OPENCLASSROOMS

Parcours Data Scientist

Projet 6 : Classifiez automatiquement des biens de consommation

Etudiant: Eric Wendling
Mentor: Julien Heiduk
Date: 15/06/2020

place de marché

Etude de faisabilité d'un moteur de classification de produits

Définition du projet

Contexte

- La société **Place de marché** est en phase de lancement d'une markeplace e-commerce

Objectifs

- Faciliter la mise en ligne et la recherche de produits

Besoins

- Automatisation de la classification de produits

Moyens

- Base de 1050 produits (image + description)
- API Amazon (pour disposer d'une base plus importante)
- Méthodes de classification

Livrable

- Notebook: Analyse exploratoire et traitement des données + Evaluation modèles de classification

Plan

Analyse

1. Analyse exploratoire des données

Traitement des données textuelles

2. Préparation du corpus
3. Traitement du corpus
4. Réduction de dimensions
5. Classification

Traitement des images

7. Prétraitement
8. Détermination des variables explicatives
9. Classification
10. Réseaux de neurones convolutifs

- *CNN*
- *Keras VGG16*
- *Transfer Learning*

Optimisation

11. Données complémentaires

- *Amazon Product Advertising API*

Conclusion

12. Synthèse
13. Optimisation
14. Prochaines étapes

Questions

1 Analyse exploratoire des données

1050 produits

	Dataset	Shape	Variable	Duplicates	Unique	Null	Type
0	df_flipkart	1050	uniq_id	0	1050	0	object
1	df_flipkart	1050	crawl_timestamp	X 901	149	0	object
2	df_flipkart	1050	product_url	X 0	1050	0	object
3	df_flipkart	1050	product_name	0	1050	0	object
4	df_flipkart	1050	product_category_tree	408	642	0	object
5	df_flipkart	1050	pid	0	1050	0	object
6	df_flipkart	1050	retail_price	695	355	1	float64
7	df_flipkart	1050	discounted_price	625	425	1	float64
8	df_flipkart	1050	image	0	1050	0	object
9	df_flipkart	1050	is_FK_Advantage_product	X 1048	2	0	bool
10	df_flipkart	1050	description	0	1050	0	object
11	df_flipkart	1050	product_rating	X 1023	27	0	object
12	df_flipkart	1050	overall_rating	X 1023	27	0	object
13	df_flipkart	1050	brand	559	491	338	object
14	df_flipkart	1050	product_specifications	65	985	1	object

Variables textuelles

Variables numériques

Variables catégorielles

Variable
uniq_id
product_name
product_category_tree
pid
retail_price
discounted_price
image
description
brand
product_specifications

remplacées par **'prod_id'**

construction de corpus

concaténation

variable **'content'**

nom_de_fichier.jpg

Variables catégorielles

Variable 'product_category_tree'

- La variable contient des valeurs de catégories délimitées par >>

```
df_cat = df_flipkart[['product_category_tree']]
df_cat.head(2).values.tolist()
```

```
[['["Home Furnishing >> Curtains & Accessories >> Curtains >> Elegance Polyester Multicolor Abstract Eyelet Do..."]'],
 ['["Baby Care >> Baby Bath & Skin >> Baby Bath Towels >> Sathiyas Baby Bath Towels >> Sathiyas Cotton Bath Towel (3 Bath Towe
l, Red, Y..."]']]
```

- On extrait les valeurs des catégories délimitées et on les stocke dans autant de variables

➢ On obtient 1 catégorie principale et 7 sous-catégories

```
df_cat.head(2)
```

	0	1	2	3	4	5	6	7
0	home furnishing	curtains & accessories	curtains	elegance polyester multicolor abstract eyelet...	NaN	NaN	NaN	NaN
1	baby care	baby bath & skin	baby bath towels	sathiyas baby bath towels	sathiyas cotton bath towel (3 bath towel, red...	NaN	NaN	NaN

Variables catégorielles

Variable 'product_category_tree'

- Modèle de classement des produits
- ➢ Variable CIBLE

- Nombre de valeurs non nulles et uniques pour chaque catégorie

Cat ID	Nombre de valeurs non nulles	Nombre de valeurs uniques	
0	1050	7	
1	1050	63	
2	1047	245	
3	679	350	X
4	405	296	X
5	127	117	X
6	57	58	X
7	0	1	X

Catégorie principale →

cat_0_id	cat_0
0	0_home furnishing
1	1_baby care
2	2_watches
3	3_home decor & festive needs
4	4_kitchen & dining
5	5_beauty and personal care
6	6_computers

Catégories contenant beaucoup de valeurs nulles

Variables catégorielles

Variable 'product_specifications'

- La variable contient des données structurées dans un dictionnaire

```
df_flipkart['product_specifications'].iloc[1]
```


```
'{"product_specification"=>[{"key"=>"Machine Washable", "value"=>"Yes"}, {"key"=>"Material", "value"=>"Cotton"}, {"key"=>"Desig
n", "value"=>"Self Design"}, {"key"=>"Brand", "value"=>"Sathiyas"}, {"key"=>"Type", "value"=>"Bath Towel"}, {"key"=>"GSM", "val
ue"=>"500"}, {"key"=>"Model Name", "value"=>"Sathiyas cotton bath towel"}, {"key"=>"Ideal For", "value"=>"Men, Women, Boys, Gir
ls"}, {"key"=>"Model ID", "value"=>"asvtwl322"}, {"key"=>"Color", "value"=>"Red, Yellow, Blue"}, {"key"=>"Size", "value"=>"Medi
am"}, {"key"=>"Length", "value"=>"30 inch"}, {"key"=>"Width", "value"=>"60 inch"}, {"key"=>"Number of Contents in Sales Packag
e", "value"=>"3"}, {"key"=>"Sales Package", "value"=>"3 Bath Towel"}]}'
```


- On extrait les valeurs des spécifications et on les stocke dans autant de variables

➢ On obtient 604 variables

➢ Avec de nombreuses valeurs nulles

```
# Nombre de valeurs nulles pour les principales variables de spécifications
pd.DataFrame(df_spec_var_all.isna().sum()).sort_values([0]).head(15)
```

	0
type	322
brand	484
sales package	539

Variables catégorielles

Variable 'product_specifications'

- On retient 9 variables de spécifications

Variable	Duplicates	Unique	Null
type	900	150	322
model name	729	321	706
sales package	779	271	539
color	975	75	543
ideal for	1007	43	582
material	974	76	640
width	864	186	668
height	909	141	756
weight	911	139	724

index	type	model name	sales package	color	ideal for	material	width	height	weight
0	eyelet	abstract polyester door curtain set of 2	2 curtains	multicolor	NaN	polyester	NaN	NaN	NaN
1	bath towel	sathiyas cotton bath towel	3 bath towel	red, yellow, blue	men, women, boys, girls	cotton	60 inch	NaN	NaN
2	face towel set	shuvam20pcftsetassorted	20 piece face towel set	assorted	boys, girls, men, women	cotton terry	9 inch	NaN	350 g

Jeu de données

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1050 entries, 0 to 1049
Data columns (total 3 columns):
 #   Column        Non-Null Count  Dtype
---  ------        --------------  -----
 0   prod_id       1050 non-null   object
 1   product_name  1050 non-null   object
 2   content       1050 non-null   object
dtypes: object(3)
memory usage: 24.7+ KB
```

Traitement des données textuelles

```
df_flipkart.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1050 entries, 0 to 1049
Data columns (total 6 columns):
 #   Column            Non-Null Count  Dtype
---  ------            --------------  -----
 0   prod_id           1050 non-null   object
 1   brand             1050 non-null   object
 2   retail_price      1050 non-null   float64
 3   discounted_price  1050 non-null   float64
 4   product_url       1050 non-null   object
 5   image             1050 non-null   object
dtypes: float64(2), object(4)
memory usage: 49.3+ KB
```

Classification (Variables explicatives)

Traitement des images

```
df_spec.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 1050 entries, 0 to 1049
Data columns (total 9 columns):
 #   Column         Non-Null Count  Dtype
---  ------         --------------  -----
 0   type           1050 non-null   object
 1   model name     1050 non-null   object
 2   sales package  1050 non-null   object
 3   color          1050 non-null   object
 4   ideal for      1050 non-null   object
 5   material       1050 non-null   object
 6   width          1050 non-null   object
 7   height         1050 non-null   object
 8   weight         1050 non-null   object
dtypes: object(9)
memory usage: 82.0+ KB
```

Classification (Variables explicatives)

```
df_cat.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 1050 entries, 0 to 1049
Data columns (total 6 columns):
 #   Column    Non-Null Count  Dtype
---  ------    --------------  -----
 0   cat_0_id  1050 non-null   int64
 1   cat_0     1050 non-null   object
 2   cat_1_id  1050 non-null   int64
 3   cat_1     1050 non-null   object
 4   cat_2_id  1050 non-null   int64
 5   cat_2     1050 non-null   object
dtypes: int64(3), object(3)
memory usage: 57.4+ KB
```

Classification (Variables cibles)

Plan

Analyse

1. Analyse exploratoire des données

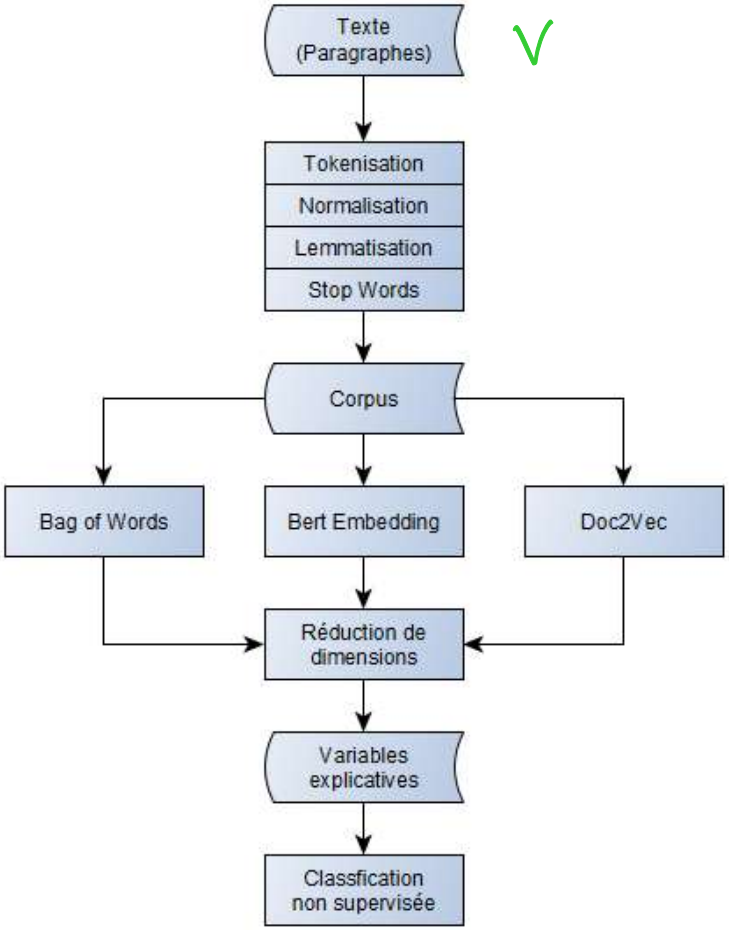
Traitement des données textuelles

2. Préparation du corpus
3. Traitement du corpus
4. Réduction de dimensions
5. Classification

Traitement des images

7. Prétraitement
8. Détermination des variables explicatives
9. Classification
10. Réseaux de neurones convolutifs

- *CNN*
- *Keras VGG16*
- *Transfer Learning*



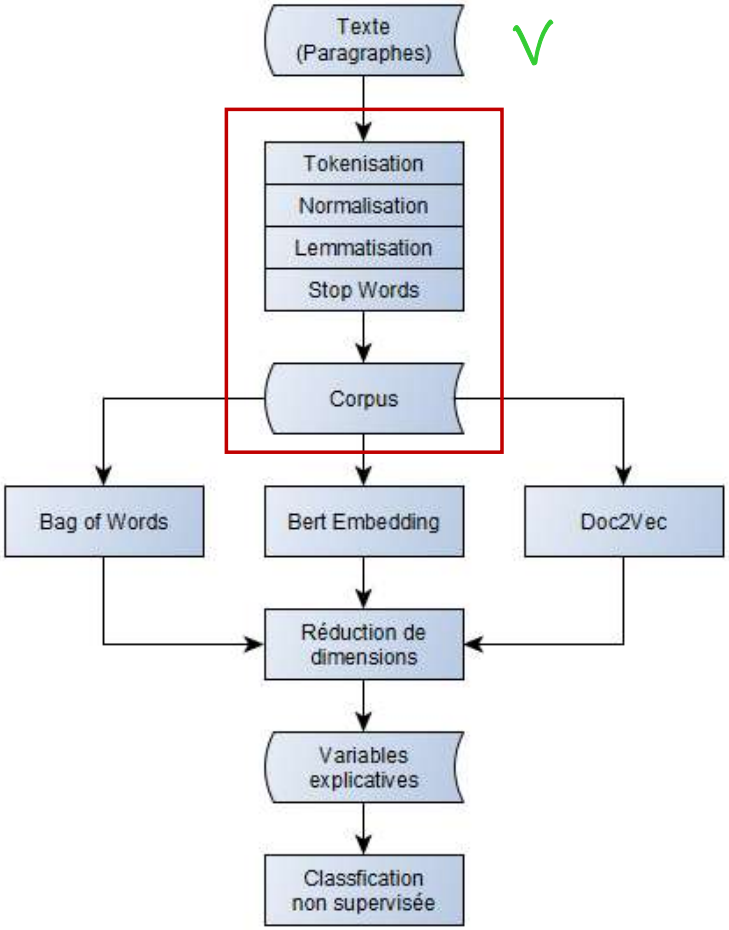
Plan

Analyse

1. Analyse exploratoire des données

Traitement des données textuelles

2. Préparation du corpus
3. Traitement du corpus
4. Réduction de dimensions
5. Classification



Configuration

Librairie

- Natural Language Toolkit (NLTK)

Variable

- 'Content'

```
df.head(2)
```

	prod_id	product_name	content
0	0	Elegance Polyester Multicolor Abstract Eyelet ...	Elegance Polyester Multicolor Abstract Eyelet ...
1	1	Sathiyas Cotton Bath Towel	Sathiyas Cotton Bath Towel Specifications of S...

Traitements

They played the match while it was raining and... They won! Congratulations!

- Tokenisation ['They', 'played', 'the', 'match', 'while', 'it', 'was', 'raining', 'and', '...', 'They', 'won', '!', 'Congratulations', '!']
- Normalisation ['they', 'played', 'the', 'match', 'while', 'it', 'was', 'raining', 'and', 'they', 'won', 'congratulations']
- Lemmatisation ['they', 'play', 'the', 'match', 'while', 'it', 'be', 'rain', 'and', 'they', 'win', 'congratulation']
- Racinisation ['they', 'play', 'the', 'match', 'while', 'it', 'wa', 'rain', 'and', 'they', 'won', 'congratul']
- Stop Words ['play', 'match', 'rain', 'win', 'congratulation']

Prétraitements

Texte original Nombre de tokens: 78

Sathiyas Cotton Bath Towel Specifications of Sathiyas Cotton Bath Towel (3 Bath Towel, Red, Yellow, Blue) Bath Towel Features Machine Washable Yes Material 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 Number of Contents in Sales Package 3 Sales Package 3 Bath Towel

Tokenisation Nombre de tokens: 88

Sathiyas Cotton Bath Towel Specifications of Sathiyas Cotton Bath Towel (3 Bath Towel , Red , Yellow , Blue) Bath Towel Features Machine Washable Yes Material 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 Number of Contents in Sales Package 3 Sales Package 3 Bath Towel

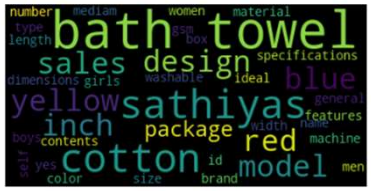
√ Tokenisation normalisée Nombre de tokens: 78

Sathiyas Cotton Bath Towel Specifications of Sathiyas Cotton Bath Towel 3 Bath Towel Red Yellow Blue Bath Towel Features Machine Washable Yes Material 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 Number of Contents in Sales Package 3 Sales Package 3 Bath Towel

√ Normalisation Nombre de tokens: 71

---

sathiyas cotton bath towel specifications of sathiyas cotton bath towel bath towel red yellow blue bath towel features machine washable yes material cotton design self design general brand sathiyas type bath towel gsm model name sathiyas cotton bath towel ideal for men women boys girls model id color red yellow blue size mediam dimensions length inch width inch in the box number of contents in sales package sales package bath towel



Prétraitements

Lemmatisation Nombre de tokens: 71

---

sathiyas cotton bath towel specification of sathiyas cotton bath towel bath towel red yellow blue bath towel feature machine wa shable yes material cotton design self design general brand sathiyas type bath towel gsm model name sathiyas cotton bath towel ideal for men woman boy girl model id color red yellow blue size mediam dimension length inch width inch in the box number of c ontent in sale package sale package bath towel

Racinisation Nombre de tokens: 71

---

sathiya cotton bath towel specif of sathiya cotton bath towel bath towel red yellow blue bath towel featur machin washabl ye ma teri cotton design self design gener brand sathiya type bath towel gsm model name sathiya cotton bath towel ideal for men women boy girl model id color red yellow blue size mediam dimens length inch width inch in the box number of content in sale packag s ale packag bath towel

Stop Words (1) Nombre de tokens: 53

---

len(stopwords.words('english'))	len(new_stop_words)	len(stop_words)
179	86	264

sathiyas cotton bath towel specification sathiyas cotton bath towel bath towel red yellow blue bath towel feature machine washa ble material cotton self brand sathiyas type bath towel gsm name sathiyas cotton bath towel ideal men woman boy girl color red yellow blue size mediam length width box number content package package bath towel

Stop Words (2) Nombre de tokens: 52

---

len(new_stop_words_2)	len(stop_words)
1023	1287

sathiyas cotton bath towel specification sathiyas cotton bath towel bath towel red yellow blue bath towel feature machine washa ble material cotton self brand sathiyas type bath towel gsm name sathiyas cotton bath towel ideal men woman boy girl color red yellow blue size length width box number content package package bath towel

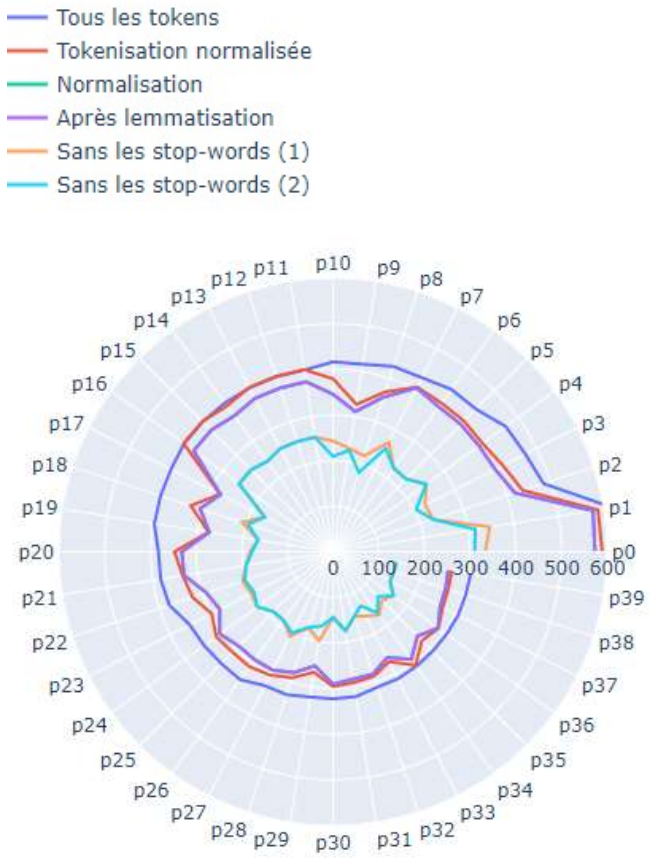
Synthèse des prétraitements

Paragraphes

Tokens

	prod_id	product_name	content	content_tok_0	content_tok_1	content_tok_2	content_tok_3	content_tok_4	content_tok_5
0	0	Elegance Polyester Multicolor Abstract Eyelet ...	Elegance Polyester Multicolor Abstract Eyelet ...	[Elegance, Polyester, Multicolor, Abstract, Ey...	[Elegance, Polyester, Multicolor, Abstract, Ey...	[elegance, polyester, multicolor, abstract, ey...	[elegance, polyester, multicolor, abstract, ey...	[elegance, polyester, multicolor, abstract, ey...	[elegance, polyester, multicolor, abstract, ey...
1	1	Sathiyas Cotton Bath Towel	Sathiyas Cotton Bath Towel Specifications of S...	[Sathiyas, Cotton, Bath, Towel, Specifications...	[Sathiyas, Cotton, Bath, Towel, Specifications...	[sathiyas, cotton, bath, towel, specifications...	[sathiyas, cotton, bath, towel, specification,...	[sathiyas, cotton, bath, towel, specification,...	[sathiyas, cotton, bath, towel, specification,...

	prod_id	words_nb_0	words_nb_1	words_nb_2	words_nb_3	words_nb_4	words_nb_5
0	0	254	243	233	233	146	134
1	1	88	78	71	71	53	52

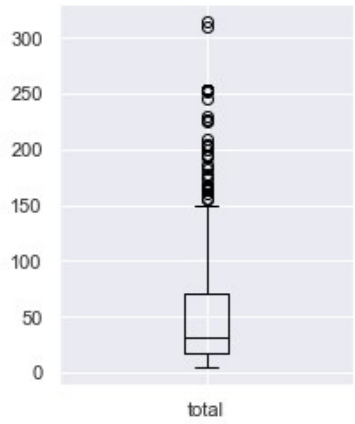


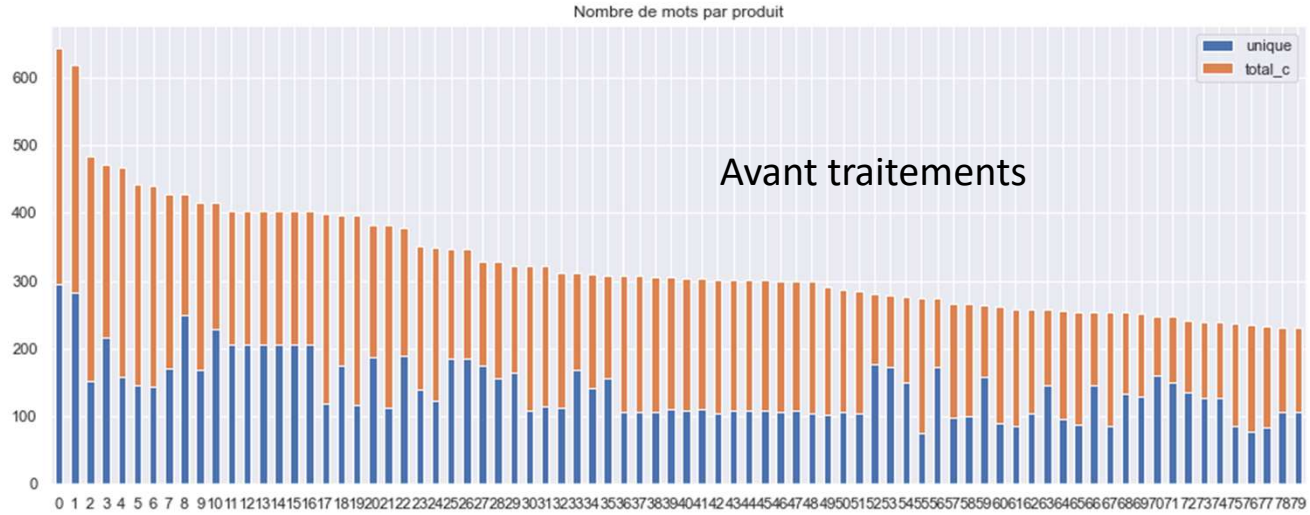
Nombre de tokens par produit après chaque traitement, pour les 40 produits ayant le plus de tokens dans leur description

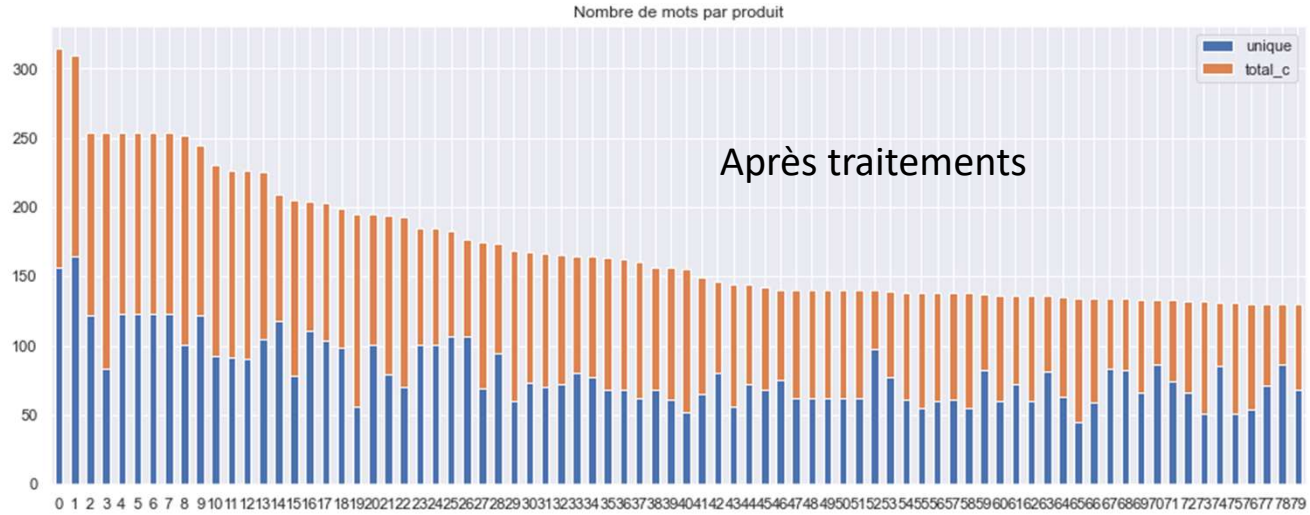
Synthèse des prétraitements

Fréquence des mots

	total
count	1050.000000
mean	49.887619
std	47.478305
min	4.000000
25%	17.000000
50%	31.000000
75%	70.000000
max	315.000000







Plan

Analyse

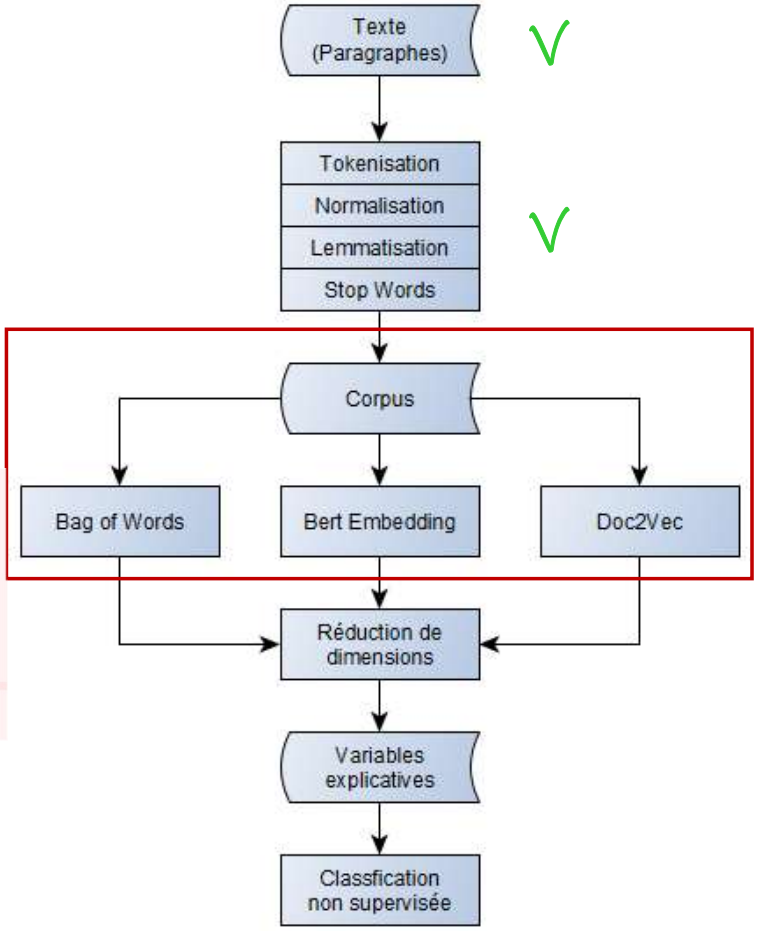
1. Analyse exploratoire des données

Traitement des données textuelles

2. Préparation du corpus
3. Traitement du corpus
4. Réduction de dimensions
5. Classification

*Le **word embedding** désigne un ensemble de techniques de machine learning qui visent à représenter les mots ou les phrases d'un texte par des vecteurs de nombres réels, décrits dans un modèle vectoriel (ou Vector Space Model).*

Source: https://dataanalyticspost.com/Lexique/word-embedding/



Bag of Words

Preprocessor_1 = Lemmatisation
Preprocessor_2 = Racinisation
stop_words = Stop Words (1) + Stop Words (2)

```
from sklearn.feature_extraction.text import CountVectorizer

c_vectorizer = CountVectorizer(analyzer = 'word',
                               preprocessor = preprocessor_1,
                               stop_words = stop_words,
                               token_pattern = r'\w+',
                               tokenizer = word_tokenize)
```

```
from sklearn.feature_extraction.text import TfidfVectorizer

t_vectorizer = TfidfVectorizer(analyzer = 'word',
                               preprocessor = preprocessor_1,
                               stop_words = stop_words,
                               token_pattern = r'\w+',
                               tokenizer = word_tokenize)
```

```
Dimensions de la matrice: (1050, 3685)
```

	abide	abilities	ability	abkl	able	abrasions	abroad	absorb	absorbency	absorbent
0	0	0	0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0	0	0	0
2	0	0	0	0	0	0	0	0	1	0
3	0	0	0	0	0	0	0	0	0	0
4	0	0	0	0	0	0	0	0	0	0
5	0	0	0	0	0	0	0	0	0	0
6	0	0	0	0	0	0	0	0	0	0
7	0	0	0	0	0	0	2	0	0	0
8	0	0	0	0	0	0	0	0	0	0
9	0	0	0	0	0	0	0	0	0	0
10	0	0	0	0	2	0	0	0	0	0
11	0	0	0	0	0	0	0	0	0	0

	abide	abilities	ability	abkl	able	abrasions	abroad	absorb	absorbency	absorbent
0	0.0	0.0	0.0	0.0	0.000000	0.0	0.00000	0.0	0.000000	0.0
1	0.0	0.0	0.0	0.0	0.000000	0.0	0.00000	0.0	0.000000	0.0
2	0.0	0.0	0.0	0.0	0.000000	0.0	0.00000	0.0	0.059665	0.0
3	0.0	0.0	0.0	0.0	0.000000	0.0	0.00000	0.0	0.000000	0.0
4	0.0	0.0	0.0	0.0	0.000000	0.0	0.00000	0.0	0.000000	0.0
5	0.0	0.0	0.0	0.0	0.000000	0.0	0.00000	0.0	0.000000	0.0
6	0.0	0.0	0.0	0.0	0.000000	0.0	0.00000	0.0	0.000000	0.0
7	0.0	0.0	0.0	0.0	0.000000	0.0	0.19236	0.0	0.000000	0.0
8	0.0	0.0	0.0	0.0	0.000000	0.0	0.00000	0.0	0.000000	0.0
9	0.0	0.0	0.0	0.0	0.000000	0.0	0.00000	0.0	0.000000	0.0
10	0.0	0.0	0.0	0.0	0.138533	0.0	0.00000	0.0	0.000000	0.0
11	0.0	0.0	0.0	0.0	0.000000	0.0	0.00000	0.0	0.000000	0.0

Doc2Vec

Méthode

- Basée sur **Word2Vec**
- Contexte des mots

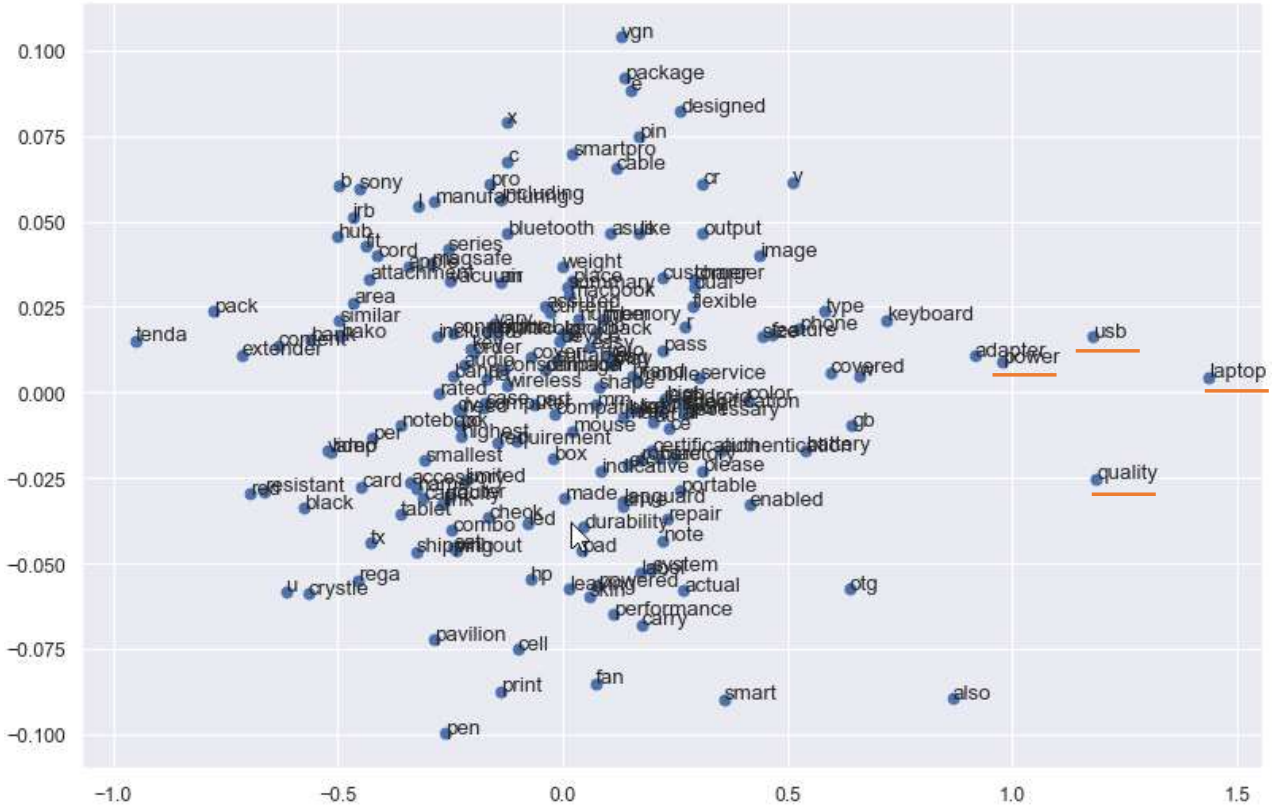
Librairie

- gensim

```
from gensim.models import doc2vec
```

```
model.most_similar('usb', topn=10)
```

```
[('factory', 0.9994964599609375),
 ('power', 0.9994903802871704),
 ('lapguard', 0.9993104934692383),
 ('assured', 0.9992811679840088),
 ('laptop', 0.9991933703422546),
 ('feature', 0.9990692734718323),
 ('port', 0.9990218281745911),
 ('like', 0.998990535736084),
 ('asus', 0.9989689588546753),
 ('portable', 0.9989268183708191)]
```



Doc2Vec

Variable

- Corpus non prétraité ('Content')
- √ Corpus prétraité

Entraînement

- Prétraitement du corpus
  - Etiqueter les paragraphes
- Instanciation

```
model = doc2vec.Doc2Vec(vector_size = 100,
                        min_count = 5,
                        epochs = 50)
```

- Entraînement

```
model.train(corpus_d2v, total_examples=model.corpus_count, epochs=model.epochs)
```

Matrice des vecteurs du corpus

```
pd.DataFrame(X_d2v).head(3)
```

	0	1	2	3	4	...
0	-1.015530	0.337350	-1.207764	-1.657250	-0.213289	...
1	-0.038099	-0.246276	-0.324762	-0.225030	0.707861	...
2	0.890005	0.225009	-0.087620	0.198586	0.314049	...

3 rows × 100 columns

Inférence

- Vecteur d'un paragraphe

Similarité

- Comparaison avec d'autres vecteurs

Doc2Vec

Evaluation du modèle

```
[['1_baby care']]
RANG 1 (1, 0.921779453754425): «sathiyas cotton bath towel specification sathiyas cotton bath towel bath towel red yellow blue
bath towel feature machine washable material cotton self brand sathiyas type bath towel gsm name sathiyas cotton bath towel ide
al men woman boy girl color red yellow blue size length width box number content package package bath towel»

[['1_baby care']]
RANG 2 (613, 0.5356090664863586): «piyo piyo four stage waterproof bib specification piyo piyo four stage waterproof bib yellow
brand piyo piyo loop type velcro material plastic name four stage waterproof bib ideal boy girl color yellow animal box number
content package package bib»

[['1_baby care']]
RANG 3 (468, 0.5283302664756775): «tks cotton hair towel key feature tks cotton hair towel size medium height gsm usage daily u
se specification tks cotton hair towel hair towel orange bath towel feature machine washable material cotton check brand tks ty
pe hair towel gsm name hair care ideal woman girl color orange size medium length width box package hair towel»

[['1_baby care']]
RANG 4 (612, 0.5206316709518433): «cheeky chompers neckerchew simple classic specification cheeky chompers neckerchew simple cl
assic sky brand cheeky chompers loop type button material cotton name neckerchew simple classic ideal boy girl cc color sky str
ip imprint box number content package package bib»

[['0_home furnishing']]
RANG 5 (104, 0.4703795611858368): «rama floral single quilt comforter yellow rama floral single quilt comforter yellow r shippi
ng»
```

Bert Embedding

Architecture

- Réseau de neurones

Librairies

- torch (gpu)
- transformers

Variable

√ • Corpus non prétraité ('Content')

Entraînement

- Chargement d'un modèle pré-entraîné
- Traitement Bert

Matrice des variables explicatives

- 768 variables

```
import torch
from torch.utils.data import DataLoader, RandomSampler, SequentialSampler, TensorDataset, Dataset
from torch import nn
from torch.nn import Module

import transformers
from transformers import BertTokenizer, BertConfig, BertPreTrainedModel, BertModel
from transformers import AlbertTokenizer, AlbertConfig, AlbertModel, AlbertPreTrainedModel
from transformers.tokenization_bert import BasicTokenizer, whitespace_tokenize
from transformers.optimization import AdamW, get_linear_schedule_with_warmup

from sentence_transformers import SentenceTransformer

print('torch:',torch.__version__)
print('transformers:',transformers.__version__)
```

```
torch: 1.0.1
transformers: 2.8.0
```

```
embedder = SentenceTransformer('bert-base-nli-mean-tokens')
```

```
X_be = embedder.encode(corpus)
```

```
pd.DataFrame(X_be).head(3)
```

	0	1	2	3	4	...
0	-0.895191	0.343897	0.122565	0.252172	0.362800	...
1	-0.681876	1.086775	-0.561551	0.335004	0.166380	...
2	-0.344509	-0.204436	0.066315	0.418061	0.178217	...

3 rows × 768 columns

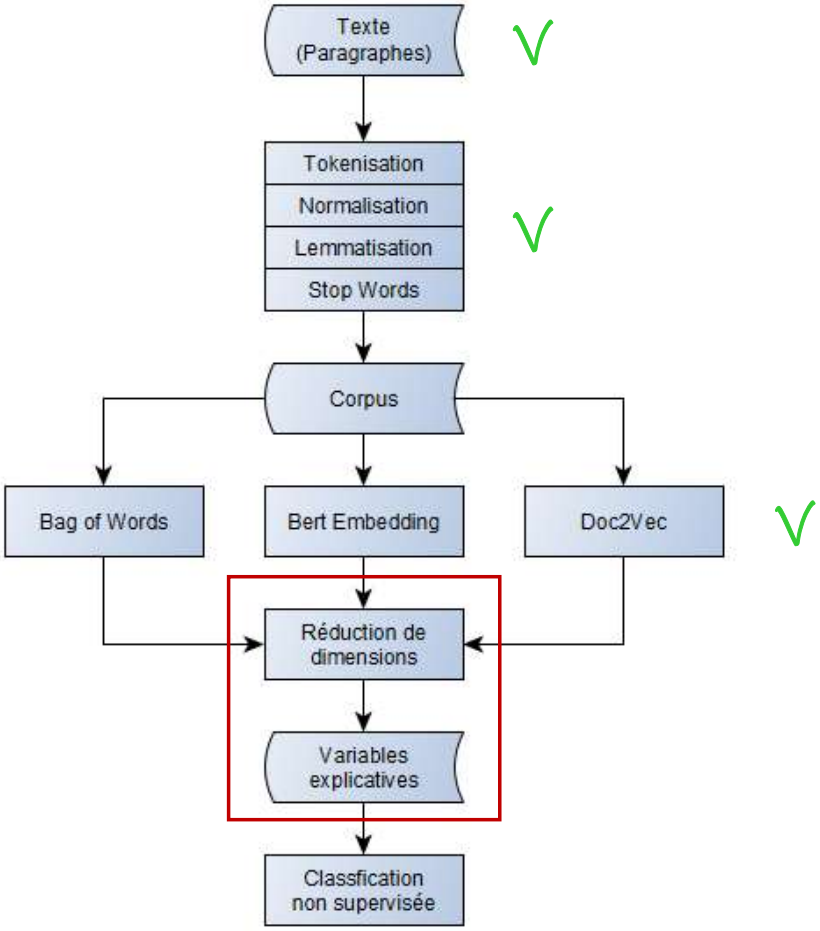
Plan

Analyse

1. Analyse exploratoire des données

Traitement des données textuelles

2. Préparation du corpus
3. Traitement du corpus
4. Réduction de dimensions
5. Classification



Pipeline

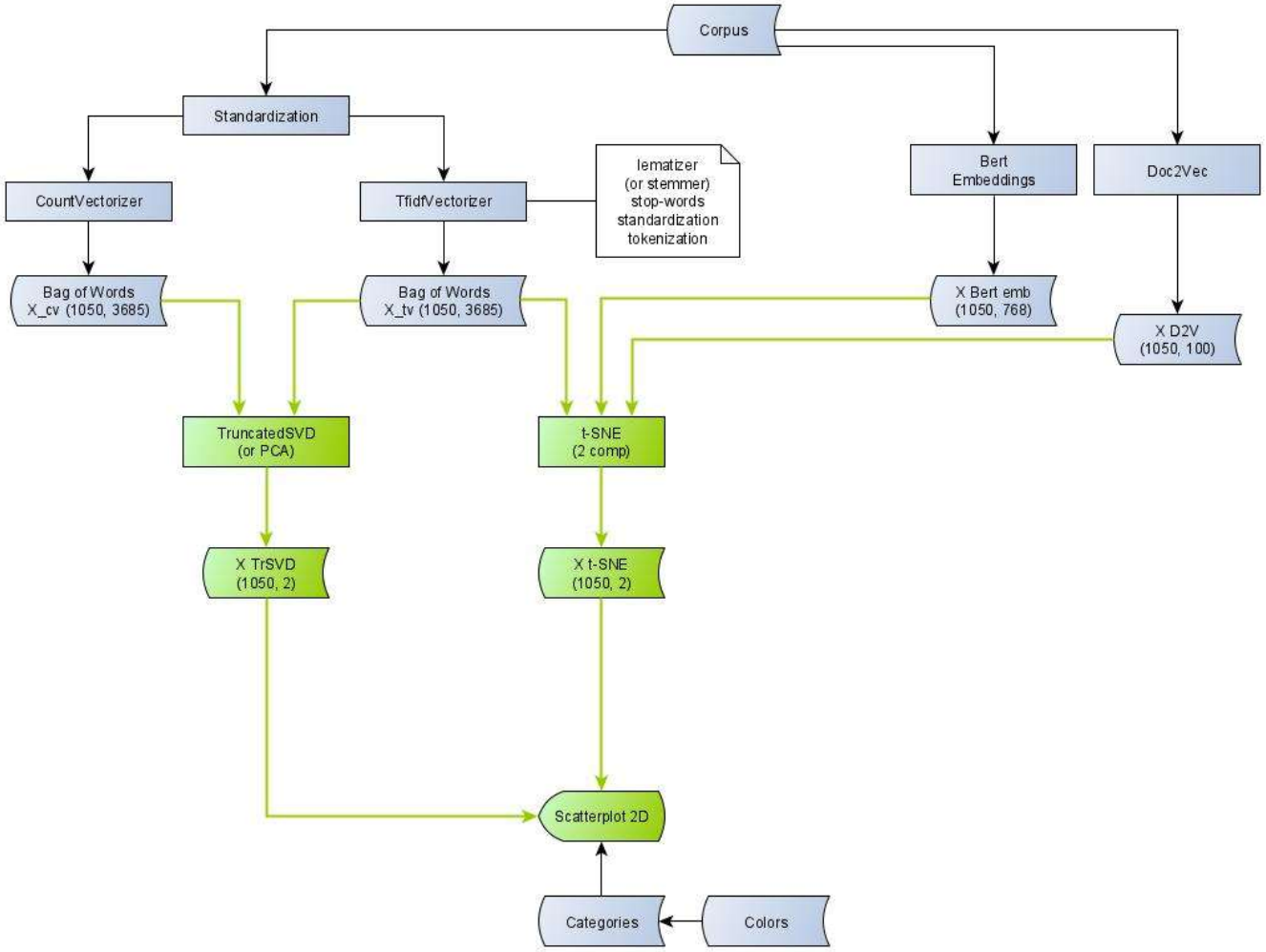
Word Embedding

Vecteurs de paragraphes

Réduction de dimensions

Variables explicatives

Visualisation

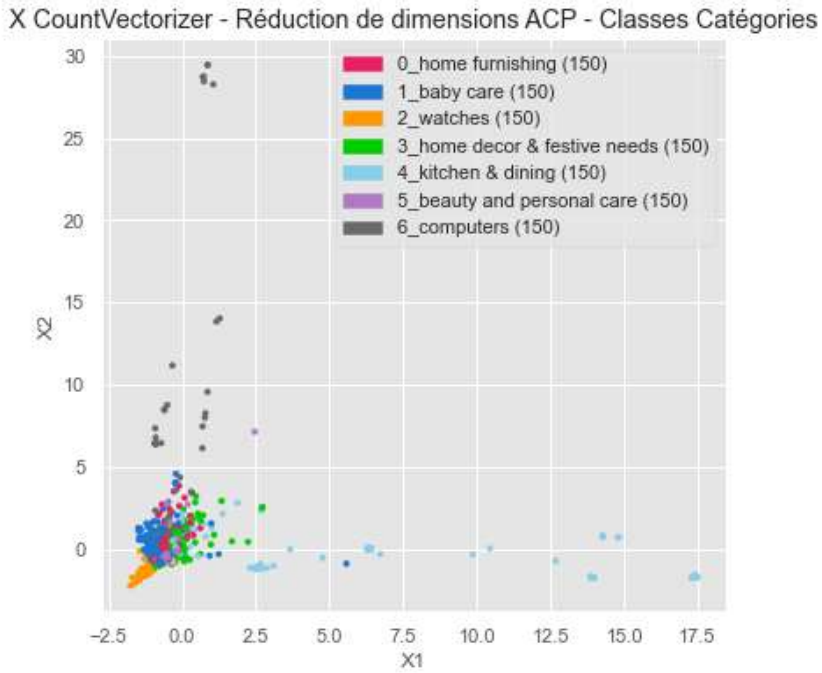


Méthodes linéaires

ACP (Analyse en Composantes Principales)

```
Nombre de dimensions: 100
Variances expliquées cumulées: 0.71
```

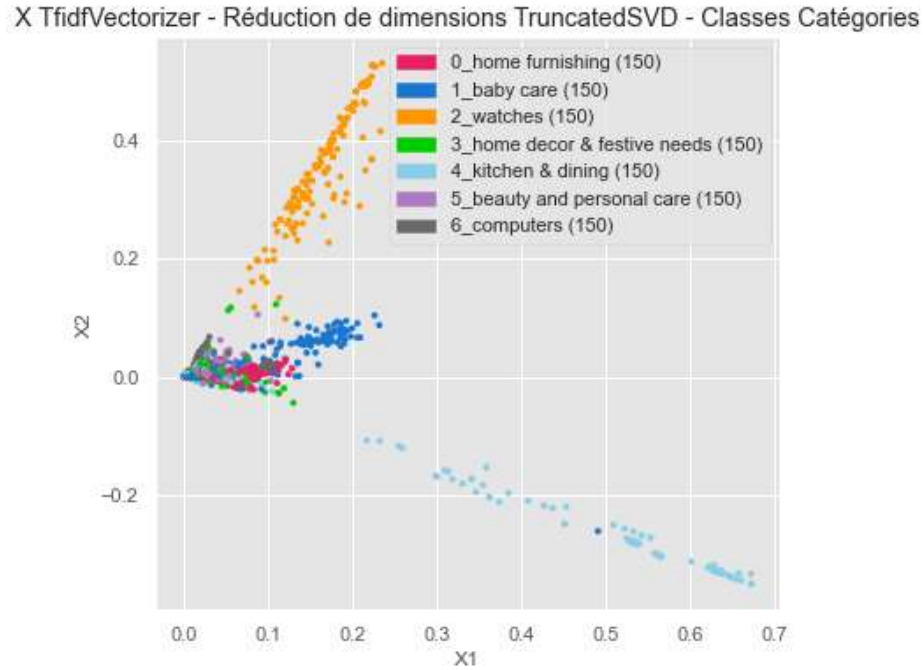
BoW



TruncatedSVD

```
Nombre de dimensions: 100
Variances expliquées cumulées: 0.46
```

BoW

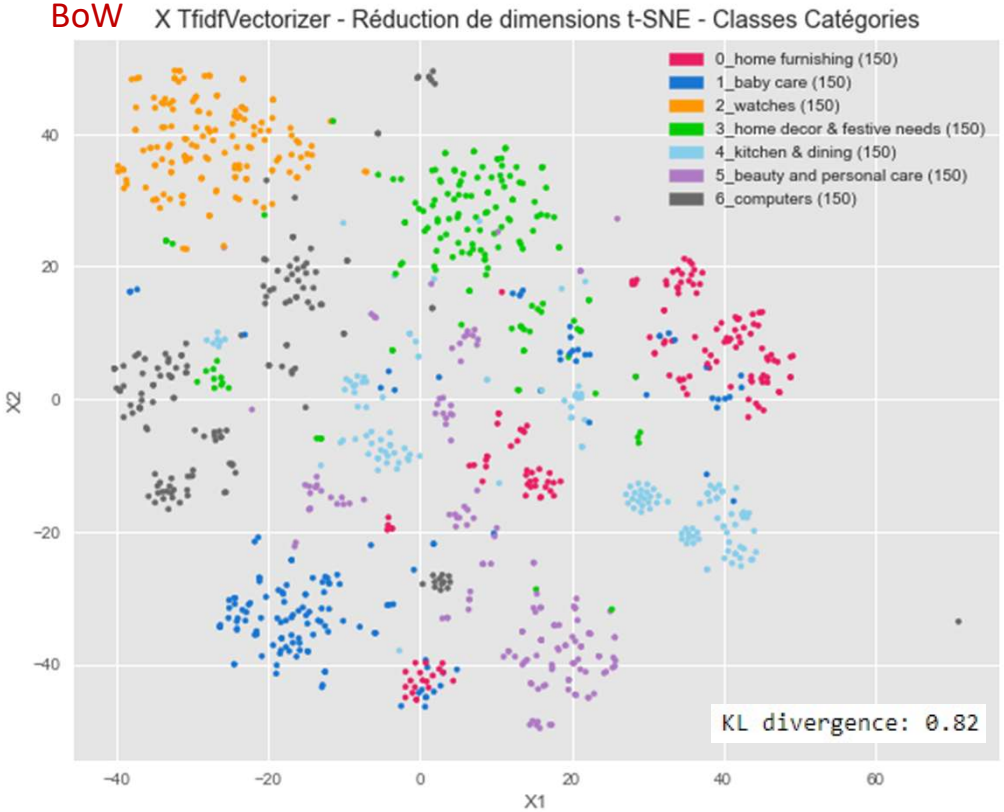


Méthode non linéaire

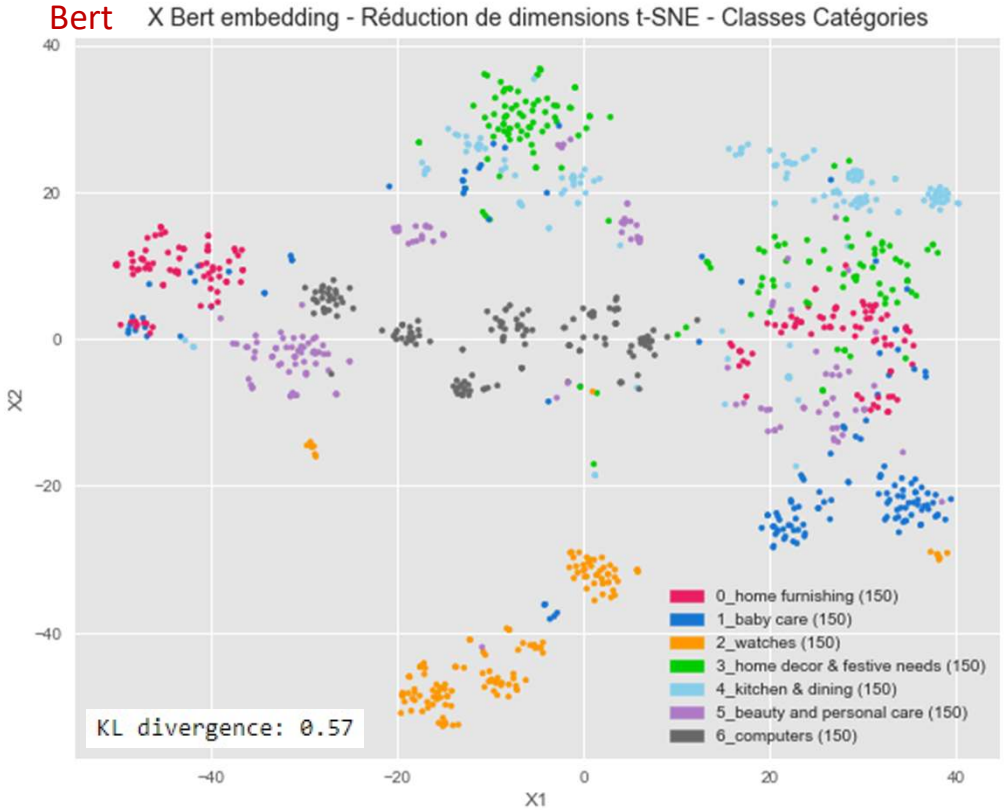
t-SNE (t-distributed Stochastic Neighbor Embedding)

Paramètres t-SNE:
{'angle': 0.5, 'early_exaggeration': 12.0, 'init': 'pca', 'learning_rate': 200.0, 'method': 'barnes_hut', 'metric': 'euclidean', 'min_grad_norm': 1e-07, 'n_components': 2, 'n_iter': 1000, 'n_iter_without_progress': 300, 'n_jobs': None, 'perplexity': 30.0, 'random_state': 42, 'verbose': 0}

BoW



Bert



Pipeline

Word Embedding

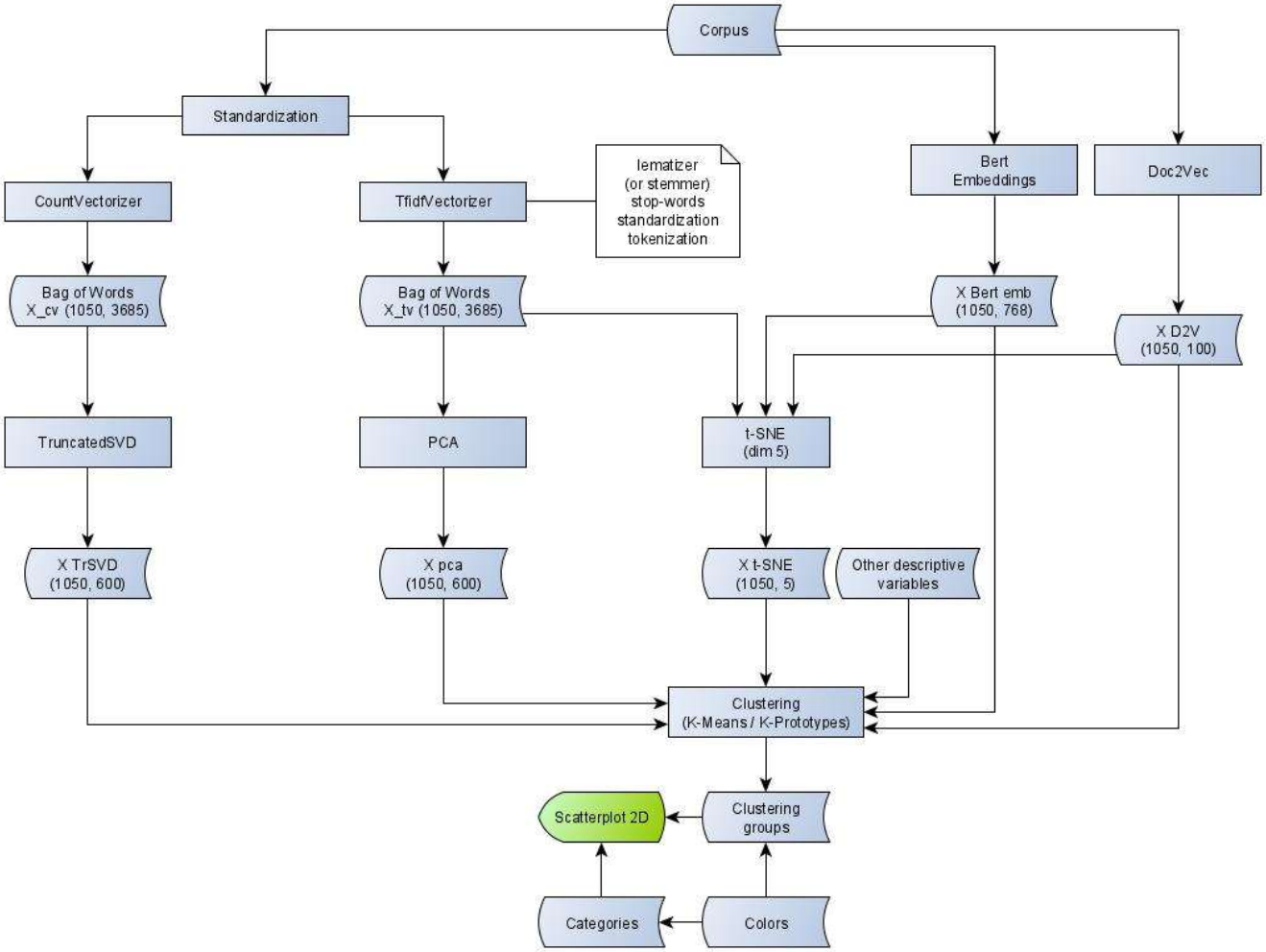
Vecteurs de paragraphes

Réduction de dimensions

Variables explicatives

Classification

Visualisation



Variables explicatives

X0 - Variables numériques (3)

X1 - Variables CountVectorizer (3685)

X2 - Variables Doc2Vec (100)

X3 - Variables Bert Embedding (768)

X4 - Variables catégorielles (4)

X1tr - Variables CountVectorizer / TruncatedSVD (600)

X1ts - Variables TfidfVectorize / t-SNE (5)

X2tr - Variables Doc2Vec / TruncatedSVD (60)

X2ts - Variables Doc2Vec / t-SNE (5)

X3tr - Variables Bert Embedding / TruncatedSVD (600)

X3ts - Variables Bert Embedding / t-SNE (5)

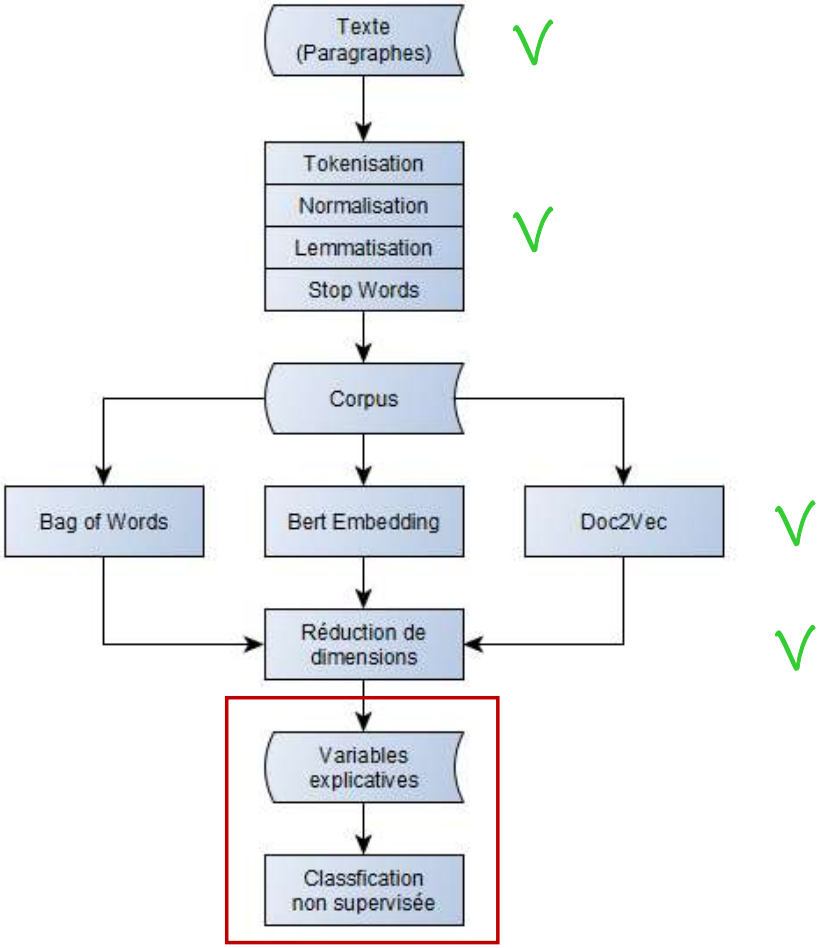
Plan

Analyse

1. Analyse exploratoire des données

Traitement des données textuelles

2. Préparation du corpus
3. Traitement du corpus
4. Réduction de dimensions
5. Classification



Variables explicatives

X1tr - Variables CountVectorizer / TruncatedSVD

	0	1	2	3	4	5	6	7	8	9	...
0	0.110333	-0.010399	0.118035	0.151303	0.216792	-0.025524	0.087112	0.029343	0.445827	-0.315743	...
1	0.114454	0.057951	0.227105	0.129096	-0.046987	-0.003585	-0.262021	0.621787	-0.075640	-0.005703	...
2	0.120606	0.030781	0.220177	0.203635	-0.116016	0.014842	-0.210891	0.435976	-0.049655	0.003849	...

3 rows × 600 columns

X1ts - Variables TfidfVectorize / t-SNE

	0	1	2	3	4
0	-12.550852	-1.246770	-4.999907	-5.360623	0.944706
1	0.452470	0.238703	-5.236247	-5.808298	-7.744123
2	3.431585	3.644700	-1.894061	-7.565708	-8.056687

X0 - Variables numériques standardisées

	brand_le	retail_price	discounted_price
0	-1.295836	-0.037357	-0.091592
1	0.278348	-0.207556	-0.151848
2	-1.246255	-0.286169	-0.211970

X2ts - Variables Doc2Vec / t-SNE

	0	1	2	3	4
0	13.713598	11.233138	-7.938262	-16.897545	-12.425076
1	-4.642823	8.261921	25.470991	10.445700	-0.549518
2	-8.725098	7.914375	24.268187	6.380171	-7.232146

X4 - Variables catégorielles

index	cat_0	type	sales package	model name	brand
0	0_home furnishing	eyelet	2 curtains	abstract polyester door curtain set of 2	Elegance
1	1_baby care	bath towel	3 bath towel	sathiyas cotton bath towel	Sathiyas
2	1_baby care	face towel set	20 piece face towel set	shuvam20pcftsetassorted	Eurospa

X3ts - Variables Bert Embedding / t-SNE

	0	1	2	3	4
0	8.745142	-3.706951	2.956133	-0.936375	-0.045379
1	9.675542	3.684541	-0.964925	-4.801901	-4.123096
2	8.664647	1.282675	2.084318	-5.329134	-3.508431

Classification par apprentissage non supervisé

Algorithmes

- K-Means
```
from sklearn.cluster import KMeans
```
- K-Prototypes
```
from kmodes.kprototypes import KPrototypes
```

Variables explicatives

- Différents jeux de données
  - Bag of Words
  - Doc2Vec
  - Bert Embedding
  - Variables numériques et catégorielles
- Standardisation des données

Bag of Words Bert Embedding

```
df_X_kmeans = df_X1ts.merge(df_X3ts, left_index=True, right_index=True)
df_X_kmeans.head(3)
```

	0_x	1_x	2_x	3_x	4_x	0_y	1_y	2_y	3_y	4_y
0	-12.550852	-1.246770	-4.999907	-5.360623	0.944706	8.745142	-3.706951	2.956133	-0.936375	-0.045379
1	0.452470	0.238703	-5.236247	-5.808298	-7.744123	9.675542	3.684541	-0.964925	-4.801901	-4.123096
2	3.431585	3.644700	-1.894061	-7.565708	-8.056687	8.664647	1.282675	2.084318	-5.329134	-3.508431

Variable catégorielle

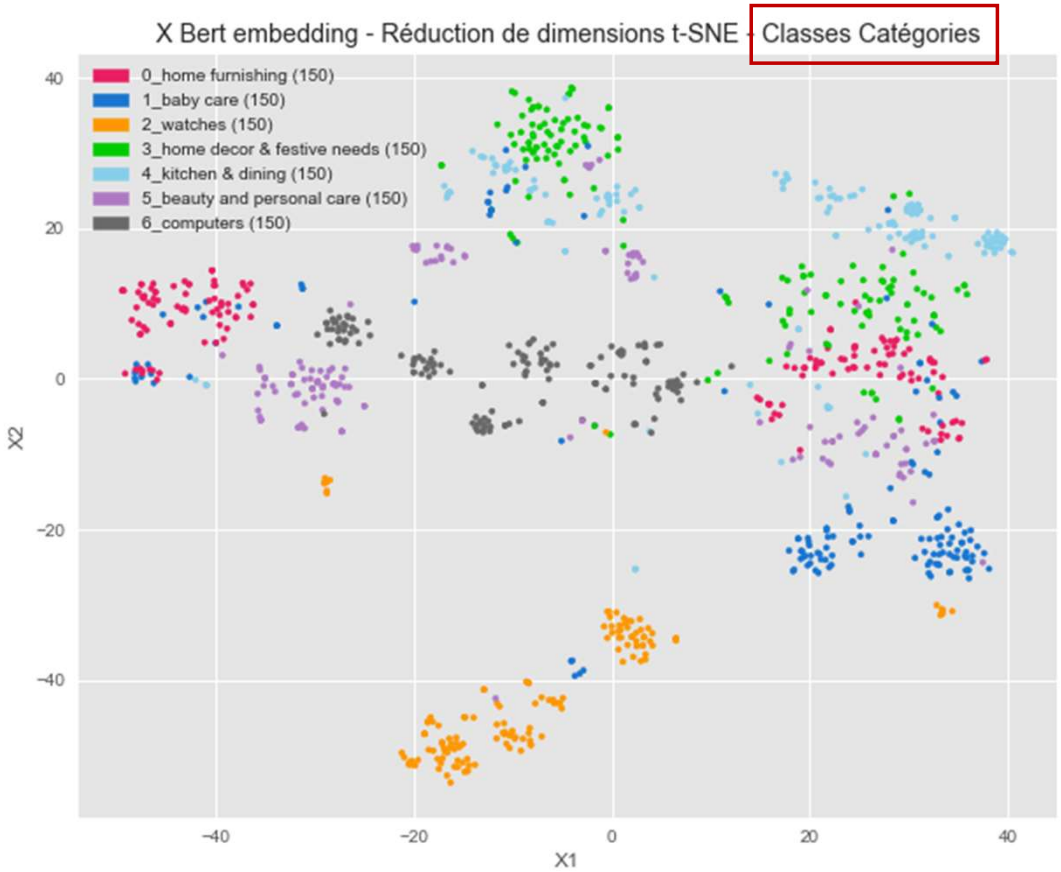
```
df_X_kpro = df_X_kmeans
df_X_kpro = df_X_kpro.merge(df_X4, left_index=True, right_index=True)
df_X_kpro.head(3)
```

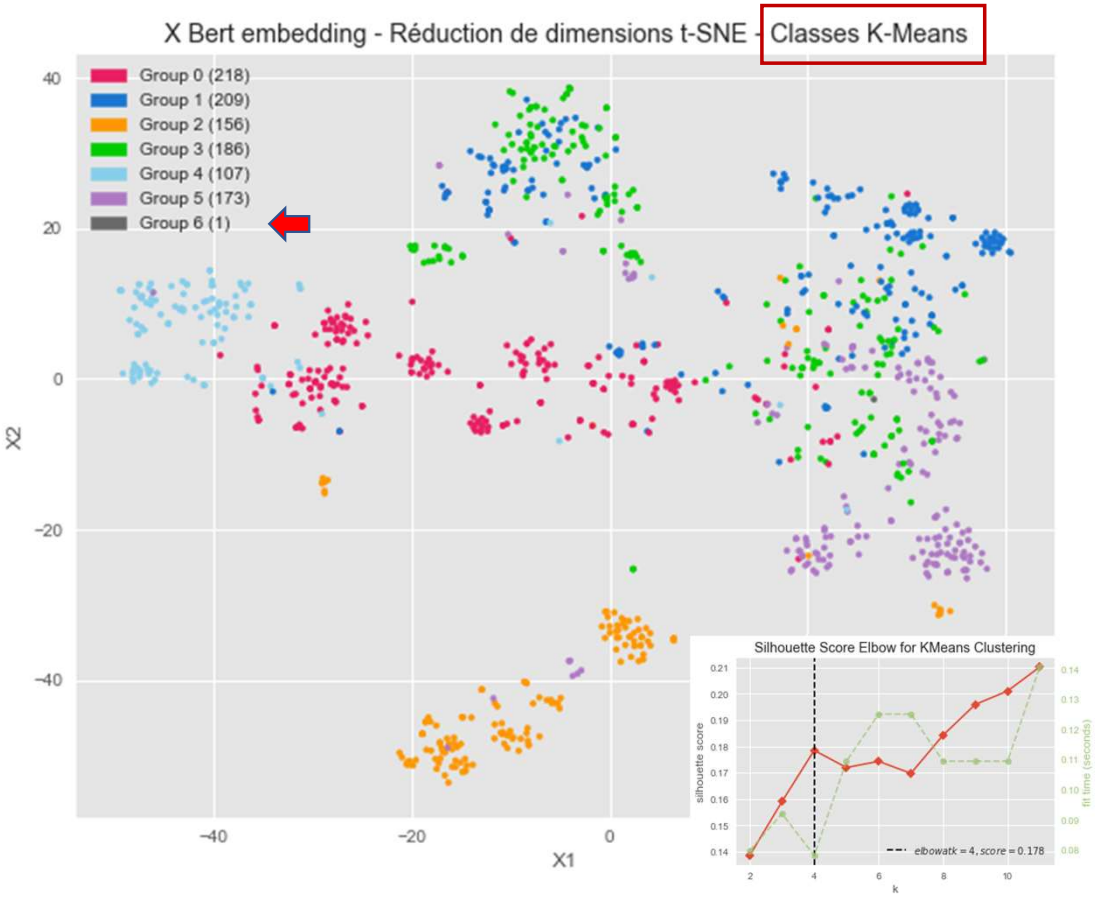
	0	1	2	3	4	5	6	7	8	9	brand
0	-1.788310	-0.200850	-0.707453	-0.931206	0.118965	1.121138	-0.664887	0.459498	-0.179621	-0.001551	Elegance
1	0.019882	0.045149	-0.740483	-1.006413	-1.251096	1.239613	0.718363	-0.167942	-0.967657	-0.839492	Sathiyas
2	0.434146	0.609191	-0.273399	-1.301645	-1.300381	1.110888	0.268876	0.319992	-1.075140	-0.713183	Eurospa

Classification

Visualisation

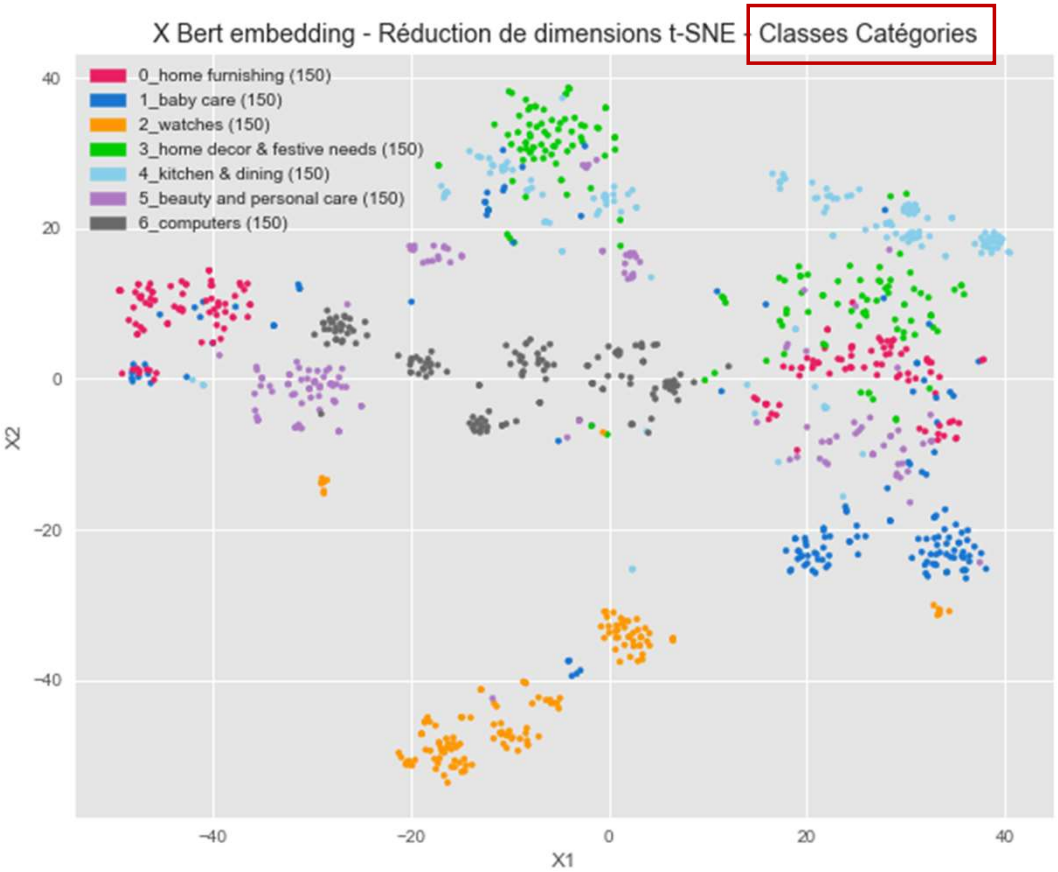
Bag of Words (TfidfVectorizer) / t-SNE (dim 5)
Bert Embedding / t-SNE (dim 5)

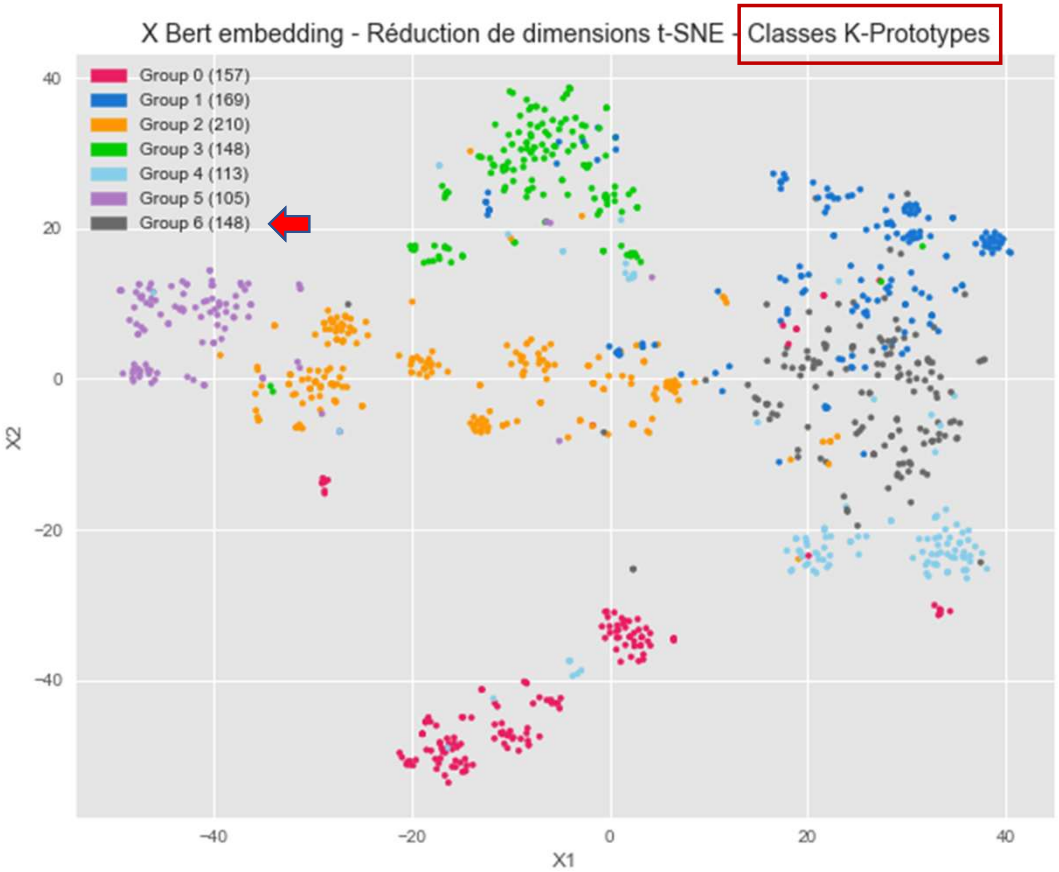




Visualisation

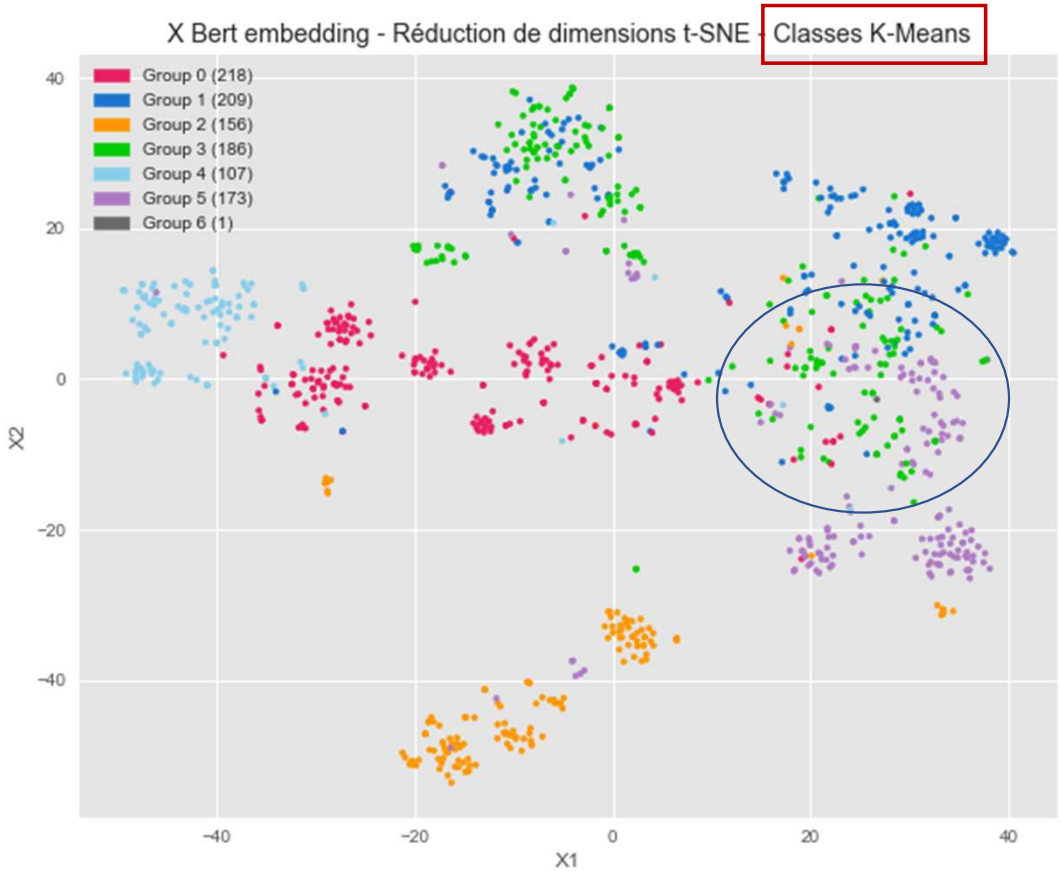
Bag of Words (TfidfVectorizer) / t-SNE (dim 5)
Bert Embedding / t-SNE (dim 5)
Variable catégorielle ('brand')

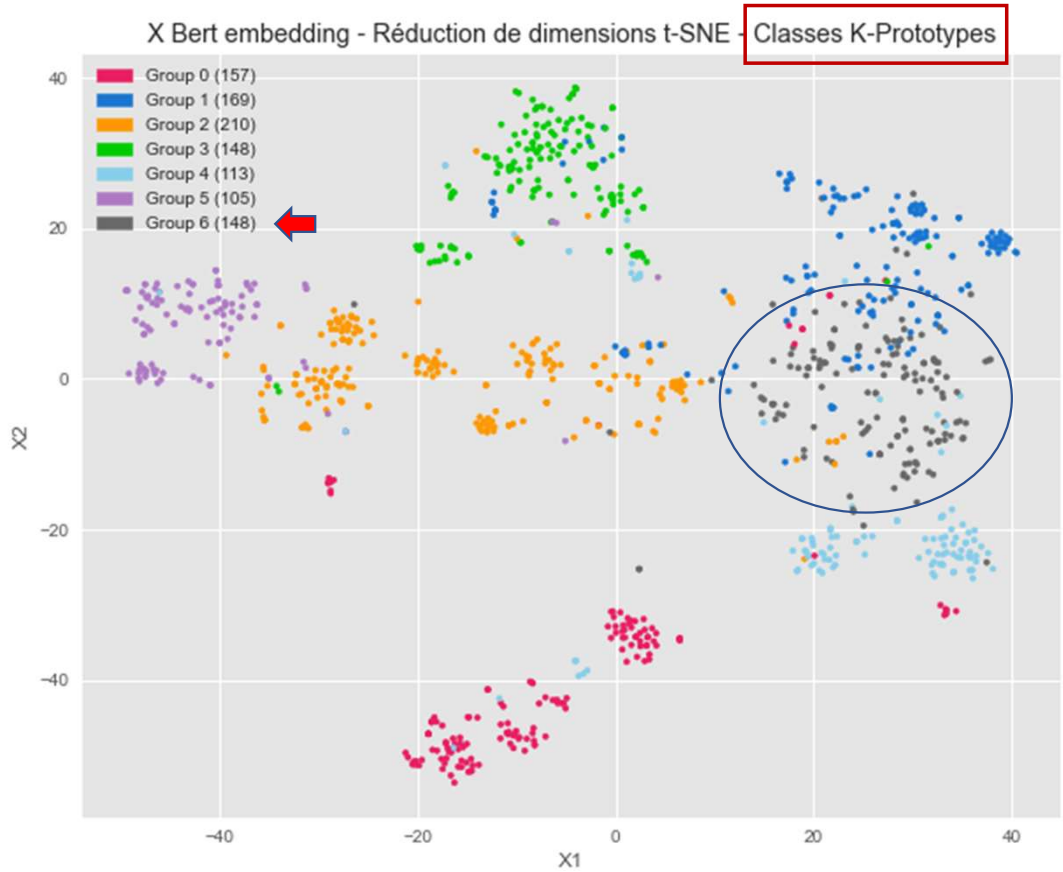




Visualisation

Bag of Words (TfidfVectorizer) / t-SNE (dim 5)
Bert Embedding / t-SNE (dim 5)
Variable catégorielle ('brand')



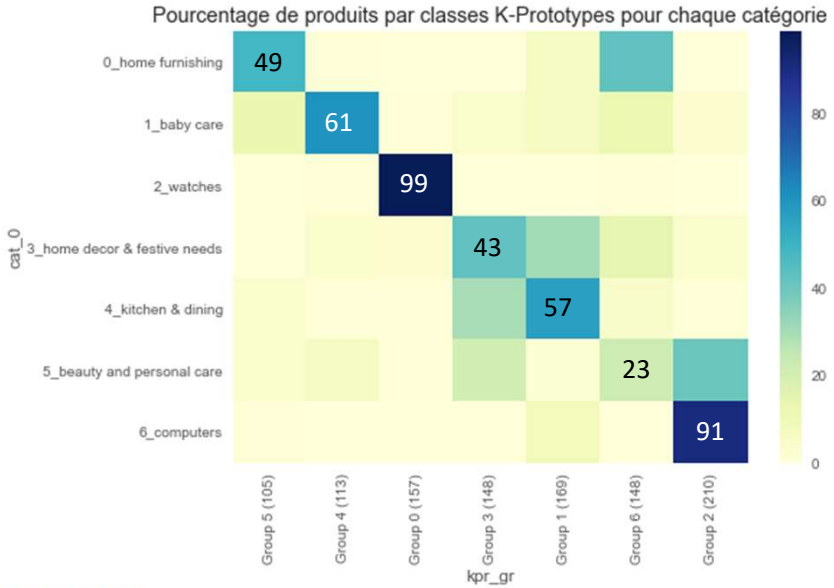


Evaluation

Indice de Rand

- Mesure de similarité entre 2 partitions

```
Indice de Rand pour K-Prototypes: 0.422
```

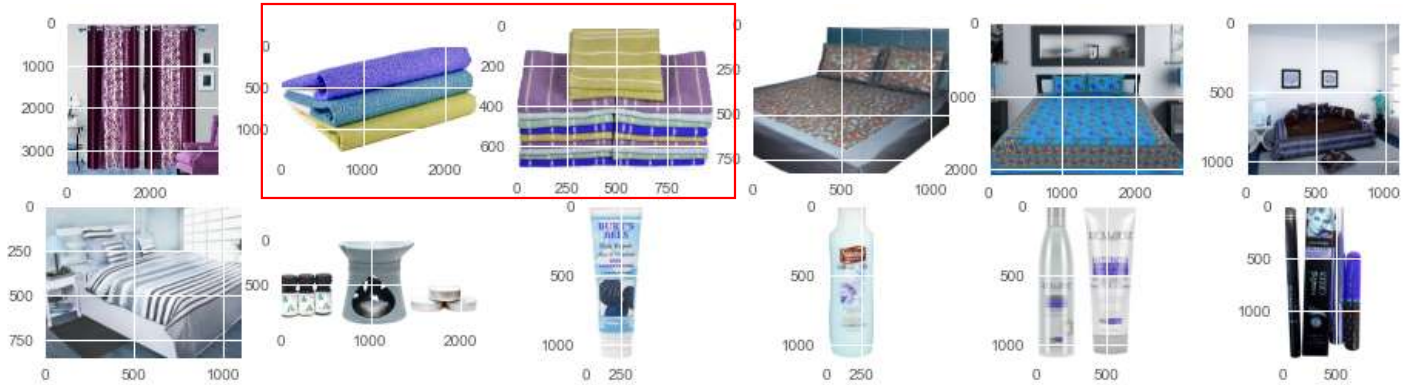



```
Pourcentage de produits par classes K-Prototypes pour chaque catégorie
----------------------------------------------------------------------
```

kpr_gr cat_0	Group 5 (105)	Group 4 (113)	Group 0 (157)	Group 3 (148)	Group 1 (169)	Group 6 (148)	Group 2 (210)
0_home furnishing	49.0	1.0	0.0	0.0	7.0	43.0	0.0
1_baby care	13.0	61.0	1.0	4.0	7.0	12.0	3.0
2_watches	0.0	1.0	99.0	0.0	0.0	1.0	0.0
3_home decor & festive needs	0.0	4.0	3.0	43.0	31.0	15.0	4.0
4_kitchen & dining	4.0	1.0	1.0	30.0	57.0	5.0	1.0
5_beauty and personal care	4.0	7.0	1.0	21.0	2.0	23.0	41.0
6_computers	1.0	0.0	0.0	0.0	9.0	0.0	91.0

Evaluation

Prédictions



	prod_id	cat_0	kpr_gr	image	product_name
0	0	0_home furnishing	Group 6 (148)	55b85ea15a1536d46b7190ad6fff8ce7.jpg	Elegance Polyester Multicolor Abstract Eyelet ...
1	1	1_baby care	Group 6 (148)	7b72c92c2f6c40268628ec5f14c6d590.jpg	Sathiyas Cotton Bath Towel
2	2	1_baby care	Group 6 (148)	64d5d4a258243731dc7bbb1eef49ad74.jpg	Eurospa Cotton Terry Face Towel Set
3	3	0_home furnishing	Group 6 (148)	d4684dcdc759dd9cdf41504698d737d8.jpg	SANTOSH ROYAL FASHION Cotton Printed King size...
4	4	0_home furnishing	Group 6 (148)	6325b6870c54cd47be6ebfbffa620ec7.jpg	Jaipur Print Cotton Floral King sized Double B...
33	33	0_home furnishing	Group 6 (148)	b85f8843bb1c297dd5712f0bd287006d.jpg	SANTOSH ROYAL FASHION Cotton Embroidered Diwan...
34	34	0_home furnishing	Group 6 (148)	a4b09aa7927c59fbb39960b3d7919909.jpg	House This Queen Cotton Duvet Cover

Evaluation

Prédictions

	prod_id	cat_0	kpr_gr	image	product_name
40	40	1_baby care	Group 4 (113)	90e56526f4c5892beac49f262b85150a.jpg	Mom and Kid Baby Girl's Printed Green Top & Py...
42	42	1_baby care	Group 4 (113)	5325213d84c80ad310b6e4b37a1ac824.jpg	Mom and Kid Baby Girl's Printed Blue, Grey Top...
63	63	1_baby care	Group 4 (113)	f4e7f269319ba9cf5507a9e2febacbf5.jpg	KOHL Wine Bag Yellow
68	68	4_kitchen & dining	Group 4 (113)	ed9817e8960a6a0d509061c54cfe5ac1.jpg	SKI WINNER GIFT SET SKYBLUE02 2 Containers Lun...
69	69	4_kitchen & dining	Group 4 (113)	5a7a1287fef0d986a5e237797beee286.jpg	GAYATRI CREATIONS ORANGE KIDS SET 2 Containers...
73	73	1_baby care	Group 4 (113)	141d2b743c477043041faddf9233c8ad.jpg	Vitamins Solid Baby Girl's Basic Shorts
74	74	1_baby care	Group 4 (113)	7a3f11f380a1bf85338ab3771ff81e9f.jpg	Vitamins Graphic Print Baby Girl's Basic Shorts

Evaluation

```
[['4_kitchen & dining']]
RANG 1 (68, 0.9305205941200256): «ski winner set skyblue container lunch box specification ski winner set skyblue container lun
ch box ml series low name winner set skyblue lunch box feature lunch box material plastic number container lunch box capacity m
l box lunch ml water bottle»
```

✓

```
[['4_kitchen & dining']]
RANG 2 (69, 0.7157008647918701): «gayatri creation orange kid set container lunch box specification gayatri creation orange kid
set container lunch box ml series name orange kid set lunch box feature lunch box material plastic number container lunch box c
apacity ml box lunch box water bottle»
```

✗

```
[['5_beauty and personal care']]
RANG 3 (684, 0.6552934646606445): «oshea herbal uvshield sunscreen gel spf pa oshea herbal uvshield sunscreen gel spf pa r lowe
st shipping»
```

✗

```
[['0_home furnishing']]
RANG 4 (754, 0.6522870659828186): «onlinemaniya abstract single quilt comforter multicolor onlinemaniya abstract single quilt c
omforter multicolor r shipping»
```

✗

```
[['2_watches']]
RANG 5 (29, 0.6344615817070007): «elantra analog watch boy men elantra analog watch boy men elantra analog watch boy men r indi
a great discount shipping»
```

	prod_id	cat_0	kpr_gr	image	product_name
40	40	1_baby care	Group 4 (113)	90e56526f4c5892beac49f262b85150a.jpg	Mom and Kid Baby Girl's Printed Green Top & Py...
42	42	1_baby care	Group 4 (113)	5325213d84c80ad310b6e4b37a1ac824.jpg	Mom and Kid Baby Girl's Printed Blue, Grey Top...
63	63	1_baby care	Group 4 (113)	f4e7f269319ba9cf5507a9e2febacbf5.jpg	KOHL Wine Bag Yellow
68	68	4_kitchen & dining	Group 4 (113)	ed9817e8960a6a0d509061c54cfe5ac1.jpg	SKI WINNER GIFT SET SKYBLUE02 2 Containers Lun...
69	69	4_kitchen & dining	Group 4 (113)	5a7a1287fef0d986a5e237797beee286.jpg	GAYATRI CREATIONS ORANGE KIDS SET 2 Containers...
73	73	1_baby care	Group 4 (113)	141d2b743c477043041faddf9233c8ad.jpg	Vitamins Solid Baby Girl's Basic Shorts
74	74	1_baby care	Group 4 (113)	7a3f11f380a1bf85338ab3771ff81e9f.jpg	Vitamins Graphic Print Baby Girl's Basic Shorts

Plan

Analyse

1. Analyse exploratoire des données

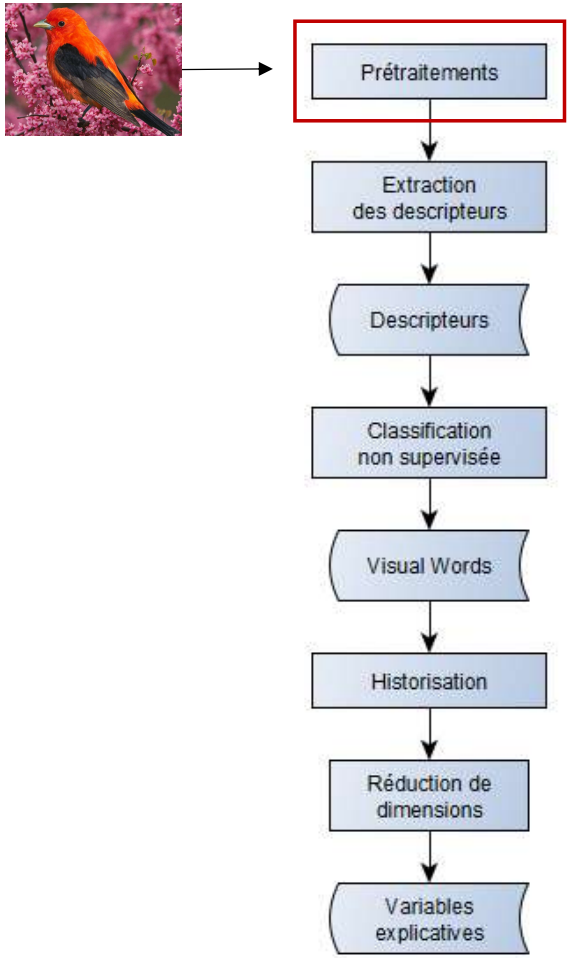
Traitement des données textuelles

2. Préparation du corpus
3. Traitement du corpus
4. Réduction de dimensions
5. Classification

Traitement des images

7. Prétraitement
8. Détermination des variables explicatives
9. Classification
10. Réseaux de neurones convolutifs

- *CNN*
- *Keras VGG16*
- *Transfer Learning*

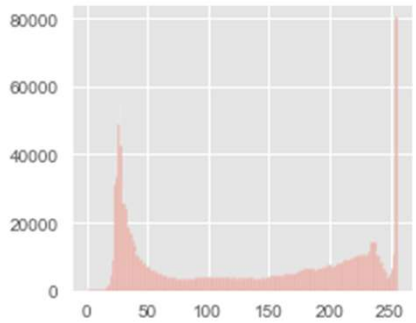


Configuration

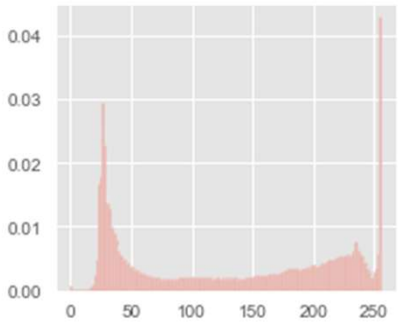
Prétraitements

	Réduction du bruit	Egalisation	
Librairies	ImageFilter.BoxBlur()	ImageOps.equalize()	PIL
	ImageFilter.GaussianBlur()	ImageOps.equalize()	
	cv2.fastNlMeansDenoisingColored()	cv2.equalizeHist()	OpenCV
	cv2.fastNlMeansDenoisingColored()	cv2.createCLAHE()	

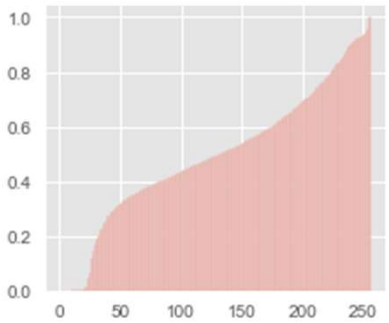
Histogrammes



Histogramme



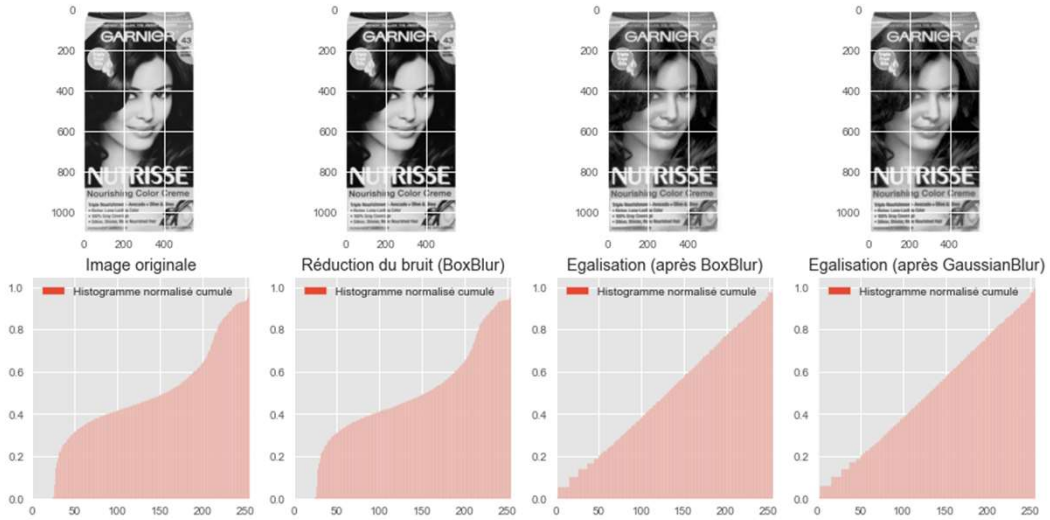
Histogramme normalisé



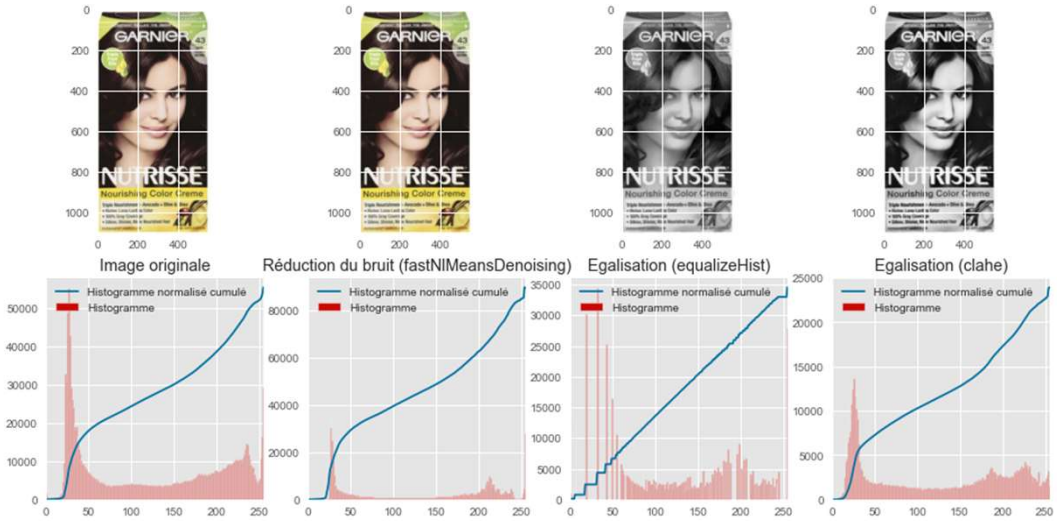
Histogramme normalisé cumulé

Evaluation

PIL



OpenCV



7 Prétraitement des images

Evaluation

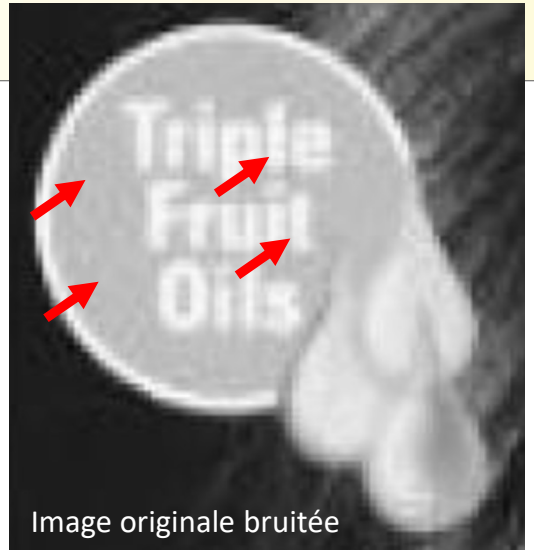
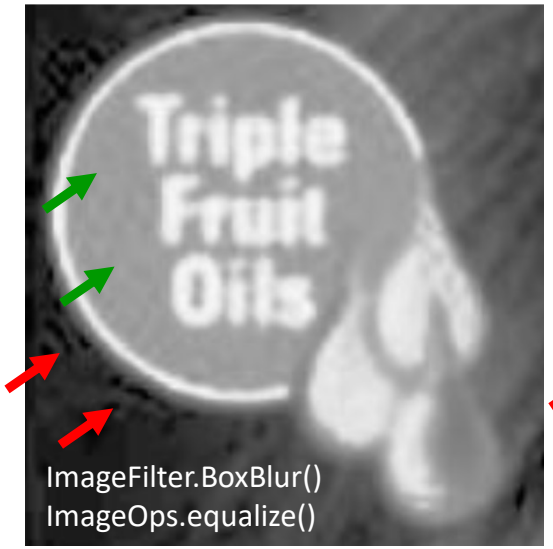
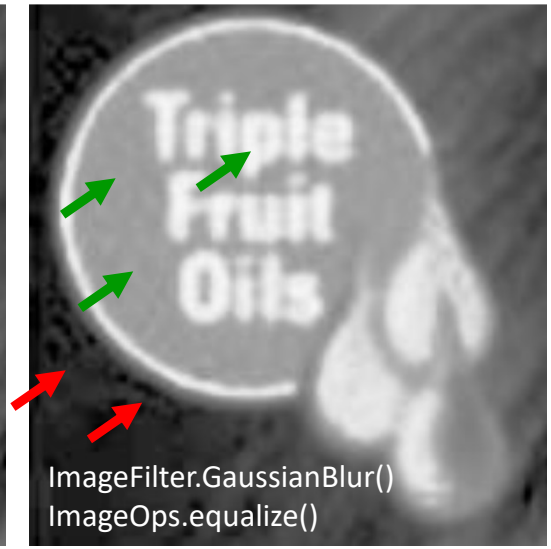


Image originale bruitée

PIL

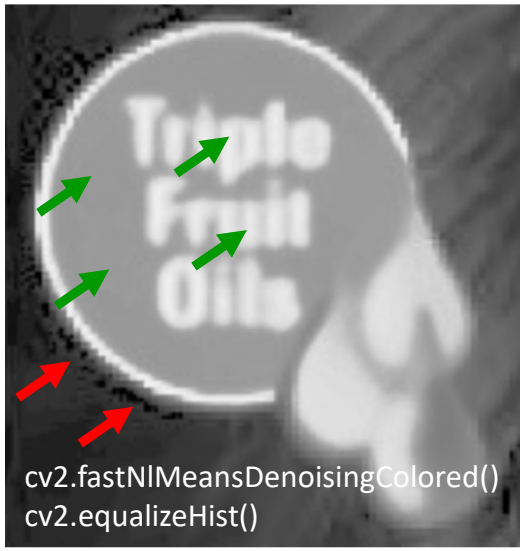


ImageFilter.BoxBlur()
ImageOps.equalize()



ImageFilter.GaussianBlur()
ImageOps.equalize()

OpenCV



cv2.fastNlMeansDenoisingColored()
cv2.equalizeHist()

cv2.fastNlMeansDenoisingColored()
cv2.createCLAHE()

Evaluation

PIL

ImageFilter.BoxBlur()
ImageOps.equalize()

ImageFilter.GaussianBlur()
ImageOps.equalize()

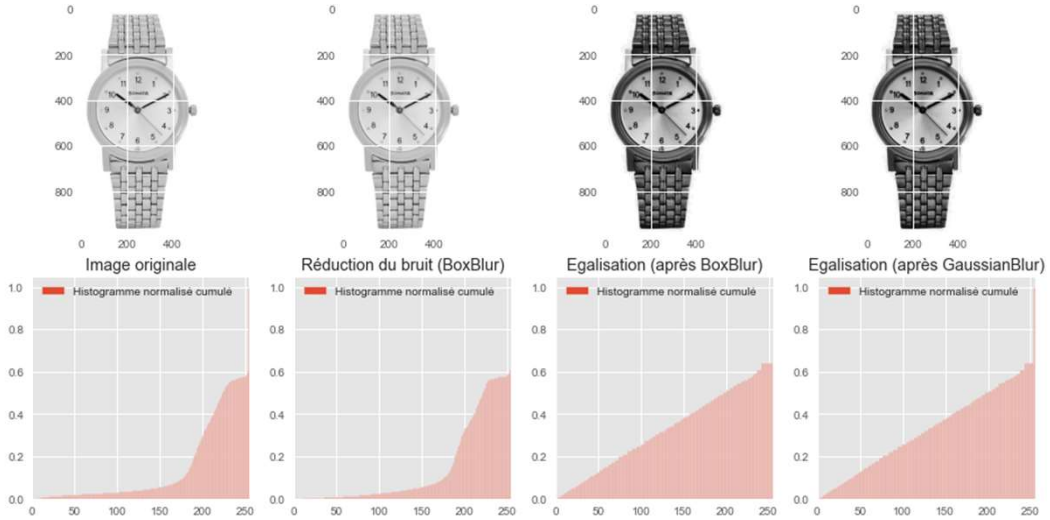
OpenCV

cv2.fastNlMeansDenoisingColored()
cv2.equalizeHist()

cv2.fastNlMeansDenoisingColored()
cv2.createCLAHE()

Evaluation

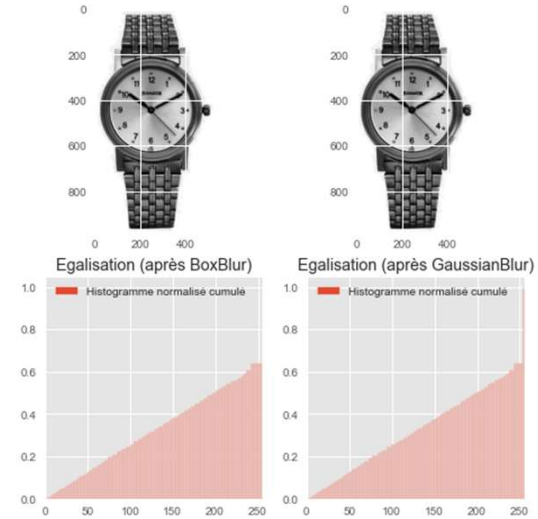
PIL



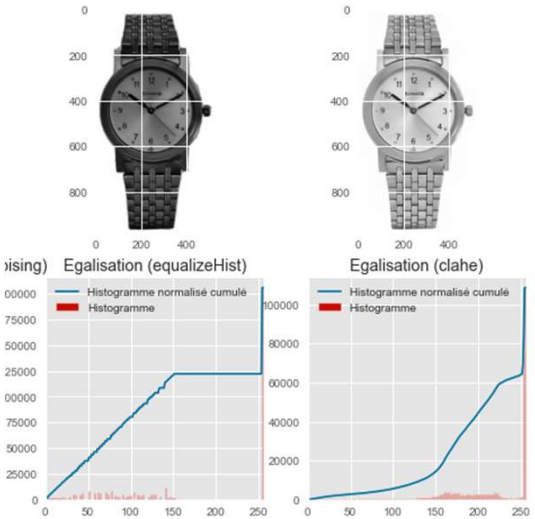
OpenCV

Evaluation

PIL



OpenCV



Configuration

Librairie retenue

- OpenCV

Méthodes retenues

✗

cv2.fastNlMeansDenoisingColored()

cv2.equalizeHist()

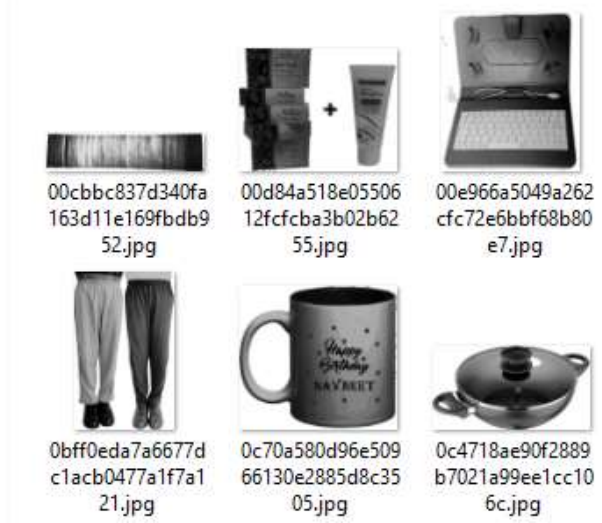
✓

cv2.fastNlMeansDenoisingColored()

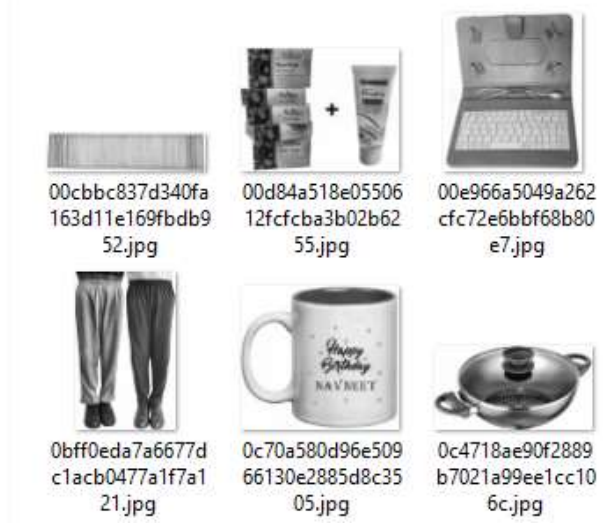
cv2.createCLAHE()

OCDS_Projet_6 › images_ori

OCDS_Projet_6 › images_pro_equ_1



OCDS_Projet_6 › images_pro_equ_2



Plan

Analyse

1. Analyse exploratoire des données

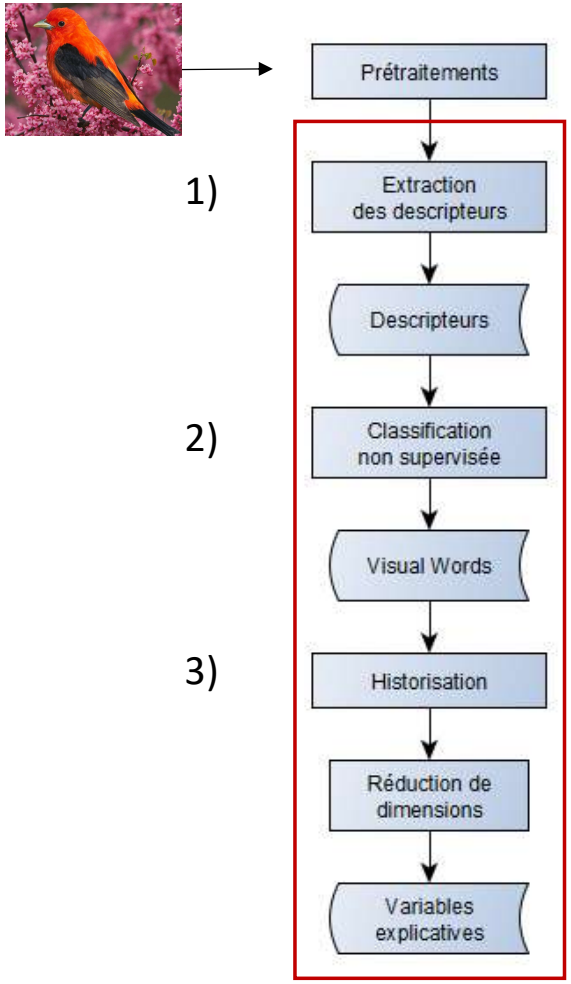
Traitement des données textuelles

2. Préparation du corpus
3. Traitement du corpus
4. Réduction de dimensions
5. Classification

Traitement des images

7. Prétraitement
8. Détermination des variables explicatives
9. Classification
10. Réseaux de neurones convolutifs

- *CNN*
- *Keras VGG16*
- *Transfer Learning*

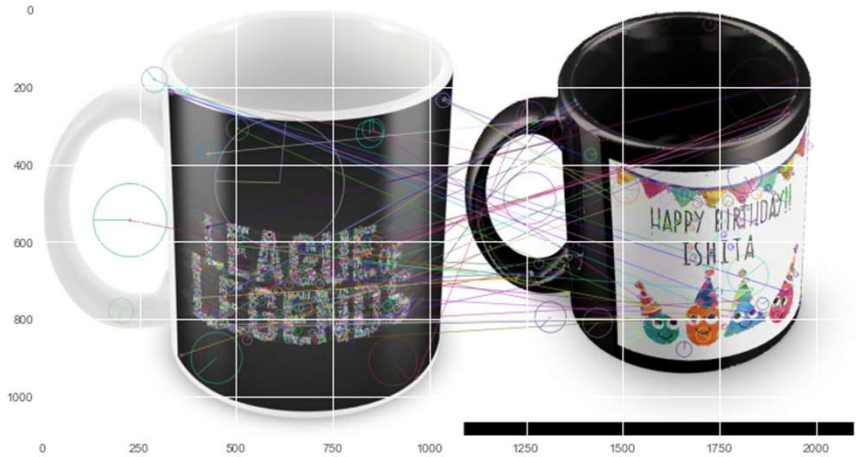


Extraction des descripteurs

Identifier des produits semblables

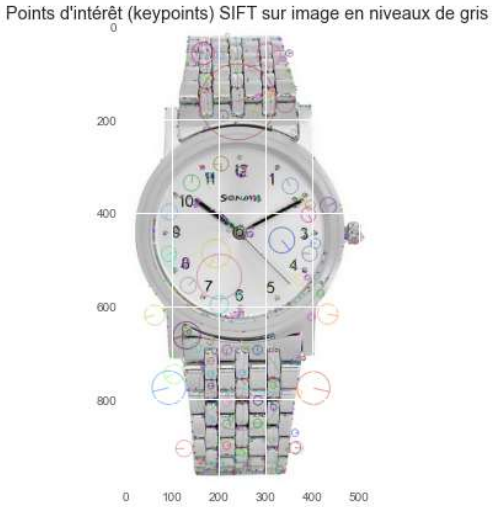
- Trouver des éléments communs
- ➢ Image Matching

```
Nombre de points d'intérêts pour l'image 1: 1100
Nombre de points d'intérêts pour l'image 2: 1066
Nombre de corrélations (matches): 291
```



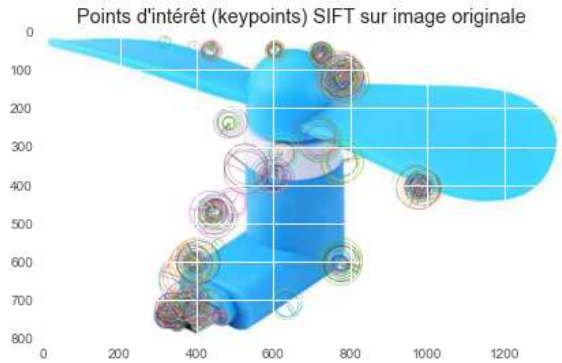
Points d'intérêt et descripteurs

- OpenCV (SIFT, SURF, ORB)
- ➢ Détection des points d'intérêt
  - Filtres de détection des bords, des coins…
- ➢ Calcul des descripteurs
  - ➢ Description du voisinage des features




```
Nombre de points d'intérêt SIFT sur image originale: 979
```

Extraction des descripteurs




```
Nombre de points d'intérêt SIFT sur image originale: 802
```

Cette image est décrite par 802 vecteurs de dimension 128

Points d'intérêt et descripteurs

- OpenCV (SIFT, SURF, ORB)

➢ Détection des points d'intérêt
  - Filtres de détection des bords, des coins…

➢ Calcul des descripteurs
  - ➢ Description de la zone d'intérêt
  - ➢ Descripteur = vecteur de dimension 128

Identifiants des images

	0	1	2	3	4	5	6	7	8	9	...	119	120	121	122	123	124	125	126	127	128
0	0.0	4.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	11.0	...	1.0	18.0	94.0	22.0	7.0	6.0	9.0	3.0	0.0	20.0
1	0.0	0.0	0.0	1.0	150.0	71.0	0.0	0.0	0.0	69.0	...	0.0	6.0	19.0	3.0	40.0	49.0	0.0	0.0	0.0	14.0
2	0.0	32.0	1.0	0.0	0.0	0.0	0.0	0.0	6.0	137.0	...	1.0	37.0	13.0	0.0	0.0	3.0	137.0	46.0	1.0	2.0
3	0.0	0.0	0.0	0.0	0.0	17.0	33.0	2.0	2.0	1.0	...	2.0	0.0	1.0	3.0	1.0	1.0	1.0	39.0	23.0	0.0
4	0.0	12.0	8.0	7.0	140.0	49.0	0.0	0.0	0.0	140.0	...	0.0	33.0	67.0	1.0	0.0	0.0	0.0	0.0	0.0	14.0
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
5979103	1049.0	0.0	28.0	60.0	24.0	3.0	0.0	0.0	0.0	0.0	...	0.0	12.0	15.0	109.0	11.0	0.0	0.0	0.0	0.0	1.0
5979104	1049.0	0.0	0.0	0.0	2.0	0.0	0.0	0.0	0.0	0.0	...	0.0	25.0	98.0	56.0	24.0	1.0	0.0	0.0	0.0	74.0
5979105	1049.0	4.0	10.0	1.0	23.0	11.0	34.0	53.0	3.0	112.0	...	25.0	20.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	2.0
5979106	1049.0	0.0	2.0	33.0	110.0	11.0	0.0	0.0	0.0	40.0	...	1.0	67.0	47.0	94.0	27.0	0.0	0.0	0.0	0.0	49.0
5979107	1049.0	2.0	0.0	0.0	9.0	62.0	9.0	25.0	35.0	27.0	...	24.0	14.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0

5979108 rows × 129 columns

Visual Words

Recherche des centroïdes

- Classification non supervisée (K-Means)
- ➢ 100 Classes de descripteurs

Identifiants des images

	0	1	2	3	4	5	6	7	8	9	...	119	120	121	122	123	124	125	126	127	128
0	0.0	4.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	11.0	...	1.0	18.0	94.0	22.0	7.0	6.0	9.0	3.0	0.0	20.0
1	0.0	0.0	0.0	1.0	150.0	71.0	0.0	0.0	0.0	69.0	...	0.0	6.0	19.0	3.0	40.0	49.0	0.0	0.0	0.0	14.0
2	0.0	32.0	1.0	0.0	0.0	0.0	0.0	0.0	6.0	137.0	...	1.0	37.0	13.0	0.0	0.0	3.0	137.0	46.0	1.0	2.0
3	0.0	0.0	0.0	0.0	0.0	17.0	33.0	2.0	2.0	1.0	...	2.0	0.0	1.0	3.0	1.0	1.0	1.0	39.0	23.0	0.0
4	0.0	12.0	8.0	7.0	140.0	49.0	0.0	0.0	0.0	140.0	...	0.0	33.0	67.0	1.0	0.0	0.0	0.0	0.0	0.0	14.0
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
5979103	1049.0	0.0	28.0	60.0	24.0	3.0	0.0	0.0	0.0	0.0	...	0.0	12.0	15.0	109.0	11.0	0.0	0.0	0.0	0.0	1.0
5979104	1049.0	0.0	0.0	0.0	2.0	0.0	0.0	0.0	0.0	0.0	...	0.0	25.0	98.0	56.0	24.0	1.0	0.0	0.0	0.0	74.0
5979105	1049.0	4.0	10.0	1.0	23.0	11.0	34.0	53.0	3.0	112.0	...	25.0	20.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	2.0
5979106	1049.0	0.0	2.0	33.0	110.0	11.0	0.0	0.0	0.0	40.0	...	1.0	67.0	47.0	94.0	27.0	0.0	0.0	0.0	0.0	49.0
5979107	1049.0	2.0	0.0	0.0	9.0	62.0	9.0	25.0	35.0	27.0	...	24.0	14.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0

5979108 rows × 129 columns

100 Visual Words

	0	1	2	3	4	5	6	7	8	9	...
0	12.804762	6.402398	14.219050	75.515265	56.903582	11.698046	9.300564	16.951997	16.264395	9.045712	...
1	34.410332	16.131904	9.360204	17.613459	41.823590	13.890978	5.495197	9.552384	124.433946	39.849277	...
2	22.713944	25.063744	27.786025	25.351235	22.316666	12.287588	10.509140	12.152560	53.967688	37.098783	...
3	9.388243	10.019488	10.276994	25.614165	71.562619	45.822325	10.340344	6.423864	46.799928	44.335472	...
4	23.081347	24.681803	20.057639	32.922950	48.505112	22.737961	7.694398	8.554271	103.528056	53.206212	...
...	...	...	...	...	...	...	...	...	...	...	...
95	14.584277	16.032448	21.469540	28.145435	23.658767	15.249362	11.841989	12.687671	28.644442	22.493801	...
96	13.710255	15.403954	23.955639	36.072052	35.865314	26.621673	17.199511	14.178906	16.491613	20.418498	...
97	83.654032	19.011834	4.082773	7.719740	25.933796	8.286295	3.719709	17.758295	43.969163	10.149853	...
98	30.032375	22.157810	17.607560	28.963706	43.162751	23.804933	13.369615	16.195700	83.140237	41.902515	...
99	14.041840	7.533766	7.037104	10.515034	12.483568	5.569893	4.114048	6.423447	95.964024	21.094173	...

100 rows × 128 columns

Historisation

Historisation

- On compare les descripteurs aux centroïdes
- Pour 1 image:
  - On prédit la classe de chaque descripteur
  - On compte le nombre de valeur par classe
  - Normalisation

Variables explicatives

- 100 variables
  - Visual Words
- Valeurs
  - Décompte normalisé des Visual Words

+1 (0)

+1 (1)

+1 (1)

Identifiants des images

	0	1	2	3	4	5	6	7	8	9	...	119	120	121	122	123	124	125	126	127	128
0	0.0	4.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	11.0	...	1.0	18.0	94.0	22.0	7.0	6.0	9.0	3.0	0.0	20.0
1	0.0	0.0	0.0	1.0	150.0	71.0	0.0	0.0	0.0	69.0	...	0.0	6.0	19.0	3.0	40.0	49.0	0.0	0.0	0.0	14.0
2	0.0	32.0	1.0	0.0	0.0	0.0	0.0	0.0	6.0	137.0	...	1.0	37.0	13.0	0.0	0.0	3.0	137.0	46.0	1.0	2.0
3	0.0	0.0	0.0	0.0	0.0	17.0	33.0	2.0	2.0	1.0	...	2.0	0.0	1.0	3.0	1.0	1.0	1.0	39.0	23.0	0.0
4	0.0	12.0	8.0	7.0	140.0	49.0	0.0	0.0	0.0	140.0	...	0.0	33.0	67.0	1.0	0.0	0.0	0.0	0.0	0.0	14.0
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
5979103	1049.0	0.0	28.0	60.0	24.0	3.0	0.0	0.0	0.0	0.0	...	0.0	12.0	15.0	109.0	11.0	0.0	0.0	0.0	0.0	1.0
5979104	1049.0	0.0	0.0	0.0	2.0	0.0	0.0	0.0	0.0	0.0	...	0.0	25.0	98.0	56.0	24.0	1.0	0.0	0.0	0.0	74.0
5979105	1049.0	4.0	10.0	1.0	23.0	11.0	34.0	53.0	3.0	112.0	...	25.0	20.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	2.0
5979106	1049.0	0.0	2.0	33.0	110.0	11.0	0.0	0.0	0.0	40.0	...	1.0	67.0	47.0	94.0	27.0	0.0	0.0	0.0	0.0	49.0
5979107	1049.0	2.0	0.0	0.0	9.0	62.0	9.0	25.0	35.0	27.0	...	24.0	14.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0

5979108 rows × 129 columns

	0	1	2	3	4	5	6	7	8	9	...
0	0.003461	0.002193	0.006471	0.002332	0.017477	0.004611	0.001741	0.002709	0.006137	0.004740	...
1	0.009312	0.008683	0.005304	0.013987	0.007976	0.014498	0.009194	0.010490	0.016855	0.010726	...
2	0.000390	0.023500	0.003116	0.005583	0.017138	0.011166	0.002986	0.005064	0.006362	0.002077	...
3	0.010726	0.021972	0.005623	0.019369	0.008747	0.008227	0.009997	0.010934	0.015620	0.018744	...
4	0.014256	0.012907	0.009825	0.007398	0.002543	0.019419	0.004354	0.010056	0.020652	0.015258	...
...	...	...	...	...	...	...	...	...	...	...	...
1045	0.008466	0.007183	0.011288	0.008722	0.009749	0.013853	0.006414	0.011288	0.009492	0.003335	...
1046	0.004768	0.002384	0.008343	0.007151	0.022646	0.008343	0.004768	0.003576	0.003576	0.010727	...
1047	0.011144	0.004261	0.006227	0.010161	0.006227	0.002950	0.022288	0.017044	0.005900	0.009177	...
1048	0.002141	0.000000	0.008565	0.002141	0.027837	0.002141	0.000000	0.000000	0.002141	0.006424	...
1049	0.012313	0.010171	0.008298	0.014989	0.006959	0.005353	0.010707	0.010439	0.006424	0.009101	...

1050 rows × 100 columns

Plan

Analyse

1. Analyse exploratoire des données

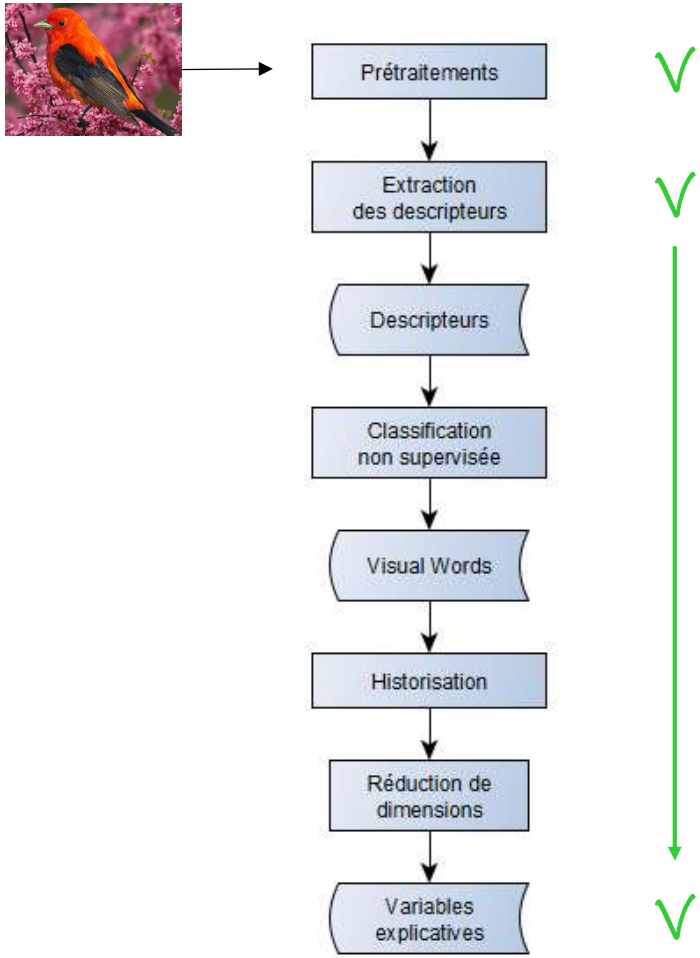
Traitement des données textuelles

2. Préparation du corpus
3. Traitement du corpus
4. Réduction de dimensions
5. Classification

Traitement des images

7. Prétraitement
8. Détermination des variables explicatives
9. Classification
10. Réseaux de neurones convolutifs

- *CNN*
- *Keras VGG16*
- *Transfer Learning*



Réduction de dimensions

ACP

```
Nombre de dimensions: 80
Variances expliquées cumulées: 0.99
```

Classification supervisée

Variable cible

- Catégories niveau 1

Jeux d'entraînement et de test

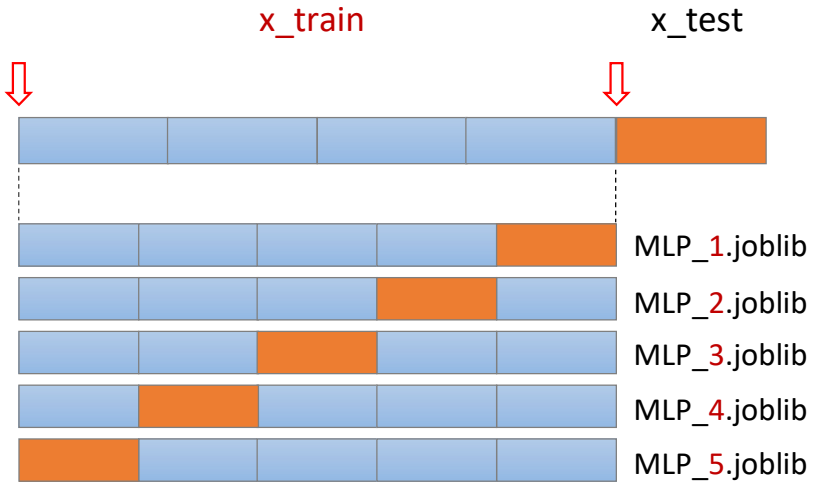
- x_train, x_test

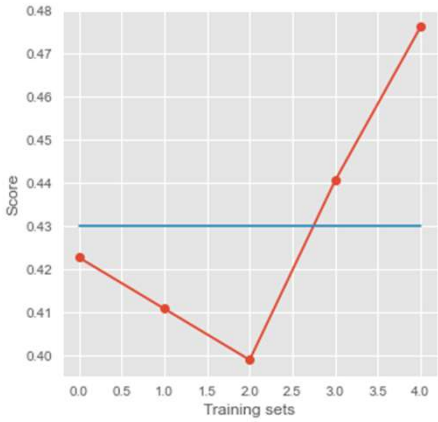
Traitement par validation croisée

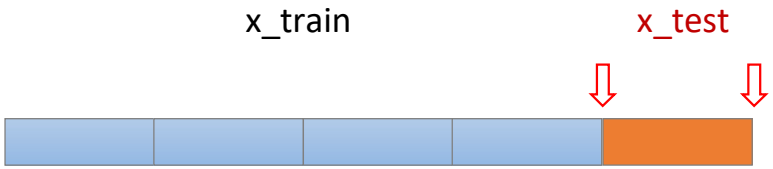
- MLPClassifier
- Export 5 modèles

Prédictions

- Import 5 modèles
- Prédictions par modèles
- ➢ Moyenne des % de prédictions par catégorie





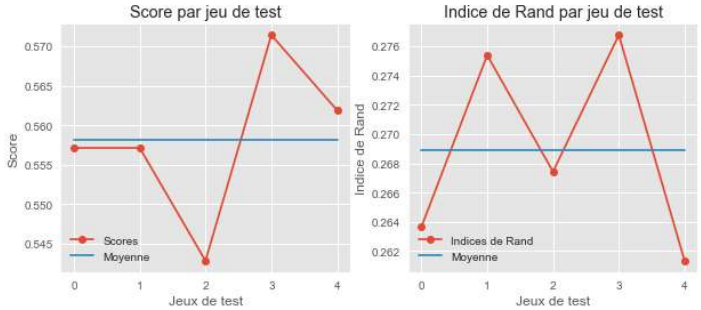


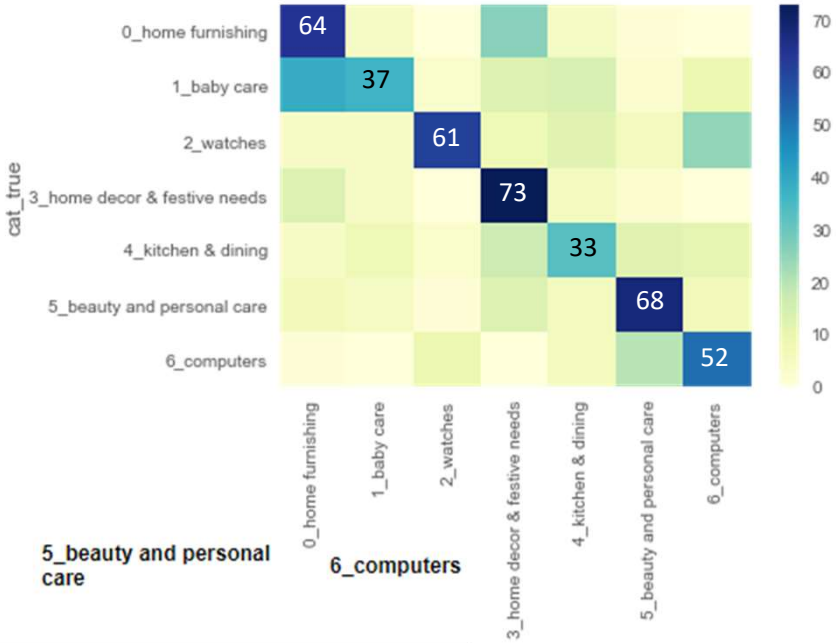
Evaluation

Prétraitements

- Images originales

Score moyen: 0.558
Indice de Rand moyen: 0.269





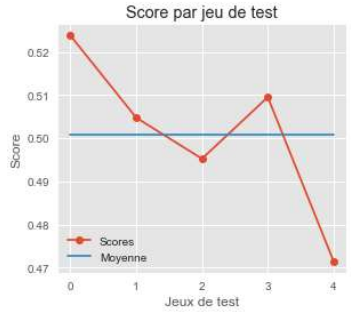
cat_true	0_home furnishing	1_baby care	2_watches	3_home decor & festive needs	4_kitchen & dining	5_beauty and personal care	6_computers
0_home furnishing	64.0	5.0	0.0	26.0	5.0	1.0	0.0
1_baby care	39.0	37.0	3.0	13.0	14.0	2.0	9.0
2_watches	4.0	4.0	61.0	8.0	12.0	6.0	25.0
3_home decor & festive needs	13.0	4.0	0.0	73.0	6.0	2.0	0.0
4_kitchen & dining	4.0	8.0	3.0	17.0	33.0	12.0	11.0
5_beauty and personal care	7.0	5.0	1.0	13.0	6.0	68.0	7.0
6_computers	1.0	0.0	9.0	0.0	6.0	20.0	52.0

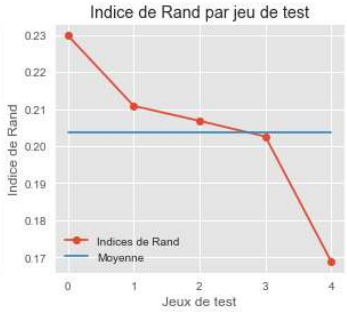
Evaluation

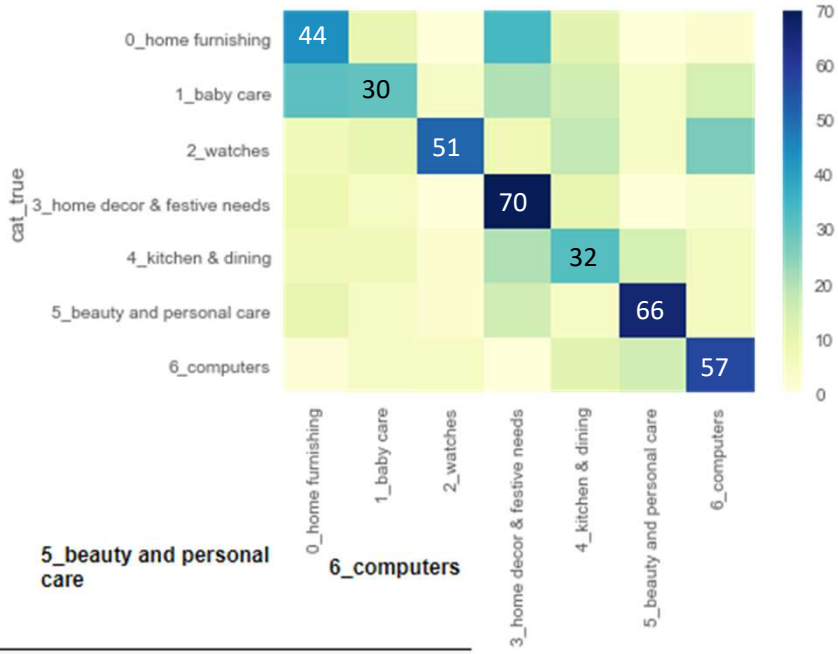
Prétraitements

- Réduction du bruit
- Egalisation

```
Score moyen: 0.501
Indice de Rand moyen: 0.204
```







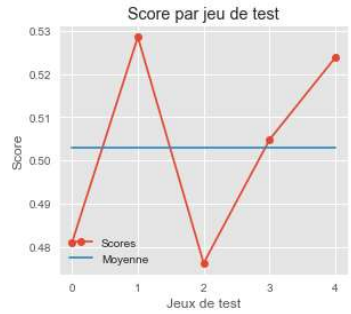
cat_true	0_home furnishing	1_baby care	2_watches	3_home decor & festive needs	4_kitchen & dining	5_beauty and personal care	6_computers
0_home furnishing	44.0	10.0	0.0	34.0	11.0	0.0	2.0
1_baby care	31.0	30.0	4.0	20.0	15.0	4.0	14.0
2_watches	7.0	10.0	51.0	8.0	18.0	4.0	27.0
3_home decor & festive needs	9.0	5.0	0.0	70.0	10.0	0.0	3.0
4_kitchen & dining	7.0	7.0	2.0	20.0	32.0	14.0	6.0
5_beauty and personal care	10.0	4.0	2.0	15.0	4.0	66.0	6.0
6_computers	1.0	4.0	5.0	0.0	11.0	15.0	57.0

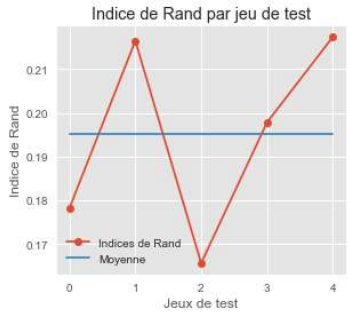
Evaluation

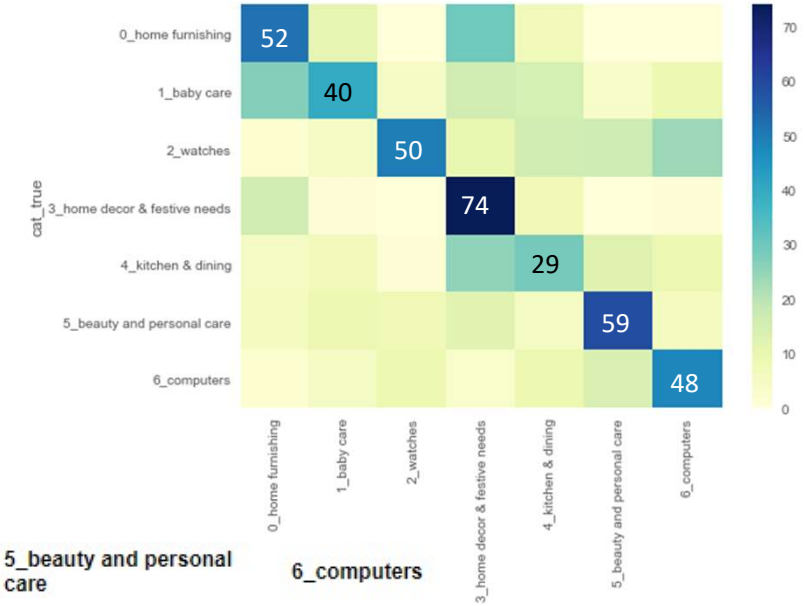
Prétraitements

- Réduction du bruit

```
Score moyen: 0.503
Indice de Rand moyen: 0.195
```





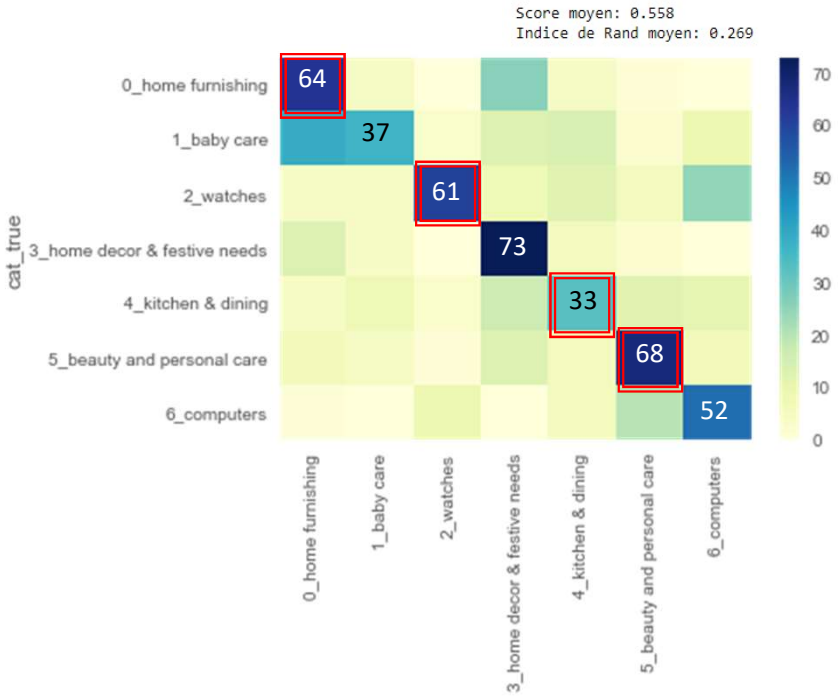


cat_true	0_home furnishing	1_baby care	2_watches	3_home decor & festive needs	4_kitchen & dining	5_beauty and personal care	6_computers
0_home furnishing	52.0	11.0	0.0	30.0	7.0	0.0	0.0
1_baby care	27.0	40.0	5.0	16.0	15.0	4.0	9.0
2_watches	2.0	5.0	50.0	10.0	16.0	17.0	24.0
3_home decor & festive needs	16.0	1.0	0.0	74.0	8.0	0.0	1.0
4_kitchen & dining	5.0	7.0	1.0	26.0	29.0	13.0	9.0
5_beauty and personal care	6.0	9.0	8.0	12.0	5.0	59.0	6.0
6_computers	2.0	5.0	9.0	3.0	9.0	14.0	48.0

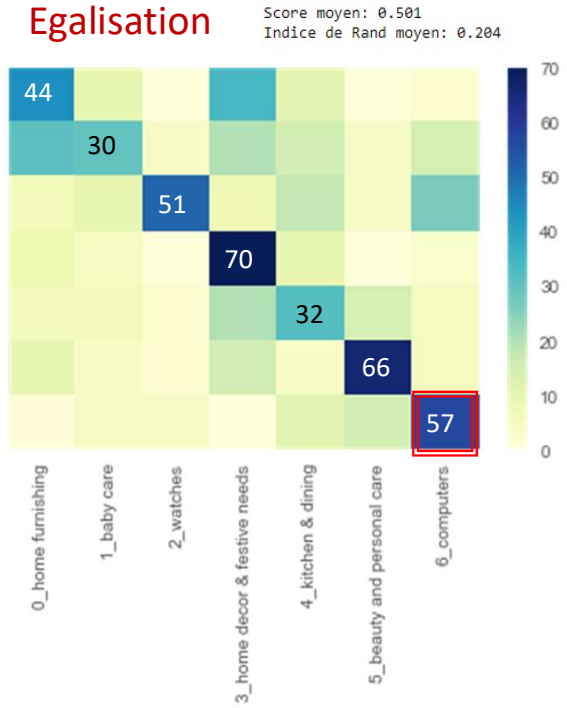
Evaluation

Comparaison suivant les sources d'images

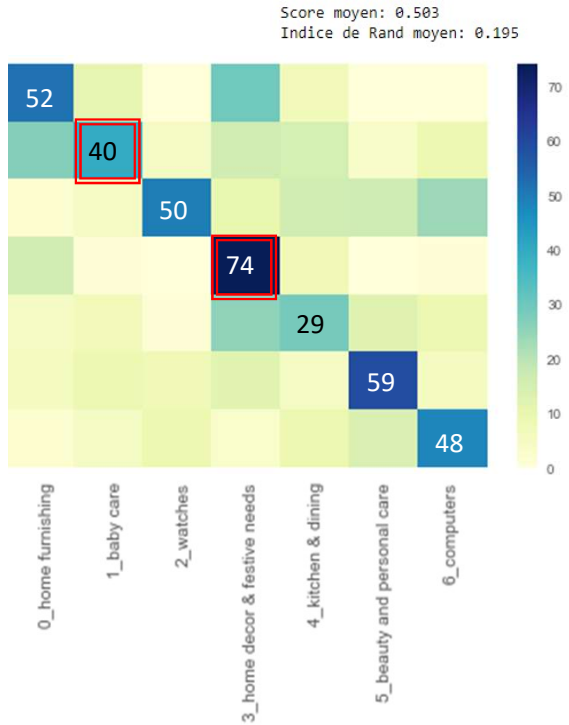
Images originales



Réduction du bruit Egalisation



Réduction du bruit



Plan

Analyse

1. Analyse exploratoire des données

Traitement des données textuelles

2. Préparation du corpus
3. Traitement du corpus
4. Réduction de dimensions
5. Classification

Traitement des images

7. Prétraitement
8. Détermination des variables explicatives
9. Classification
10. Réseaux de neurones convolutifs

- *CNN*
- *Keras VGG16*
- *Transfer Learning*

Définition

Fonction

- Classification par apprentissage supervisé

Spécificité

- Apprentissage automatique des features
  - Extraction et description
  - ➢ Optimisation des paramètres et des features lors de l'entraînement

Architecture

- Empilage multicouches de Perceptron
  - Couche de convolution (repérage des features)
  - Pooling (réduction de la taille des images)
  - Relu (traitement correctif)
  - Fully-connected (classification)

CNN

Processus

- Numérisation des images

```
if version_ima_num_ == 'v0': img_path = '../OCDS_Projet_6/images_ori/'
elif version_ima_num_ == 'v1': img_path = '../OCDS_Projet_6/images_pro_equ_1/'
elif version_ima_num_ == 'v2': img_path = '../OCDS_Projet_6/images_pro_equ_2/'
elif version_ima_num_ == 'v3': img_path = '../OCDS_Projet_6/images_pro_equ_3/'
```

- Jeux d'entraînement et de test

```
X_train (840, 224, 224, 3)
y_train (840,)
X_test (210, 224, 224, 3)
y_test (210,)
```

- Configuration du CNN
  - Empilement des couches
  - Compilation du modèle

```
# Compilation du modèle
model_cnn1.compile(loss='categorical_crossentropy', optimizer='adam', metrics=['accuracy'])
```

- Entraînement

```
# Apprentissage
if model_cnn1_ == 1:
    model_cnn1.fit(X_train, y_train, validation_data=(X_test, y_test), epochs=20, batch_size=200)
    joblib.dump(model_cnn1, 'model_cnn1_'+version_ima_num_+'.joblib')
```

CNN

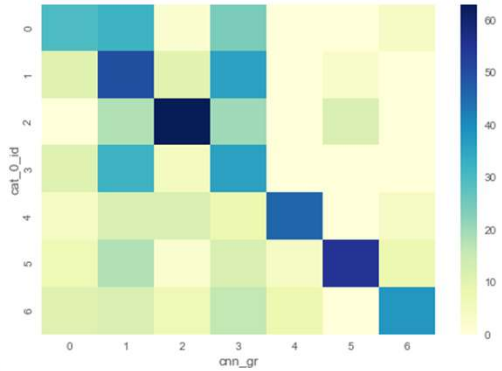
Evaluation

Images originales

Score : 47.14%
Taux d'erreurs : 52.86%

epochs=10,

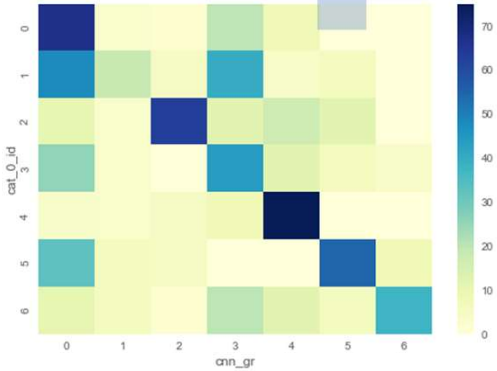
cat_0_id \ cnn_gr	0	1	2	3	4	5	6
0	30.0	32.0	2.0	24.0	0.0	0.0	4.0
1	11.0	50.0	10.0	36.0	0.0	3.0	0.0
2	0.0	18.0	63.0	20.0	0.0	12.0	0.0
3	11.0	32.0	5.0	36.0	0.0	0.0	0.0
4	4.0	12.0	12.0	8.0	46.0	0.0	4.0
5	7.0	18.0	2.0	12.0	4.0	55.0	8.0
6	11.0	12.0	7.0	16.0	8.0	0.0	38.0



Score : 50.95%
Taux d'erreurs : 49.05%

epochs=20,

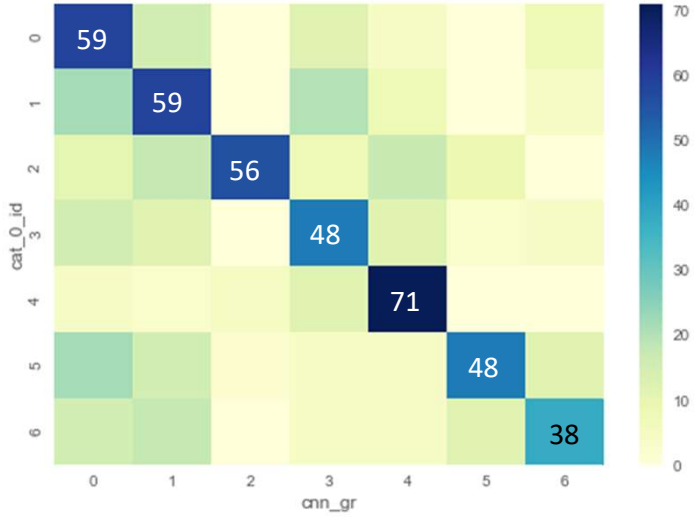
cat_0_id \ cnn_gr	0	1	2	3	4	5	6
0	67.0	3.0	2.0	20.0	8.0	0.0	0.0
1	48.0	18.0	5.0	40.0	4.0	6.0	0.0
2	11.0	3.0	63.0	12.0	17.0	12.0	0.0
3	26.0	3.0	0.0	44.0	12.0	6.0	4.0
4	4.0	3.0	5.0	8.0	75.0	0.0	0.0
5	33.0	6.0	5.0	0.0	0.0	55.0	8.0
6	11.0	6.0	2.0	20.0	12.0	6.0	38.0



Avec réduction du bruit

Score : 54.29%
Taux d'erreurs : 45.71%

cat_0_id \ cnn_gr	0	1	2	3	4	5	6
0	59.0	15.0	0.0	12.0	4.0	0.0	8.0
1	22.0	59.0	0.0	20.0	8.0	0.0	4.0
2	11.0	18.0	56.0	8.0	17.0	9.0	0.0
3	15.0	12.0	0.0	48.0	12.0	3.0	4.0
4	4.0	3.0	5.0	12.0	71.0	0.0	0.0
5	22.0	15.0	2.0	4.0	4.0	48.0	12.0
6	15.0	18.0	0.0	4.0	4.0	12.0	38.0

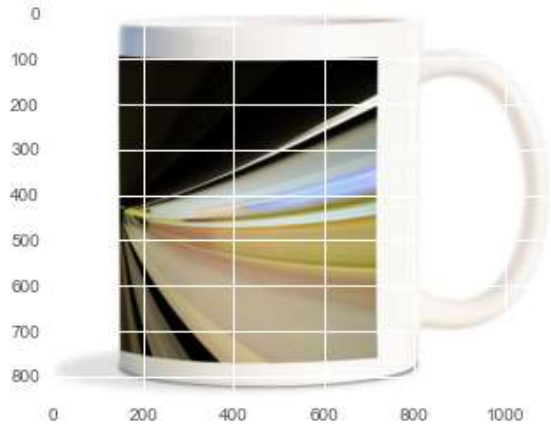


Keras VGG16

Prédictions

```
from keras.applications.vgg16 import decode_predictions

# Afficher les 3 classes les plus probables
print('Top 3 :', decode_predictions(y, top=3)[0])

img = Image.open('../OCDS_Projet_6/images_ori/'+ima_id+'.jpg')
plt.imshow(img)
plt.show()
```

```
Top 3 : [('n03063599', 'coffee_mug', 0.59327143), ('n07930864', 'cup', 0.34781334), ('n03916031', 'perfume', 0.006643412)]
```



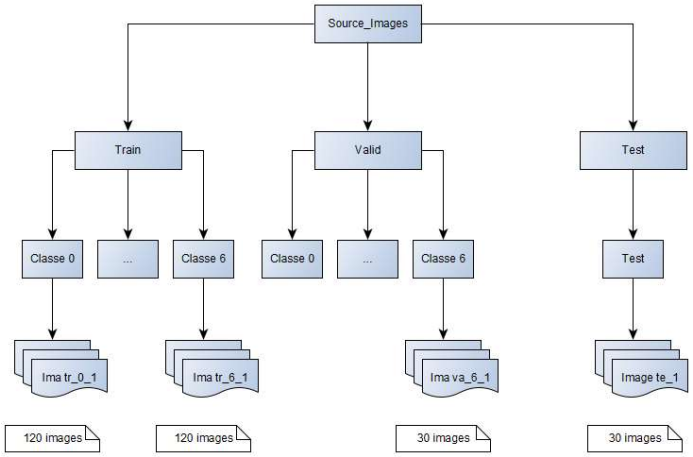
Transfer Learning

Architecture

- Réseau de neurones pré-entrainé
  - MobilNet
- Remplacement des dernières couches
  - 7 classes

Création d'une structure de répertoires

- Niveau 1: Train / Valid / Test
- Niveau 2:
  - Train / Valid: 7 sous-répertoires (1 par catégorie)
  - Test : 1 sous-répertoire




```
Nombre d'images par sous-répertoire de catégorie pour le jeu de test

0_home furnishing : 30 files
1_baby care : 30 files
2_watches : 30 files
3_home decor & festive needs : 30 files
4_kitchen & dining : 30 files
5_beauty and personal care : 30 files
6_computers : 30 files

Nombre d'images par sous-répertoire de catégorie pour le jeu de validation

0_home furnishing Number of files: 30 files
1_baby care Number of files: 30 files
2_watches Number of files: 30 files
3_home decor & festive needs Number of files: 30 files
4_kitchen & dining Number of files: 30 files
5_beauty and personal care Number of files: 30 files
6_computers Number of files: 30 files

Nombre d'images par sous-répertoire de catégorie pour le jeu d'entraînement

0_home furnishing : 120 files
1_baby care : 120 files
2_watches : 120 files
3_home decor & festive needs : 120 files
4_kitchen & dining : 120 files
5_beauty and personal care : 120 files
6_computers : 120 files
```

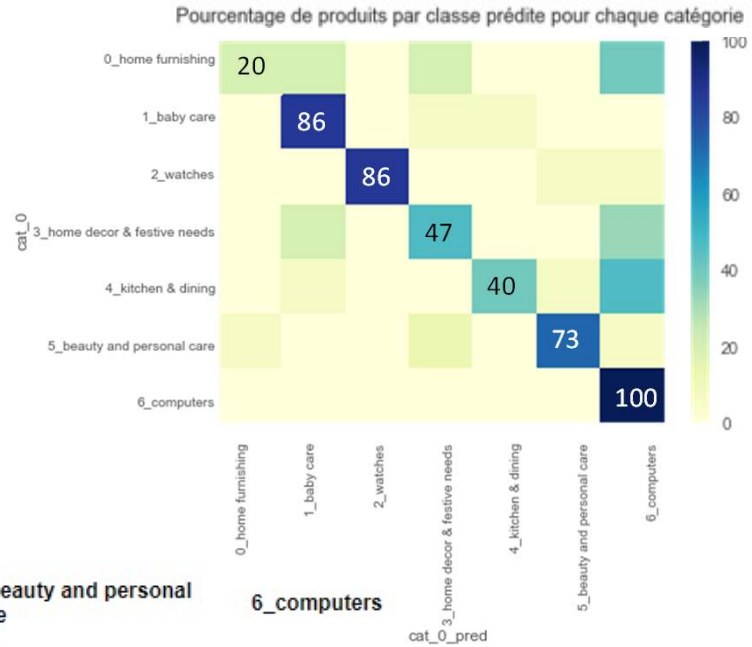
Transfer Learning

Evaluation

- Prétraitements
  - Images originales
- Entraînement
  - Dernières couches CNN

```
model.evaluate_generator(generator=valid_generator)
```

```
[2.604249954223633, 0.6285714507102966]
```



Pourcentage de produits par classe prédite pour chaque catégorie

cat_0_pred cat_0	0_home furnishing	1_baby care	2_watches	3_home decor & festive needs	4_kitchen & dining	5_beauty and personal care	6_computers
0_home furnishing	20.0	20.0	0.0	20.0	0.0	0.0	40.0
1_baby care	0.0	86.0	0.0	7.0	7.0	0.0	0.0
2_watches	0.0	0.0	86.0	0.0	0.0	7.0	7.0
3_home decor & festive needs	0.0	20.0	0.0	47.0	0.0	0.0	33.0
4_kitchen & dining	0.0	7.0	0.0	0.0	40.0	7.0	47.0
5_beauty and personal care	7.0	0.0	0.0	13.0	0.0	73.0	7.0
6_computers	0.0	0.0	0.0	0.0	0.0	0.0	100.0

```
model_layers_trainable_ = 2

if model_layers_trainable_ == 0:
    for layer in model.layers:
        layer.trainable=False

elif model_layers_trainable_ == 1:
    for layer in model.layers:
        layer.trainable=True

# 20 premières couches non entrainées
elif model_layers_trainable_ == 2:
    for layer in model.layers[:20]:
        layer.trainable=False
    for layer in model.layers[20:]:
        layer.trainable=True
```

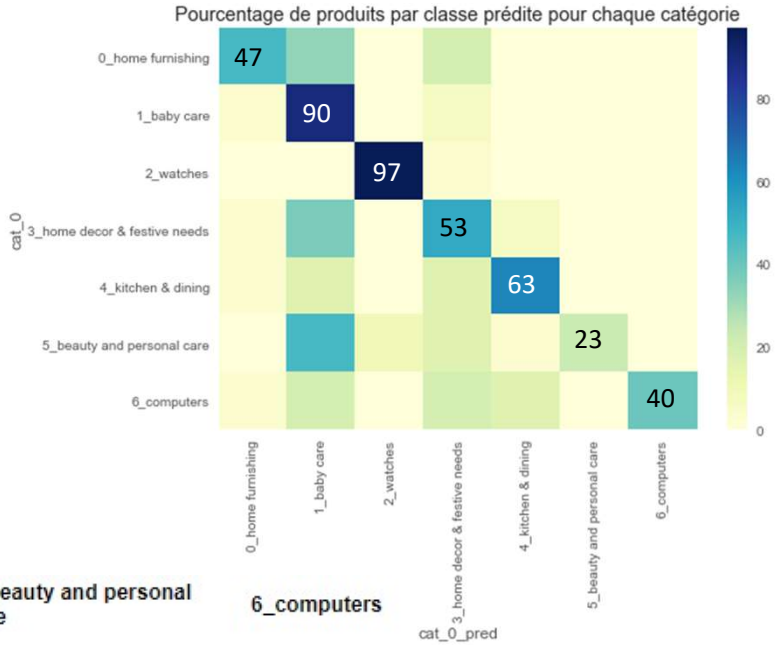
Transfer Learning

Evaluation

- Prétraitements
  - Images originales
- Entraînement
  - Toutes les couches CNN

```
model.evaluate_generator(generator=valid_generator)
```

```
[0.40858227014541626, 0.8095238208770752]
```



Pourcentage de produits par classe prédite pour chaque catégorie

cat_0_pred cat_0	0_home furnishing	1_baby care	2_watches	3_home decor & festive needs	4_kitchen & dining	5_beauty and personal care	6_computers
0_home furnishing	47.0	33.0	0.0	20.0	0.0	0.0	0.0
1_baby care	3.0	90.0	0.0	7.0	0.0	0.0	0.0
2_watches	0.0	0.0	97.0	3.0	0.0	0.0	0.0
3_home decor & festive needs	3.0	37.0	0.0	53.0	7.0	0.0	0.0
4_kitchen & dining	3.0	17.0	0.0	17.0	63.0	0.0	0.0
5_beauty and personal care	0.0	47.0	10.0	17.0	3.0	23.0	0.0
6_computers	3.0	20.0	0.0	20.0	17.0	0.0	40.0

```
model_layers_trainable_ = 1

if model_layers_trainable_ == 0:
    for layer in model.layers:
        layer.trainable=False

elif model_layers_trainable_ == 1:
    for layer in model.layers:
        layer.trainable=True

# 20 premières couches non entrainées
elif model_layers_trainable_ == 2:
    for layer in model.layers[:20]:
        layer.trainable=False
    for layer in model.layers[20:]:
        layer.trainable=True
```

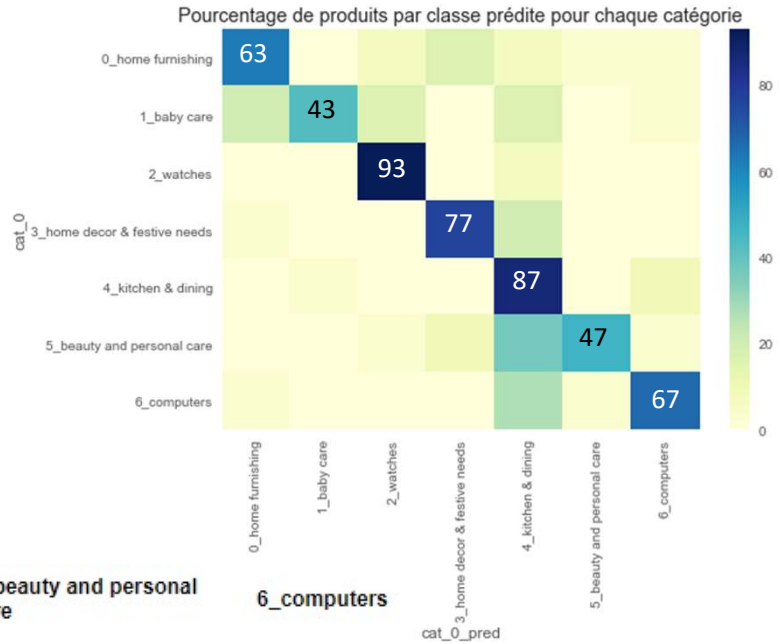
Transfer Learning

Evaluation

- Prétraitements
  - Réduction du bruit
- Entraînement
  - Toutes les couches CNN

```
model.evaluate_generator(generator=valid_generator)
```

```
[0.7438859343528748, 0.8666666746139526]
```

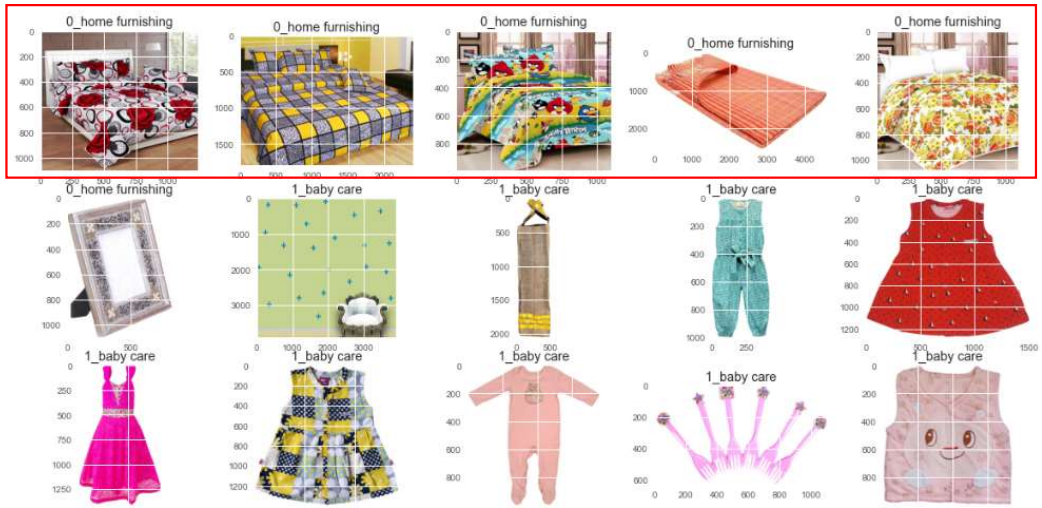


Pourcentage de produits par classe prédite pour chaque catégorie

cat_0_pred cat_0	0_home furnishing	1_baby care	2_watches	3_home decor & festive needs	4_kitchen & dining	5_beauty and personal care	6_computers
0_home furnishing	63.0	0.0	7.0	17.0	7.0	3.0	3.0
1_baby care	20.0	43.0	17.0	0.0	17.0	0.0	3.0
2_watches	0.0	0.0	93.0	0.0	7.0	0.0	0.0
3_home decor & festive needs	3.0	0.0	0.0	77.0	20.0	0.0	0.0
4_kitchen & dining	0.0	3.0	0.0	0.0	87.0	0.0	10.0
5_beauty and personal care	0.0	0.0	3.0	10.0	37.0	47.0	3.0
6_computers	3.0	0.0	0.0	0.0	27.0	3.0	67.0

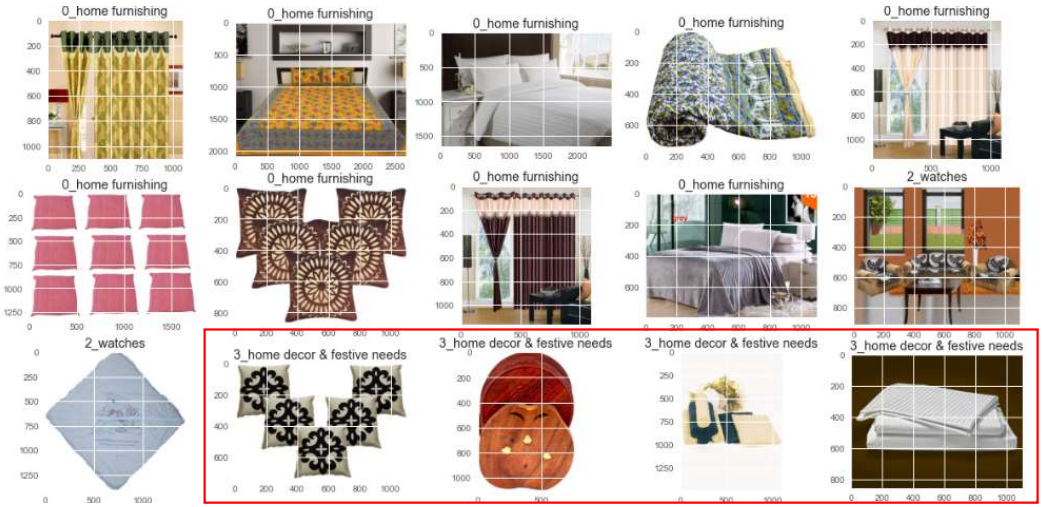
Transfer Learning

Prédictions



Catégorie baby care

Catégorie home furnishing



Catégorie home decor & festive needs

Plan

Analyse

1. Analyse exploratoire des données

Traitement des données textuelles

2. Préparation du corpus
3. Traitement du corpus
4. Réduction de dimensions
5. Classification

Traitement des images

7. Prétraitement
8. Détermination des variables explicatives
9. Classification
10. Réseaux de neurones convolutifs

- *CNN*
- *Keras VGG16*
- *Transfer Learning*

Optimisation

11. Données complémentaires

- *Amazon Product Advertising API*

Conclusion

12. Synthèse
13. Optimisation
14. Prochaines étapes

Questions

Amazon Product Advertising API

Interface d'accès au catalogue de produits

Amazon's Product Advertising API, formerly **Amazon Associates Web Service** (**A2S**) and before that known as **Amazon E-Commerce Service** (**ECS**), is a web service and application programming interface (API) that gives application programmers access to Amazon's product catalog data. Accessible via either the SOAP or REST protocols it enables products to be listed and/or sold through third-party websites and applications. It is a product of Amazon Services, not to be confused with Amazon Web Services.

Source: https://en.wikipedia.org/wiki/Amazon_Product_Advertising_API

- Documentation
  - https://webservices.amazon.com/paapi5/documentation/

Product Advertising API 5.0 Documentation

- SDK

Python

Latest Stable 1.0.0 | python 2.7 | python 3.4+

1. Download **paapi5-python-sdk-and-samples** archive and unzip it.

Amazon Product Advertising API

Recherche de produits

- Fonction search_items()

```python
from paapi5_python_sdk.api.default_api import DefaultApi
from paapi5_python_sdk.models.partner_type import PartnerType
from paapi5_python_sdk.models.search_items_request import SearchItemsRequest
from paapi5_python_sdk.models.search_items_resource import SearchItemsResource
from paapi5_python_sdk.rest import ApiException

def search_items():

    access_key = "<YOUR ACCESS KEY>"
    secret_key = "<YOUR SECRET KEY>"
    partner_tag = "<YOUR PARTNER TAG>"

    # Hôte et région de destination de la requête
    host = "webservices.amazon.com"
    region = "us-east-1"

    # Déclaration API
    default_api = DefaultApi(access_key=access_key, secret_key=secret_key, host=host, region=region)

    # Initialisation de la requête

    # Mots clés
    keywords = None

    # Categorie dans laquelle faire la requête
    # https://webservices.amazon.com/paapi5/documentation/use-cases/organization-of-items-on-amazon/search-index.html
    search_index = "Beauty"
    # Nous avons choisi une catégorie parmi la liste des références locales us
    # https://webservices.amazon.com/paapi5/documentation/locale-reference/united-states.html

    # Nombre d'items à retourner
    item_count = 1000
```

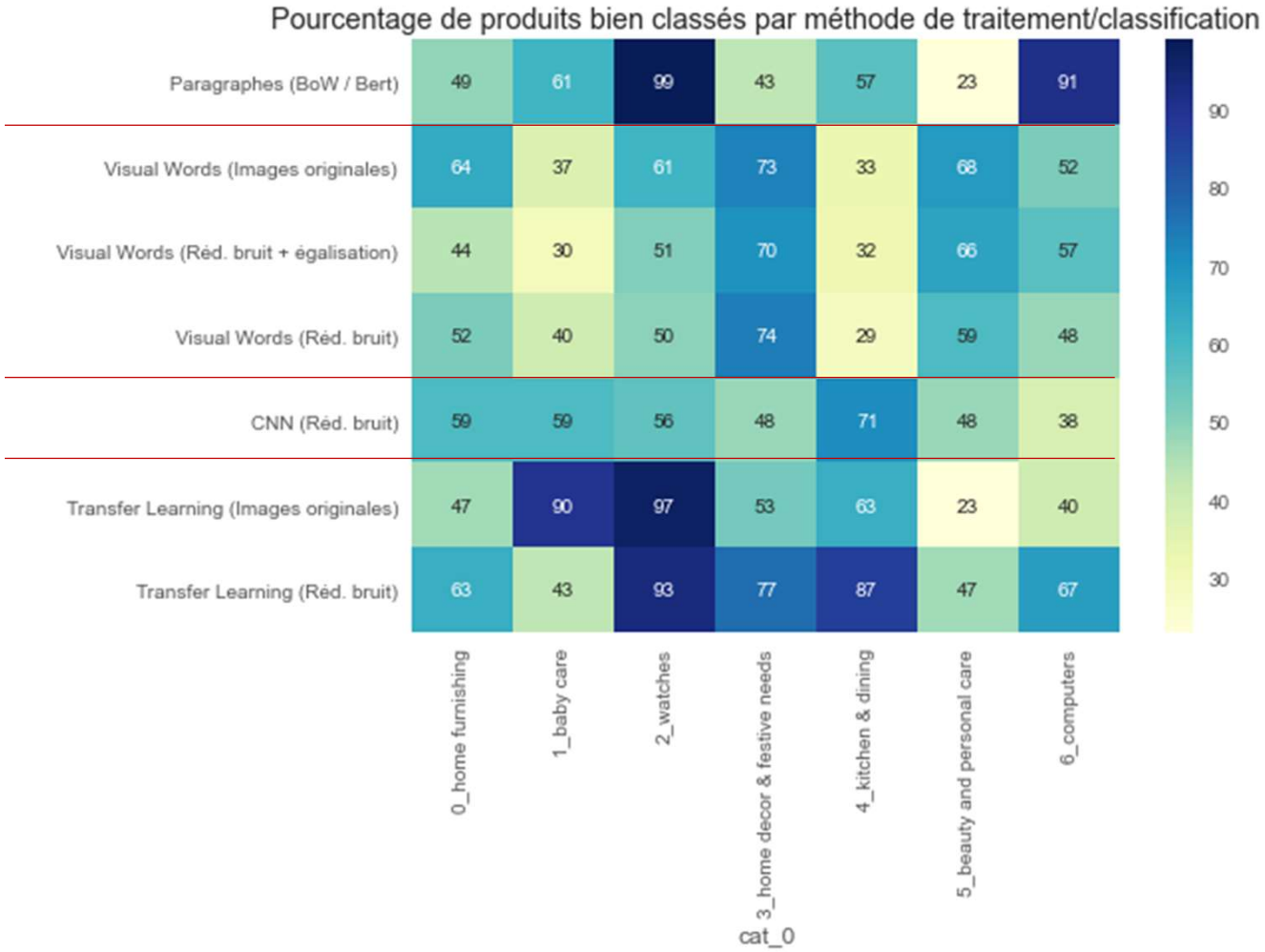
Amazon Product Advertising API

Recherche de produits

- Fonction search_items()

```
# Choix des 'ressources' c'est-à-dire de la nature des données
# https://webservices.amazon.com/paapi5/documentation/search-items.html#resources-parameter
search_items_resource = [SearchItemsResource.ITEMINFO_TITLE, # Nom du produit
                         SearchItemsResource.ITEMINFO_FEATURES, # Description
                         SearchItemsResource.IMAGES_PRIMARY_LARGE, # Image
                         SearchItemsResource.OFFERS_LISTINGS_PRICE] # Prix

# Exécution de la requête
try:
    search_items_request = SearchItemsRequest(partner_tag=partner_tag,
                                              partner_type=PartnerType.ASSOCIATES,
                                              keywords=keywords,
                                              search_index=search_index,
                                              item_count=item_count,
                                              resources=search_items_resource)

except ValueError as exception:
    print("Error in forming SearchItemsRequest: ", exception)
    return

try:
    """ Sending request """
    response = default_api.search_items(search_items_request)

    print("API called Successfully")
    print("Complete Response:", response)

    """ Parse response """
```

12 Synthèse



Pourcentages de classement moyens par processus (méthodes)

Paragraphes (BoW / Bert)	60.0
Visual Words (Images originales)	55.0
Visual Words (Réd. bruit + égalisation)	50.0
Visual Words (Réd. bruit)	50.0
CNN (Réd. bruit)	54.0
Transfer Learning (Images originales)	59.0
Transfer Learning (Réd. bruit)	68.0

Pourcentages de classement moyen par catégorie

cat_0	
0_home furnishing	54.0
1_baby care	51.0
2_watches	72.0
3_home decor & festive needs	63.0
4_kitchen & dining	53.0
5_beauty and personal care	48.0
6_computers	56.0

13 Optimisation

Prétraitements

Données et variables explicatives

- Minimiser les valeurs nulles
- Niveaux de catégories
- Augmentation de l'échantillon d'images (Amazon AP API)
- Combiner les variables (Textes + Images)

Méthodes et modèles

- Nouvelles méthodes (n-gram embedding)
- Réduction de dimensions
- Algorithmes de classification
  - Optimisation des paramètres, configuration CNN (couches)…
  - Tester d'autres algorithmes
  - Utiliser VGG16
    - Associer les catégories VGG16 aux produits
    - Etablir la correspondance avec les catégories Flipkart
    - ➢ Prédiction: faire le lien entre les catégories VGG16 pour identifier les catégories Flipkart

14 Recommandations et prochaines étapes

Sources de données

- Obtenir plus d'observations (Amazon AP API…)
- Obtenir plus de variables (minimiser les valeurs manquantes)

Plan de classement

- Revoir les structures de catégories

Prétraitements

- Tests spécifiques pour identifier les méthodes optimales

Méthodes, modèles et technologies

- Tests et optimisations
- Recommandations
  - Textes: Bag of Words - Bert Embedding
  - Images: Transfer Learning

Moteur de classification

- Conception

14 Recommandations et prochaines étapes

Je me pose des questions…

Sources de données

- Obtenir plus d'observations (Amazon AP API…)
- Obtenir plus de variables (minimiser les valeurs manquantes)

Plan de classement

- Revoir les structures de catégories

Prétraitements

- Tests spécifiques pour identifier les méthodes optimales

Méthodes, modèles et technologies

- Tests et optimisations
- Recommandations
  - Textes: Bag of Words - Bert Embedding
  - Images: Transfer Learning

Moteur de classification

- Conception