Projet 6 Openclassrooms

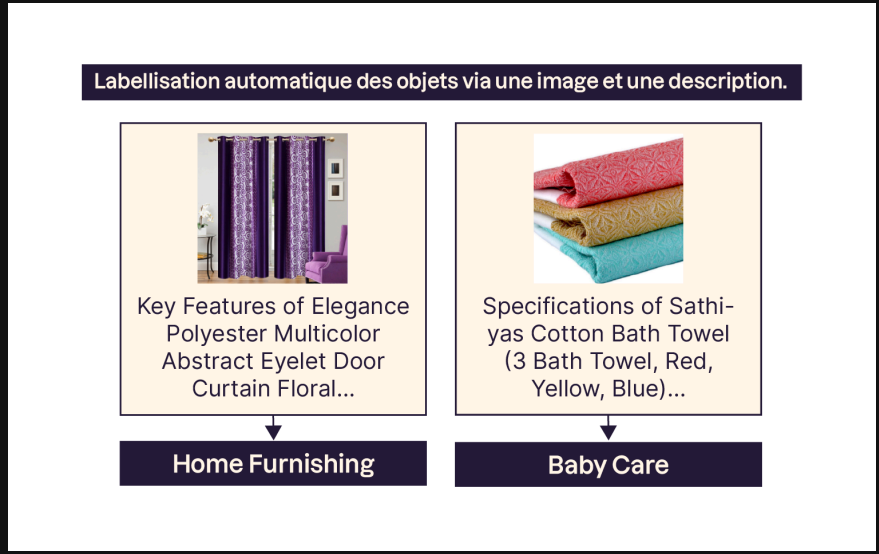
Classifiez automatiquement des biens de consommation

I. Présentation du contexte projet et du jeu de données

Contexte Projet

- L'entreprise **Place de marché** souhaite lancer une markeplace e-commerce
- **Problématique:** actuellement l'attribution de la catégorie du produit est effectuée manuellement par le vendeur, et est donc peu fiable.

Notre mission :
Faire une étude de faisabilité d'un moteur de classification pour l'automatisation de l'attribution de la catégorie de l'article en fonction de la description et de l'image de l'article.



Présentation du jeu de données

Fichier CSV

- **1050 lignes** contenant des articles.
- **15 colonnes** fournissant des informations sur chaque produit :
- Identifiant unique du produit
- **Nom du produit**
- Marque du produit
- URL du produit
- **Arborescence de la catégorie du produit (7 niveaux)**
- Prix
- **Description du produit**
- **Nom de l'image**

Nettoyage des données : Très peu de données manquantes, aucune absence dans les champs utilisés.

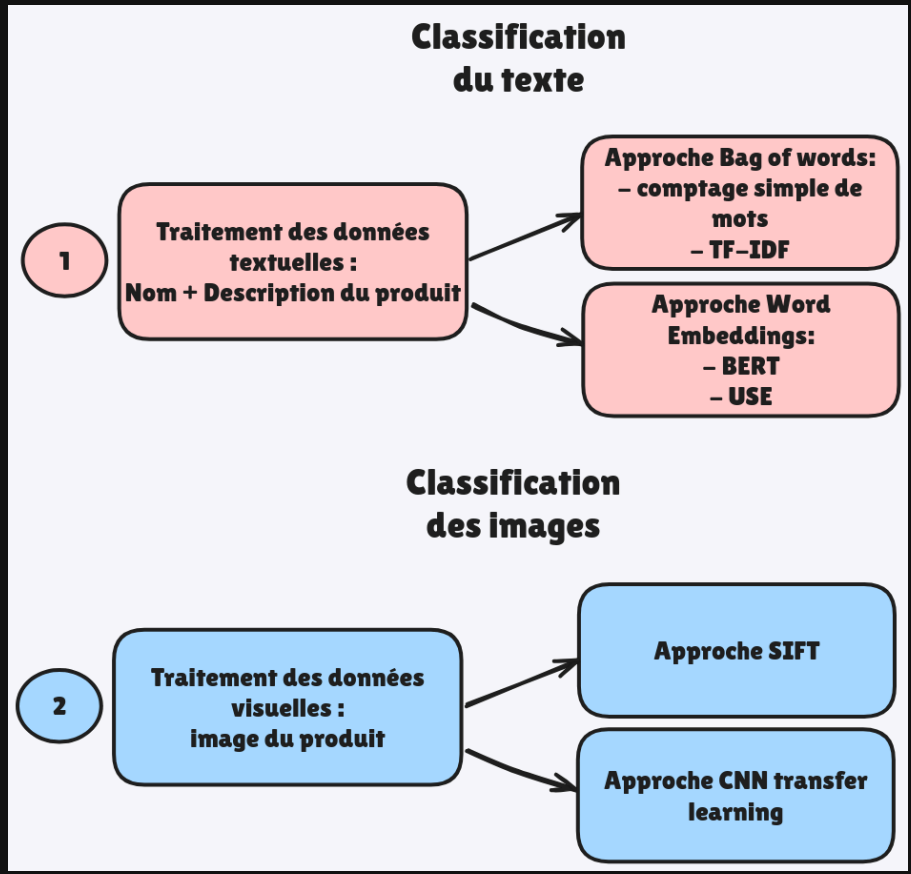
Dossier d'images

- Vérification de la corruption et du format des images

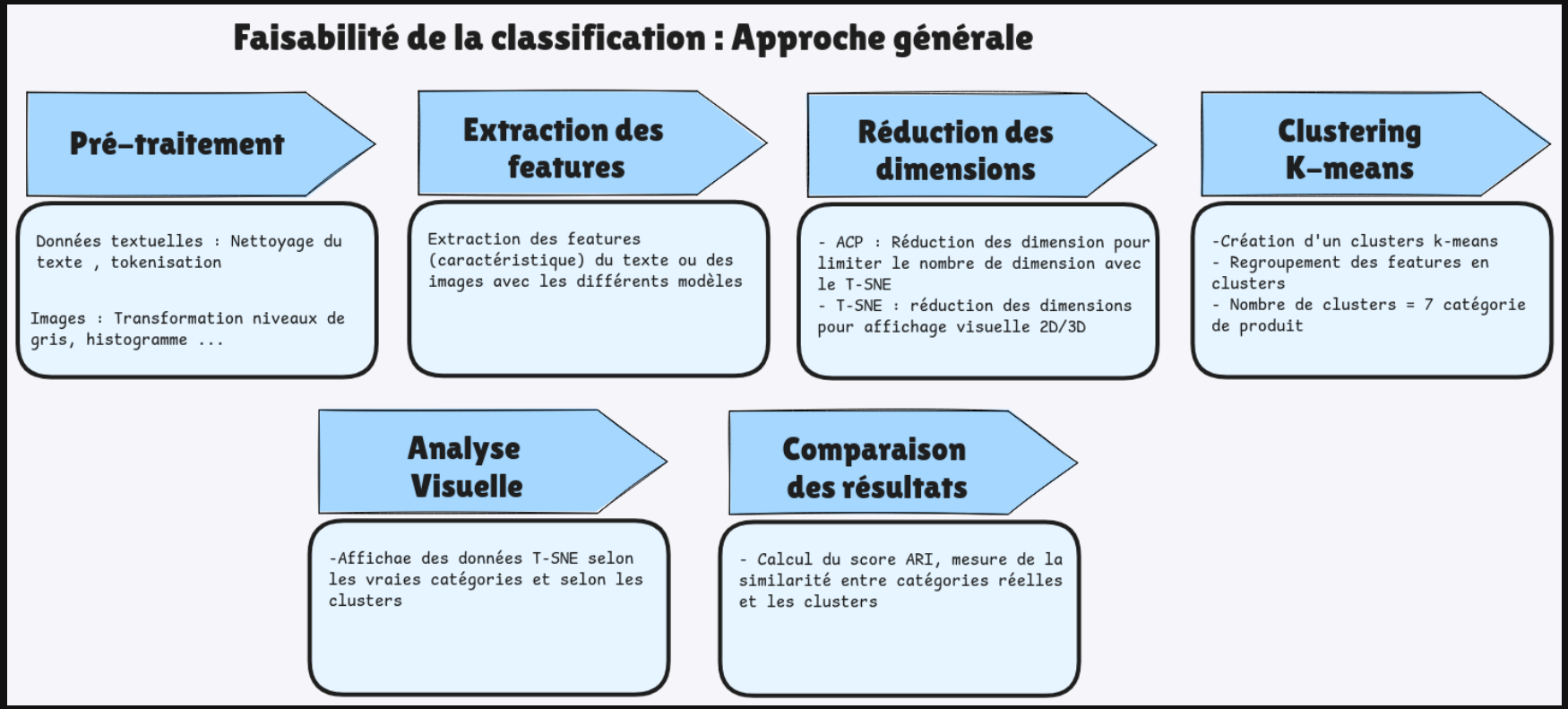
1050 image pour chaque produit

II.Présentation des approches de modélisation de classification et résultats

Approche de modélisation



Approche de modélisation générale



III.Présentation des approches de modélisation de classification du Texte

Traitement des données textuelles

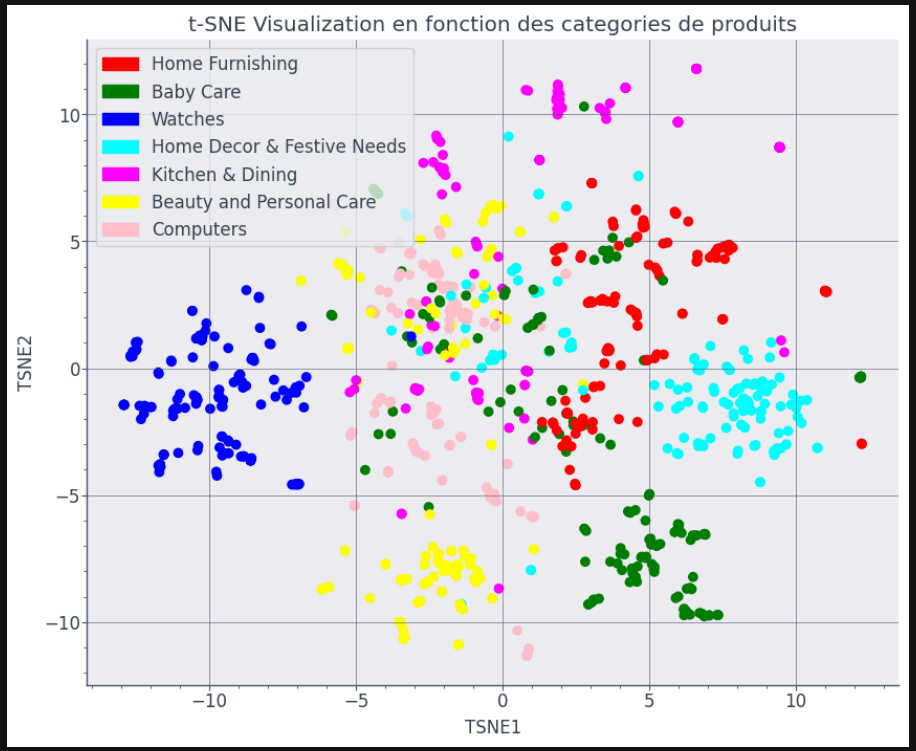
Nettoyage des données textuelles

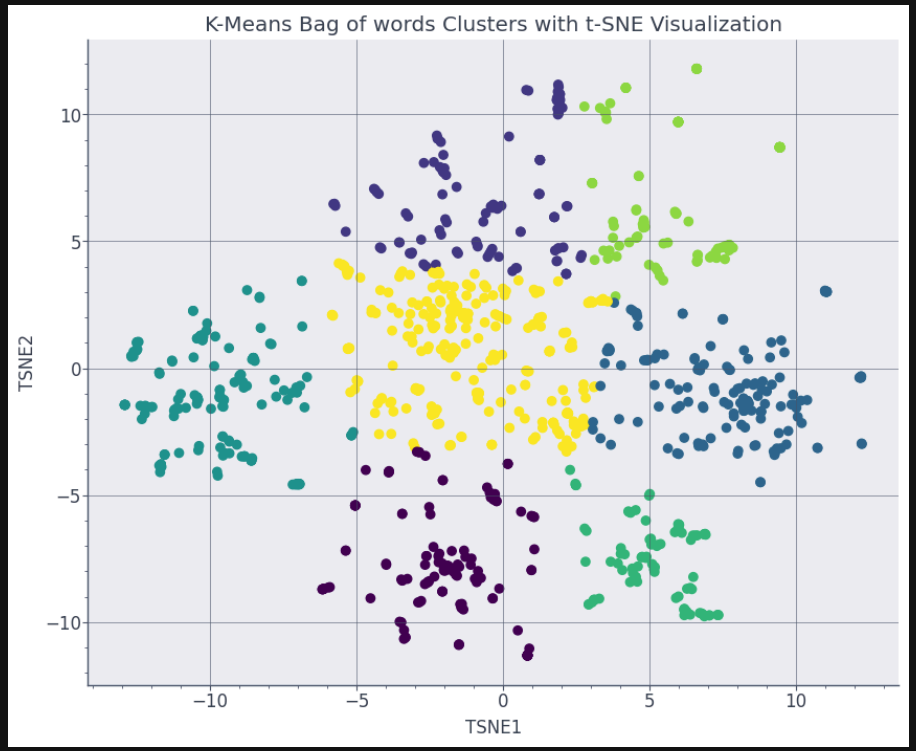
- Normalisation du texte sur le texte `product_name` et `description` :

1. Conversion du texte en minuscules et suppression de la ponctuation
2. Tokenisation : Division du texte par mots
3. Suppression des stopwords (articles,pronoms...)
4. Ajout du texte `product_name` et `description` dans la même phrase

Approche Bag of words : comptage simple

- Représentation de chaque document en **fonction de la fréquence des mots** (count vectorizer)
- Création d'un vecteur pour chaque document rassemblé dans une **matrice de comptage**

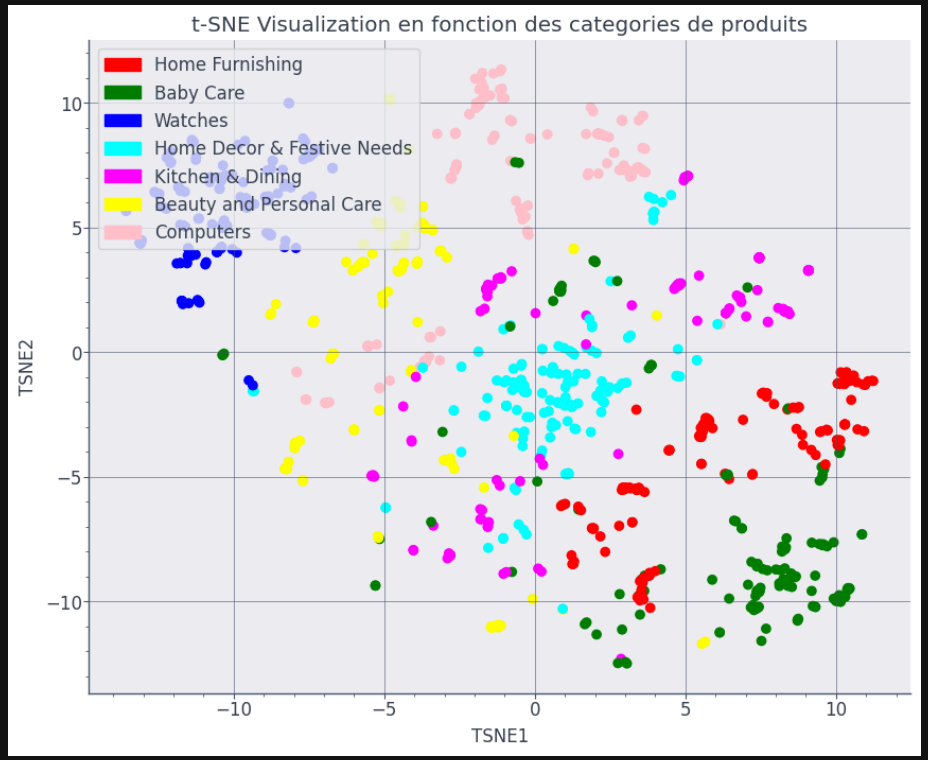


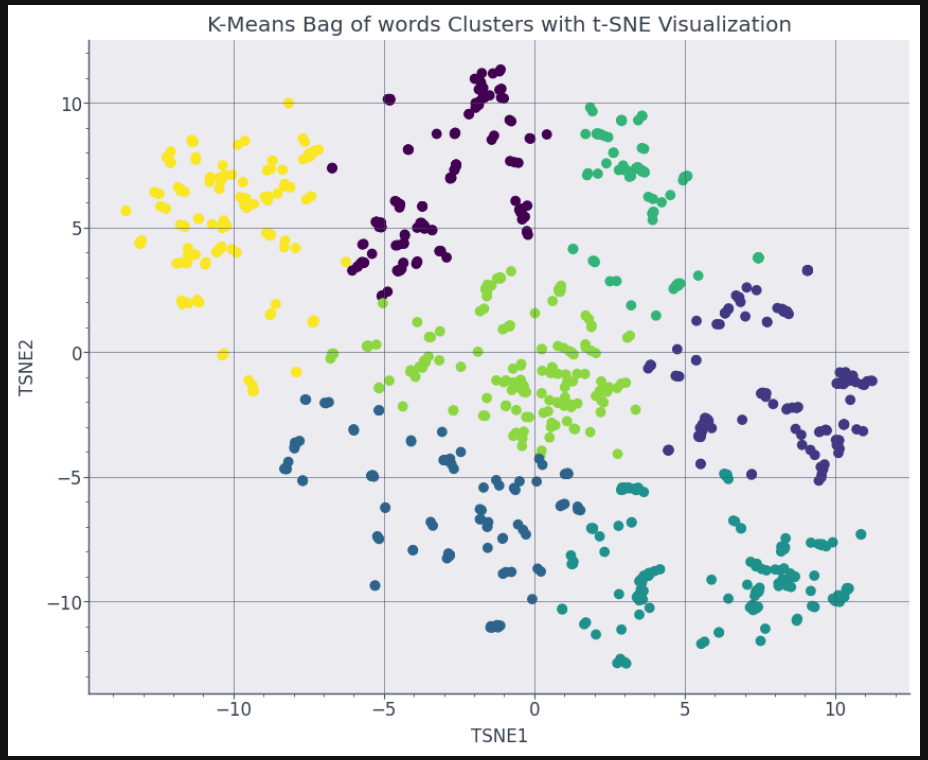


Score ARI : 0.3347 Séparation partielle des catégories

Approche Bag of words : TF-IDF (Term Frequency-Inverse Document Frequency)

- TF (Term Frequency) : Fréquence d'un mot dans le document
- IDF (Inverse Document Frequency) : Réduit l'importance des mots communs qui apparaissent dans de nombreux documents.

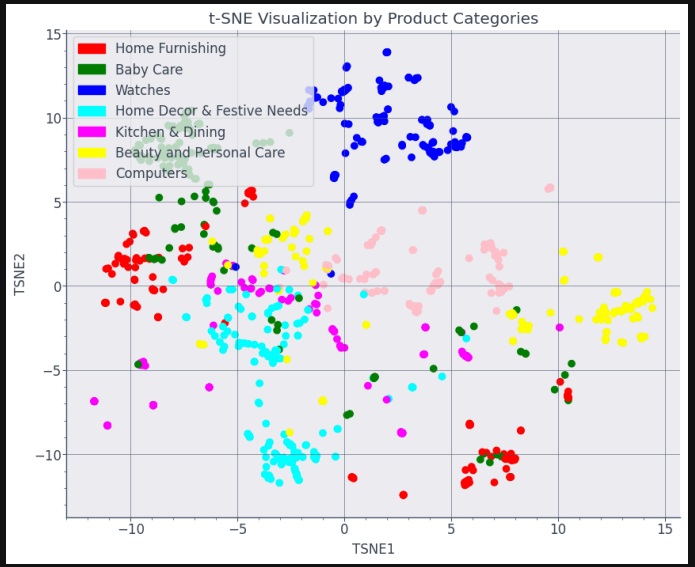




Score ARI : 0.4092 Assez bonne séparation des catégories

Approche Bag of words : Word2Vec

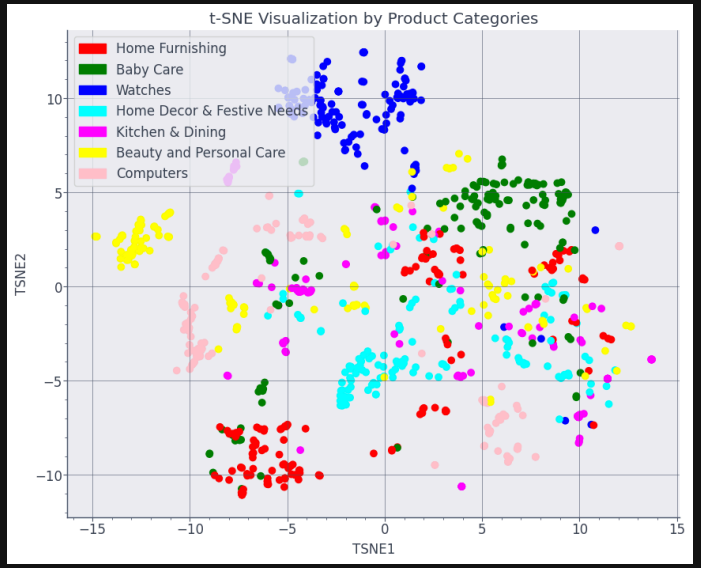
- Word2Vec transforme chaque mot d'un texte en un vecteur de nombres, capturant des **caractéristiques sémantiques**.
- **similarité sémantique** : Les mots ayant des contextes similaires se retrouvent proches dans l'espace vectoriel

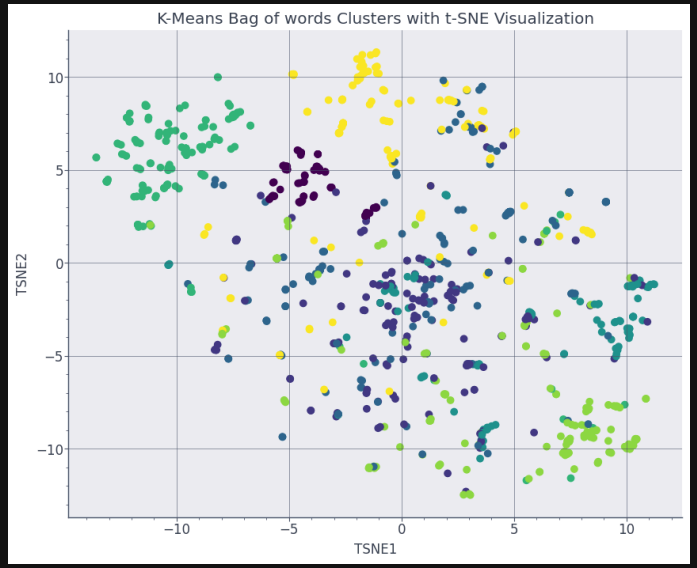


Score ARI : 0.4364 Assez bonne séparation des catégories

Approche Bag of words : BERT (Bidirectional Encoder Representations from Transformers)

- Réseau de neuronne pré-entrainé basé sur l'architecture transformers
- **Pré-entraînement bidirectionnel** : BERT apprend le contexte des mots à la fois avant et après chaque mot.Ce qui permet de mieux capture le contexte et sens des phrase.

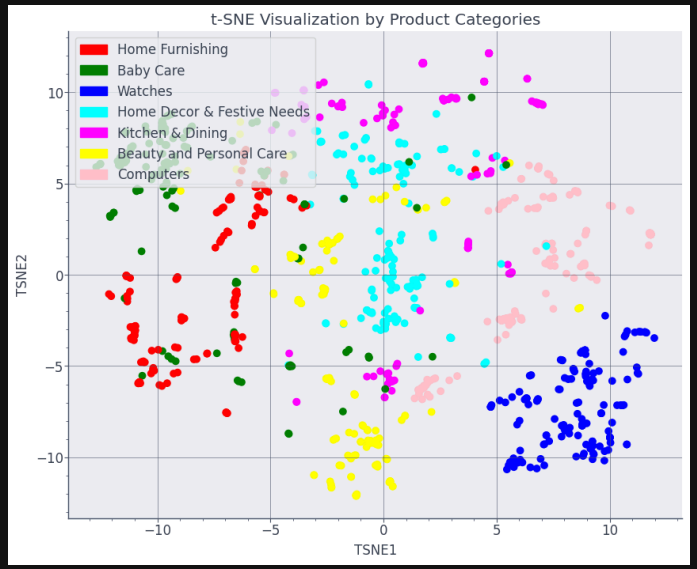


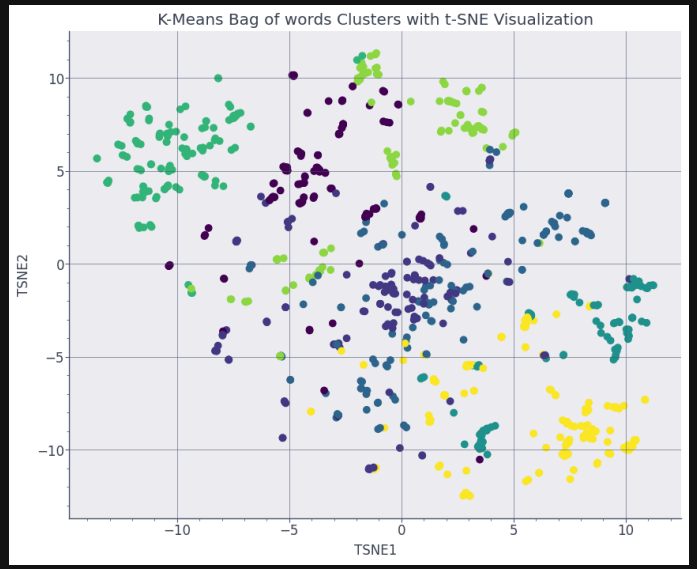


Score ARI : 0.3251 Séparation partielle des catégories

Approche Bag of words : USE (Universal Sentence Encoder)

- USE produit une représentation vectorielle dense de chaque phrase
- **Représentation sémantique** : Les vecteurs produits par USE sont créés pour que les phrases similaires (sémantiquement proches) aient des vecteurs proches dans l'espace vectoriel.



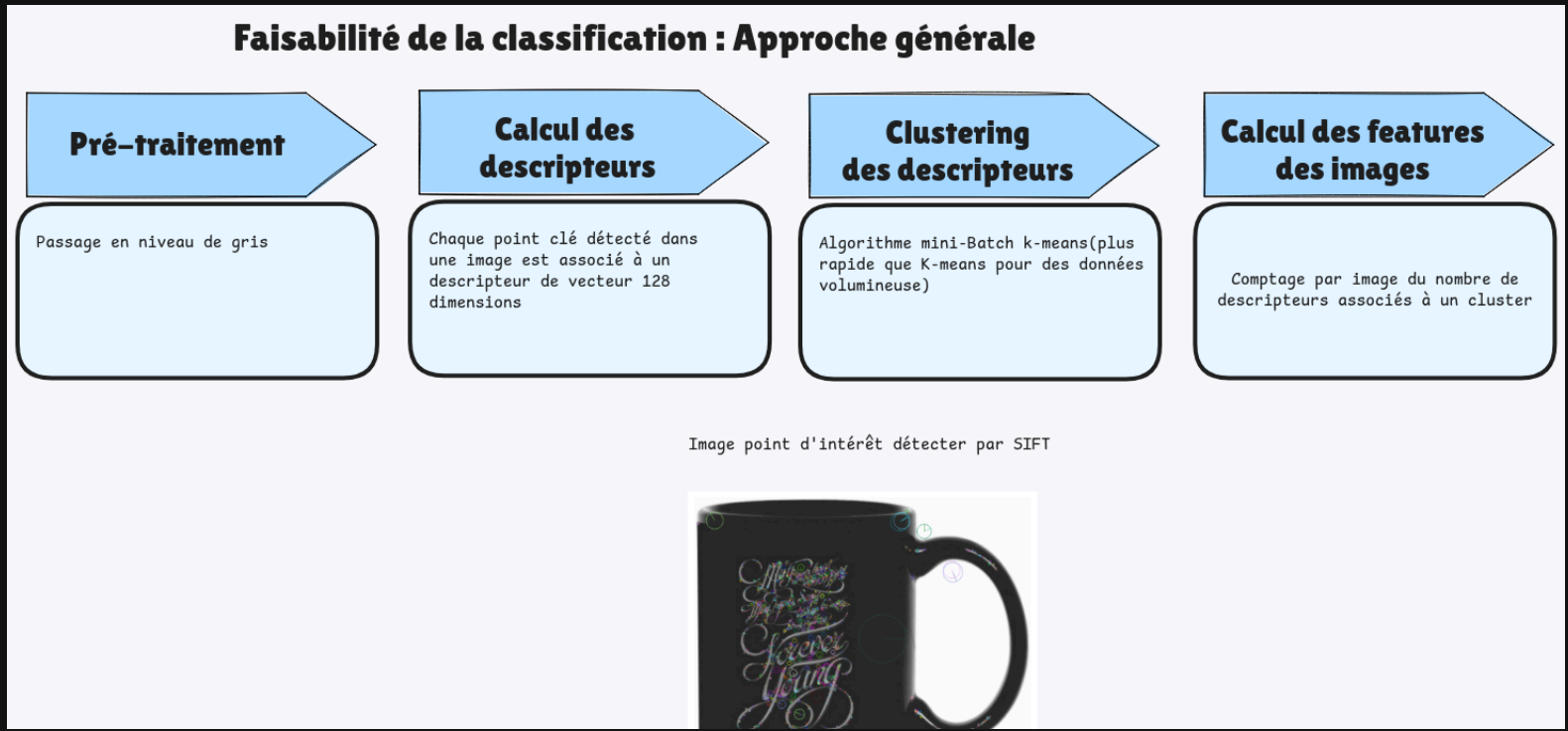


Score ARI : 0.5122 Bonne séparation des catégories

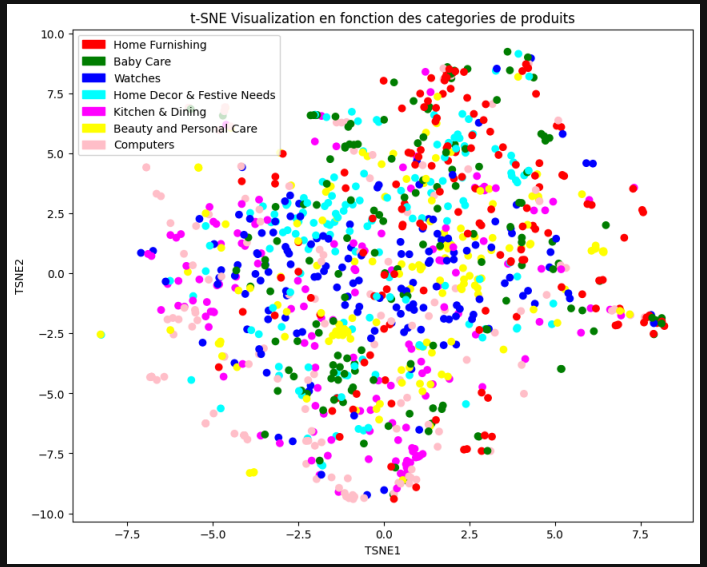
III.Présentation des approches de modélisation de classification des images

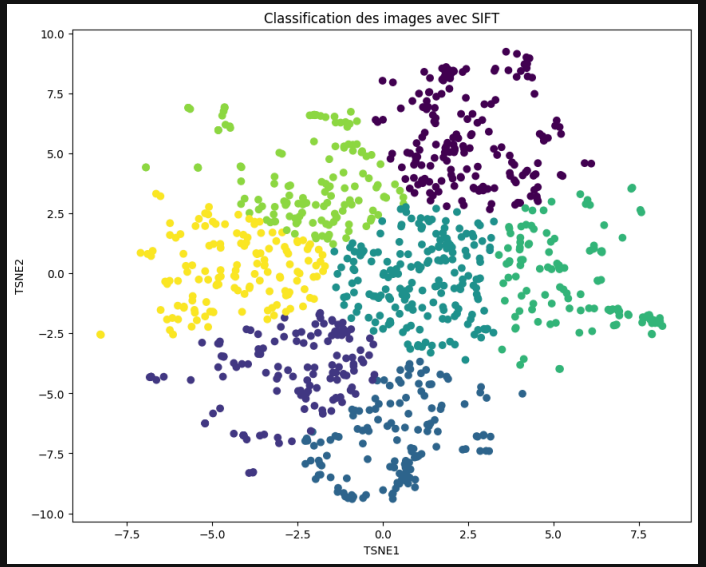
Approche SIFT

- **SIFT** : alorithme du domaine de la vision par ordinateur de reconnaissance de caractéristiques(feature détection).Il permet de **détecter et d'extraire des descripteurs de points clés dans une image** (bord,contours et point d'intérêt)qui sont invariant aux variations d'échelle et à la rotation.



Approche Bag of words : SIFT (Bidirectional Encoder Representations from Transformers)





Score ARI : 0.04989 SIFT ne permet pas la séparation des catégories

Approche CNN transfer Learning et data augmentation avec VGG 16

- **CNN réseau de neurones** conçu pour traiter des données ayant une structure de grille, comme les images:
  - **Couches de convolution**: elles extraient automatiquement des caractéristiques importantes des images (comme les bords, textures, motifs).
  - **Couches de pooling** : elles réduisent la taille des images pour diminuer le nombre de calculs, tout en gardant les informations principales.
  - **Couches entièrement connectées** : en fin de réseau, elles combinent les caractéristiques extraites pour classer ou interpréter l'image.

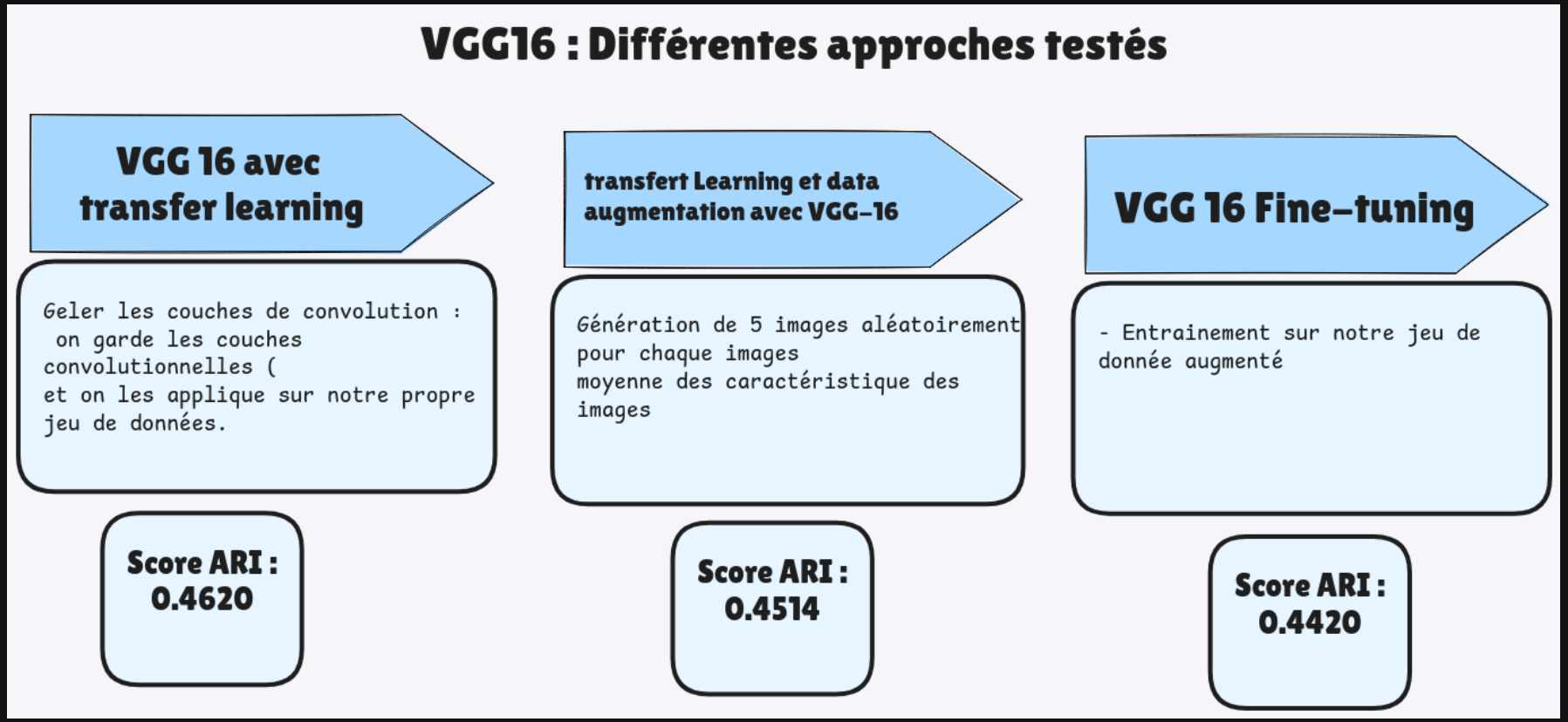
VGG16: 13 couches de convolution et 3 couches entièrement connectées entraîné sur l'ensemble des données ImageNET

Transfer Learning : Le Transfer Learning consiste à utiliser un modèle pré-entraîné et à adapter ce modèle pour une tâche spécifique avec moins de données et de temps d'entraînement.

Data Augmentation : La Data Augmentation consiste à créer de nouvelles images d'entraînement en appliquant des transformations sur les images existantes:

- Rotations, translations, zooms, et inversions.
- Changements de luminosité ou d'échelle de couleurs.
- Découpes aléatoires ou ajouts de bruit.

Approche CNN transfer Learning et data augmentation avec VGG 16



IV.Utilisation de l’api pour récupération de produit

Création du script avec api pour récupération des produits

```
# Script for API

# Term to search
query = 'champagne'

# URL of API
url = 'https://api.edamam.com/api/food-database/v2/parser'

# parameter query
params = {
    'app_id': app_id,
    'app_key': app_key,
    'ingr': query,

}

# Send request GET
response = requests.get(url, params=params)

# Verify statut of response
if response.status_code == 200:
    data = response.json()
    # Extract first 10 products
    hints = data.get('hints', [])[:10]
    # Prepare data
    products = []
    for hint in hints:
        food = hint.get('food', {})
        products.append({
            'foodId': food.get('foodId', ''),
            'label': food.get('label', ''),
            'category': food.get('category', ''),
            'foodContentsLabel': food.get('foodContentsLabel', ''),
            'image': food.get('image', '')
        })
    # Create dataframe
    df = pd.DataFrame(products)
    # Save dataframe into csv file
    df.to_csv('edamam_products.csv', index=False)
    print('Les données ont été enregistrées dans edamam_products.csv')
else:
    print(f'Erreur {response.status_code} : {response.text}')
```

V.Conclusion du Projet

Faisabilité du moteur de classification

- L’analyse graphique et du score ARI nous permis qu’il est réailisable de séparer automatiquement les produits selon leurs vraies catégories avec leurs nom/description et des images