

OpenClassrooms



PROJET 6

Classifier automatiquement des biens de consommations

Kilian Alliot

Parcours Data Scientist

PLAN

01 Introduction

02 Texte

03 Image

04 Conclusion

PROBLÉMATIQUE

Je suis Data Scientist au sein de l'entreprise "Place de marché", qui 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.

Pour rendre l'expérience utilisateur des vendeurs (faciliter la mise en ligne de nouveaux articles) et des acheteurs (faciliter la recherche de produits) la plus fluide possible, et dans l'optique d'un passage à l'échelle, il devient nécessaire d'automatiser cette tâche.

Linda, Lead Data Scientist, me demande donc d'étudier la faisabilité d'un moteur de classification des articles en différentes catégories, avec un niveau de précision suffisant.



LES DONNÉES

1 unique dataframe de 1050 lignes et 15 colonnes

- id
- timestamp
- product url
- product name
- product category tree
- pid
- retail price
- discounted price
- image
- is FK advantage product
- description
- product rating
- overall rating
- brand
- prodcut specifications

1050 images au format jpg

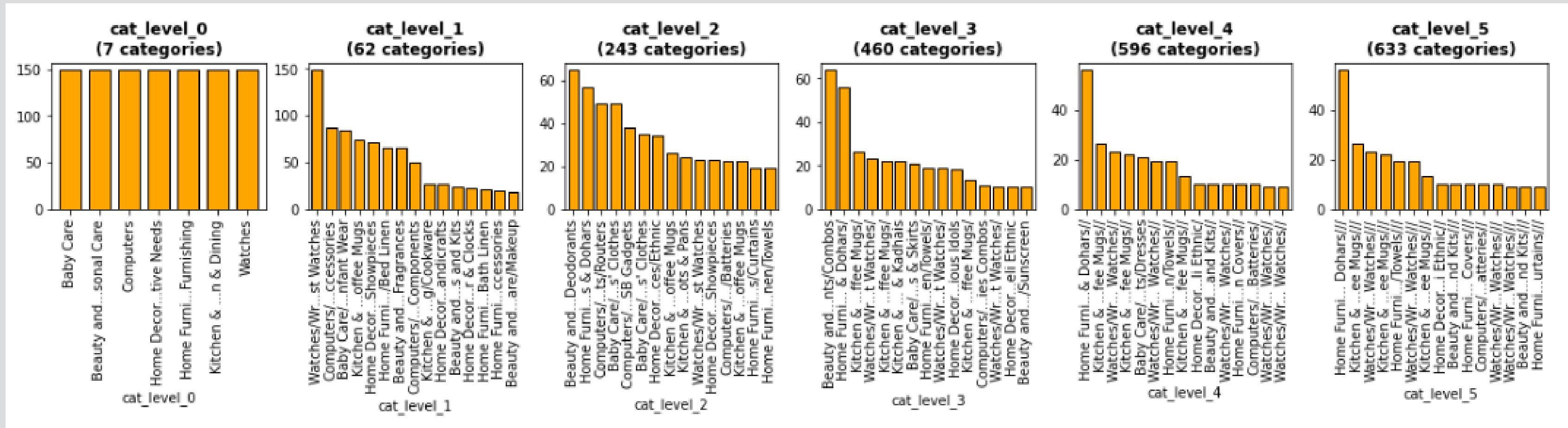
PRÉTRAITEMENT

Garder uniquement les colonnes/variables pertinentes pour le projet :

- image
- description
- category_tree



- category
- category_code



TEXTE

Objectif :

1. Etude de faisabilité
2. Classification supervisée

Approches :

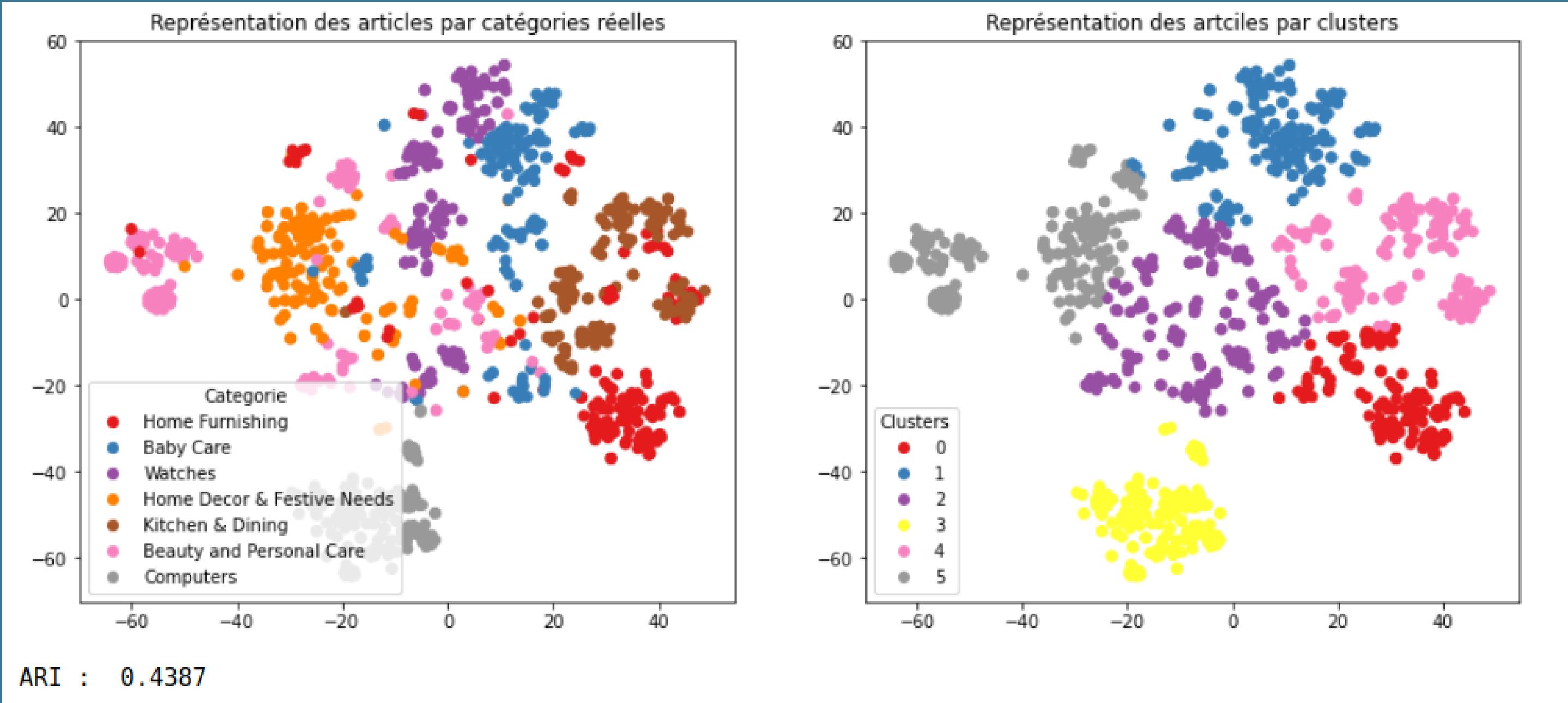
- deux approches de type “bag-of-words”,
comptage simple de mots et Tf-idf ;
- une approche de type word/sentence embedding
classique avec Word2Vec;
- une approche de type word/sentence embedding
avec BERT ;
- une approche de type word/sentence embedding
avec USE (Universal Sentence Encoder)

Nettoyage :

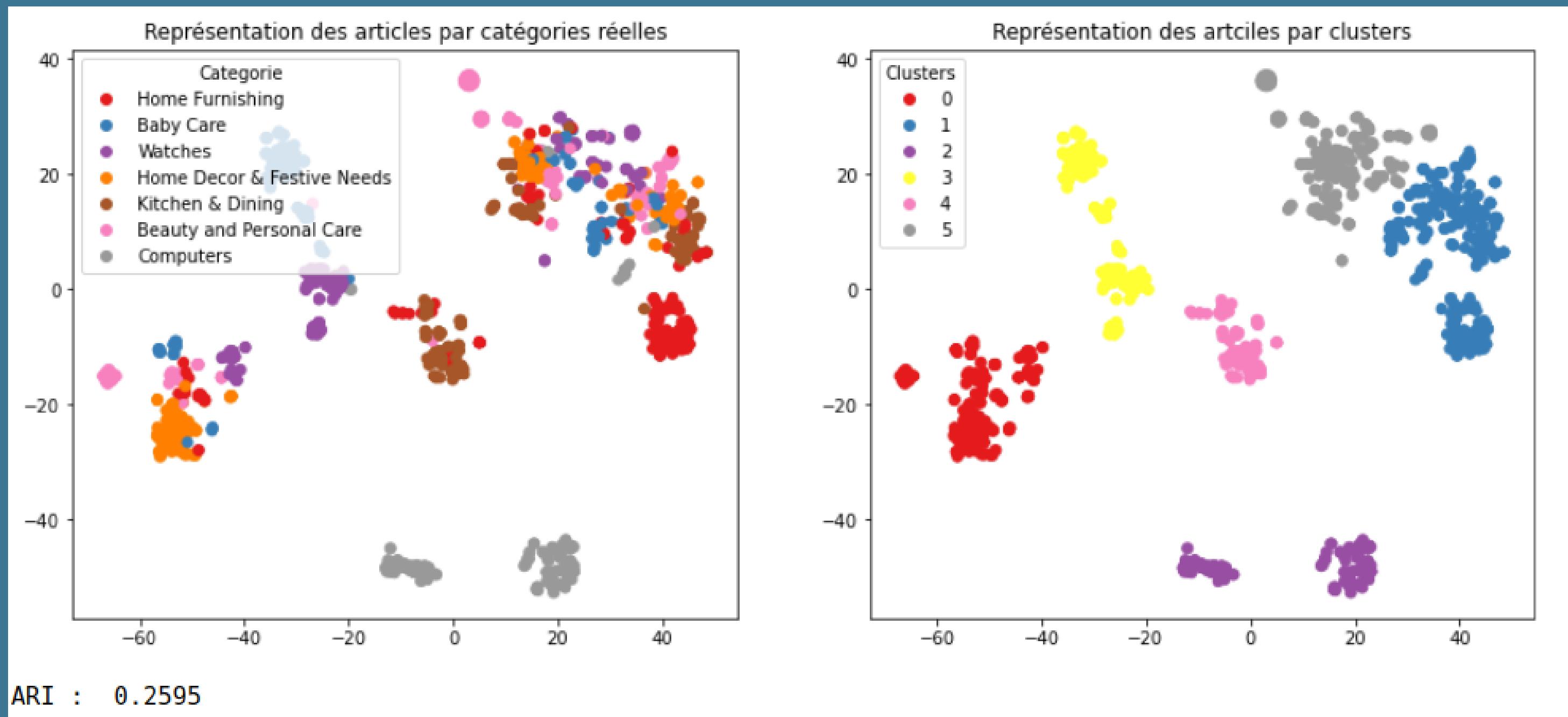
- tokenizer
- lower
- stop word
- lemmatizer

```
max length bow : 379  
max length dl : 632
```

BAG OF WORD - TF-IDF



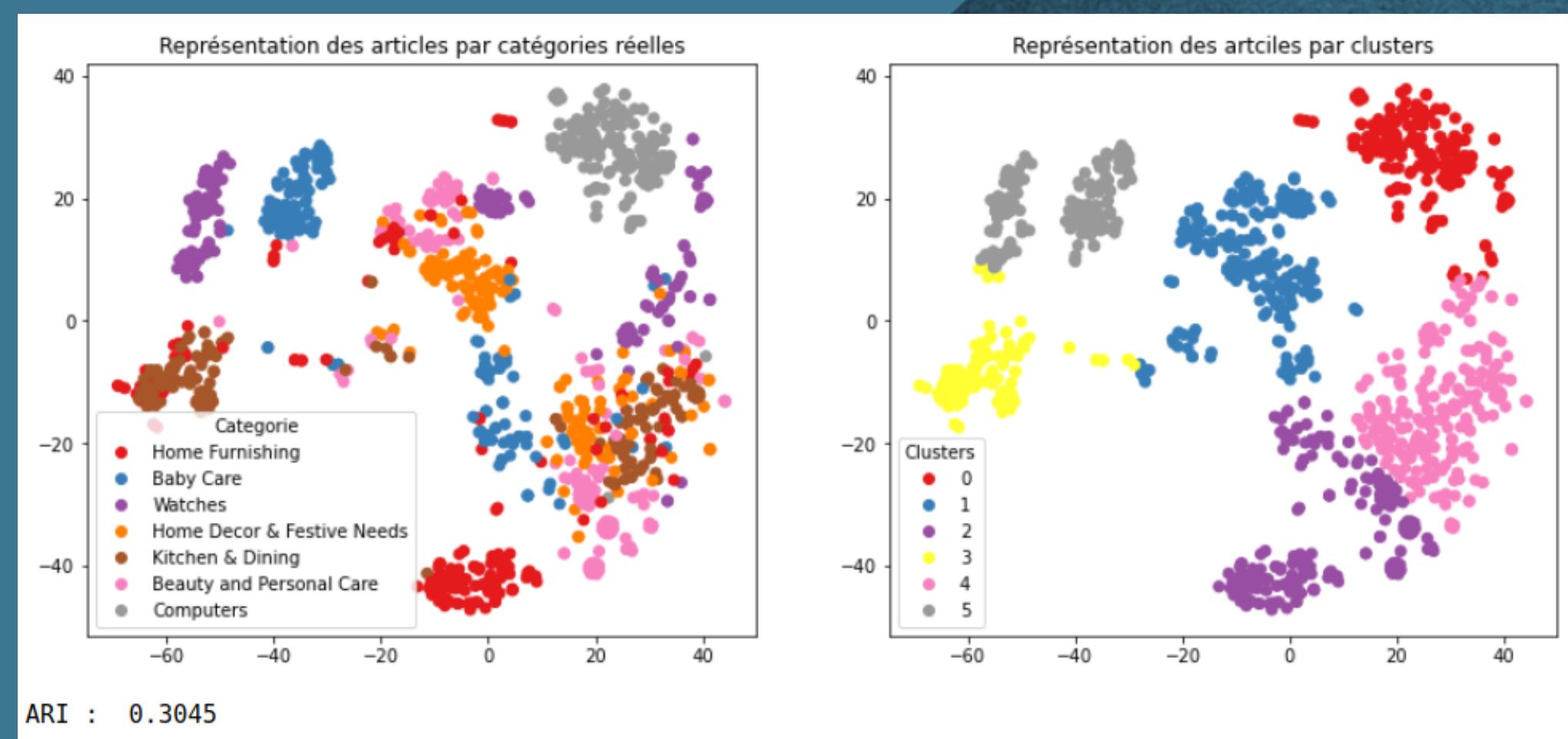
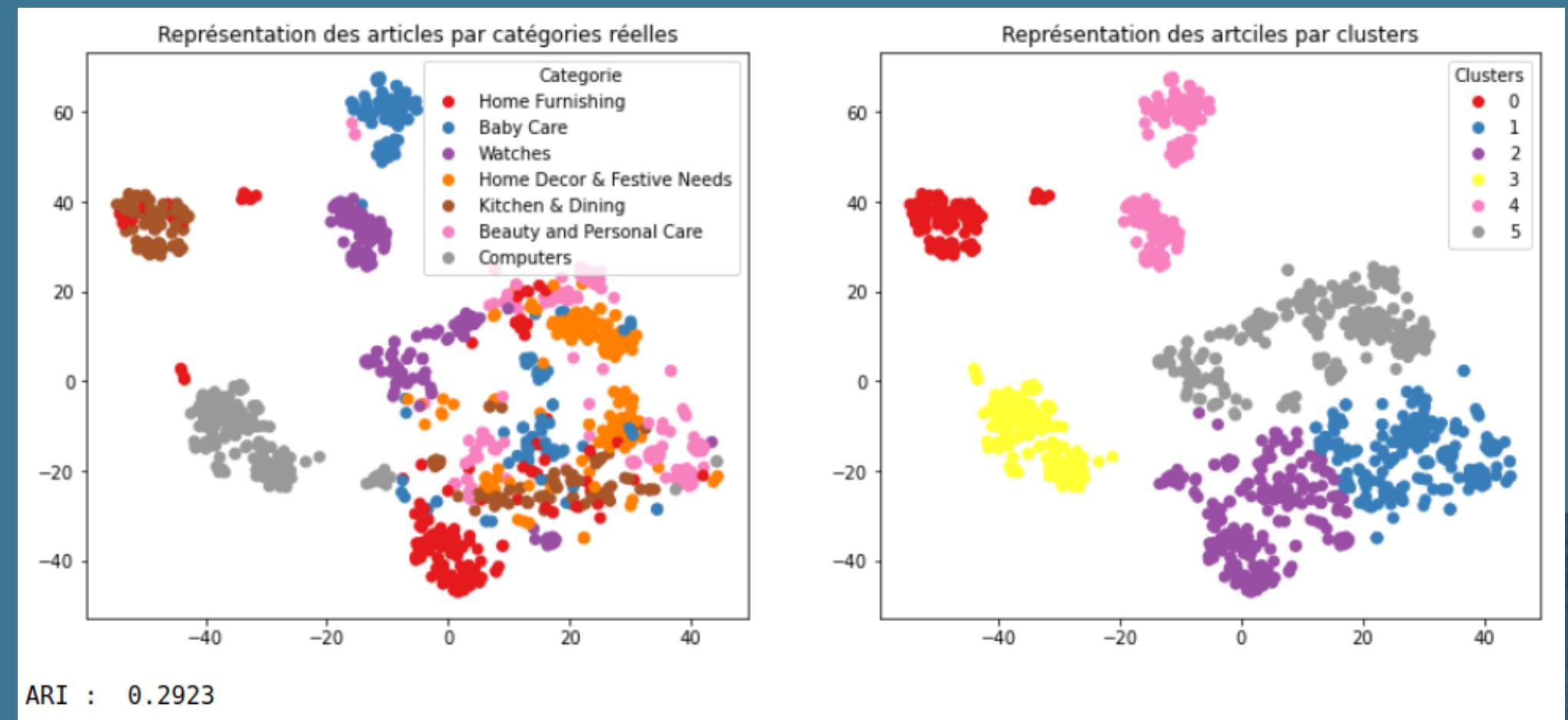
WORD2VEC



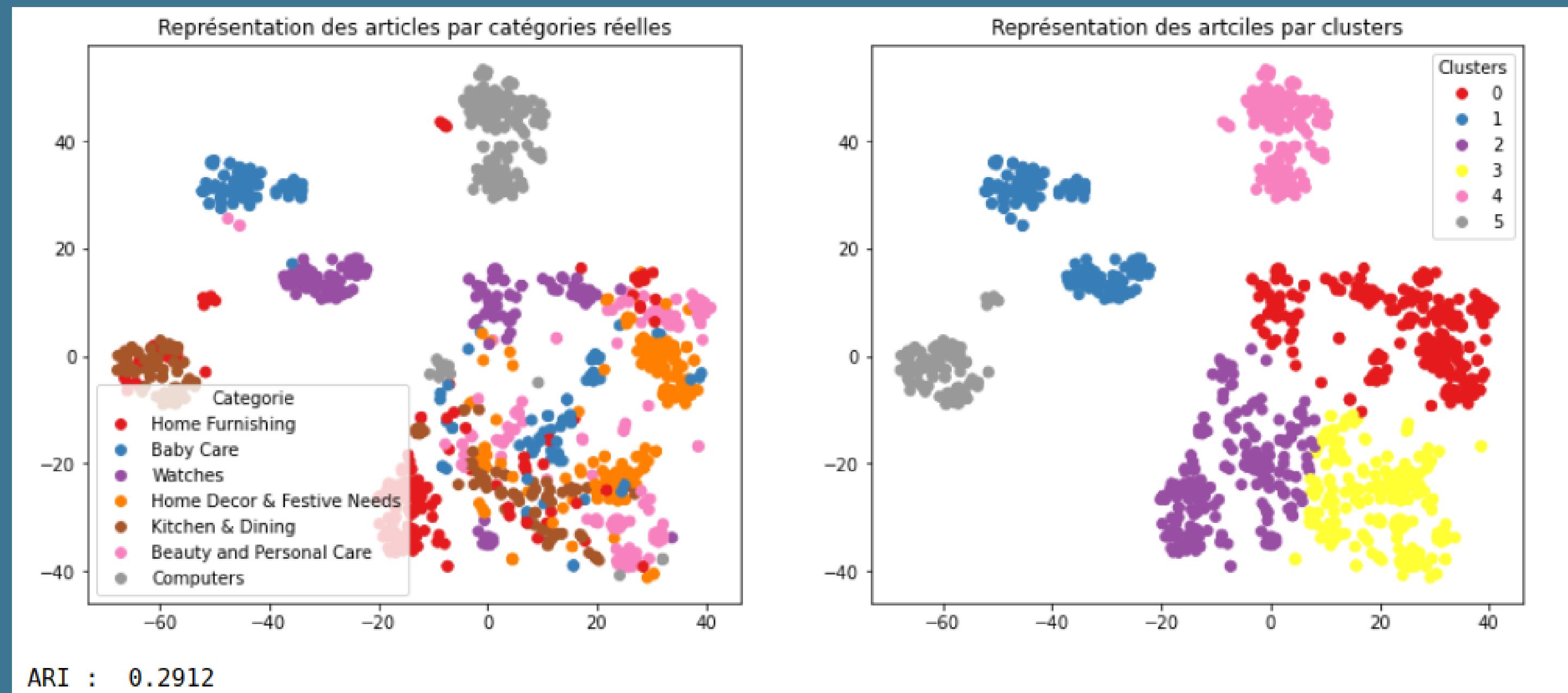
bert-base-uncased

BERT

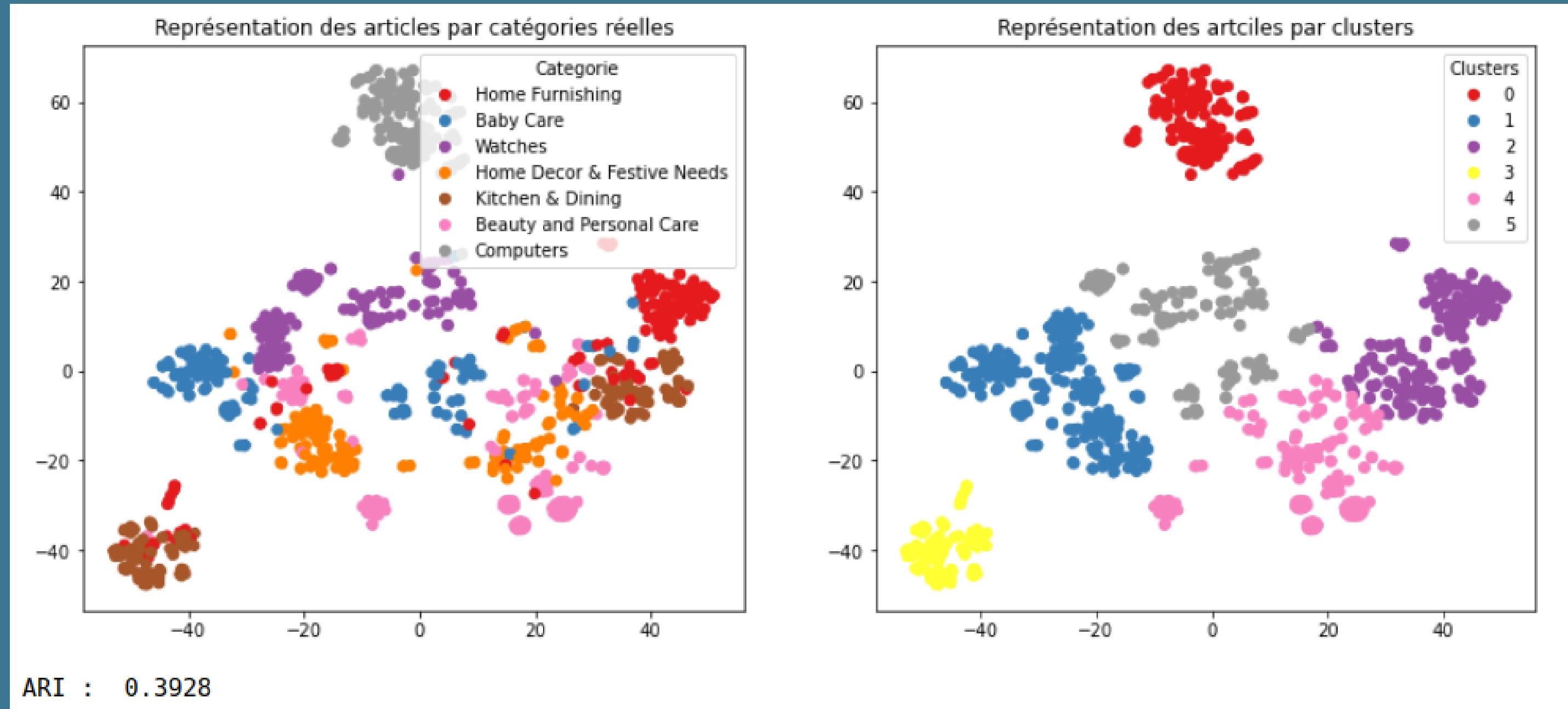
roberta-base



BERT HUB TENSORFLOW



USE - UNIVERSAL SENTENCE ENCODER



CLASSIFICATION

A partir de l'extraction de features du Tf-idf et USE

- Train Test split sur les features
- SVC et Random Forest
- Entrainement
- Prédiction des labels/catégories
- Mesurer la précision de la classification

USE :

- 512 features
- Accuracy de 0.93

Tf-idf :

- 5324 features
- Accuracy de 0.95

IMAGE

Objectif:

1. Etude de faisabilité
2. Classification supervisée

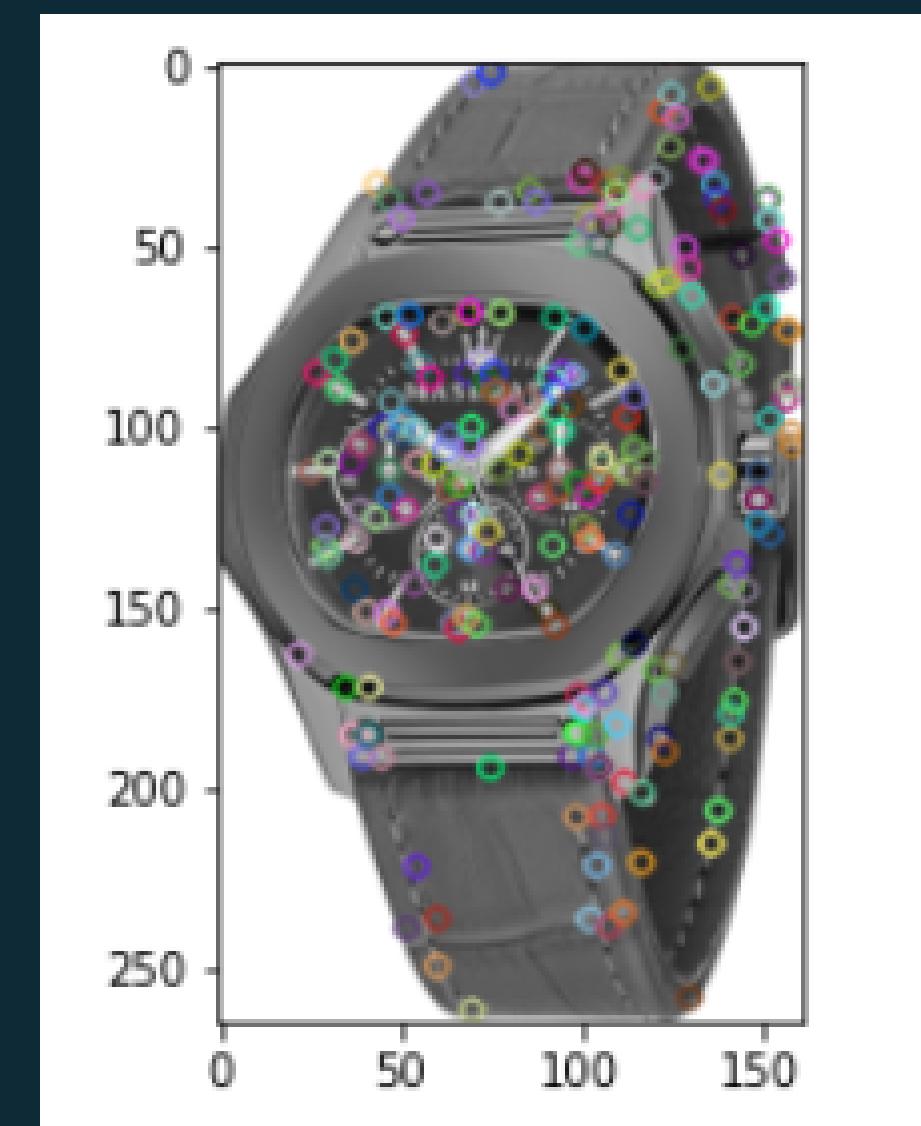
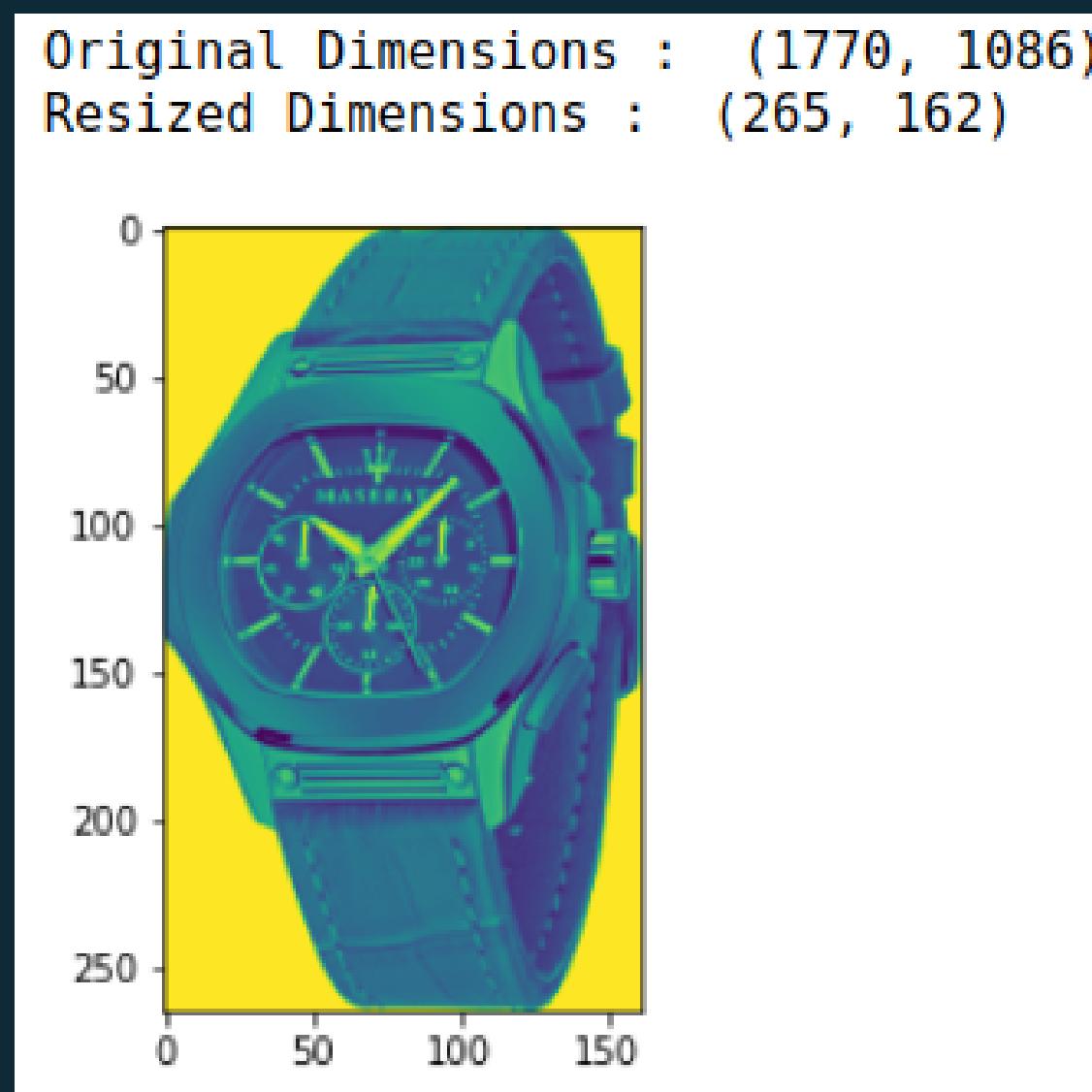
Approches :

- un algorithme de type SIFT / ORB / SURF ;
- un algorithme de type CNN Transfer Learning
(convolutional neural network)



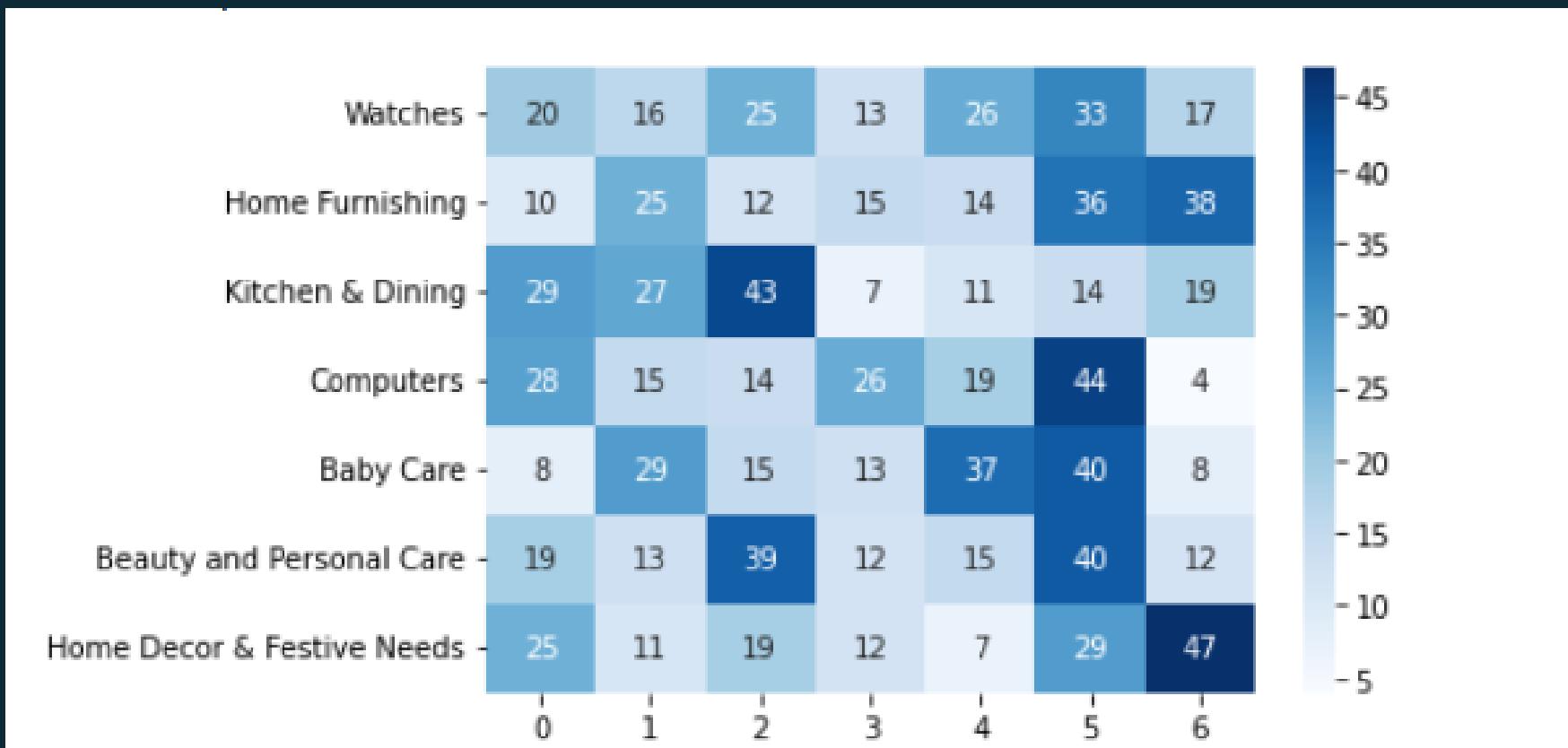
SIFT

- L'image originale contient 7k (montre) comme 47k (serviettes) descripteurs
- Il convient de réduire la résolution de l'image pour réduire le nombre de descripteurs et alléger les calculs
- Chaque descripteur est un vecteur de longueur 128



SIFT

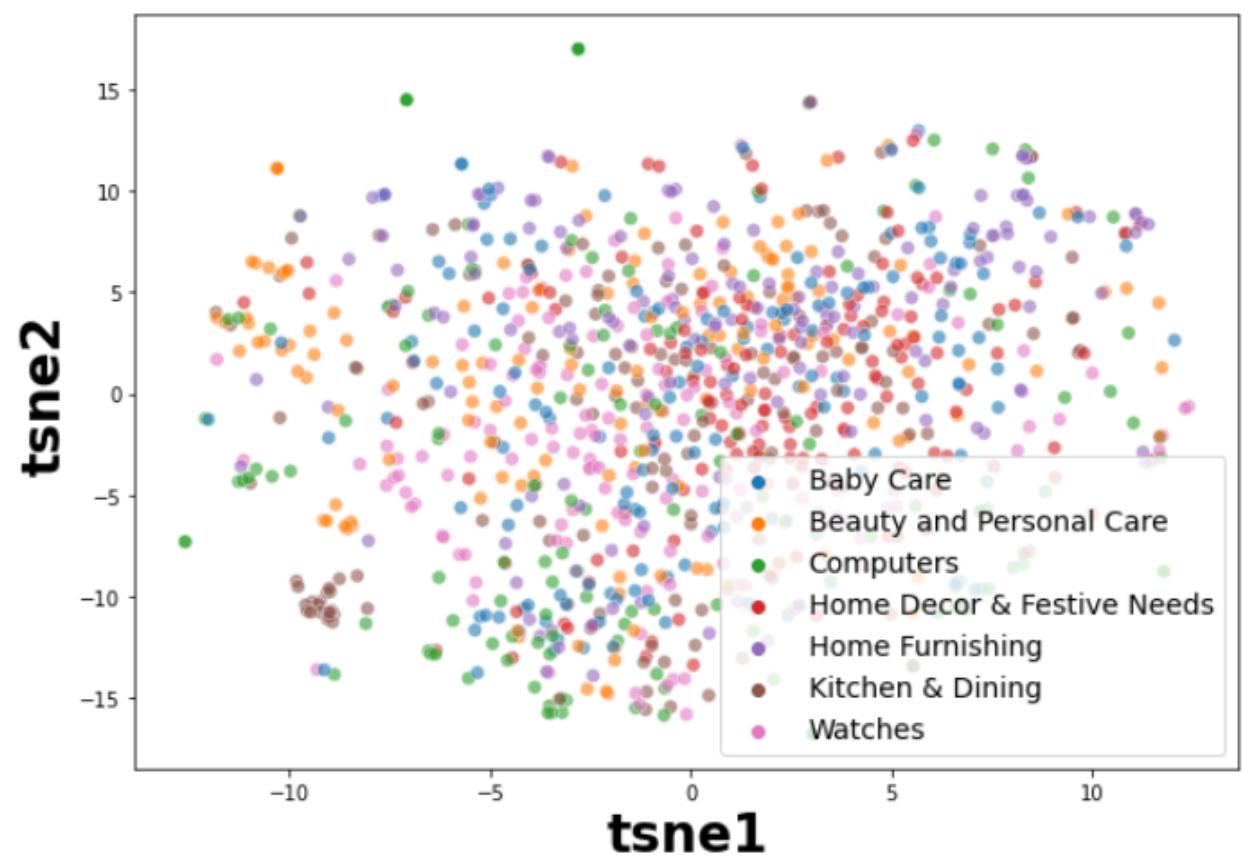
- Iteration sur toutes les images pour identifier les descripteurs associés
- Clustering avec MiniBatchKMeans (460 clusters créés)
- ACP (variance expliquée 99%)
- T-SNE (2 composantes)
- ARI entre vraies classes et clusters (0.02)



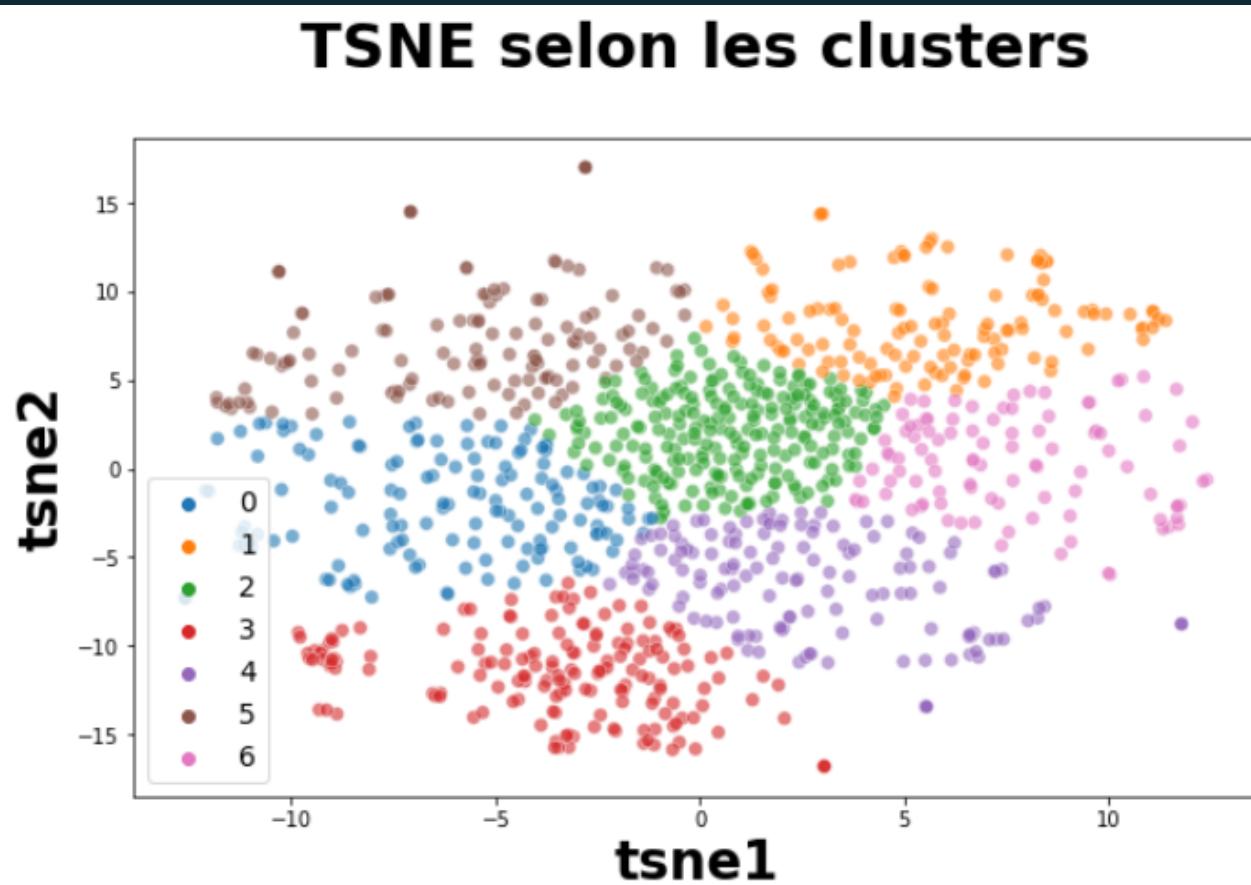
On en conclue que le clustering n'est pas efficace, pour cause les raisons suivantes :

- grosse réduction des images en terme de résolution
- algorithme SIFT performe moyennement ?
- nos features sont réduites par une PCA
- puis TSNE sur la PCA

TSNE selon les vraies classes

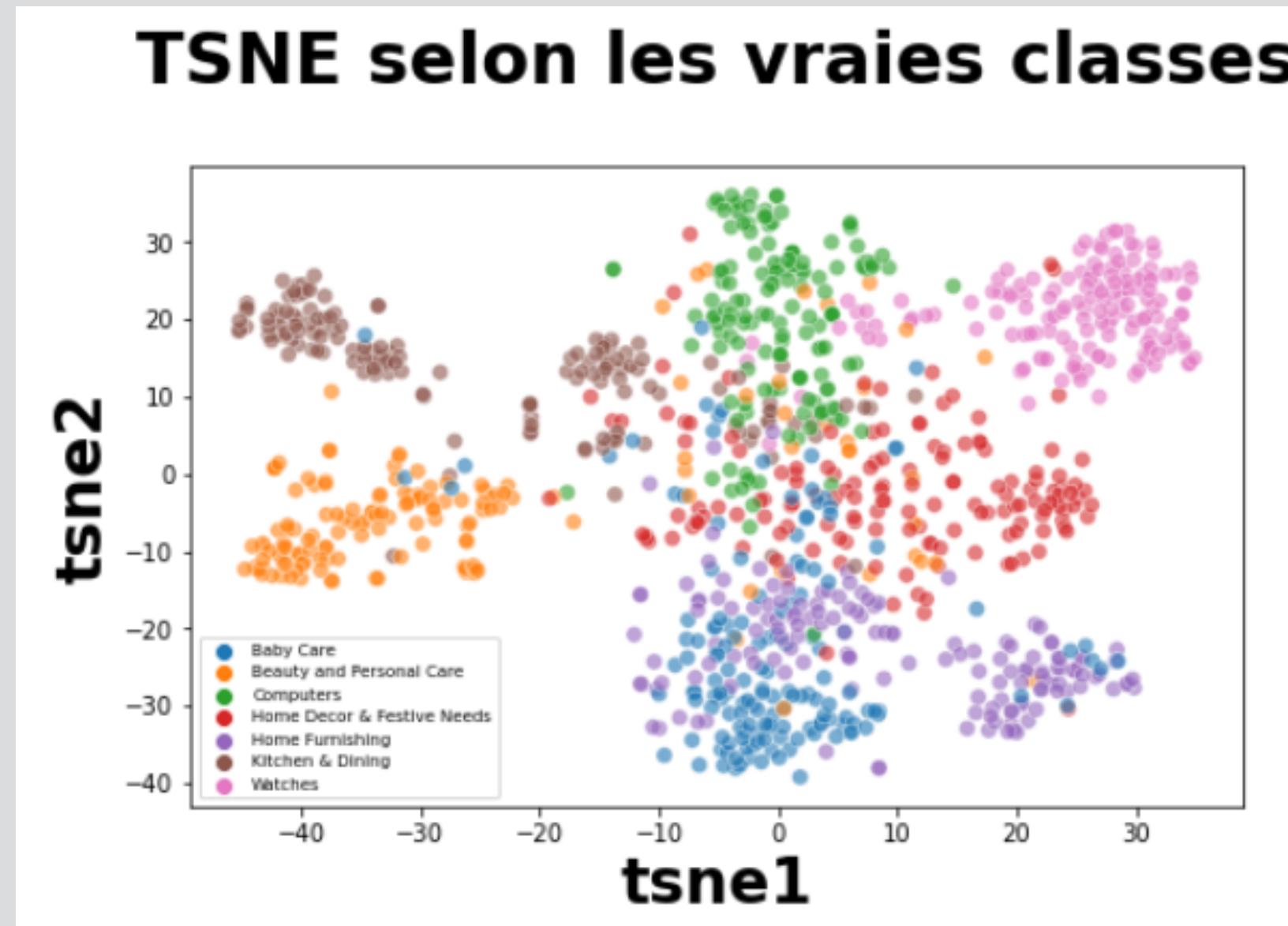


TSNE selon les clusters



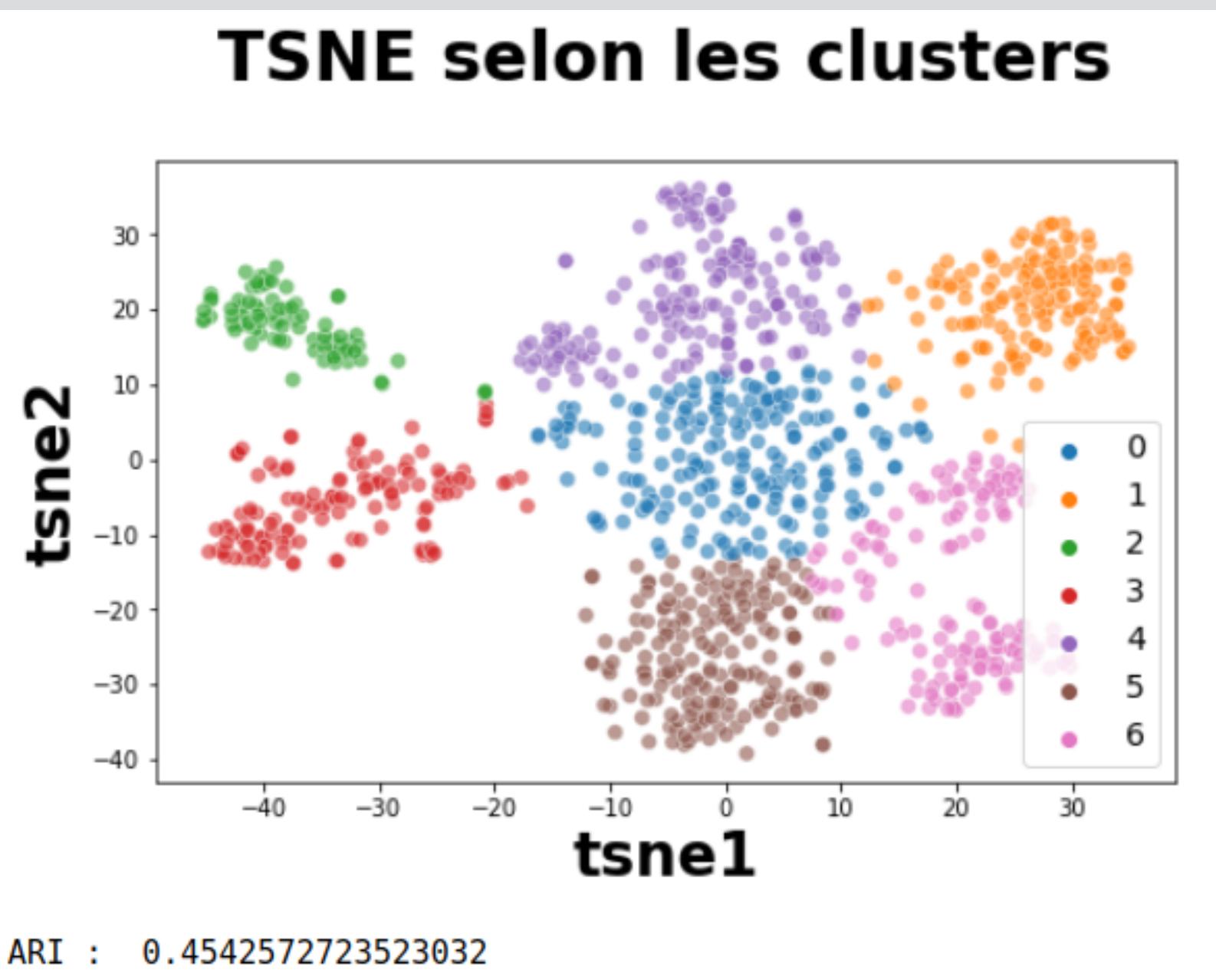
CNN

- Crédit du modèle pré-entraîné VGG16
- Crédit des features images (4096 features)
- ACP (99% de variance expliquée - 803 features)
- T-SNE (2 composantes)

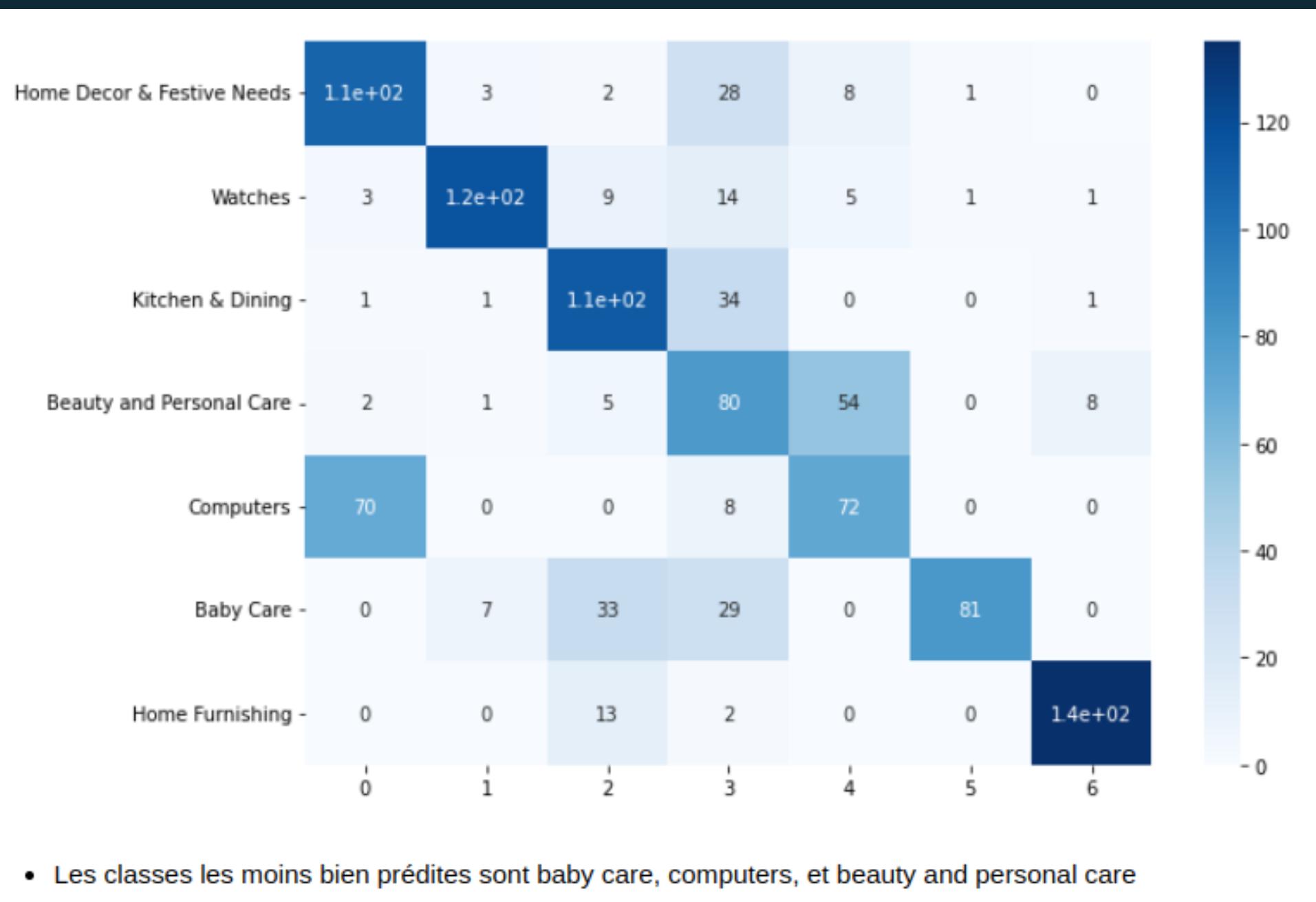


- L'analyse graphique montre visuellement qu'il est réalisable de séparer automatiquement les images selon leurs vraies classes
- Ceci suffit à démontrer la faisabilité de réaliser ultérieurement une classification supervisée pour déterminer automatiquement les classes des images
- Cette étape 1 est très rapide à mettre en oeuvre. Une conclusion négative sur la faisabilité aurait évité de réaliser des traitements beaucoup plus lourd de classification supervisée
- Cette démarche en 2 étapes (1. Faisabilité, 2. Classification supervisée si étape 1 OK) s'inscrit dans une démarche agile de tout projet Data

Création de clusters à partir du T-SNE et affichage des images selon clusters :



- Attention : ici, il ne s'agit pas de faire une classification non supervisée, mais simplement, par une mesure de l'ARI, de conforter l'analyse graphique précédente qui démontre la faisabilité de réaliser ultérieurement une classification supervisée. Cette mesure de l'ARI nécessite de créer des clusters théoriques via KMeans
- Il s'agit donc de réaliser une mesure de ce que nous voyons graphiquement, donc à partir des données en sortie du t-sne
- Pour réaliser une classification non supervisée, il aurait fallu repartir des données avant t-sne
- Dans la démarche en 2 étapes, il n'est pas utile de réaliser une classification non supervisée, une classification supervisée est bien plus performante. Même le calcul de l'ARI n'est pas indispensable, nous pourrions passer directement du graphique t-sne précédent à l'étape 2 de classification supervisée
- Il n'est donc pas utile de passer du temps à optimiser l'ARI, un ordre de grandeur suffit pour conforter le 1er graphique t-sne. D'ailleurs la meilleure solution de feature engineering ne génère pas toujours le meilleur ARI. L'analyse graphique t-sne est bien plus riche d'enseignement



- Analyse : le modèle pré-entraîné confond "Computer" avec une valise de maquillage ...



CLASSIFICATION

4 approches sont présentées :

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

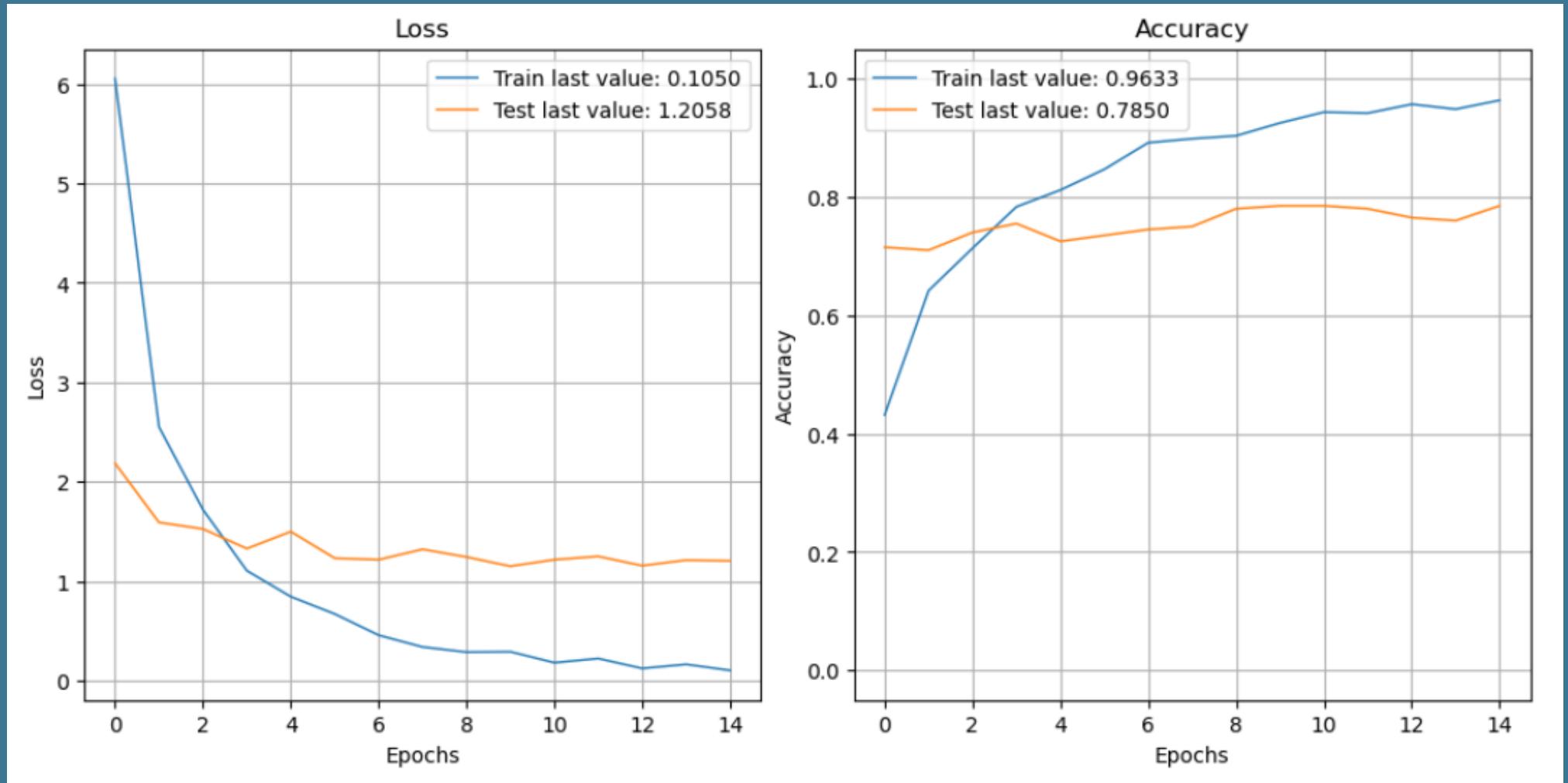
Dataset d'entraînement :

- 800 lignes
- test size 0.25

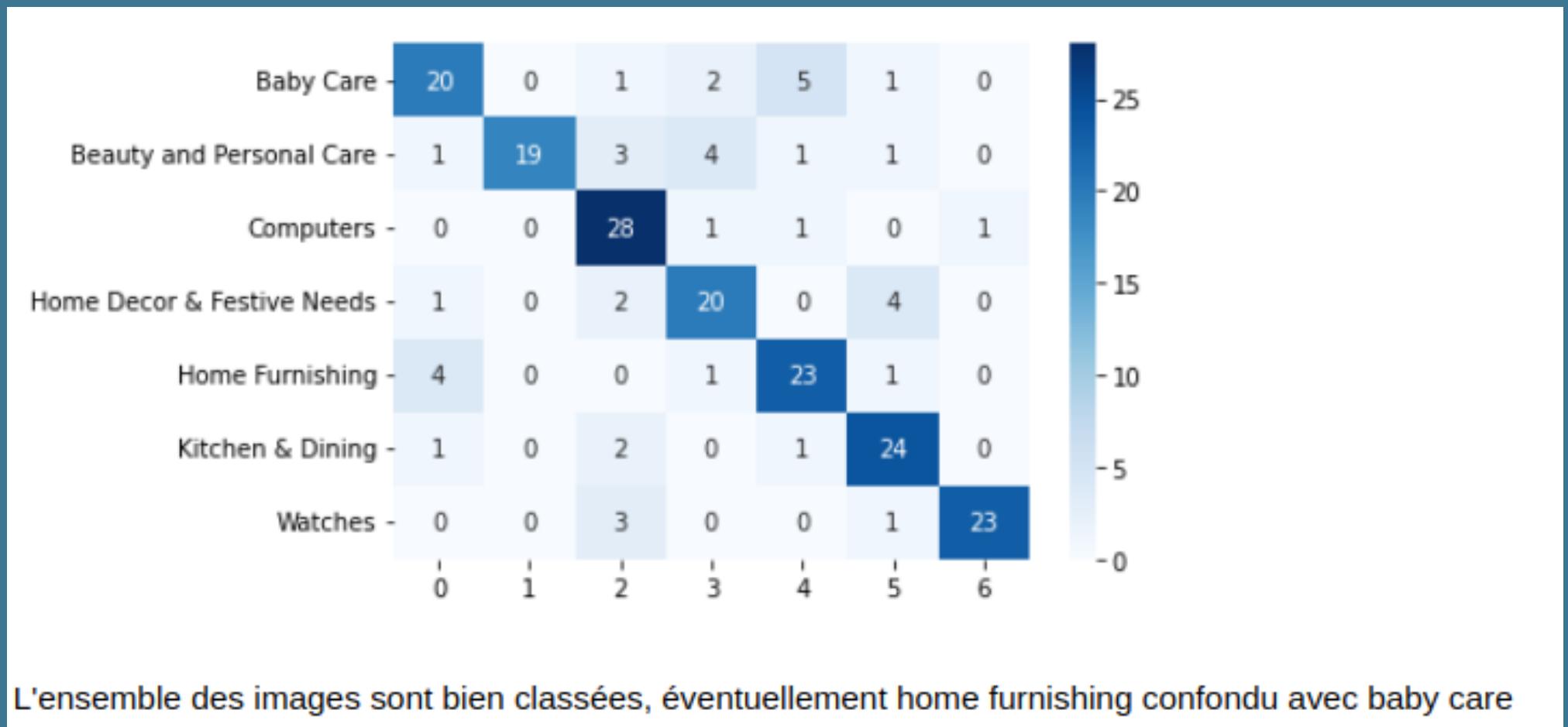
Dataset de test :

- 250 lignes

APPROCHE SIMPLE



Training Accuracy : 0.9917
Validation Accuracy : 0.7850
Test Accuracy : 0.8560



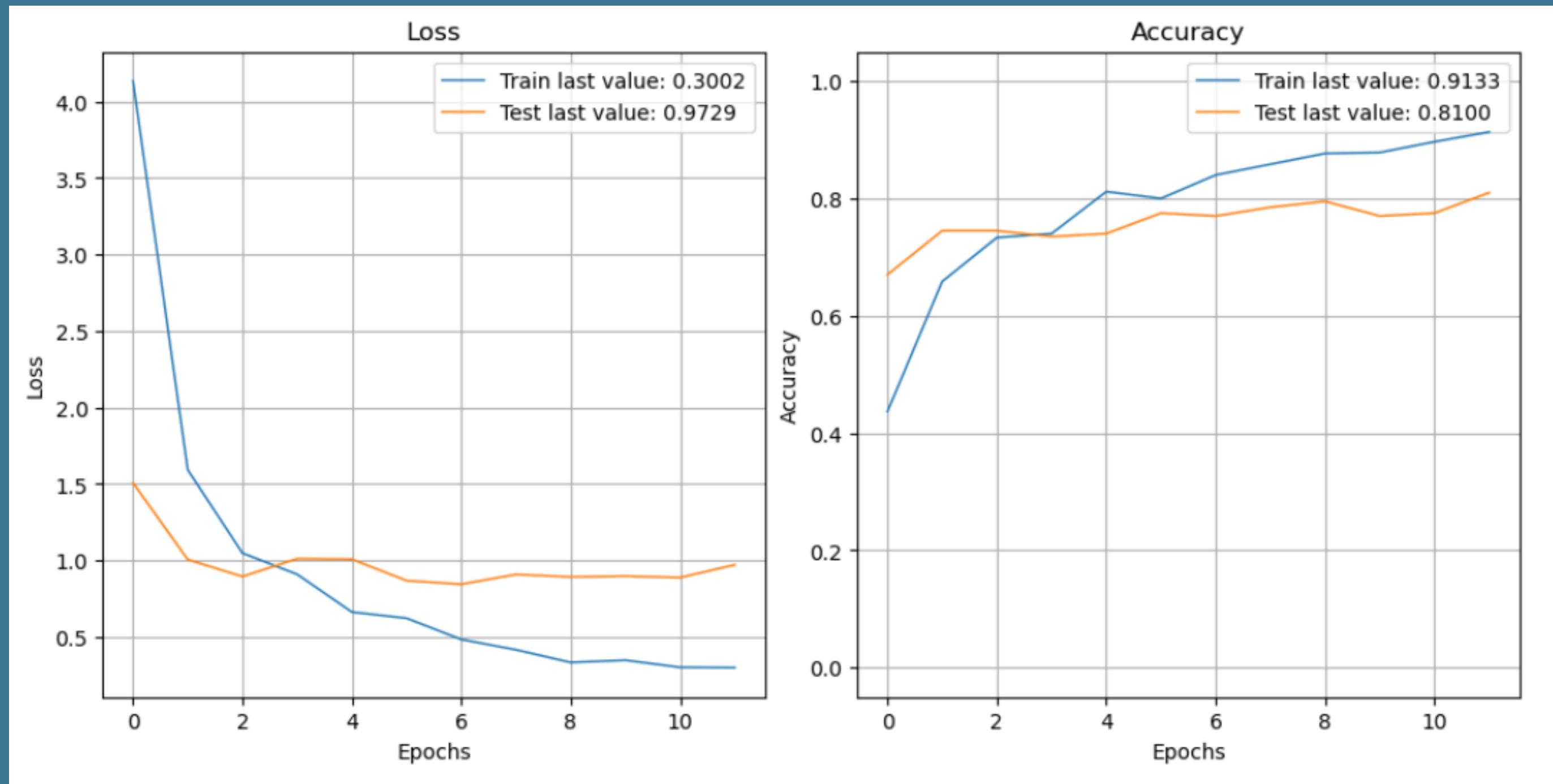
ImageDataGenerator (rotation, flip, shift)

Training accuracy 0.98

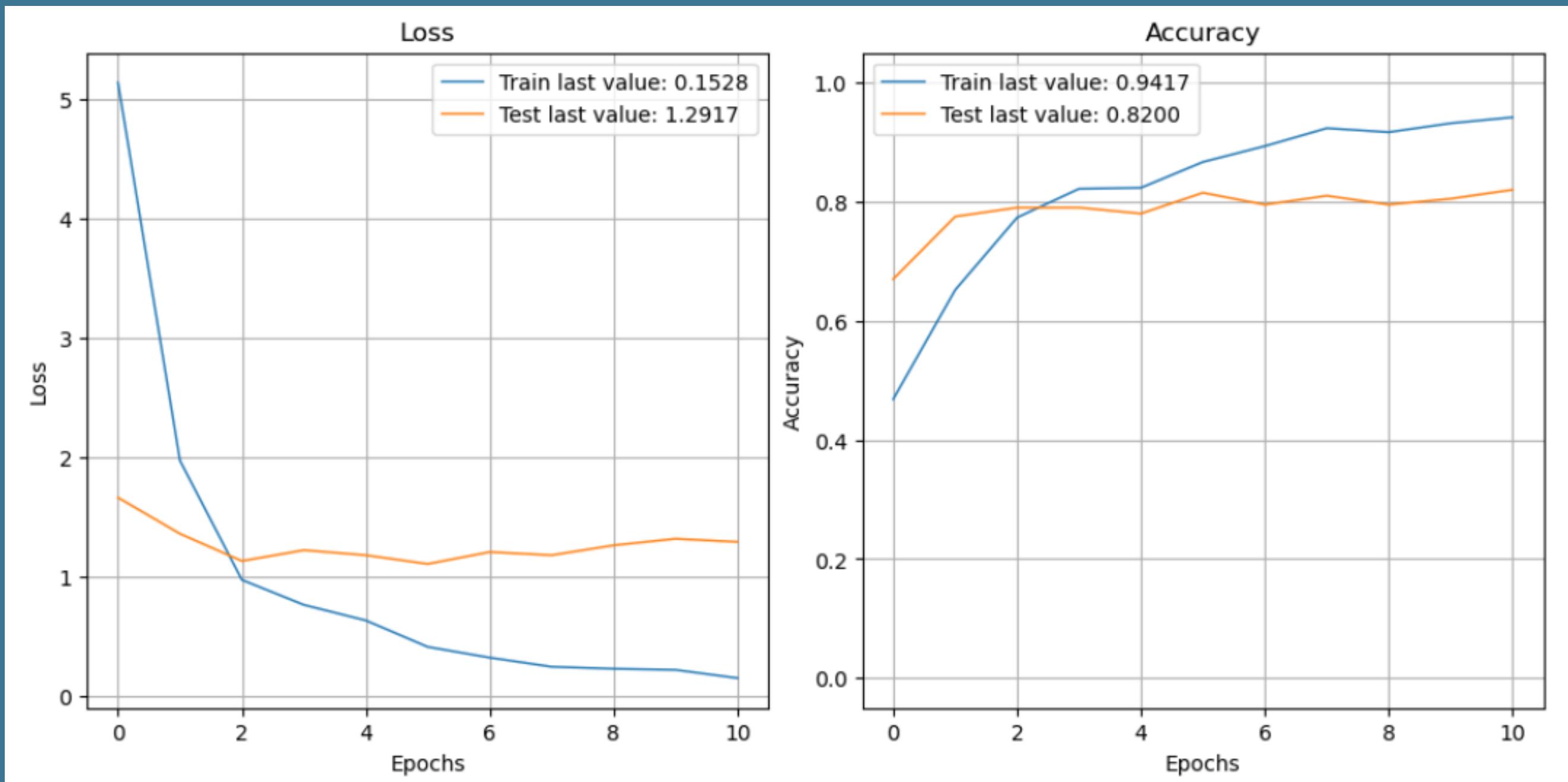
Validation accuracy 0.78

Test accuracy 0.85

DATAGENERATOR AVEC DATA AUGMENTATION

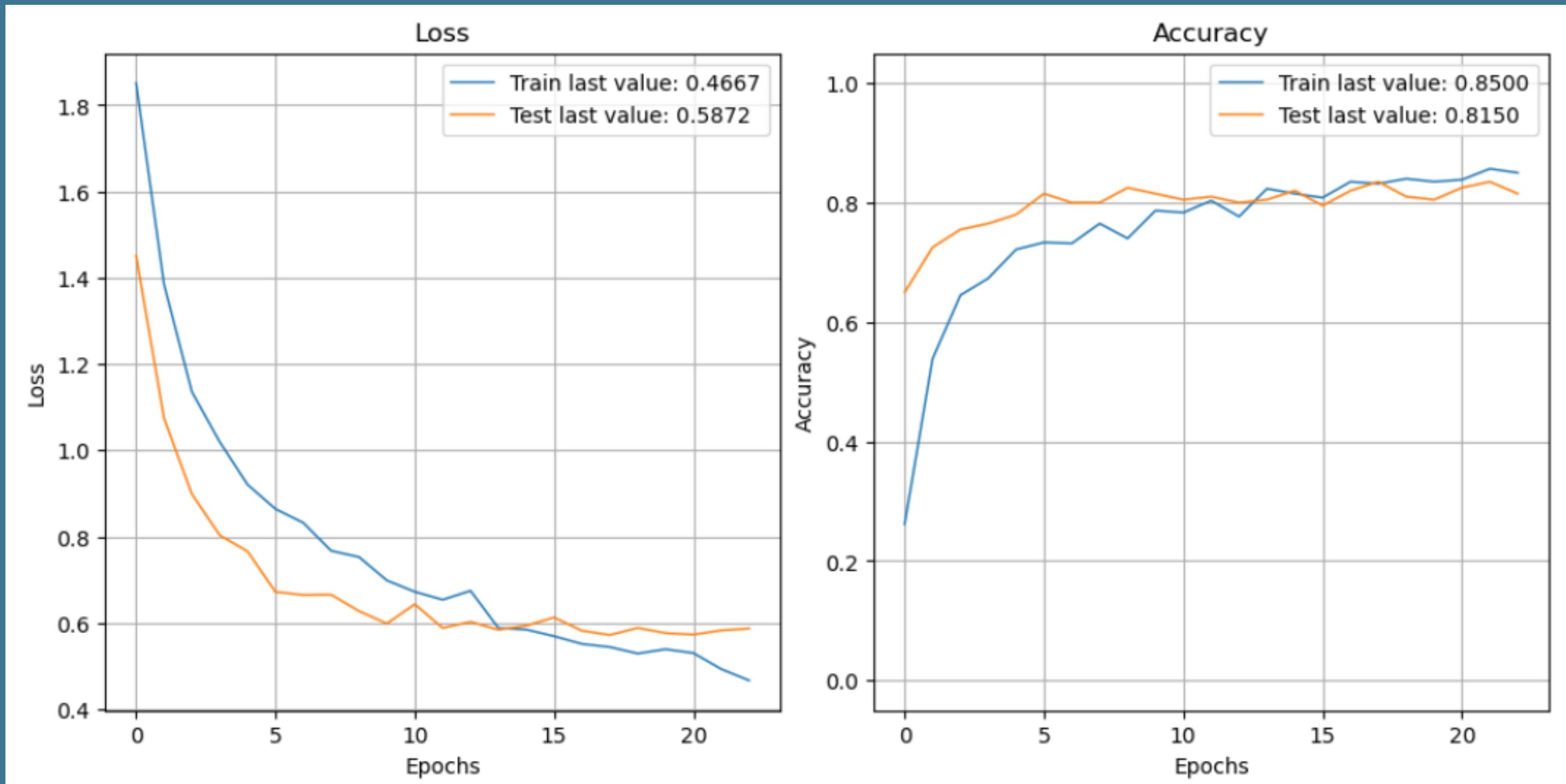


APPROCHE PAR DATASET SANS DATA AUGMENTATION



Training accuracy 0.99
Validation accuracy 0.82
Test accuracy 0.83

APPROCHE PAR DATASET AVEC DATA AUGMENTATION INTÉGRÉE AU MODÈLE



Training accuracy 0.90
Validation accuracy 0.81
Test accuracy 0.80

CONCLUSION

Les différentes méthodes de classification abordées nous permettent de conclure sur la faisabilité de l'automatisation de la tâche qui nous importe : déterminer automatiquement la catégorie d'un produit à partir de la description et de l'image donné par le vendeur.

Nous avons obtenu une précision de prédiction à la hauteur de nos attentes, aussi bien à l'aide des données textuelles qu'avec des données visuelles.

On pourrait combiner l'utilisation de deux modèles, un pour le texte et l'autre pour l'image, ou bien créer un nouveau modèle prenant en entrée toutes nos données brutes (plus complexe).



MERCI DE VOTRE ATTENTION