# Classifiez automatiquement des biens de consommation

PLE Coline

# Plan de la présentation

- Présentation de la problématique
- Présentation du jeu de données
- Etude de faisabilité
- Classification supervisée
- Test de l'API
- Discussion

# RAPPEL DE LA PROBLEMATIQUE


## PROBLEMATIQUE ET MISSIONS

**Contexte:**
Proposition par des vendeurs d'articles à des acheteurs en postant une photo et une description.

**Problème:**
Attribution manuelle par les vendeurs pour la catégorie d'un article d'où un manque de fiabilité.

**Besoin:**
Automatisation de la tâche afin de faciliter l'expérience utilisateur des vendeurs et des acheteurs.

**Missions:**
1. Etude de faisabilité d'un moteur de classification pour la catégorisation des produits selon la description du produit et/ou de son image
2. Classification supervisée à partir des images des produits
3. Importer des produits *via* une API

# PRESENTATION DU JEU DE DONNEES

```
# Informations sur le jeu de données
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1050 entries, 0 to 1049
Data columns (total 15 columns):
 #   Column                  Non-Null Count  Dtype
---  ------                  --------------  -----
 0   uniq_id                 1050 non-null   object
 1   crawl_timestamp         1050 non-null   object
 2   product_url             1050 non-null   object
 3   product_name            1050 non-null   object
 4   product_category_tree   1050 non-null   object
 5   pid                     1050 non-null   object
 6   retail_price            1049 non-null   float64
 7   discounted_price        1049 non-null   float64
 8   image                   1050 non-null   object
 9   is_FK_Advantage_product 1050 non-null   bool
 10  description             1050 non-null   object
 11  product_rating          1050 non-null   object
 12  overall_rating          1050 non-null   object
 13  brand                   712 non-null    object
 14  product_specifications  1049 non-null   object
dtypes: bool(1), float64(2), object(12)
memory usage: 116.0+ KB
```

## Les informations à retenir sur le dataset:

1. Présence de 1050 articles (absence de doublon)
2. 4 variables d'intérêt
3. 7 catégories principales (1ère branche de l'arbre des catégories)

| product_category_tree |
| --- |
| Home Furnishing >> Curtains & Accessories >> C... |
| Baby Care >> Baby Bath & Skin >> Baby Bath Tow... |

## Un dossier image:

1. Présence de 1050 images en format .jpg
2. Une image par produit

## Information importante:

Textes et images ne relevant pas d'une propriété intellectuelle dont l'utilisation ou la modification est interdite

PS : J'ai bien vérifié qu'il n'y avait aucune contrainte de propriété intellectuelle sur les données et les images.

# ETUDE DE FAISABILITE SUR LE TEXTE:
# Prétraitement des données de la target

## 1ère étape: Création de la variable 'main_category'

- Elimination des crochets
- Elimination des guillemets
- Split de l'arbre afin d'extraire la 1ère branche

```
# Elimination des crochets, des guillemets et création de la variable 'main_category'
data["product_category_tree"] = data["product_category_tree"].str.replace("[", "")
data["product_category_tree"] = data["product_category_tree"].str.replace("]", "")
data["product_category_tree"] = data["product_category_tree"].str.replace('"', '')
data["main_category"] = data["product_category_tree"].str.split('>>').str[0]
```

## 2ème étape: Léger traitement du texte

- Mise en minuscule et remplacement des & par des and
- Vérification du contenu textuel de la variable

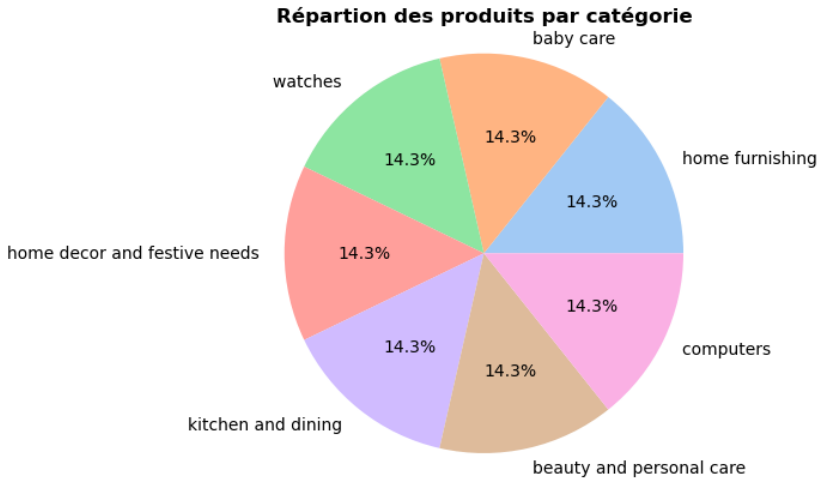
```
print(data["main_category"].unique().tolist())
```

```
['home furnishing ', 'baby care ', 'watches ', 'home decor and festive needs ', 'kitchen and dining ', 'beauty and personal car
e ', 'computers ']
```

## 3ème étape : Analyse de la target

```
# Comptage du nombre d'articles
data["main_category"].value_counts()
```

```
home furnishing                150
baby care                      150
watches                        150
home decor and festive needs   150
kitchen and dining             150
beauty and personal care       150
computers                      150
Name: main_category, dtype: int64
```



# ETUDE DE FAISABILITE SUR LE TEXTE:
# Prétraitement des données sur la description des produits

## 1ère étape: Choix de ne conserver que la variable 'description'

- Variable plus détaillée que la variable 'product_name'
- Contenu de la variable 'product_name' présent dans la variable 'description'

```
Contenu de la variable 'product_name' pour le deuxième produit
------------------------------------------------------------
Sathiyas Cotton Bath Towel

Contenu de la variable 'description' pour le deuxième produit
------------------------------------------------------------
Specifications of Sathiyas Cotton Bath Towel (3 Bath Towel, Red, Yellow, Blue) Bath Towel Features Machine Washable Yes Materia
l Cotton Design Self Design General Brand Sathiyas Type Bath Towel GSM 500 Model Name Sathiyas cotton bath towel Ideal For Men,
Women, Boys, Girls Model ID asvtwl322 Color Red, Yellow, Blue Size Mediam Dimensions Length 30 inch Width 60 inch In the Box Nu
mber of Contents in Sales Package 3 Sales Package 3 Bath Towel
```

## 2ème étape: Traitement du texte

- Mise en minuscule
- Elimination des URL, des éléments HTML et des caractères non ascii
- Tokenization *via* le RegexpTokenizer
- Elimination des stopwords anglais et des mots non anglais (words.words())
- Elimination des tokens de moins de 3 caractères
- Elimination des mots présents qu'une seule fois
- Elimination des mots possédant un mélange de chiffres et de lettres
- Elimination des stopwords 'spécial projet' = mots communs aux 7 catégories
- Lemmatisation
- Jointure des tokens

- Tentative infructueuse: Conservation unique des noms et des verbes

# ETUDE DE FAISABILITE SUR LE TEXTE:
# Feature extraction et faisabilité

## APPROCHE DEMANDEE

**1ère étape: Extraction des features et encodage par diverses méthodes**

- **Le Bag of Words** : Comptage simple (term frequency) et tf-idf (Term Frequency- Inverse Document Frequency)
- **Le Word Embedding** : Word2Vec, BERT (HuggingFace et Hub Tensorflow) et USE

**2ème étape: Réduction dimensionnelle**

- **PCA avec 99% de la variance expliquée** (décorrélation des variables entre elles et accélération de la T-SNE)
- **T-SNE à 2 composantes** (visualisation graphique)

**3ème étape: Analyse de la faisabilité**

- **Visualisation selon les vraies catégories** (résultats obtenus après T-SNE)
- **K-means avec 7 clusters** (similarité classes réelles/classes obtenues après segmentation)
- **Calcul du score ARI** (confirmation de l'analyse visuelle)

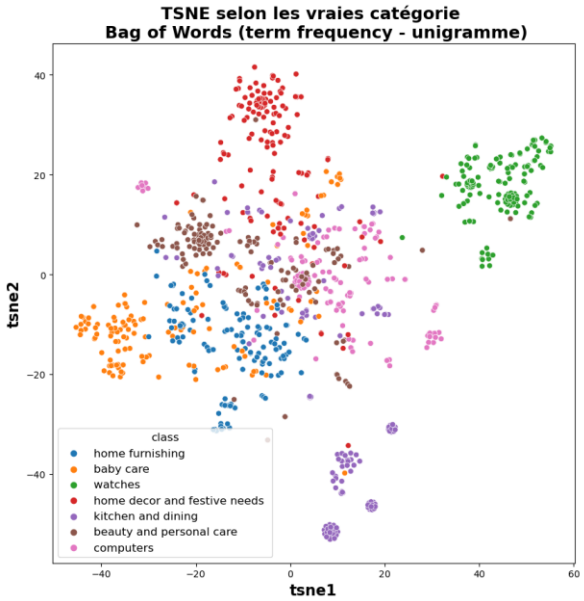
**A NOTER: Comptage réalisé en unigramme ou bigramme**

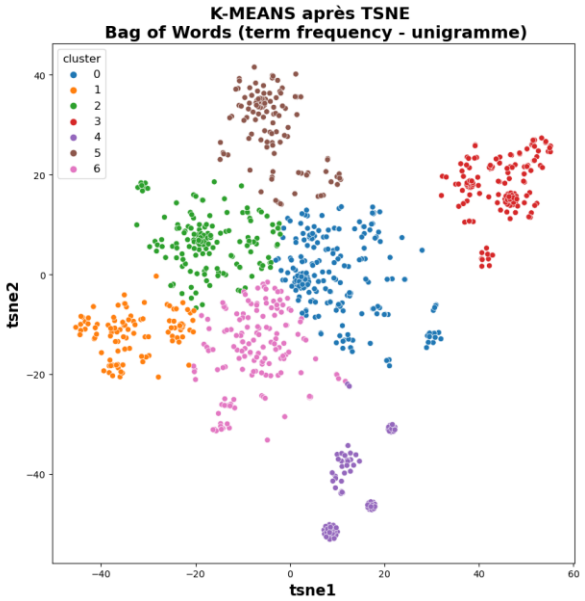
- **Meilleurs résultats en unigramme**
- **Présentation uniquement des résultats obtenus en unigramme**

## APPROCHE INCONTOURNABLE:
## Algorithmes de classification supervisée

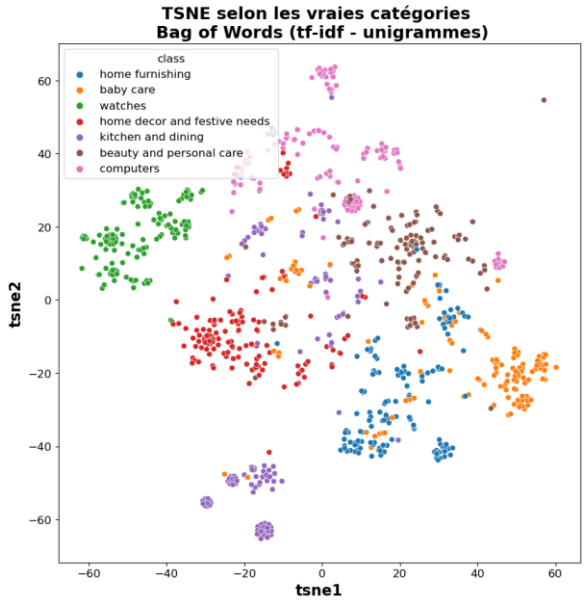
- **LogisticRegression**
- **LinearSVC**
- **MultinomialNB**
- **KNeighborsClassifier**
- **RandomForestClassifier**

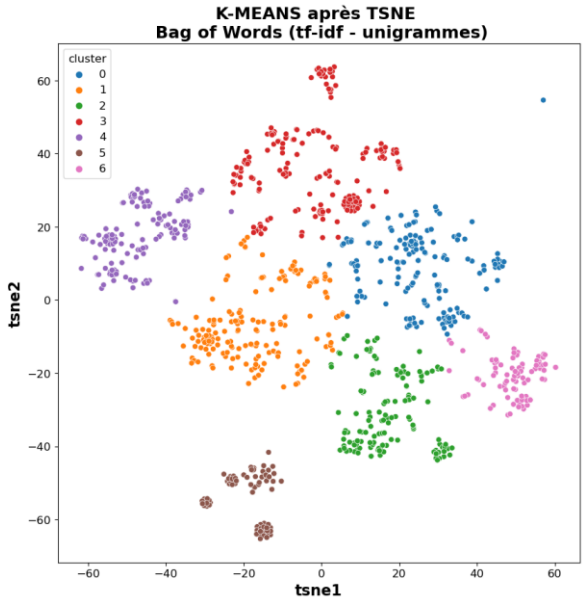
# ETUDE DE FAISABILITE SUR LE TEXTE:
## Le Bag of Words




**ARI après simple comptage**
**0.485**

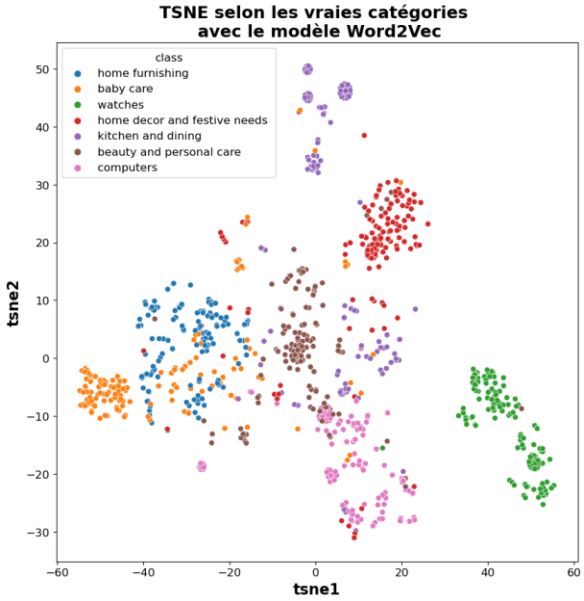


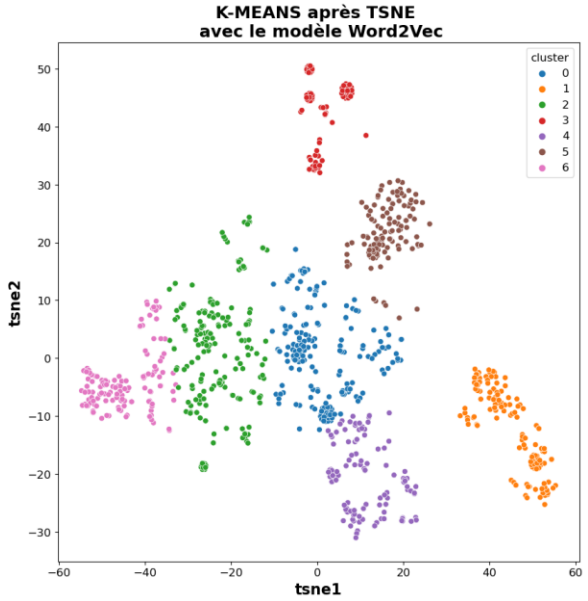




**ARI après tf-idf**
**0.570**

Très satisfaisant…
Quid avec du Word Embedding?

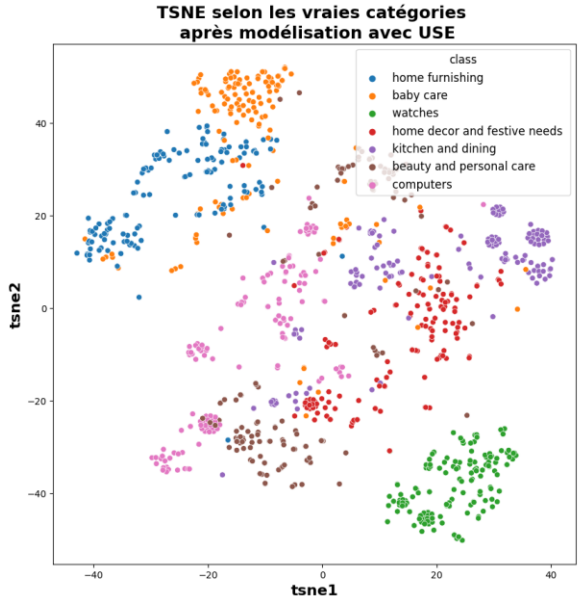
# ETUDE DE FAISABILITE SUR LE TEXTE:
# Le Word Embedding: Word2Vec et USE

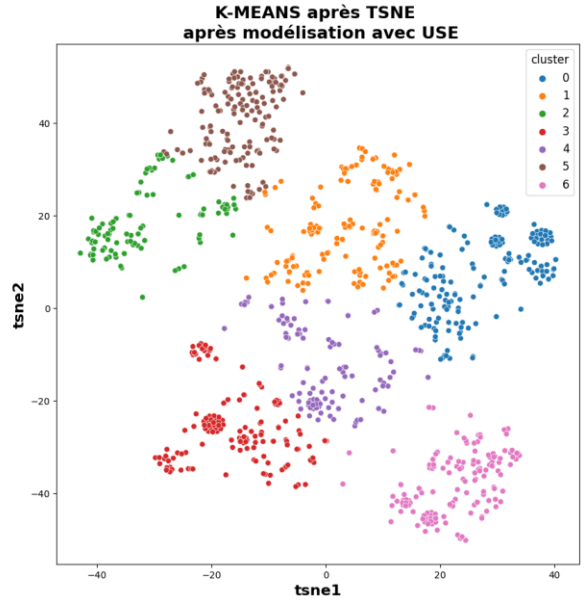


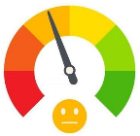

**ARI après Word2Vec**
**0.500**

Résultat moins bon que le tf-idf (0.570)




**ARI après USE**
**0.420**



Quid des algorithmes de classification supervisée

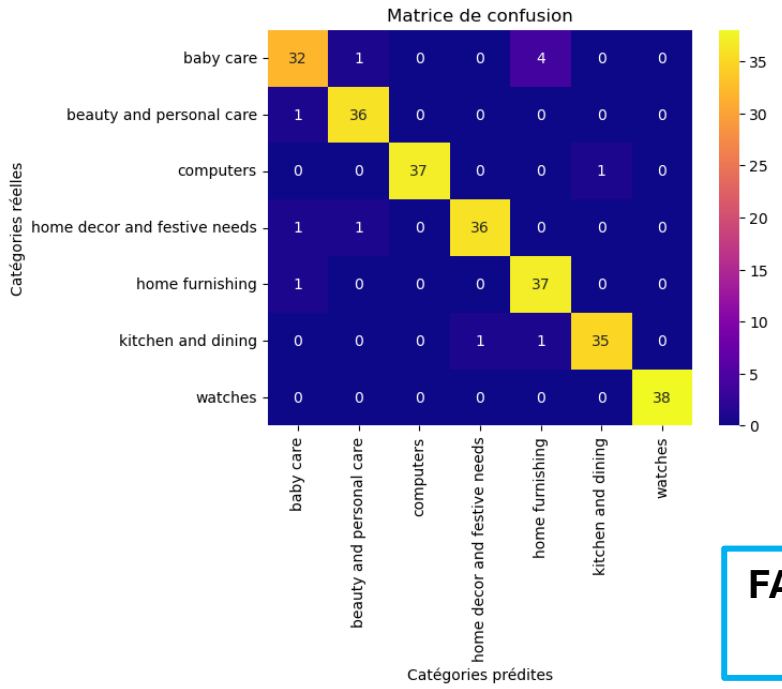
# ETUDE DE FAISABILITE SUR LE TEXTE:
# Les algorithmes de classification supervisée

Comparaison des différents modèles pour la prédiction de la classification des produits selon leur catégorie

| | LogisticRegression | LinearSVC | MultinomialNB | KNeighborsClassifier | RandomForestClassifier |
|---|---|---|---|---|---|
| Training Time | 0.47918 | 0.721695 | 0.893906 | 1.077331 | 3.047266 |
| Grid Training Time | 0.562278 | 0.818171 | 1.006015 | 1.198639 | 5.17468 |
| Train Score | 97.8399 | 98.7294 | 96.4422 | 93.0114 | 99.1105 |
| Grid Train Score | 97.8399 | 98.094 | 97.2046 | 99.1105 | 97.967 |
| Train erreurs (%) | 2.1601 | 1.2706 | 3.5578 | 6.9886 | 0.8895 |
| Grid Train erreurs (%) | 2.1601 | 1.906 | 2.7954 | 0.8895 | 2.033 |
| Test Score | 95.4373 | 94.2966 | 92.7757 | 90.4943 | 90.4943 |
| Grid Test Score | 95.4373 | 94.2966 | 91.635 | 91.635 | 92.7757 |
| Test erreurs (%) | 4.5627 | 5.7034 | 7.2243 | 9.5057 | 9.5057 |
| Grid Test erreurs (%) | 4.5627 | 5.7034 | 8.365 | 8.365 | 7.2243 |

## La régression logistique

- Meilleur apprenant
- Pas de sur-apprentissage
- Résultat surpassant les précédentes méthodes




## Le rapport de classification

| | precision | recall | f1-score | support |
|---|---|---|---|---|
| baby care | 0.91 | 0.86 | 0.89 | 37 |
| beauty and personal care | 0.95 | 0.97 | 0.96 | 37 |
| computers | 1.00 | 0.97 | 0.99 | 38 |
| home decor and festive needs | 0.97 | 0.95 | 0.96 | 38 |
| home furnishing | 0.88 | 0.97 | 0.93 | 38 |
| kitchen and dining | 0.97 | 0.95 | 0.96 | 37 |
| watches | 1.00 | 1.00 | 1.00 | 38 |
| accuracy | | | 0.95 | 263 |
| macro avg | 0.96 | 0.95 | 0.95 | 263 |
| weighted avg | 0.96 | 0.95 | 0.95 | 263 |

**FAISABILITE DE CLASSIFICATION SUR LA DESCRIPTION**

**OUI**

# ETUDE DE FAISABILITE SUR LES IMAGES:
## Test de prétraitement des images avec Pillow

**Image d'origine**

**Passage en nuance de gris**

**Egalisation de l'image (Contraste)**

**Etirement de l'image**

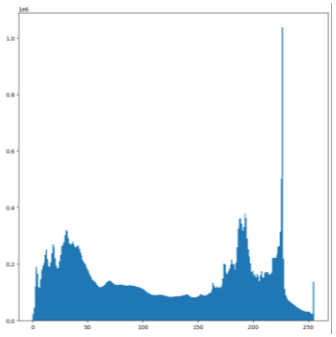


Image assez bien exposée

**Filtre gaussien (Radius 20)**

**Filtre gaussien (Radius 2)**

**Filtre médian (Taille 3)**

Très satisfaisant…
Feature extraction

# ETUDE DE FAISABILITE SUR LES IMAGES:
## Extraction de features via la génération de descripteurs (SIFT)

**1ère étape:**

Traitement des images avec Pillow (sans passage en nuances de gris)

**openCV** **Pillow**

**2ème étape:**

Création des descripteurs
Création des clusters de descripteurs
Extraction des features

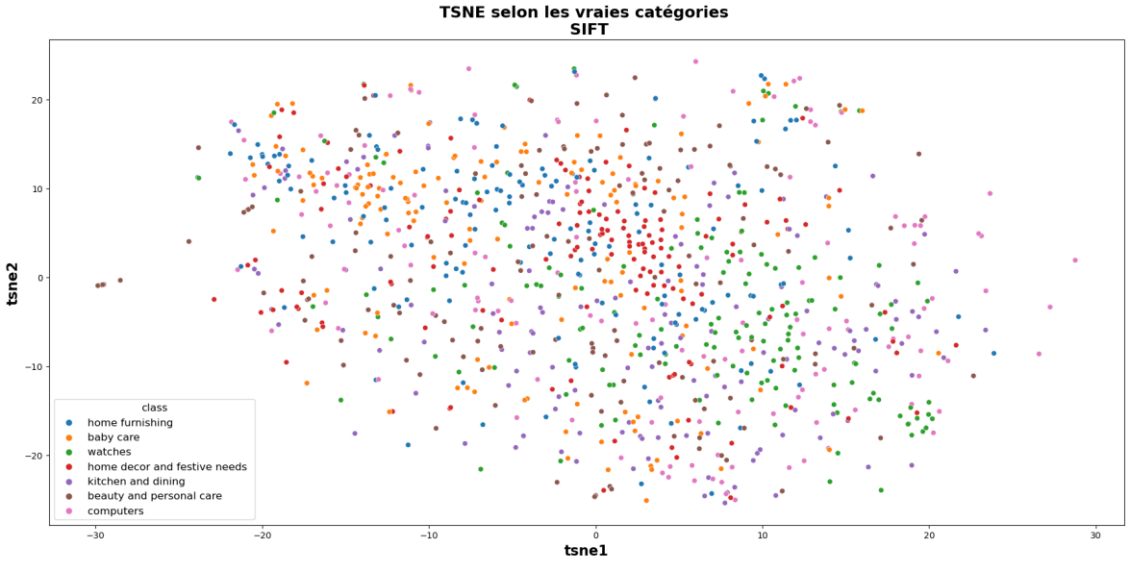
**3ème étape:**

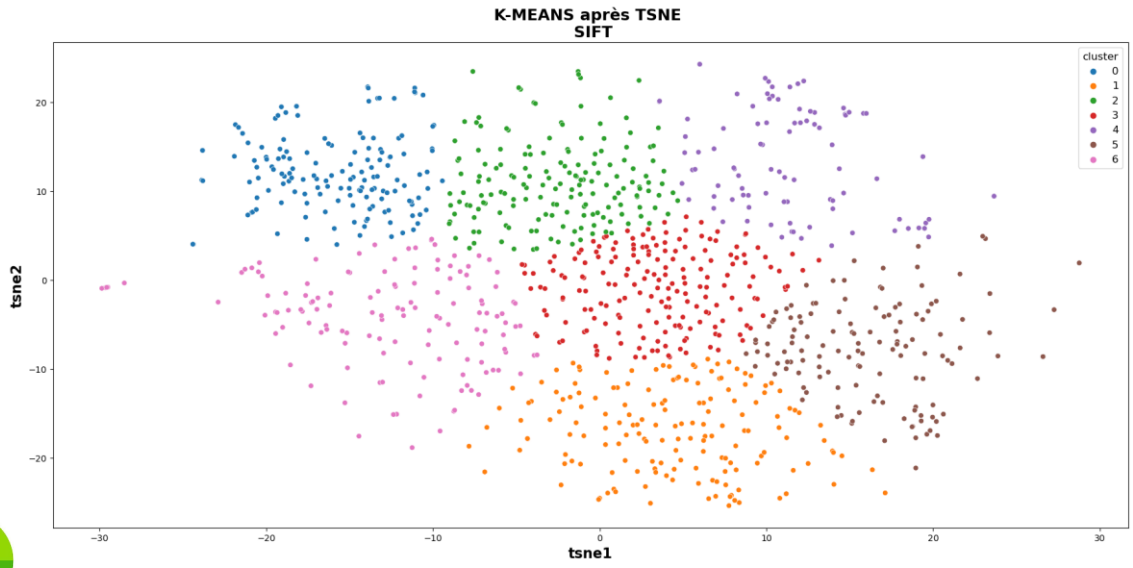
Réduction dimensionnelle PCA+T-SNE

**4ème étape:**

- Visualisation selon les vraies catégories (T-SNE)
- K-means avec 7 clusters
- Calcul du score ARI

Algorithme de Transfer Learning basé sur des réseaux de neurones (CNN)???




**ARI après SIFT: 0.039**

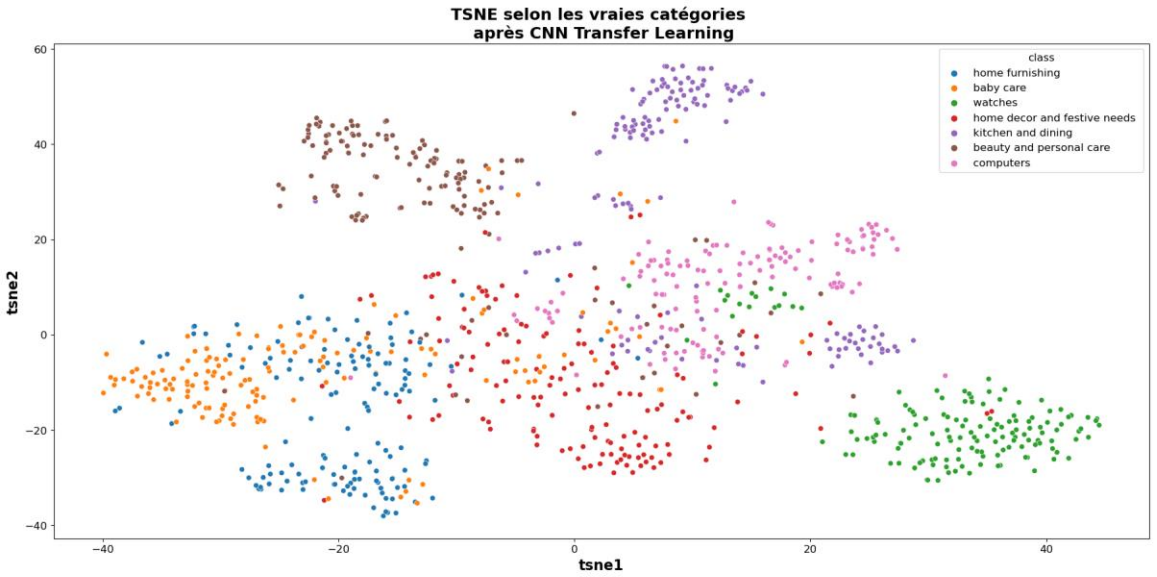
# ETUDE DE FAISABILITE SUR LES IMAGES:
## Extraction de features via Transfer Learning (VGG16)

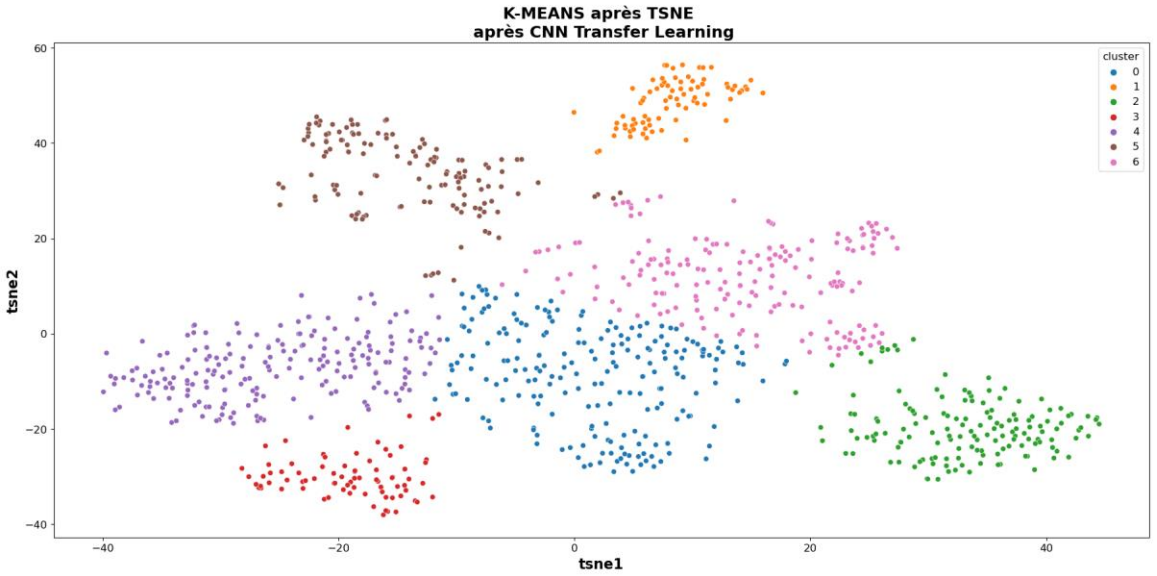
**1ère étape:**
Création d'un modèle pré-entraîne

**2ème étape:**
Création des features des images

**3ème étape:**
Réduction dimensionnelle PCA+T-SNE

**4ème étape:**
Visualisation selon les vraies catégories (T-SNE)
K-means avec 7 clusters
Calcul du score ARI




**FAISABILITE DE CLASSIFICATION SUR L'IMAGE**
**OUI**

**ARI après Transfer Learning : 0.464**

# CLASSIFICATION SUPERVISEE SUR LES IMAGES:
# Utilisation du VGG16 avec ou sans data augmentation

**Contexte:** Développement d'un modèle de classification de produits grâce aux images

**Modèle pré-entrainé utilisé:** VGG16 avec imagenet

- imagenet: Regroupement de 14 millions d'images appartenant à 1000 classes
- Précision du test dans ImageNet : 92.7%

## Stratégie de séparation du dataset

- 70% dans le jeu d'entrainement : 735 images
- 15% dans le jeu de validation: 157 images
- 15% dans le jeu test (~nouvelles données): 158 images

## Paramètres testés

- epoch de 50 avec possibilité de stopper avant (patience sur la val_loss à 5)
- Taille du batch size: 32 ou 64 (nombre d'échantillons utilisés lors d'une itération)
- Activation : softmax (plus de 2 classes en sortie)
- Loss: categorical_crossentropy (plus de 2 classes en sortie)
- Optimizer: rmsprop ou adam
- Dropout (métrique évitant l'over-fitting comprise entre 0 et 1) : 0,5
- Dense Layer de 256 unités

## Paramètres additionnels pour la data augmentation

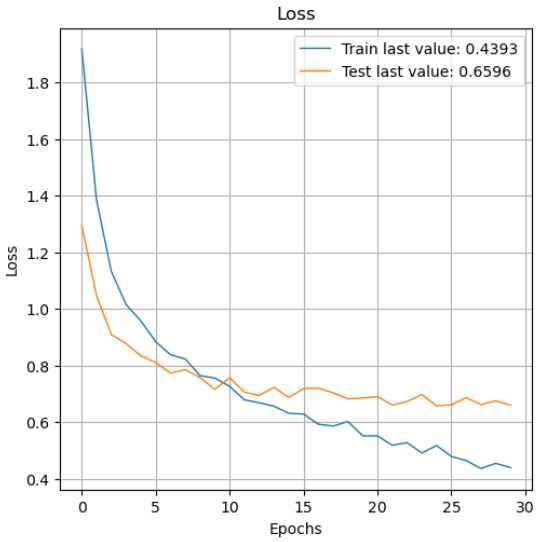
- Rotation de l'image: 0,1
- Zoom de l'image: x0,1
- Retournement de l'image (Flip): horizontal
- Redimensionnement de l'image : 1/127,5

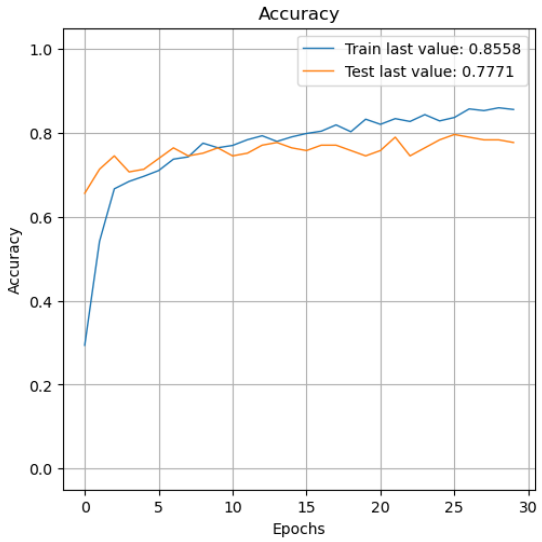
# CLASSIFICATION SUPERVISEE SUR LES IMAGES:
## Tableau récapitulatif des différents tests

| Optimizer | Batchs size | Data augmentation | Validation accuracy / loss | Test accuracy / Loss |
|---|---|---|---|---|
| rmsprop | 64 | Non | 80,89% / 0.9137 | 82,91% / 0.7279 |
| | | Oui | 78,34% / 0.6575 | 86,08% / 0.5099 |
| | 32 | Non | 80,89% / 0.6907 | 80,38% / 0.7117 |
| | | Oui | 78,34% / 0.6508 | 82,28% / 0.5178 |
| adam | 64 | Non | 83,44% / 0.8869 | 82,28% / 0.7255 |
| | | Oui | 78,34% / 0.6762 | 82,28% / 0.5693 |
| | 32 | Non | 80,89% / 0.9473 | 83,54% / 0.6439 |
| | | Oui | 77,71% / 0.6742 | 84,81% / 0.5366 |

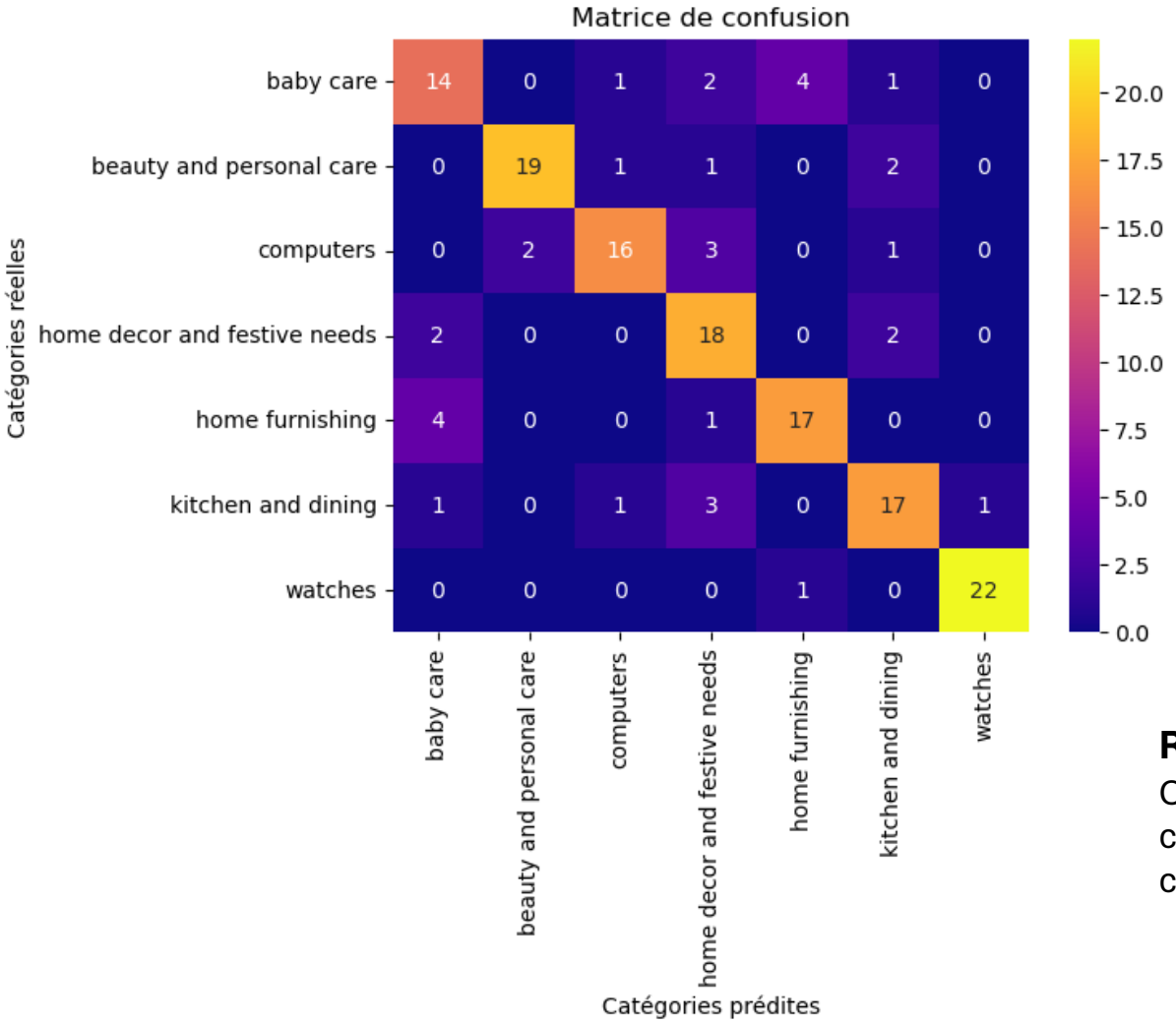

**Points importants:**

- Modèle retenu : VGG16 avec data augmentation, l'optimizer rmsprop et des batch de 64
- Résultat correct au regard du nombre d'images
- La data augmentation permet surtout d'éviter le sur-apprentissage



# CLASSIFICATION SUPERVISEE SUR LES IMAGES:
## Le modèle VGG16 choisi




**Remarques**

Confusion la plus importante entre les catégories ‘baby care’ et ‘home furnishing’ comme pour les données textuelles

# ELARGISSEMENT DE LA GAMME DE PRODUITS:
## Extraction de produits via l'API

**Contexte:** Elargissement de la gamme de produits 'épicerie fine' sur place de marché

**Objectifs:** Extraction des 10 premiers produits à base de champagne d'une API

**API et Database:** Rapid API / Edamam Food and Grocery Database (~900 000 aliments)

Edamam provides free Food API access with its basic plan for developers, startups and non-profits alike.

Enterprise customers are charged a very low monthly and per call fee based on usage. Custom packages are also available.

| Column1 | Column2 | Column3 | Column4 | Column5 |
|---|---|---|---|---|
| Food Id | Label | Category | Food Contents Label | image |
| food_a656mk2a5dmqb2adiamu6beihduu | Champagne | Generic foods | N/A | https://www.edamam.com/food-img/a71/a718cf3c52a... |
| food_b753ithamdb8psbt0w2k9aquo06c | Champagne Vinaigrette, Champagne | Packaged foods | OLIVE OIL; BALSAMIC VINEGAR; CHAMPAGNE VINEGAR... | N/A |
| food_b3dyababjo54xobm6r8jzbghjgqe | Champagne Vinaigrette, Champagne | Packaged foods | INGREDIENTS: WATER; CANOLA OIL; CHAMPAGNE VINE... | https://www.edamam.com/food-img/d88/d88b64d973... |
| food_a9e0ghsamvoc45bwa2ybsa3gken9 | Champagne Vinaigrette, Champagne | Packaged foods | CANOLA AND SOYBEAN OIL; WHITE WINE (CONTAINS S... | N/A |
| food_an4jjueaucpus2a3u1ni8auhe7q9 | Champagne Vinaigrette, Champagne | Packaged foods | WATER; CANOLA AND SOYBEAN OIL; WHITE WINE (CO... | N/A |
| food_bmu5dmkazwuvpaa5prh1daa8jxs0 | Champagne Dressing, Champagne | Packaged foods | SOYBEAN OIL; WHITE WINE (PRESERVED WITH SULFITE... | https://www.edamam.com/food-img/ab2/ab2459fc2a... |
| food_alpl44taoyv11ra0lic1qa8xculi | Champagne Buttercream | Generic meals | sugar; butter; shortening; vanilla; champagne; milk | N/A |
| food_byap67hab6evc3a0f9w1oag3s0qf | Champagne Sorbet | Generic meals | Sugar; Lemon juice; brandy; Champagne; Peach | N/A |
| food_am5egz6aq3fpjlaf8xpkdbc2asis | Champagne Truffles | Generic meals | butter; cocoa; sweetened condensed milk; vanilla extra... | N/A |
| food_bcz8rhiajk1fuva0vkfmeakbouc0 | Champagne Vinaigrette | Generic meals | champagne vinegar; olive oil; Dijon mustard; shallot; ho... | N/A |

**Catégorie à renommer**

**Peu d'images**

**Données textuelles OK**

# CONCLUSION

**Faisabilité de la classification des produits selon leur description**

- Faisabilité prouvée par 3 approches différentes:
    1. Le Bag of Words avec comptage tf-idf (unigramme)
    2. Un Word Embedding (Word2Vec)
    3. La Régression logistique (un modèle de classification supervisée): Performance de 95%
- Amélioration possible *via* l'élimination de mots communs pour les catégories présentant le plus d'erreurs

**Faisabilité de la classification des produits selon leur image :** OUI

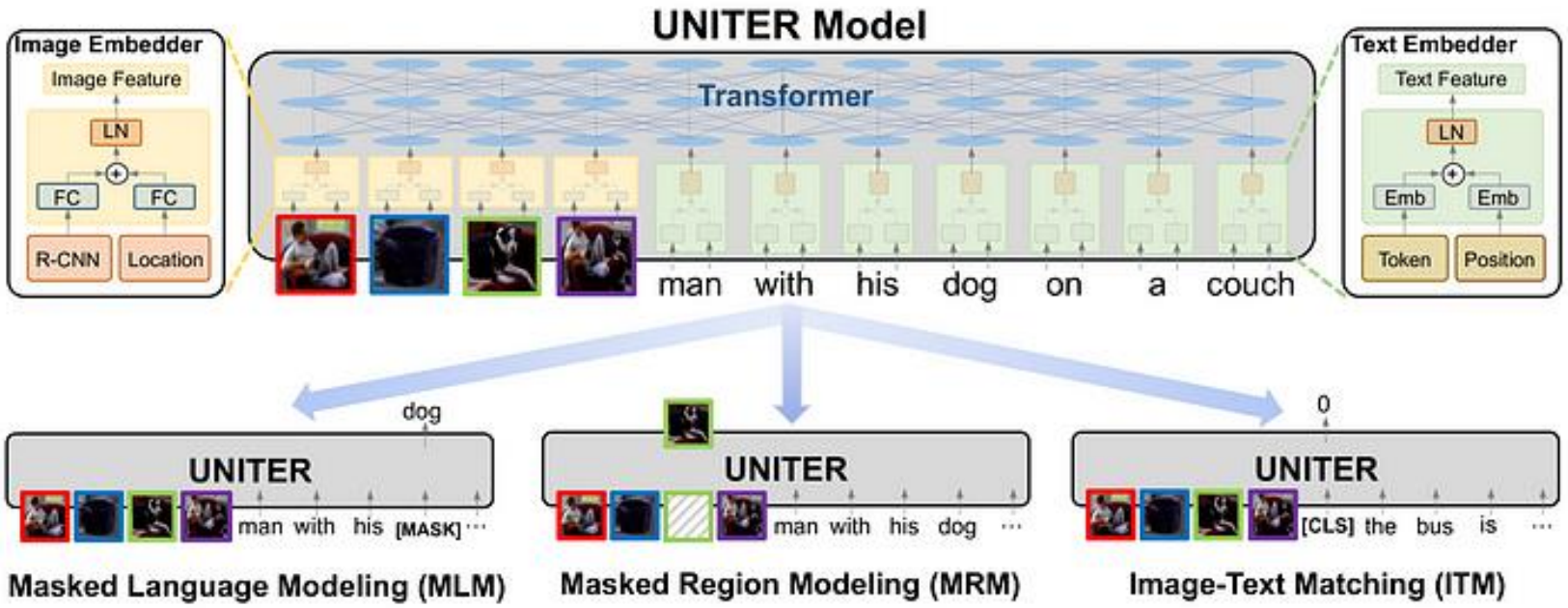
- Prouvée uniquement par DeepLearning (CNN Transfer Learning)
- Opinion: Atteinte d'une moins bonne classification vu le peu d'images à notre disposition, et ce même avec une data augmentation
    1. Résultats obtenus avec le modèle VGG16 (batch de 64, data augmentation, optimizer 'rmsprop et une couche dense de 256 neurones): 78% en validation et 86% en test.
    2. Possibilité d'amélioration par recherche des meilleurs paramètres (tuner de Keras)
    3. Possibilité d'amélioration par une approche Vision Transformer (ViT) (source: https://viso.ai/deep-learning/vision-transformer-vit/)

**Extension de la gamme de produits via RapidApi :**

- Envisageable pour les descriptions de produits
- Plus difficile pour les images
- PAYANT

# POUR ALLER PLUS LOIN
## Combinaison des données textuelles et des images

### UNITER (UNiversal Image-TExt Representation Learning)

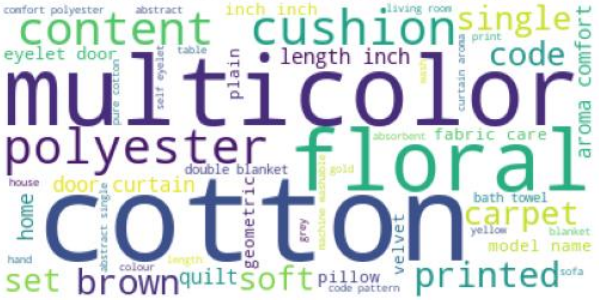



(sources : https://arxiv.org/abs/1909.11740 , https://towardsdatascience.com/uniter-d979e2d838f0 )

# - Débat / Réflexion -

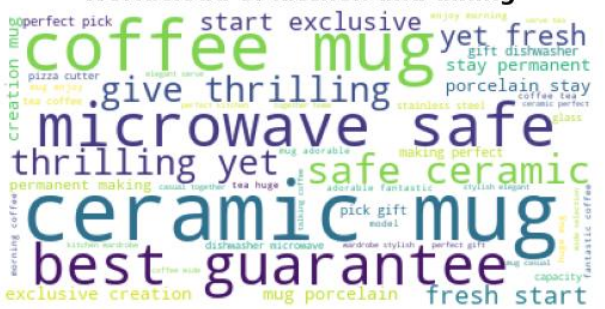
# Informations supplémentaires: WordCloud des 50 mots les plus utilisés par catégorie

### Worldcloud of home furnishing




### Worldcloud of home decor and festive needs


### Worldcloud of kitchen and dining

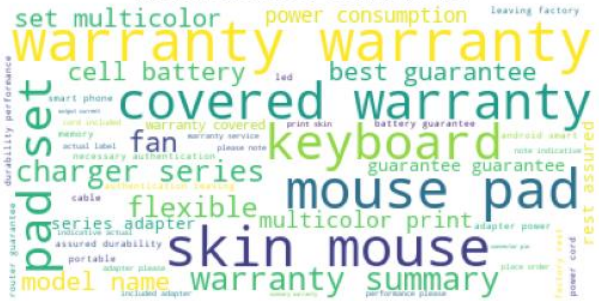



### Worldcloud of baby care

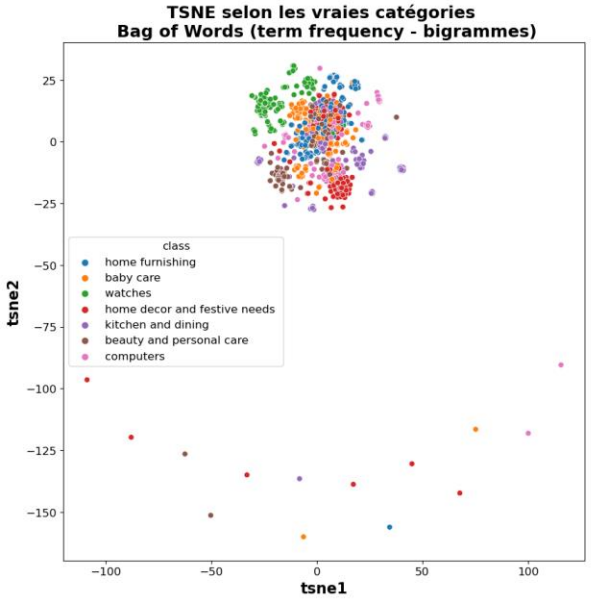

### Worldcloud of beauty and personal care

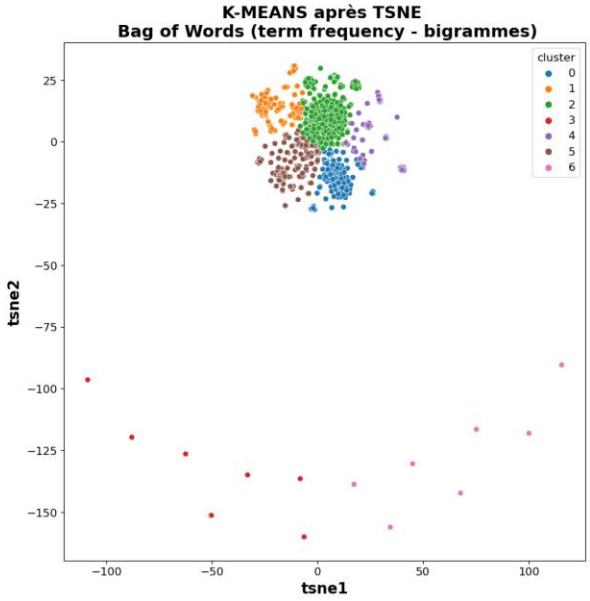

### Worldcloud of watches


### Worldcloud of computers

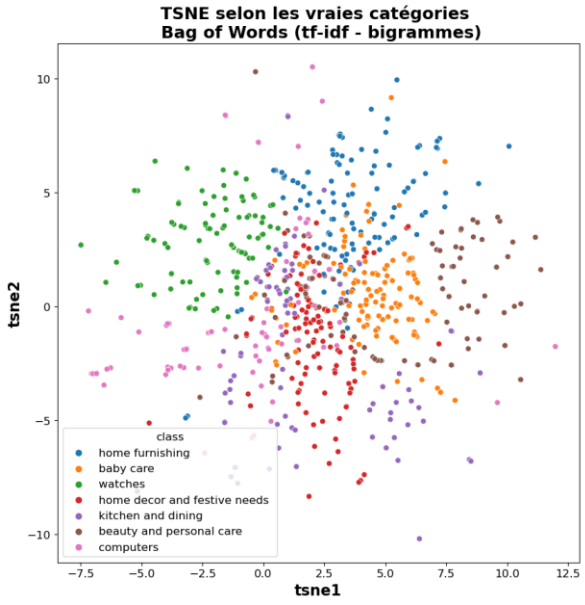


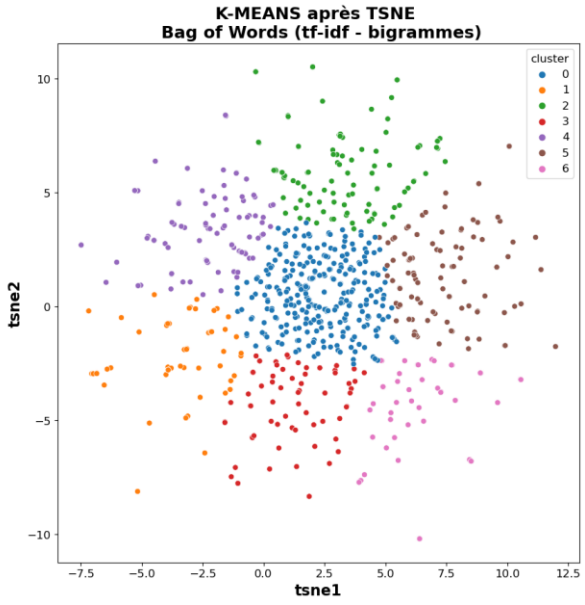
# Informations supplémentaires:
# Le Bag of Words sur les bigrammes



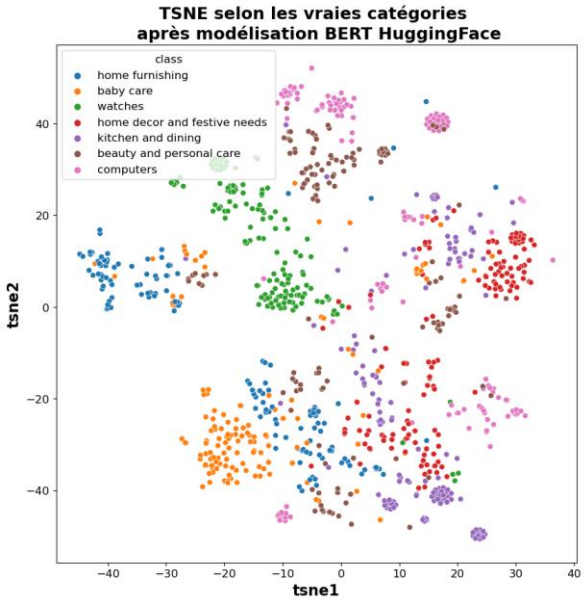

**ARI après simple comptage :**
**0.146**

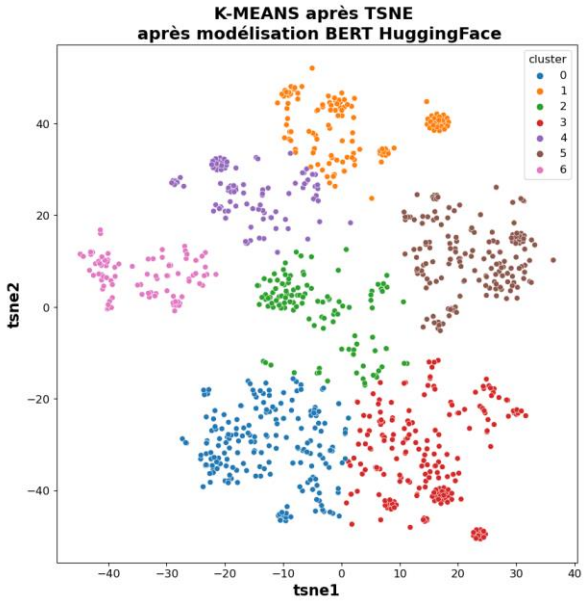


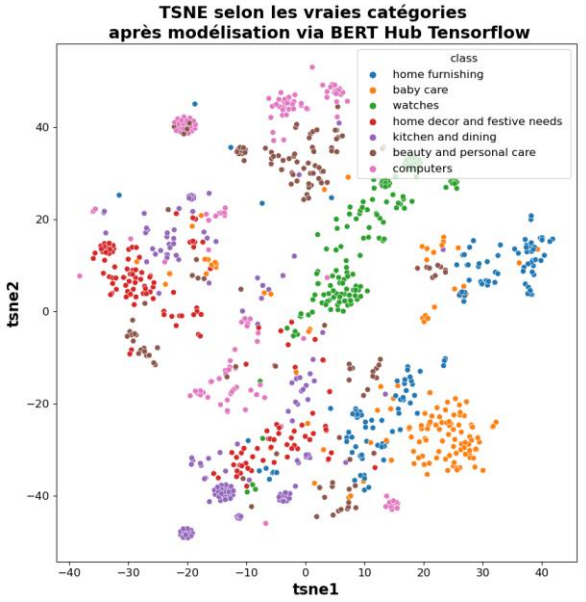

**ARI après tf-idf:**
**0.237**

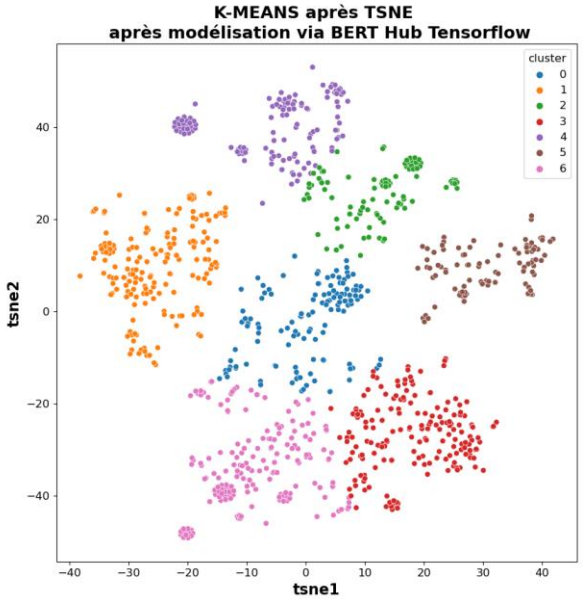
# Informations supplémentaires:
# Le Word Embedding avec BERT




**ARI après BERT HuggingFace**
**0.250**




**ARI après BERT Hub TensorFlow**
**0.250**

# Informations supplémentaires:
# Les méthodes non supervisées pour le texte: LDA et NMF

## LDA (LatentDirichletAllocution)

- Chaque document du corpus est un ensemble de mots sans ordre (bag-of-words)
- Détermination catégories de produits en fonction des distributions de chaque mot dans le corpus (term frequency)

```
display_topics(lda, count_vectorizer.get_feature_names(), 15)
```

```
Topic 0:
guarantee set ceramic mug sticker best safe gift fresh wall making dishwasher perfect exclusive microwave
Topic 1:
mug coffee warranty ceramic adapter perfect tea power bring charger make gift elegant series happy
Topic 2:
baby cotton fabric girl dress pattern printed boy wash content length single neck sleeve care
Topic 3:
inch towel set art glass best bath paper hand beautiful cotton showpiece finish home collection
Topic 4:
skin warranty set led power model multicolor print pad mouse flexible easy use fan apple
Topic 5:
watch guarantee showpiece men battery dial best strap brass gold model round resistant steel digital
Topic 6:
pizza cutter table pot model carpet steel soft medium brown pyjama home drawer inch lamp
```

## NMF (Negative Matrix Factorisation)

- Chaque document du corpus est un ensemble de mots sans ordre (bag-of-words)
- Détermination catégories de produits en considérant la fréquence d'apparition des mots dans le corpus (tf-idf)

```
display_topics(nmf, bow_tfidf1_feature_names, 15)
```

```
Topic 0:
watch men dial strap sonata maximum resistant guarantee round clasp buckle case digital gold steel
Topic 1:
guarantee best cell battery router wireless pavilion link led watch band dual extender bulb compatible
Topic 2:
mug ceramic coffee perfect gift safe microwave creation permanent thrilling porcelain stay start making dishwasher
Topic 3:
baby girl cotton fabric dress boy neck sleeve printed pattern shirt content wash casual occasion
Topic 4:
showpiece best guarantee brass exotic statue decorative lord elephant gift raja stone wall wooden musician
Topic 5:
abstract single blanket double multicolor floral home brown geometric inch sheet checkered grey polyester curtain
Topic 6:
set cotton skin towel warranty playboy bath anna gift print dark pad mouse guarantee pyjama
```

**RESULTATS NON CONCLUANTS**

# Informations supplémentaires:
# Traitement des images avec Pillow avant l'extraction des feaures

## Autre image traitée avec Pillow
## Filtre médian de taille 3

**Image d'origine**

**Image traitée**

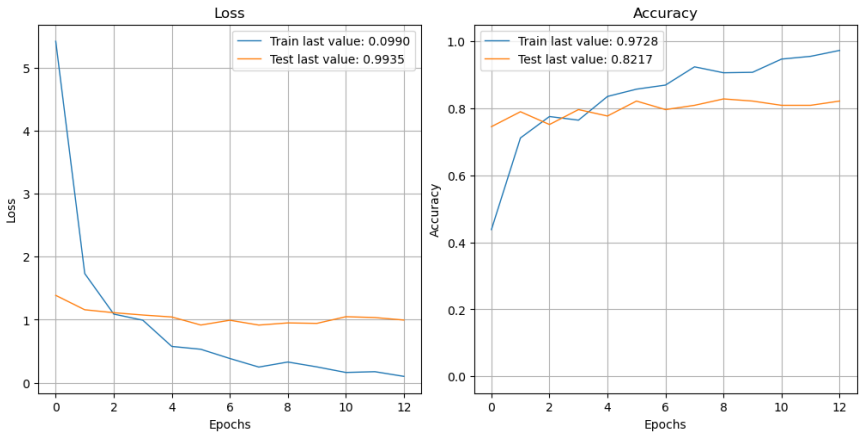
## Traitement d'images avec openCV


## Traitement d'images avec Pillow

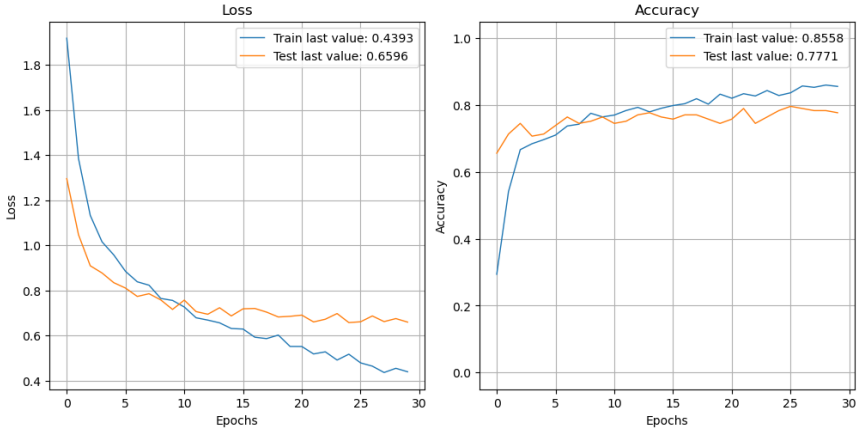
# Informations supplémentaires: VGG16 avec l'optimizer rmsprop

## BATCH SIZE DE 64

### Sans Data Augmentation

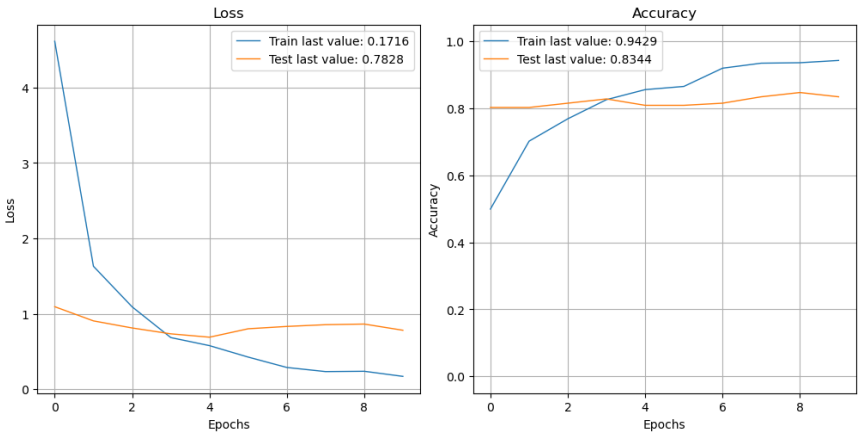



### Avec Data Augmentation

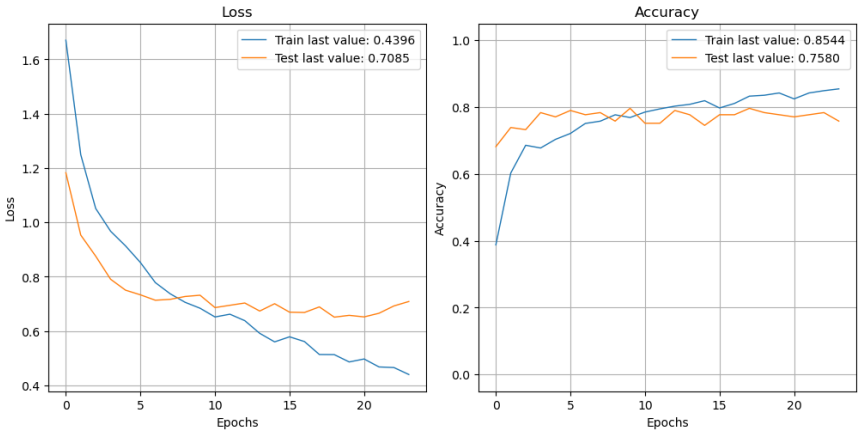



## BATCH SIZE DE 32

### Sans Data Augmentation



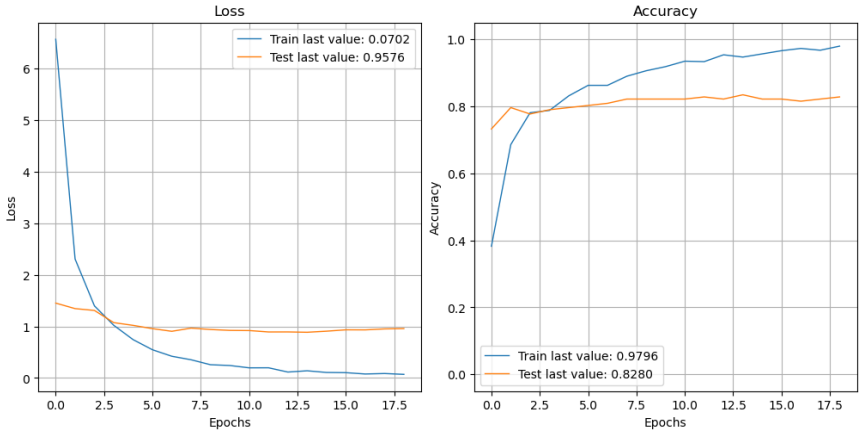

### Avec Data Augmentation



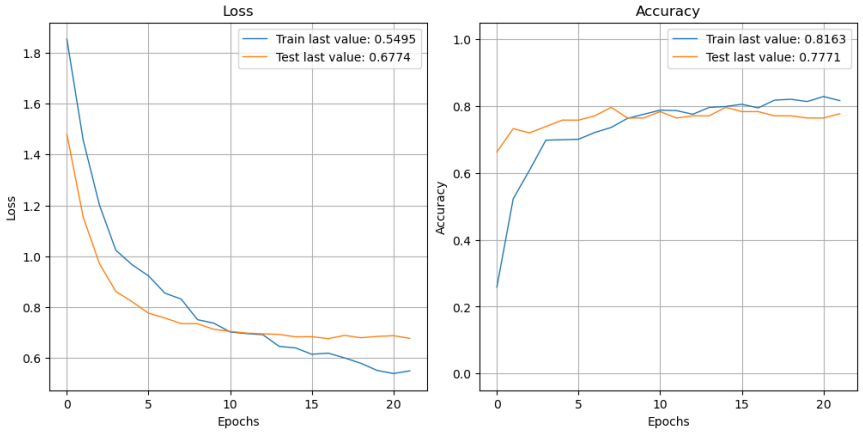
# Informations supplémentaires: VGG16 avec l'optimizer adam

## BATCH SIZE DE 64

### Sans Data Augmentation

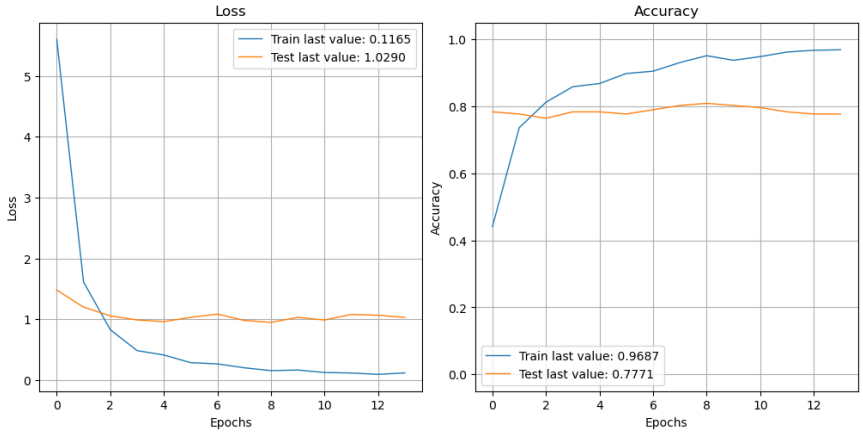



### Avec Data Augmentation




## BATCH SIZE DE 32

### Sans Data Augmentation




### Avec Data Augmentation