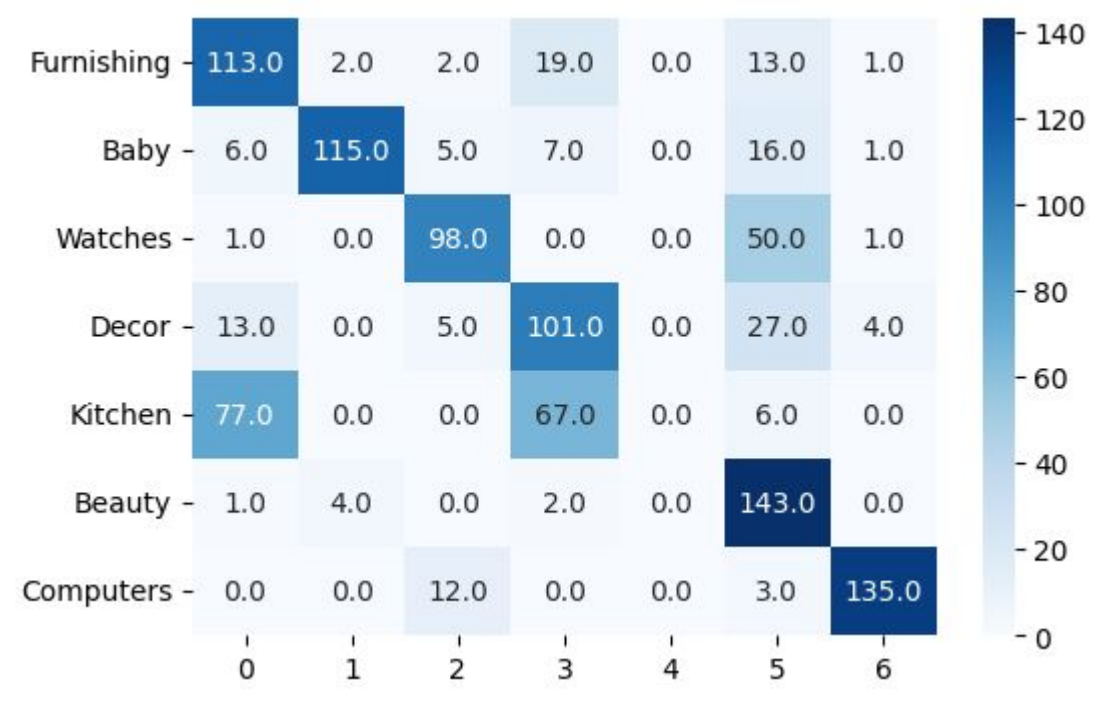





```
ARI     : 0.440
ARI ACP : 0.437
```

Classification des données d'images

Modèle pré-entraîné - VGG 16 non supervisé



	precision	recall	f1-score	support
0	0.54	0.75	0.63	150
1	0.95	0.77	0.85	150
2	0.80	0.65	0.72	150
3	0.52	0.67	0.58	150
4	0.00	0.00	0.00	150
5	0.55	0.95	0.70	150
6	0.95	0.90	0.92	150
accuracy			0.67	1050
macro avg	0.62	0.67	0.63	1050
weighted avg	0.62	0.67	0.63	1050

$$precision = \frac{True\ Positive}{True\ Positive + False\ Positive}$$

$$F1\ Score = 2 \times \frac{recall \times precision}{recall + precision}$$

$$recall = \frac{True\ Positive}{True\ Positive + False\ Negative}$$

Classification supervisée des données d'images

Modèle pré-entraîné - VGG 16 supervisé

- Prétraitement des images
- Split des données (Train - 75% /Test - 25%)
- Entraînement avec le modèle
- Évaluation du modèle

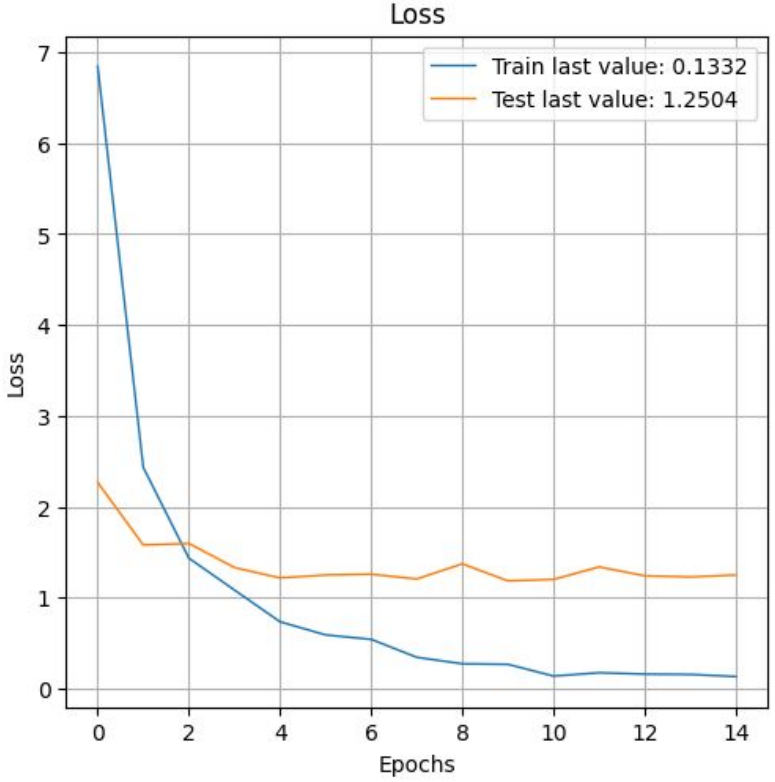
Différentes Approches :

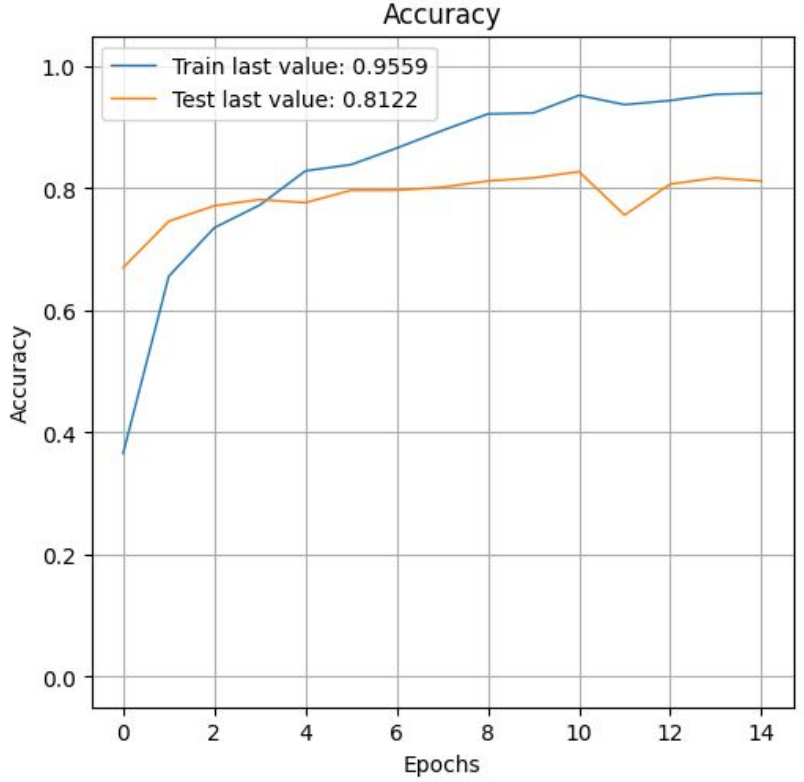
- Approche simple par préparation initiale de l'ensemble des images avant classification supervisée
- Approche par data generator, permettant facilement la data augmentation. Les images sont directement récupérées à la volée dans le répertoire des images
- Approche récente proposée par Tensorflow.org par DataSet, sans data augmentation
- Approche par DataSet, avec data augmentation intégrée au modèle : layer en début de modèle

Classification supervisée des données d'images

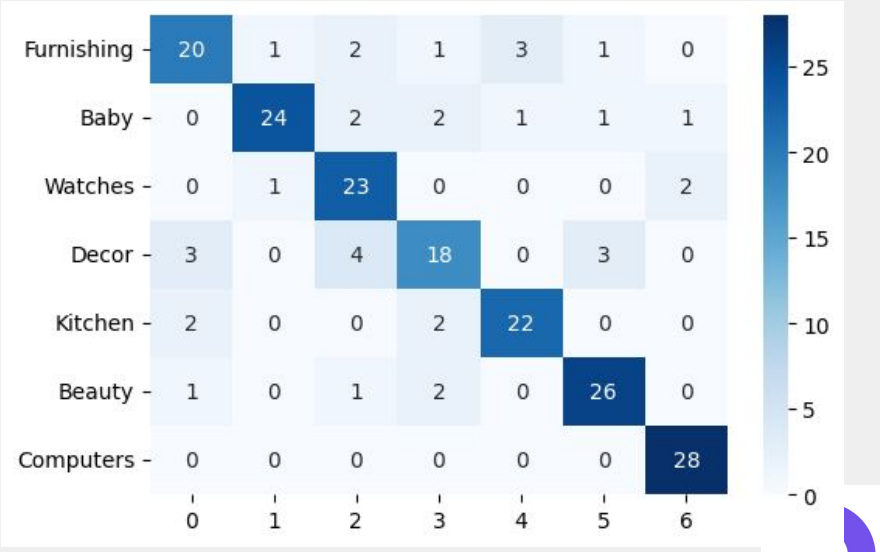
Approche simple par préparation initiale de l'ensemble des images avant classification supervisée

Temps de traitement VGG descriptor : 5018.80 secondes



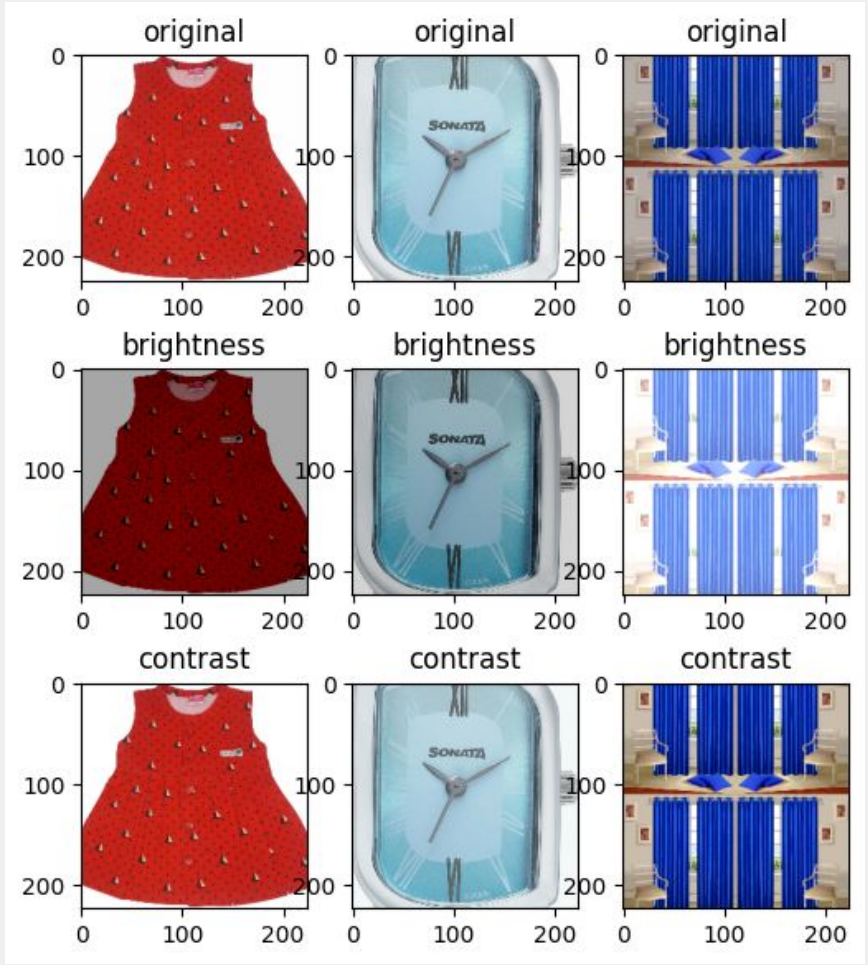


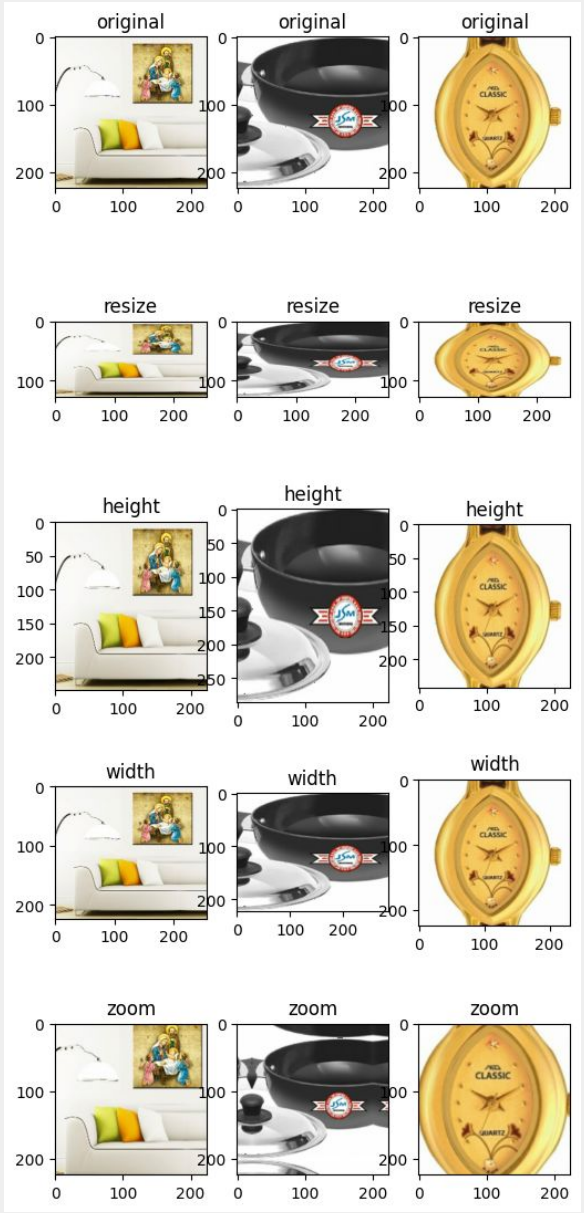
	precision	recall	f1-score	support
0	0.77	0.71	0.74	28
1	0.92	0.77	0.84	31
2	0.72	0.88	0.79	26
3	0.72	0.64	0.68	28
4	0.85	0.85	0.85	26
5	0.84	0.87	0.85	30
6	0.90	1.00	0.95	28
accuracy			0.82	197
macro avg	0.82	0.82	0.81	197
weighted avg	0.82	0.82	0.82	197

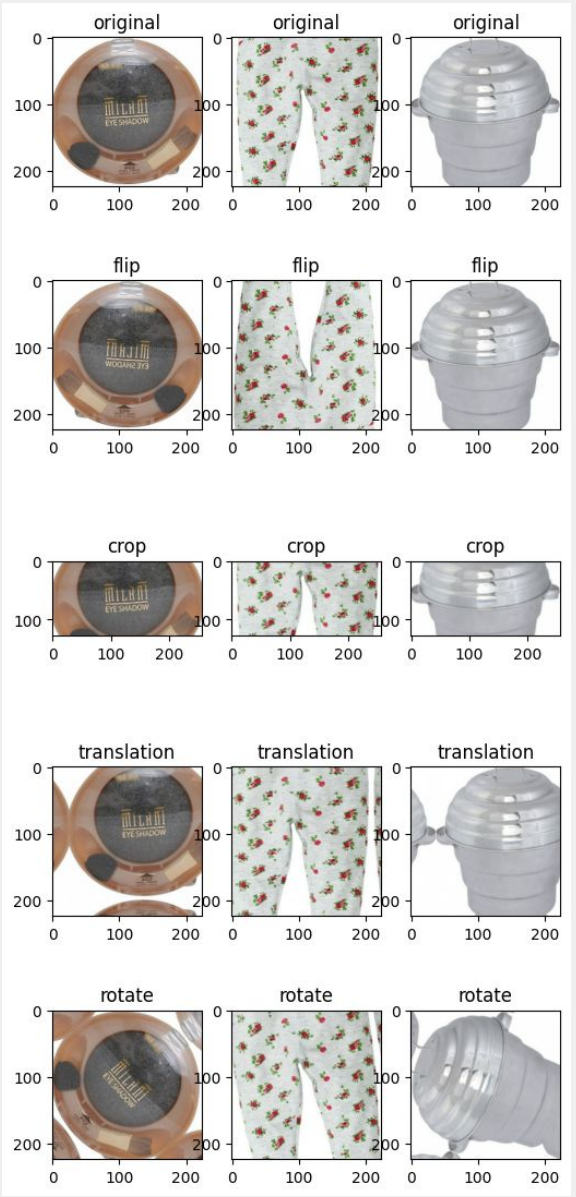


Classification supervisée des données d'images

Approche par ImageDataGenerator (Data Augmentation)



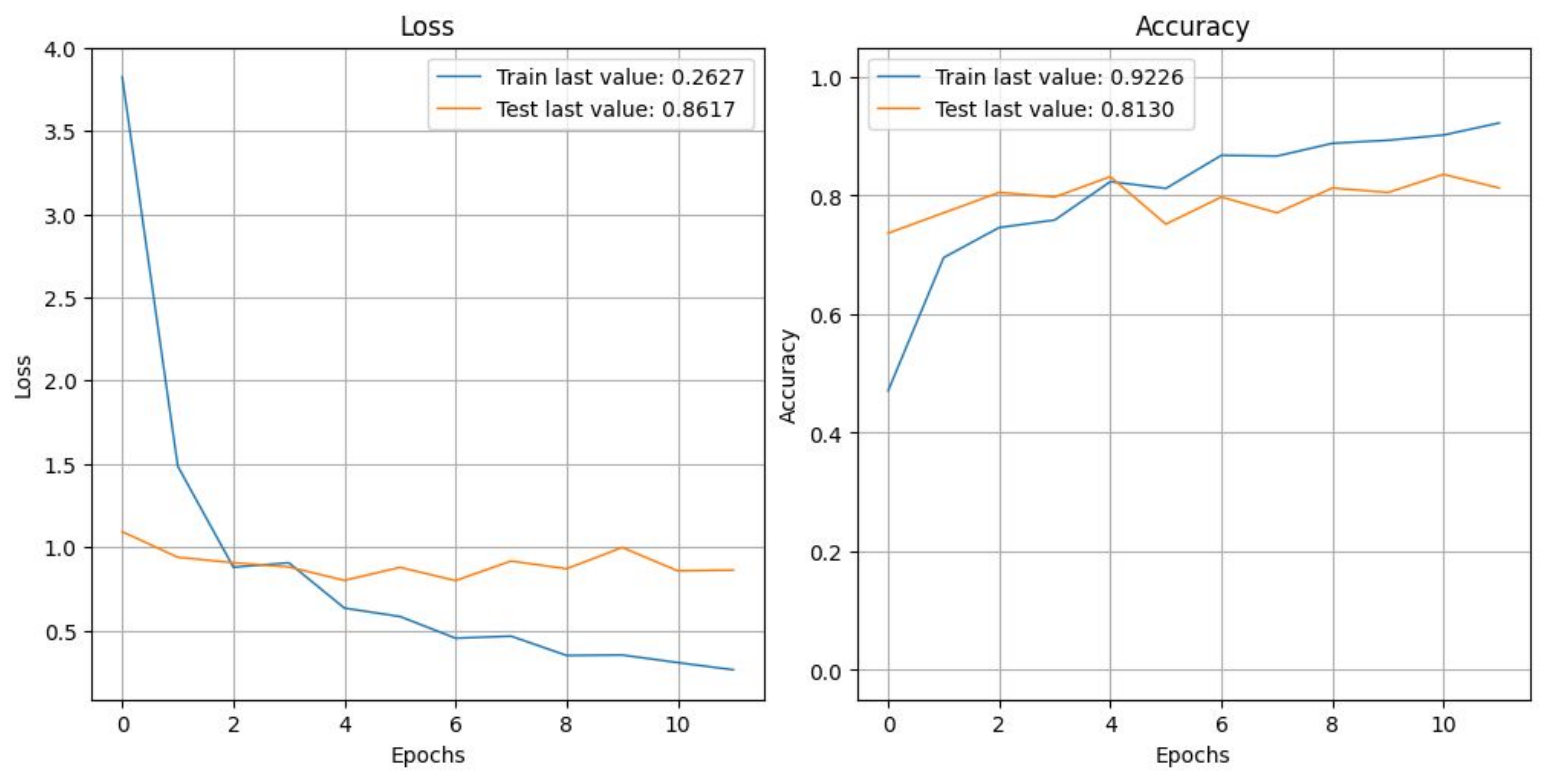




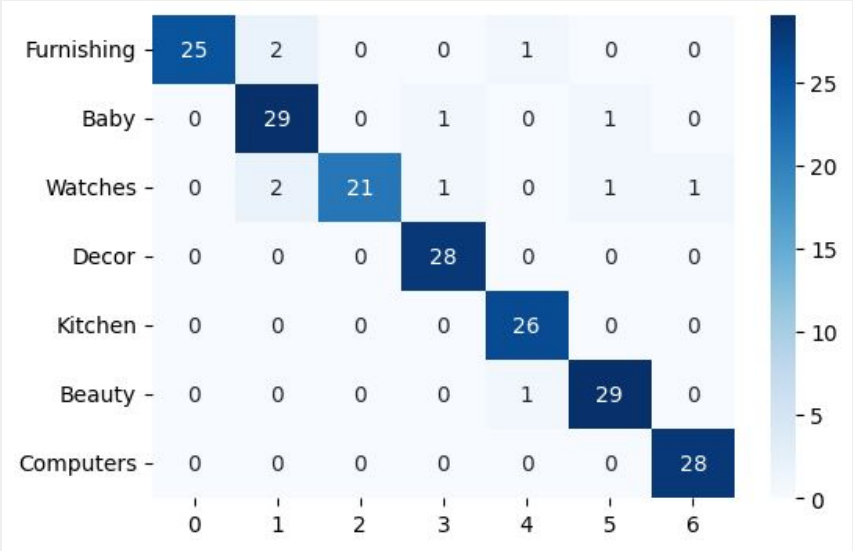
Classification supervisée des données d'images

Approche par ImageDataGenerator (Data Augmentation)

Temps de traitement VGG descriptor : 10372.43 secondes



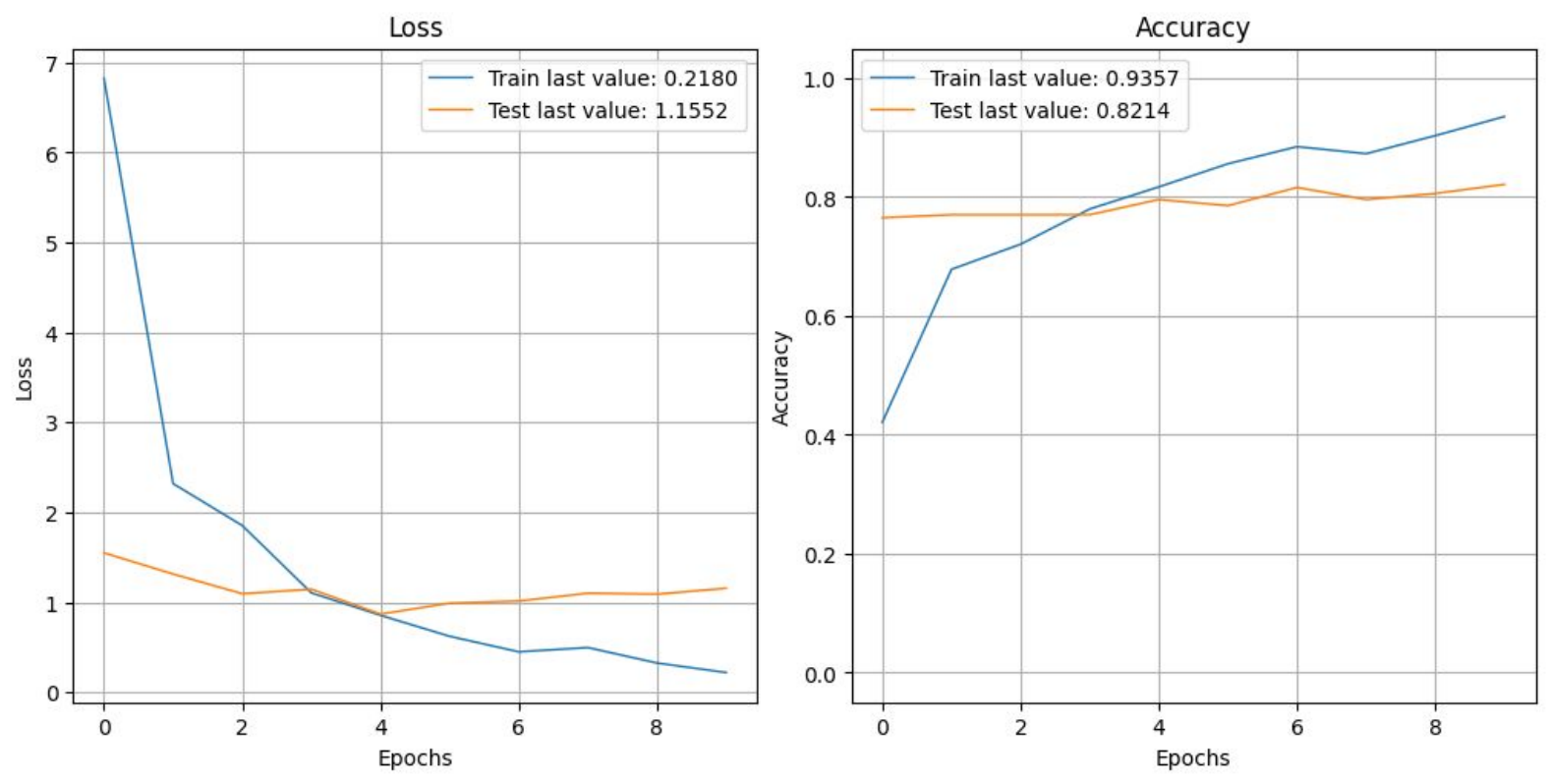
	precision	recall	f1-score	support
0	1.00	0.89	0.94	28
1	0.88	0.94	0.91	31
2	1.00	0.81	0.89	26
3	0.93	1.00	0.97	28
4	0.93	1.00	0.96	26
5	0.94	0.97	0.95	30
6	0.97	1.00	0.98	28
accuracy			0.94	197
macro avg	0.95	0.94	0.94	197
weighted avg	0.95	0.94	0.94	197



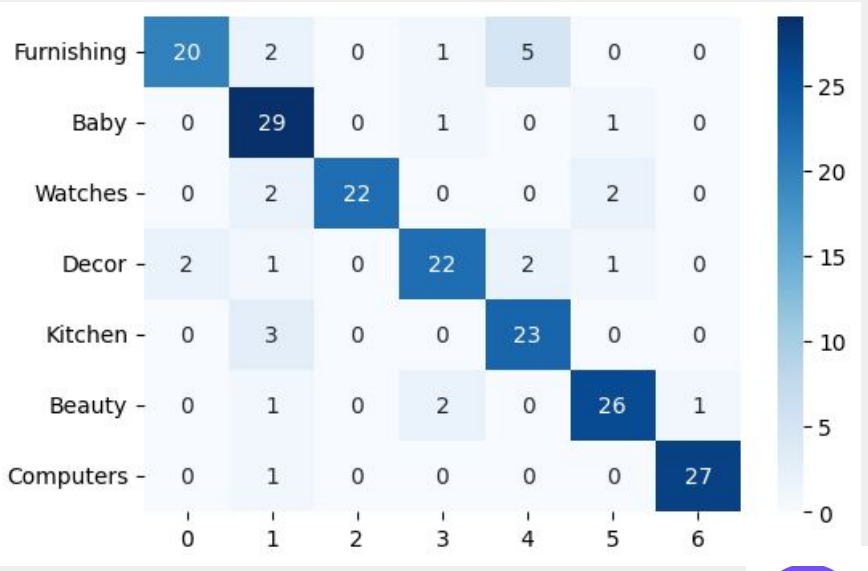
Classification supervisée des données d'images

Approche récente proposée par Tensorflow.org par DataSet, sans data augmentation

Temps de traitement VGG descriptor : 1843.02 secondes



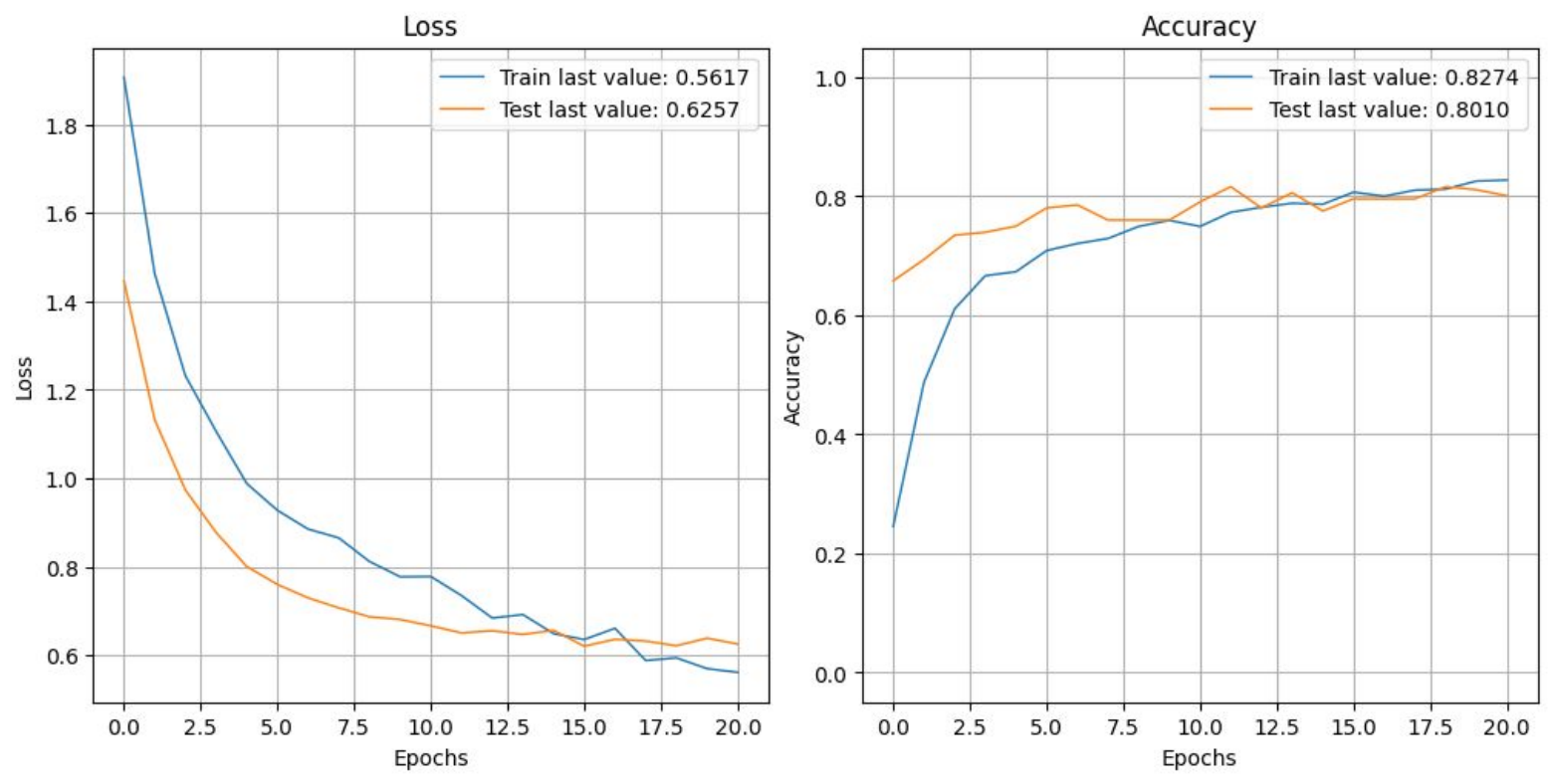
	precision	recall	f1-score	support
0	0.91	0.71	0.80	28
1	0.74	0.94	0.83	31
2	1.00	0.85	0.92	26
3	0.85	0.79	0.81	28
4	0.77	0.88	0.82	26
5	0.87	0.87	0.87	30
6	0.96	0.96	0.96	28
accuracy			0.86	197
macro avg	0.87	0.86	0.86	197
weighted avg	0.87	0.86	0.86	197



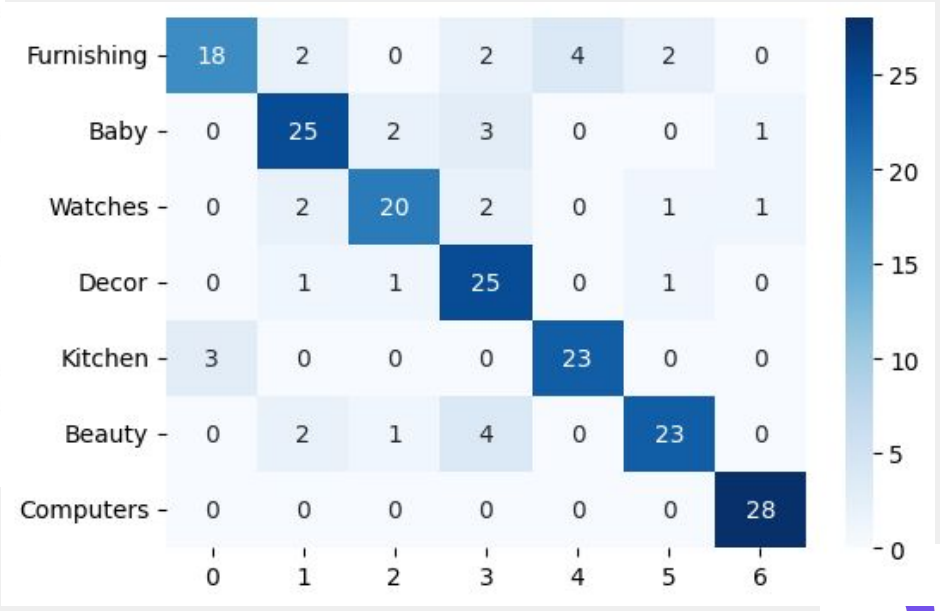
Classification supervisée des données d'images

Approche par DataSet, avec data augmentation intégrée au modèle

Temps de traitement VGG descriptor : 3986.17 secondes



	precision	recall	f1-score	support
0	0.86	0.64	0.73	28
1	0.78	0.81	0.79	31
2	0.83	0.77	0.80	26
3	0.69	0.89	0.78	28
4	0.85	0.88	0.87	26
5	0.85	0.77	0.81	30
6	0.93	1.00	0.97	28
accuracy			0.82	197
macro avg	0.83	0.82	0.82	197
weighted avg	0.83	0.82	0.82	197



Classification supervisée des données d'images

Tableau Résumé comparant les différentes approches du modèle supervisé :

Modèle	Temps d'entraînement (secondes)	F1-Score	Loss Train	Loss Test	Accuracy Train	Accuracy Test	Différence Accuracy Train/Test
VGG Simple	5018.80	0.82	0.1332	1.2504	0.9559	0.8122	0.1437
VGG ImageDataGenerator	10372.43	0.94	0.2627	0.8617	0.9226	0.8130	0.1096
VGG DataSet	1843.02	0.86	0.2180	1.1552	0.9357	0.8214	0.1143
VGG DataSet avec Data Augmentation	3986.17	0.82	0.5617	0.6257	0.8214	0.8010	0.0204

Choix entre VGG ImageDataGenerator et VGG DataSet avec Data Augmentation ⟶ Temps d'entraînement déterminant.

Récupération des données API

- **Site web :** https://rapidapi.com/edamam/api/edamam-food-and-grocery-database
- Flatten dictionary records
- Récupérer les produits à base de “Champagne”
- Choisir et nommer les colonnes

	foodId	label	foodContentsLabel	image
0	food_a656mk2a5dmqb2adiamu6beihduu	Champagne	NaN	https://www.edamam.com/food-img/a71/a718cf3c52...
1	food_b753ithamdb8psbt0w2k9aquo06c	Champagne Vinaigrette, Champagne	OLIVE OIL; BALSAMIC VINEGAR; CHAMPAGNE VINEGAR...	NaN
2	food_b3dyababjo54xobm6r8jzbghjgqe	Champagne Vinaigrette, Champagne	INGREDIENTS: WATER; CANOLA OIL; CHAMPAGNE VINE...	https://www.edamam.com/food-img/d88/d88b64d973...

- Sauvegarder les 10 premiers produits

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

Conclusion

1. Étude sur fichiers images et données textuelles
2. Classification des données avec le texte : Faisable - meilleur résultats du score ARI avec USE
3. Classification des fichiers images : Non faisable avec du clustering non supervisé en utilisant SIFT/ORB mais faisable avec un algorithme de réseau de neurones VGG 16
4. Classification supervisée des données :

   - Étude avec ou sans data augmentation

   - Paramètres d'évaluation : F1-Score, Temps d'entraînement, Loss et Accuracy (Train et Test)

   - Meilleur modèle à continuer à améliorer : VGG16 DataSet avec Data Augmentation.

5. Apprentissage de récupération des données sur une API et extraction des données utiles en fichier .csv

Merci pour votre attention.

OPENCLASSROOMS