Projet 6 – Parcours Data Scientist

Classifiez automatiquement des biens de consommation

OPENCLASSROOMS

Robin Perbet – août 2022

Sommaire

1. Introduction

Introduction

Problématique du projet

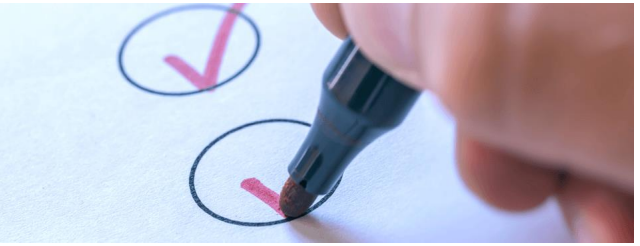
Présentation de la problématique

Le lancement d'une plateforme e-commece

L'entreprise "Place de marché" souhaite lancer une marketplace e-commerce sur laquelle des vendeurs proposent des articles à des acheteurs en postant une photo et une description

Une catégorisation des produits non optimisée

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



Une possibilité d'amélioration avec le machine learning

Pour améliorer l'expérience utilisateur il devient nécessaire d'automatiser la classification des articles

→ L'objectif sera d'étudier la faisabilité d'un moteur de classification des articles en différentes catégories avec un algorithme de machine learning

Introduction

Présentation du jeu de données

Données textuelles

- Jeu de données décrivant 1 050 différents produits recensés sur le site avec 15 variables descriptives :
  - uniq_id
  - crawl_timestamp
  - product_url
  - product_name
  - **product_category_tree**
  - pid
  - retail_price
  - discounted_price
  - image
  - is_FK_Advantage_product
  - **description**
  - product_rating
  - overall_rating
  - brand
  - product_specifications

Images

- Une image par produit, ci-dessous quelques exemples :




2. Analyse des données textuelles

Analyse des données textuelles

Méthodologie générale

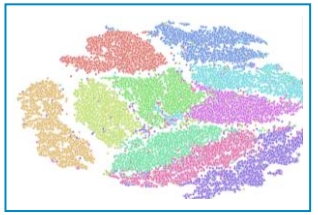
Méthodologie

1 Extraction des features

- Chaque description textuelle sera représentée par un vecteur. Différentes techniques sont utilisées (détaillées ultérieurement) :
  - Bag of words : CountVectorizer et Tf-idf
  - Plongement de mots : Word2vec, SpaCy
  - Modèles de langage : USE, BERT

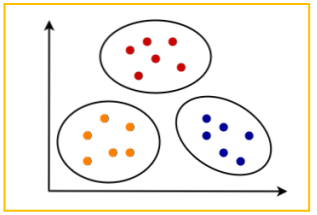
2 Réduction de dimension

- Réduction de ces features à 2 dimensions via un t-SNE



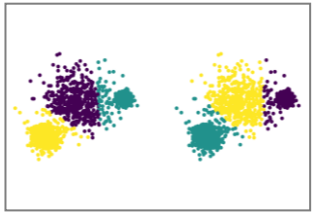
3 Clustering

- Réalisation d'un clustering via un KMeans avec k égal au nombre de catégories de produits (7 catégories) sur les features réduites par le t-SNE



5 Analyse de similarité

- Comparaison des catégories prédites par le KMeans pour chaque produit par rapport aux catégories réelles via un ARI
- Un score élevé déterminera la faisabilité de la mise en place d'un moteur de classification à partir de la description des produits



Analyse des données textuelles

Bag of words

Méthodologie d'extraction de features de texte avec le bag of words

1 Prétraitement des données textuelles

- Afin de réduire le nombre de mots utilisés tout en conservant les informations les plus importantes des documents, des étapes de prétraitement sont nécessaires :
  - Tokenisation
  - Suppression des stop words
  - Suppression des mots rares
  - Suppression des mots trop courts
  - Suppression des caractères numériques
  - Lemmatisation (ou stematisation)
- Deux librairies différentes ont été utilisées dans cette optique : SpaCy et NLTK (même étapes mais certains traitements donnent des résultats différents, notamment la liste des stopwords et la lemmatisation)

2 Réalisation d'un « bag of words »

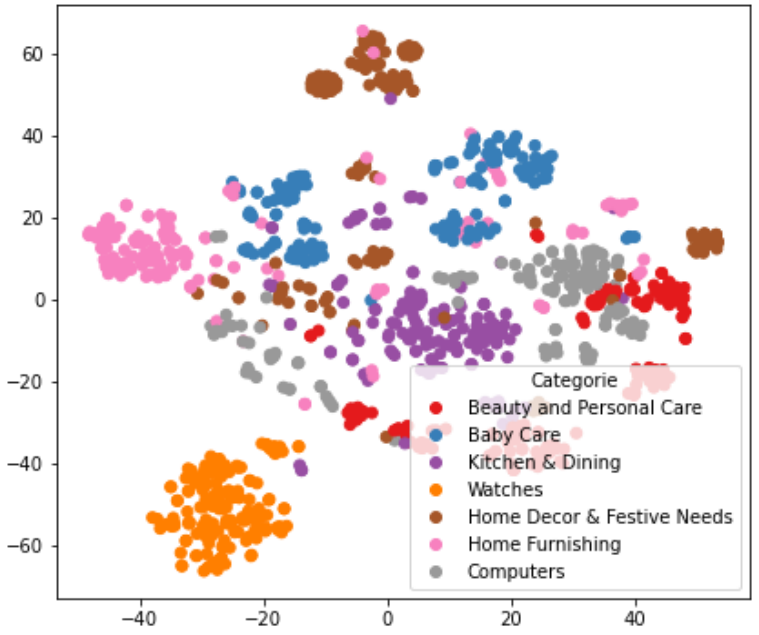
- Nous avons utilisé deux méthodes de vectorisation distinctes : CountVectorizer et Tf-idf
- CountVectorizer :
  - Chaque mot unique du corpus de texte constitue une variable
  - Pour chaque document, nous comptons le nombre d'occurrences de chacun des mots
  - Nous obtenons donc pour chaque document un vecteur de la taille du nombre de mots unique du corpus et avec pour valeurs les occurrences de ces mots dans le document
- Tf-idf :
  - Principe similaire au CountVectorizer mais avec une différence de calcul des valeurs pour chaque mot
  - La valeur est la fréquence d'apparition d'un mot pondérée par sa rareté dans le corpus
  - La formule est le produit TF (nombre de fois où le mot est dans le document / nombre de mots dans le document) * IDF (nombre de documents / nombre de documents où apparaît le mot)

Analyse des données textuelles

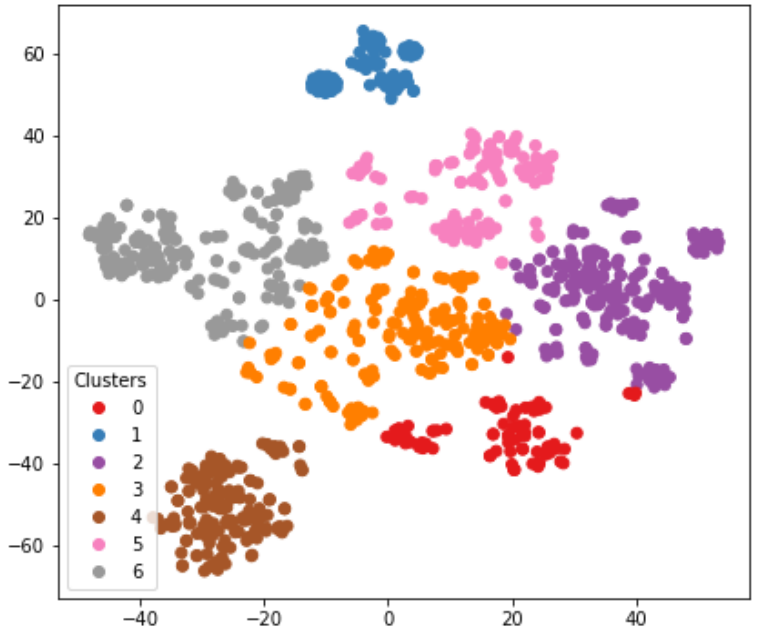
Bag of words - NLTK

Résultats de l'analyse de similarité entre classes réelles et le clustering réalisé sur les features

Visualisation t-SNE des classes réelles



Visualisation t-SNE des clusters



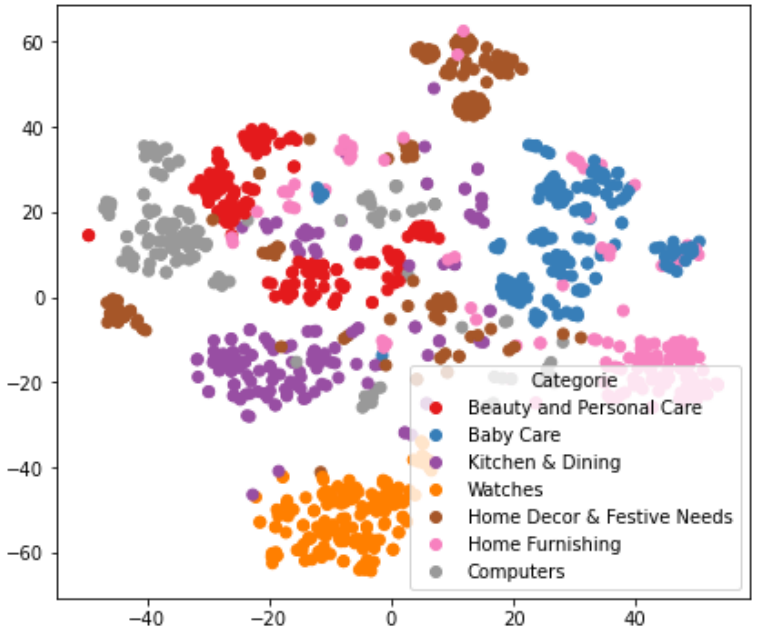
ARI : 0,41

Analyse des données textuelles

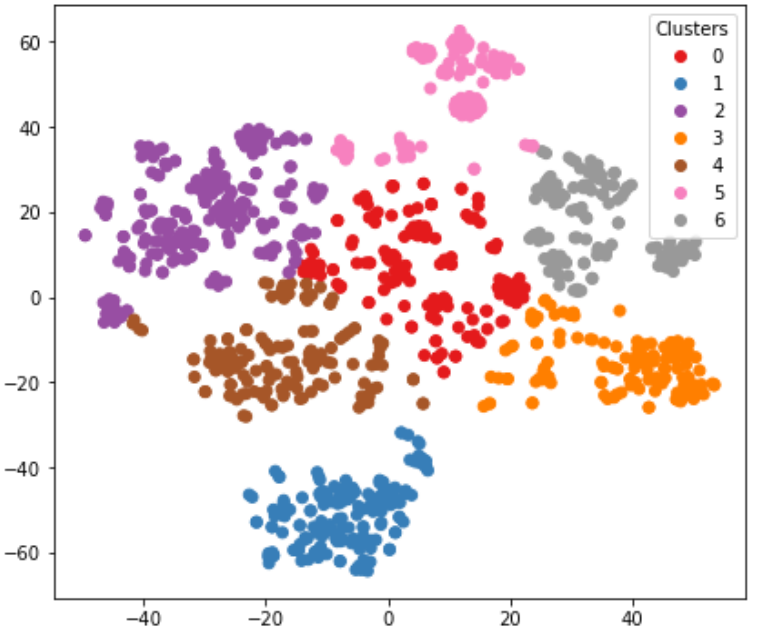
Bag of words - SpaCy

Résultats de l'analyse de similarité entre classes réelles et le clustering réalisé sur les features

Visualisation t-SNE des classes réelles



Visualisation t-SNE des clusters



ARI : 0,46

Analyse des données textuelles

Plongements de mots et modèles de langage

Présentation des modèles utilisés et de leurs caractéristiques

Fonctionnement général des plongements de mots et modèles de langage

- Réseaux de neurones entrainés sur des corpus de texte très larges
- Création pour chacun des mots de vecteurs permettant de mesurer la similarité des différents mots (chaque mot dispose d'un vecteur de même taille)
- Deux mots avec une signification proche auront des vecteurs proches dans l'espace défini (les poids des vecteurs tiennent compte du contexte d'utilisation du mot et les mots proches seront généralement utilisés dans les mêmes contextes)

	Word2vec	SpaCy	USE	BERT
Principale source d'entrainement	Google News	OntoNotes	Wikipédia	Wikipédia
Architecture	■ Réseau de neurone classique avec une couche cachée (sans fonction d'activation) ■ Le vecteur retenu représente les poids de la couche cachée	■ Architecture non détaillée par SpaCy	■ Réseau de neurones basé sur l'architecture transformers ■ USE fonctionne sur un plongement de phrases et non de mots	■ Réseau de neurones basé sur l'architecture transformers ■ Le principal avantage de ce modèle est d'être bidirectionnel (prend en compte aussi bien les mots précédents et succédant le mot cible)
Dimension des vecteurs (modèle initial)	300	96	512	768

Plongements de mots

- Vecteurs « statiques » : chacun des mots sera représenté par un unique vecteur

Modèles de langage

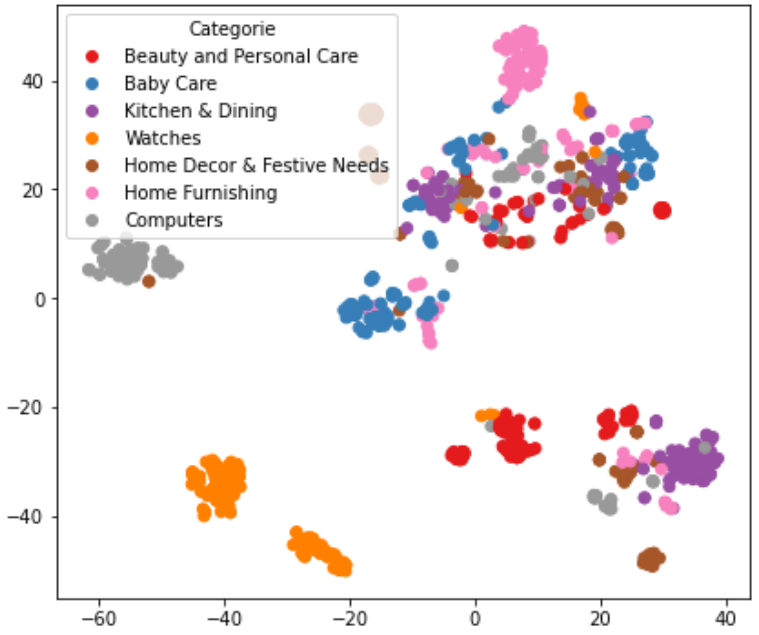
- Vecteurs « dynamiques » : un même mot pourra être représenté par différents vecteurs en fonction du contexte actuel du mot

Analyse des données textuelles

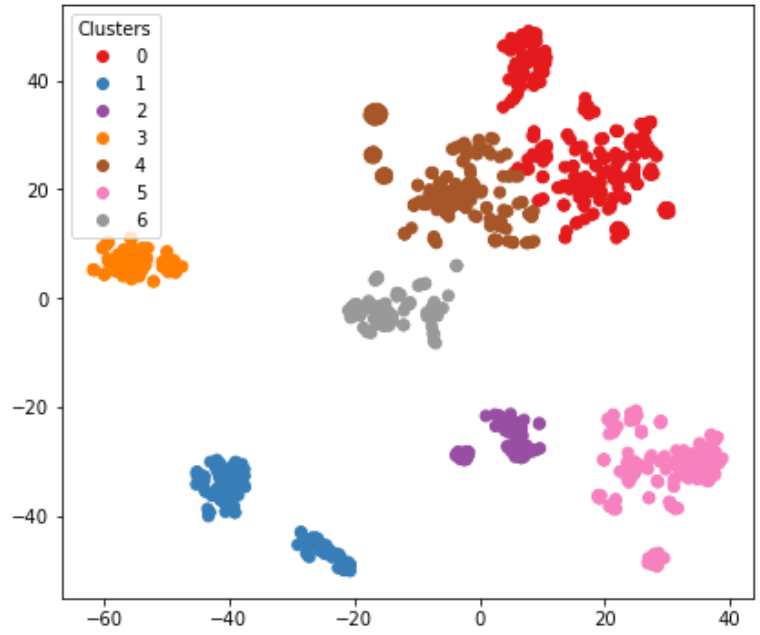
Plongement de mots - Word2vec

Résultats de l'analyse de similarité entre classes réelles et le clustering réalisé sur les features

Visualisation t-SNE des classes réelles



Visualisation t-SNE des clusters



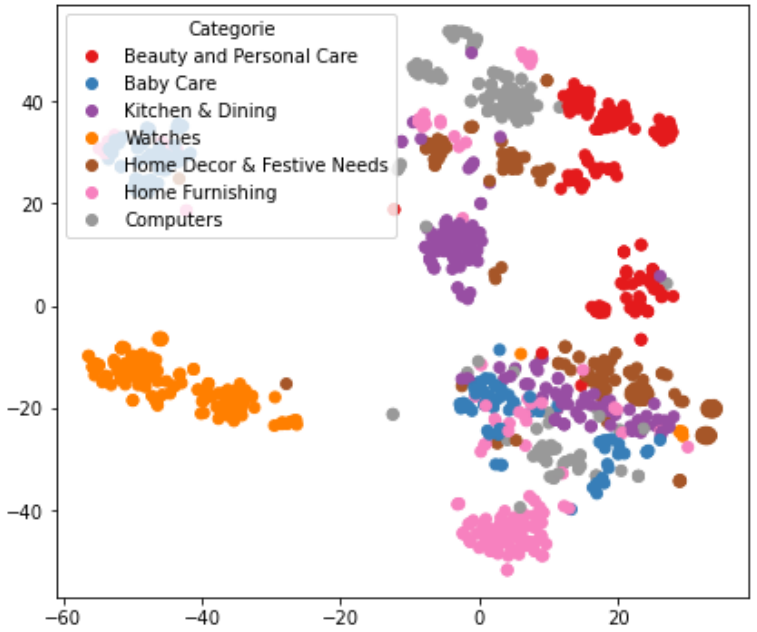
ARI : 0,27

Analyse des données textuelles

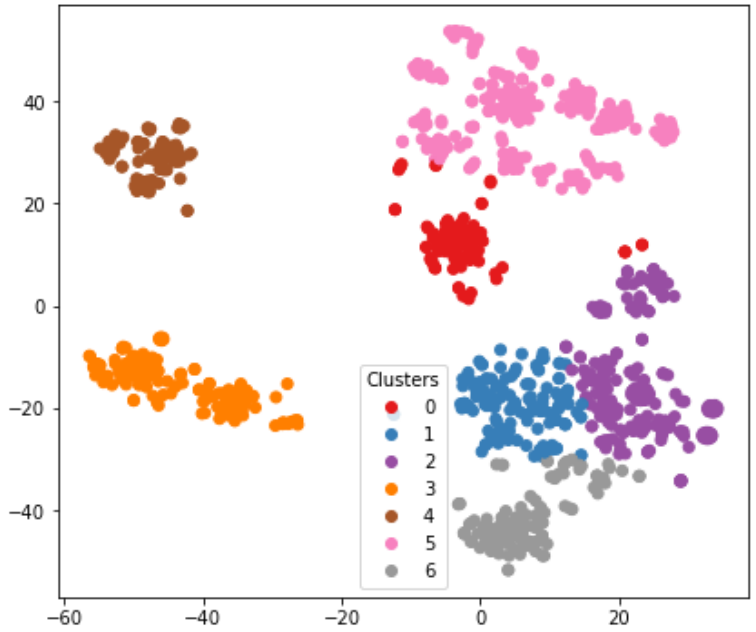
Plongement de mots - SpaCy

Résultats de l'analyse de similarité entre classes réelles et le clustering réalisé sur les features

Visualisation t-SNE des classes réelles



Visualisation t-SNE des clusters



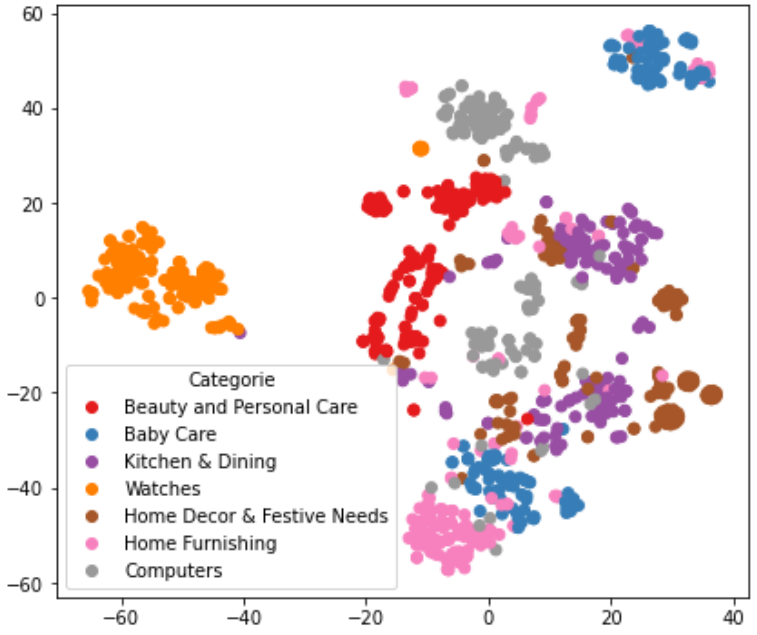
ARI : 0,35

Analyse des données textuelles

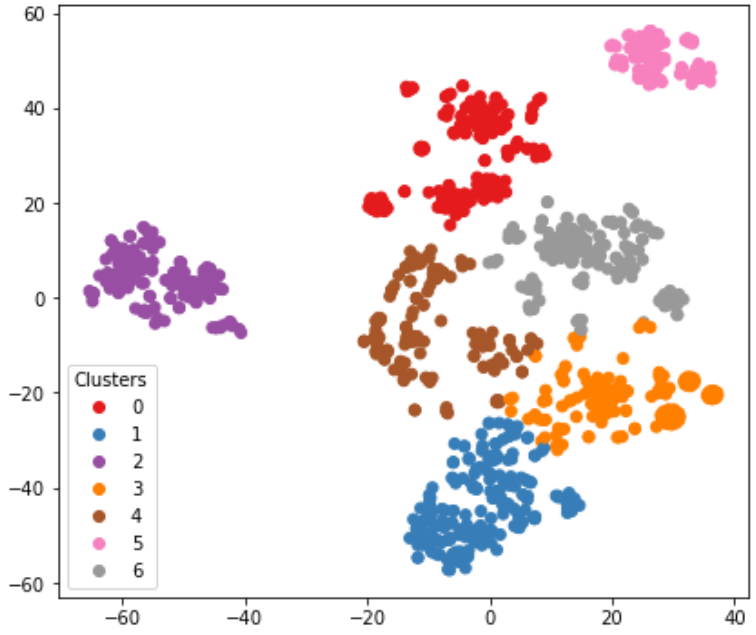
Modèles de langage - USE

Résultats de l'analyse de similarité entre classes réelles et le clustering réalisé sur les features

Visualisation t-SNE des classes réelles



Visualisation t-SNE des clusters



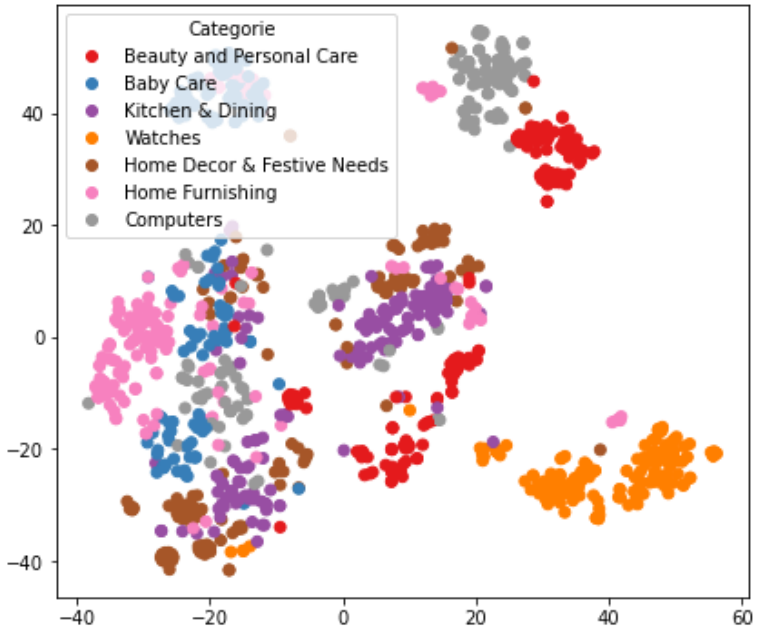
ARI : 0,41

Analyse des données textuelles

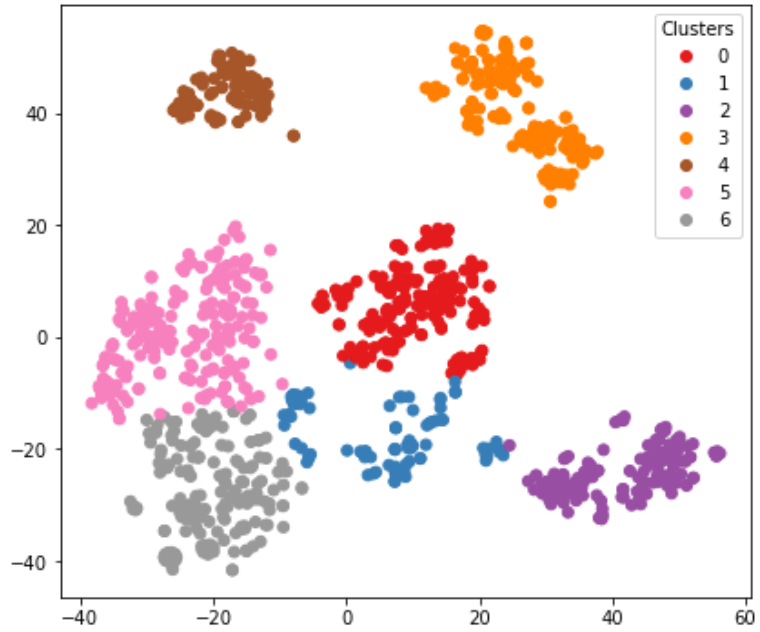
Modèles de langage - BERT

Résultats de l'analyse de similarité entre classes réelles et le clustering réalisé sur les features

Visualisation t-SNE des classes réelles



Visualisation t-SNE des clusters



ARI : 0,33

3. Analyse des images

Analyse des images

Méthodologie générale

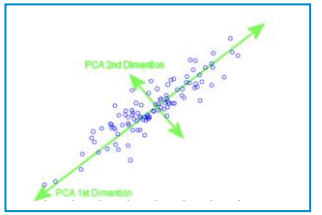
Méthodologie

1 Extraction des features

- Chaque image sera représentée par un vecteur. Différentes techniques sont utilisées (détaillées ultérieurement) :
  - Descripteurs SIFT puis application d'un bag of visual words
  - Réseau de neurones convolutif pré-entraîné

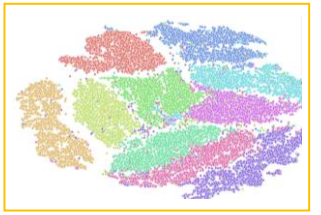
2 Réduction de dimension 1

- Réduction de dimension de ces features avec une ACP en retenant 99% de la variance
- Cela permet d'accélérer l'étape suivante et de ne conserver que les features pertinentes



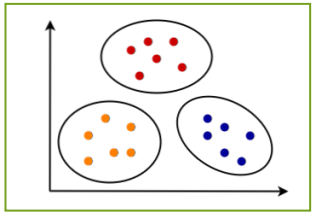
3 Réduction de dimension 2

- Réduction des composantes principales de l'ACP à 2 dimensions via un t-SNE



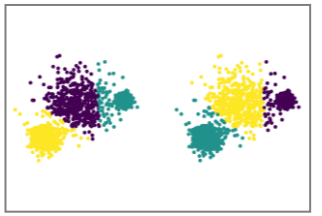
4 Clustering

- Réalisation d'un clustering via un KMeans avec k égal au nombre de catégories de produits (7 catégories) sur les features réduites par le t-SNE



5 Analyse de similarité

- Comparaison des catégories prédites par le KMeans pour chaque produit par rapport aux catégories réelles via un ARI
- Un score élevé déterminera la faisabilité de la mise en place d'un moteur de classification à partir des images des produits



Analyse des images

Descripteurs SIFT

Méthodologie d'extraction de features d'images avec les descripteurs SIFT

1 Génération des descripteurs SIFT

- Scale-invariant feature transform (SIFT) : algorithme permettant de définir les points d'intérêts d'une image (descripteurs)
- Ces points correspondent à des bords ou coins d'une image : zones autour desquelles ont observe de fortes variations d'intensité ou de couleur des pixels, qui indiquent donc la jonction entre des objets différents sur l'image
- Les descripteurs constituent des vecteurs qui décrivent le voisinage de la feature à laquelle ils sont associés
- Les descripteurs du SIFT ont l'avantage d'être invariants par rotation, par changement d'échelle et par exposition

2 Réalisation d'un « bag of visual words »

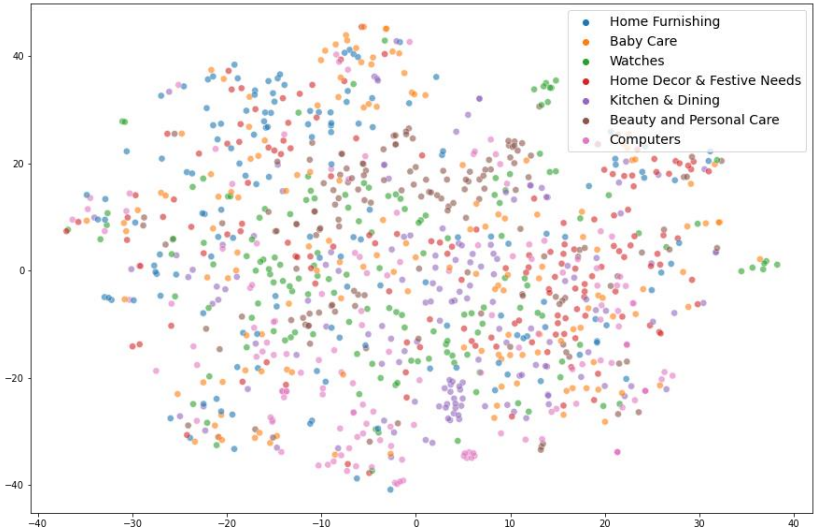
- La taille du vecteur de chacun des descripteur est identique, en revanche le nombre de descripteurs varie pour chacune des images, il n'est donc pas possible d'utiliser directement les descripteurs comme features pour une classification
- Afin de pallier cela, nous avons appliqué un « bag of visual words » :
  - Clustering des descripteurs via un KMeans (k = racine carrée du nombre total de descripteurs)
  - Pour chaque image nous allons déterminer le nombre de descripteurs par cluster : chaque image disposera donc d'un vecteur de taille k avec pour valeurs le nombre d'occurrences pour chacun des clusters
- Ces vecteurs seront nos features finales auxquelles nous appliqueront les étapes décrites précédemment afin d'obtenir un ARI

Analyse des images

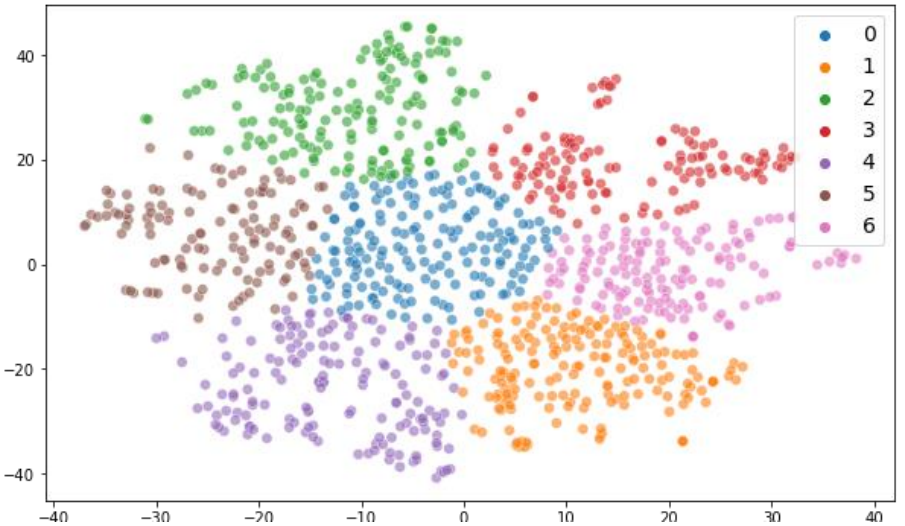
Descripteurs SIFT

Résultats de l'analyse de similarité entre classes réelles et le clustering réalisé sur les features

Visualisation t-SNE des classes réelles



Visualisation t-SNE des clusters

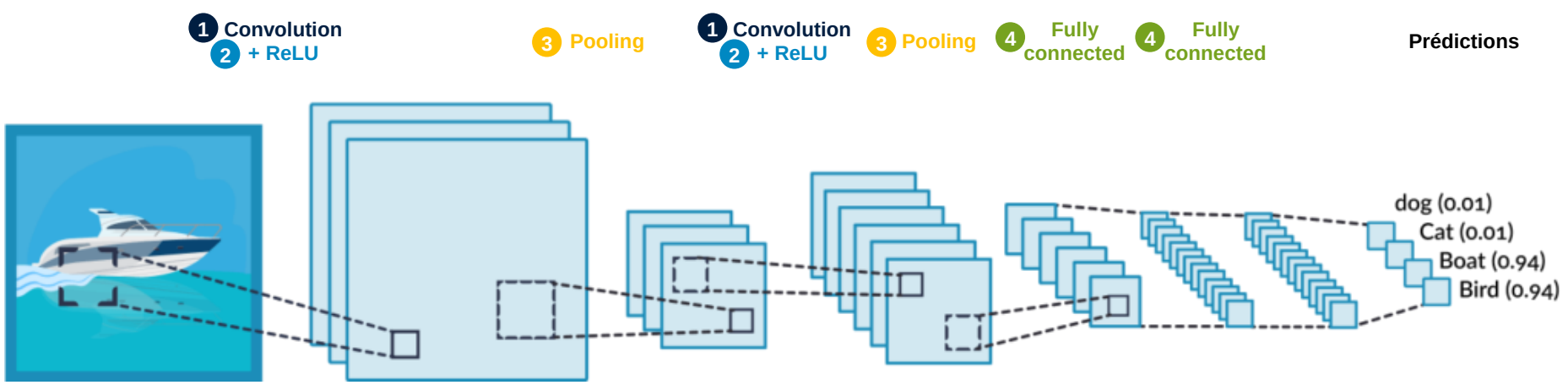


ARI : 0,04

Analyse des images

Réseaux de neurones convolutifs

Description de l'architecture d'un réseau de neurones convolutif



1 Couche de convolution

- Application d'un filtre qui va « glisser » sur toute l'image avec un pas déterminé et va calculer à chaque pas le produit de convolution entre le filtre et l'image
- Plus le produit sera proche de 1, plus le filtre ressemble à la partie de l'image étudiée
- La composition du filtre est initialisée aléatoirement puis est modifiée avec rétropropagation du gradient

2 Fonction d'activation ReLU

- La fonction d'activation des différentes couches de convolution est une fonction ReLU
- Elle retient pour chaque valeur le maximum entre 0 et la valeur concernée (ce qui élimine donc les nombres négatifs)

3 Couche de pooling

- Placée après la couche de convolution, elle permet de réduire la taille des images, tout en préservant leurs caractéristiques importantes
- L'image est découpée en cellules de petite taille et l'on va conserver la valeur maximale (maxpooling) ou moyenne (meanpooling) de la cellule
- Cela permet donc de ne conserver qu'une valeur par cellule

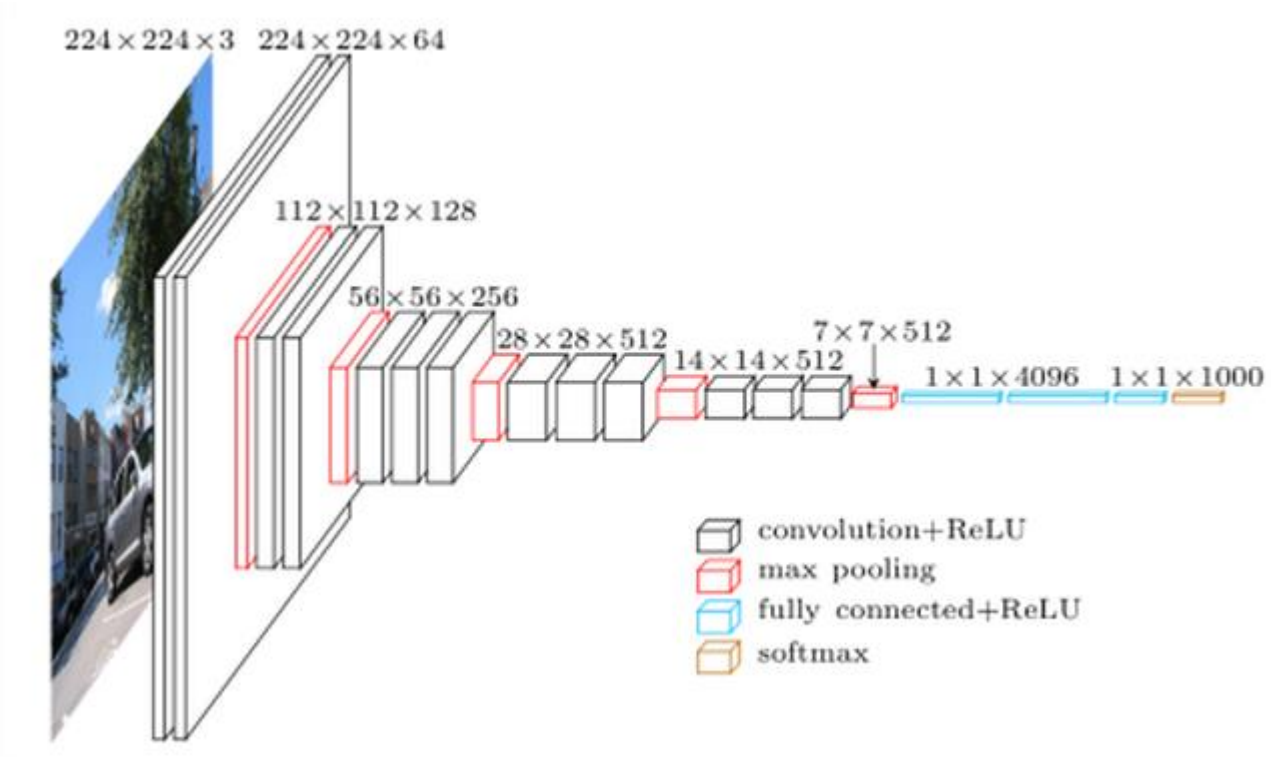
4 Couche fully-connected

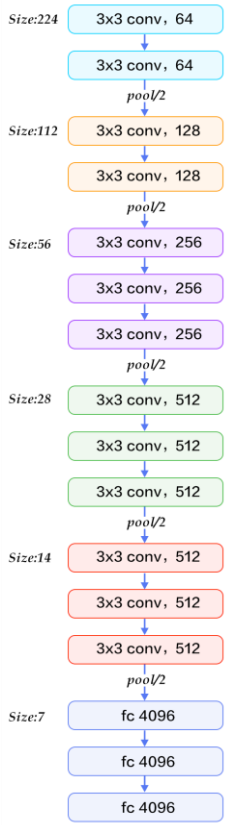
- Dernière couche qui reprend le fonctionnement classique d'un réseau de neurones
- Elle prend pour input les features extraites par la convolution + pooling et applique une combinaison linéaire
- La fonction d'activation de la dernière couche permet la classification (logistique ou softmax)

Analyse des images

Réseaux de neurones convolutifs

Architecture du réseau pré-entraîné utilisé : VGG16





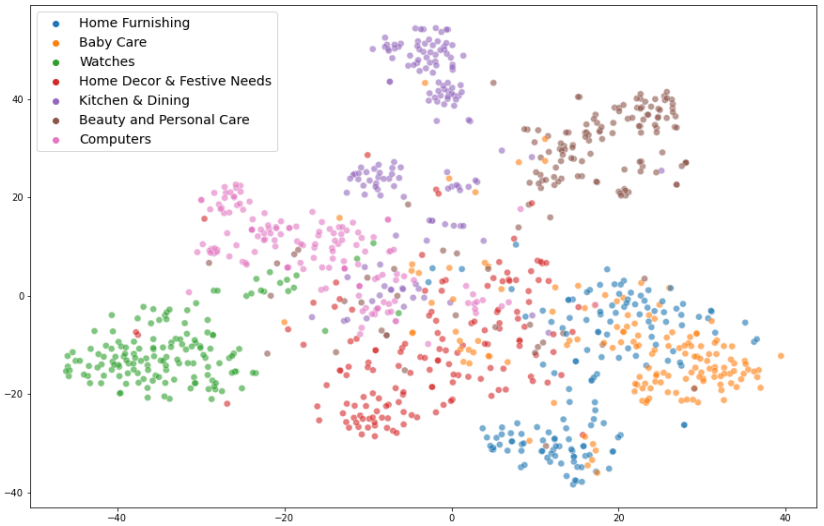
- Notre objectif n'étant pas de réaliser directement une classification, mais d'extraire des features afin d'étudier la faisabilité d'une future classification, la dernière couche fully-connected (pour la classification) a été retirée afin d'obtenir seulement pour chaque image le vecteur de features auquel sera ensuite appliqué les mêmes opérations que pour SIFT

Analyse des images

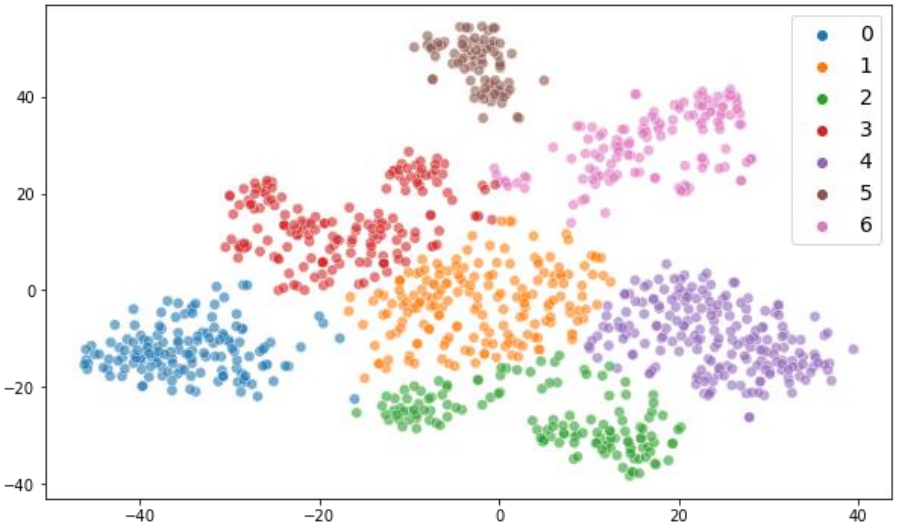
Réseaux de neurones convolutifs

Résultats de l'analyse de similarité entre classes réelles et le clustering réalisé sur les features

Visualisation t-SNE des classes réelles



Visualisation t-SNE des clusters



ARI : 0,45

4. Conclusion

Conclusion

Analyse de faisabilité de la création d'un moteur de classification

	Analyse du texte						Analyse des images	
	BoW NLTK	**BoW SpaCy**	**Word2vec**	**SpaCy vec**	**USE**	**BERT**	**SIFT**	**VGG16**
ARI	0,41	**0,46**	0,27	0,35	0,41	0,33	0,04	**0,45**
Structure	Bag of words	Bag of words	Plongement de mots statique	Plongement de mots statique	Plongement de phrases	Plongement de mots dynamique	Descripteurs	Réseau de neurones convolutif

Conclusion

- ✓ Nous obtenons des ARI supérieurs à 0,4 aussi bien pour l'analyse du texte que l'analyse d'images (BoW SpaCy et VGG16)
- ✓ Ces scores laissent envisager une possible mise en œuvre d'un moteur de classification automatique avec des algorithmes supervisés entraînés sur notre jeu de données et optimisés
- × Néanmoins, il est probable que même après entraînement et optimisation ces algorithmes ne soient pas d'une précision parfaite et que quelques erreurs de classification subsistent
- × Par ailleurs, l'analyse a été réalisée sur la base des catégories larges de produits, ne permettant pas un classement fin (l'utilisation de sous-catégories n'est pour l'instant pas réaliste au regard du peu de produits présents dans certaines sous-catégories)