



Parcours OpenClassRooms

Data Scientist

P6 Classifiez des biens de consommations

Développé sur un Notebook Jupyter Colaboratory



Pictures used for educational purpose only

Sommaire

I. Introduction

II. Traitement des images

III. Traitement des textes

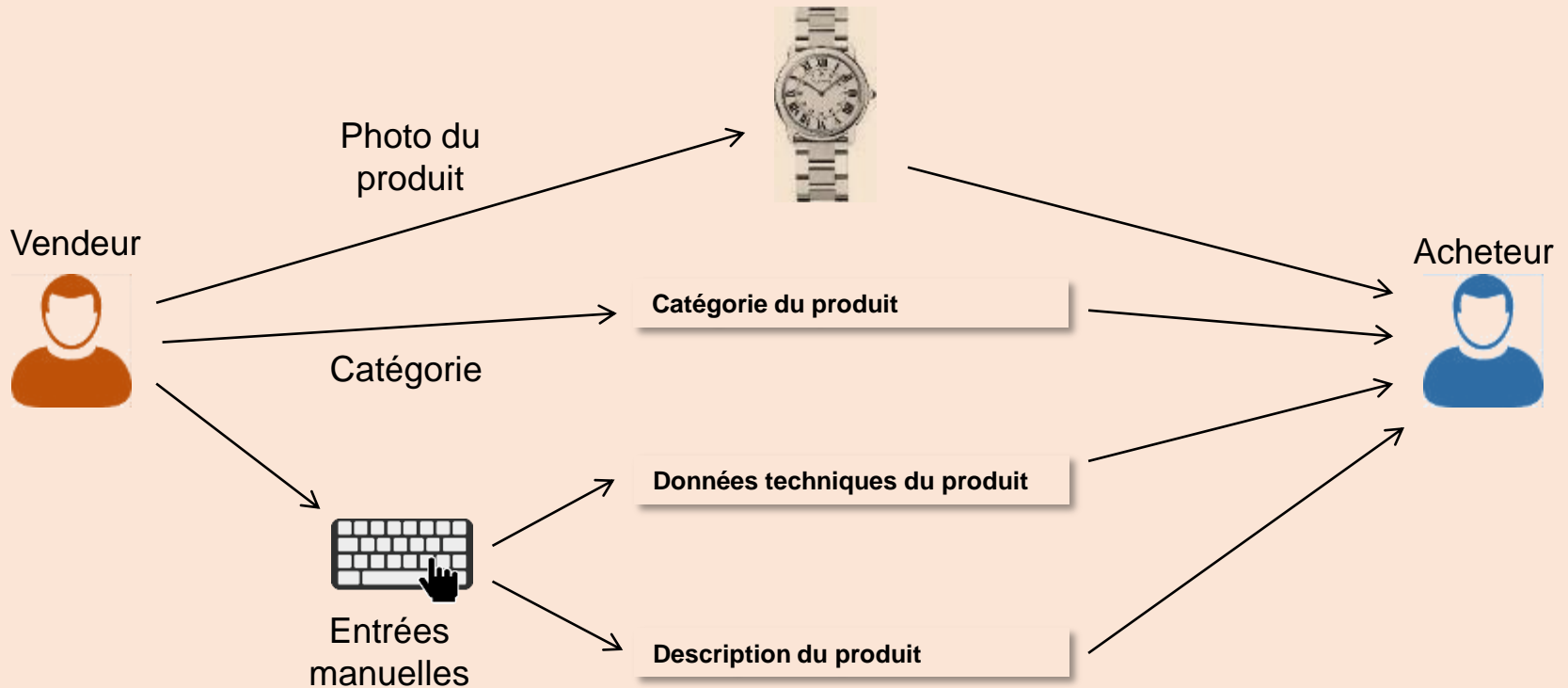
IV. Segmentation

V. Bilan et perspectives

I. Introduction

1. L'entreprise et son besoin

Activité de l'entreprise : une place de marché



Problématique

Difficulté à garantir la **fiabilité des entrées manuelles**.



Projet

Etudier la faisabilité d'une **classification** des produits, avec un niveau de précision suffisant.

I. Introduction

2. Le livrable recherché

Nouveau produit



product_name
Carrier W6701005 Analog Watch -
For Boys, Men
description
Carrier W6701005 Analog Watch -
For Boys, Men - Buy Carrier
W6701005 Analog Watch - For
...
crawl_timestamp
2015-12-04 07:29:36 +0000
retail_price
201000.0

Algorithme de
classification

« Avec un niveau de précision suffisant » :
→ fixer une **limite** au nombre d'**erreurs**
d'affectation

Groupe 1



Groupe 2



Groupe 3



Groupe 4



Groupe 5



I. Introduction

3. Le jeu de données

15 caractéristiques

La table principale

Dates, prix, descriptions, identifiants, marques, ...

	crawl_timestamp	product_name	product_category_tree	retail_price	discounted_price	description	brand
0	2016-04-30 03:22:56 +0000	Elegance Polyester Multicolor Abstract Eyelet ...	[*Home Furnishing >> Curtains & Accessories >>...	1899.0	899.0	Key Features of Elegance Polyester Multicolor ...	Elegance

La banque d'image associée



I. Introduction

4. Les différents types de données (features)

Données textuelles

product_name

description

product_category_tree
["Watches >> Wrist Watches >>
Cartier Wrist Watches"]

product_specifications
{ "product_specification" => { "key" =
> "Chronograph", "value" => "No" },
{ "key" => "Altimeter",
"value" => "No" },

Données numériques

retail_price

Dates

crawl_timestamp

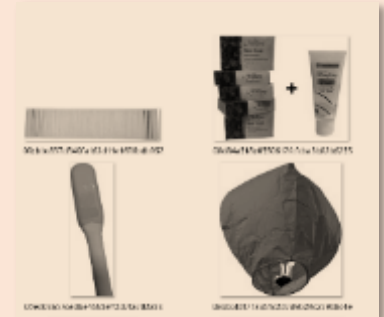
Données booléennes

is_FK_Advantage_product

Identifiants

uniq_id

Images



Certaines **données textuelles** sont « à tiroir », elles peuvent contenir :

- Une **arborescence**, avec des **embranchements**,
- Un **dictionnaire**, avec des **clés** propres à chaque produit (≈ .json).

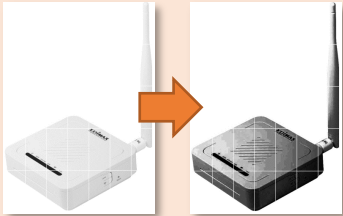
Après un traitement approprié, ces sous-éléments vont fournir de **nouvelles caractéristiques**. 6

I. Introduction

5. Les outils à disposition

Traitement d'image

Égalisation



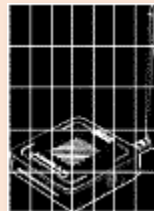
Filtre gaussien



Filtre de Canny



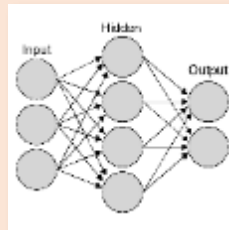
Gradient



Descripteurs



Réseaux de neurones



Traitement de texte

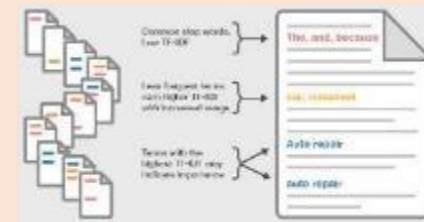
OneHotEncoding

Color	Red	Yellow	Green
Red	1	0	0
Red	1	0	0
Yellow	0	1	0
Green	0	0	1
Yellow	0	1	0

Specific encoding

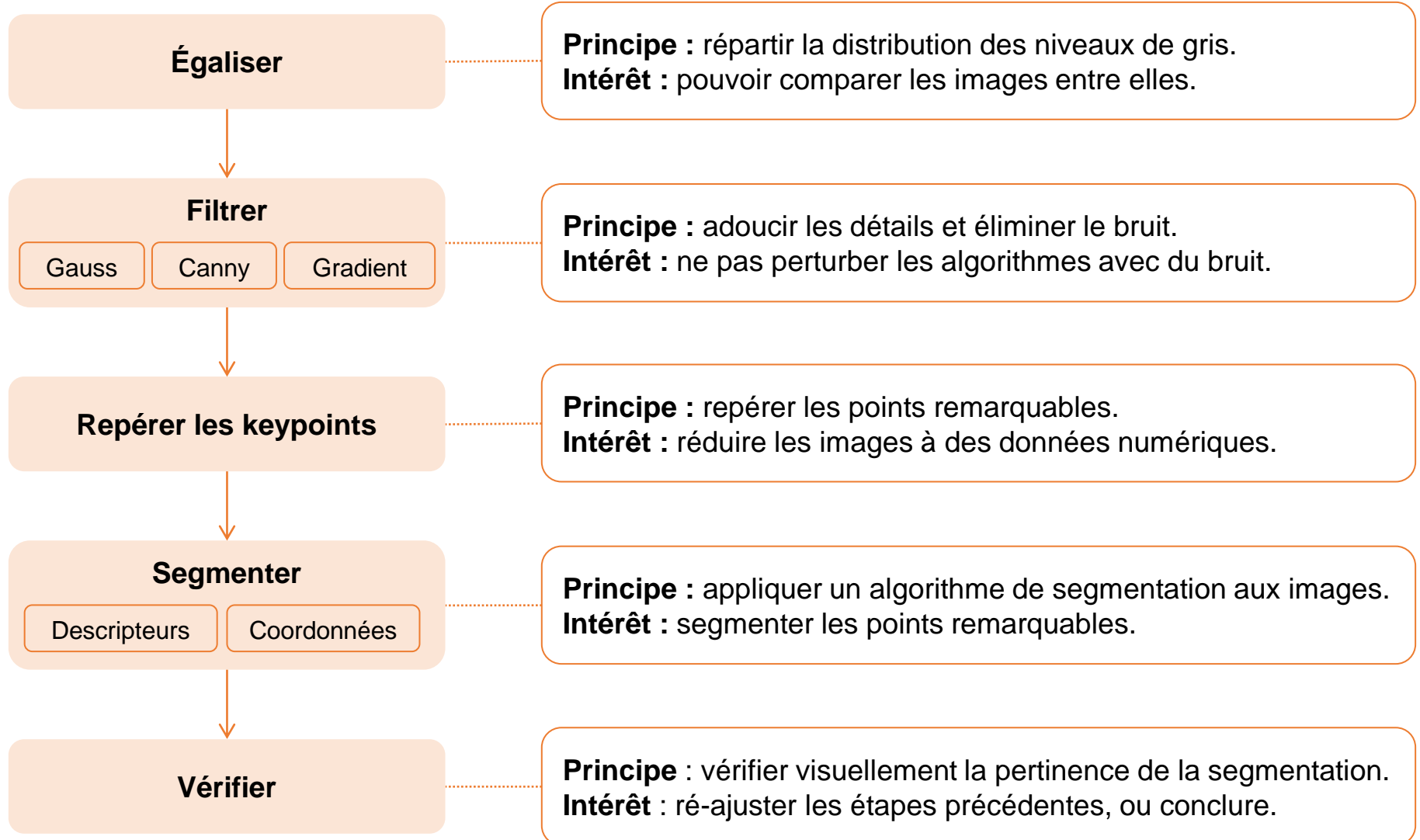
Color	Red	Yellow
Red	1	0
Red	1	0
Yellow	0	1
Green	0	0
Yellow	0	1

Poids tf-idf



II. Traitement des images

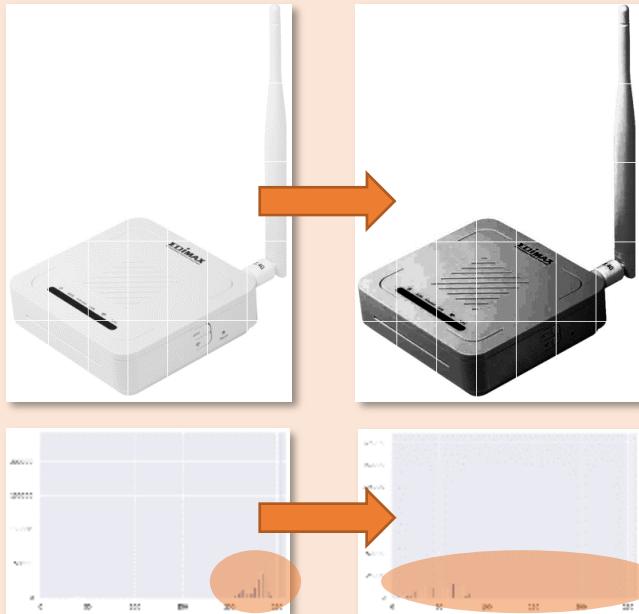
1. Les étapes de traitement



II. Traitement des images

2. Les pré-traitements

1. Egaliser



→ Sur plusieurs images, on compare visuellement les résultats en faisant varier **filtre gaussien** et **filtre de Canny**.

2. Filtrer

Gauss

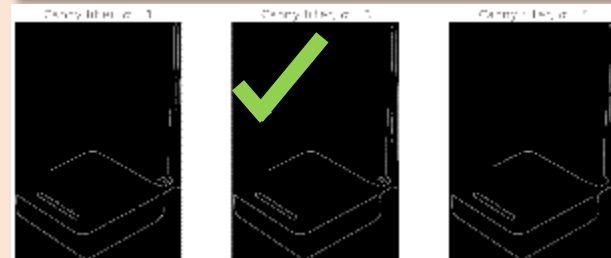
Canny

Gradient

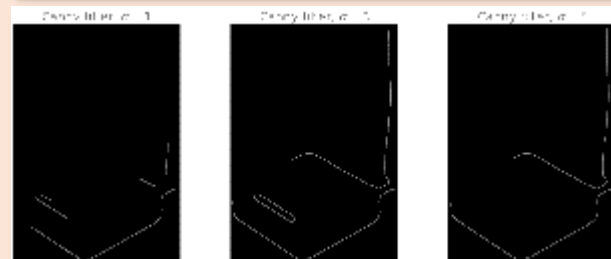
2



5



10

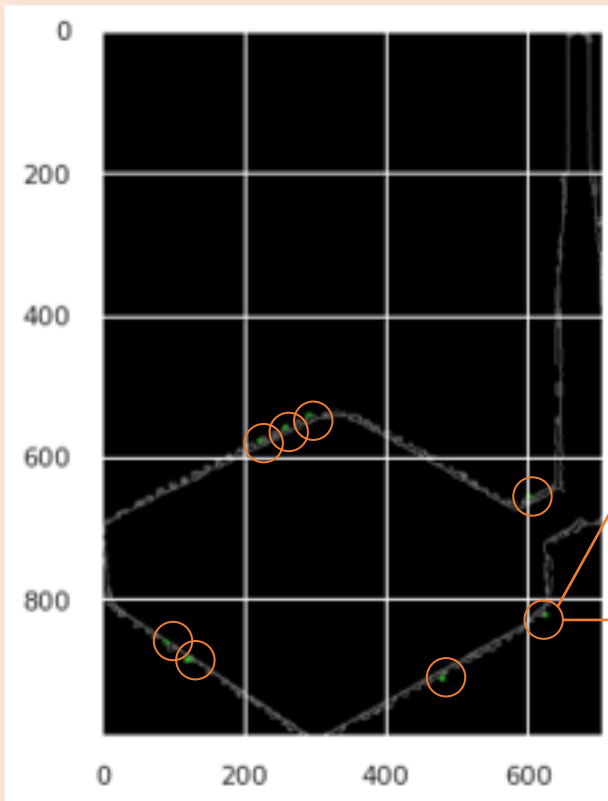


→ Un filtre gaussien de 5 et un filtre de Canny de 3 seront utilisés pour l'ensemble du jeu de données.

II. Traitement des images

3. Les keypoints

Points remarquables repérés par ORB



Le nombre de points est modifiable.

Les coordonnées du point remarquable

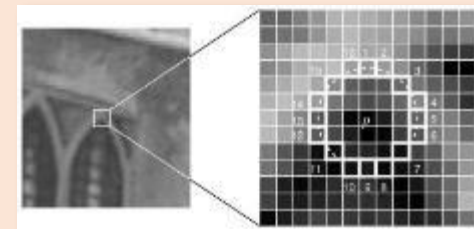
```
array([[318.9032 , 591.22504]], dtype=float32)
```

Abcisse du
point remarquable

Ordonnée du
point remarquable

Le descripteur du point remarquable

```
array([[ 6, 50, 143, 130, 70, 115, 184, 72, 174, 245, 147, 255, 76,  
238, 127, 10, 190, 133, 238, 183, 149, 243, 28, 192, 24, 20,  
203, 189, 205, 39, 45, 212]], dtype=uint8)
```



Source

<https://medium.com/data-breach/introduction-to-orb-oriented-fast-and-rotated-brief-4220e8ec40cf>

→ Mieux vaut-il utiliser les coordonnées ou le descripteur ?

II. Traitement des images

4. La segmentation

→ Environ 22 groupes identifiés visuellement.

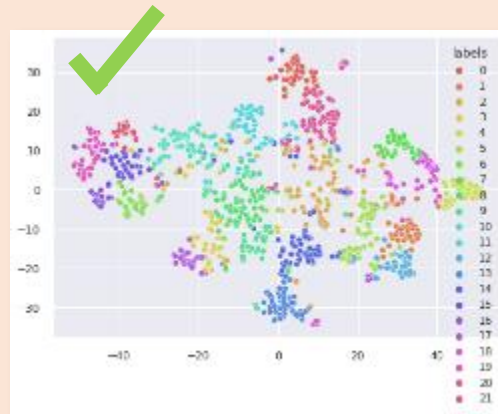
Segmenter les points remarquables via k-means

Représenter dans le plan t-SNE

Compter le nombre d'étiquettes par images

Avec les coordonnées

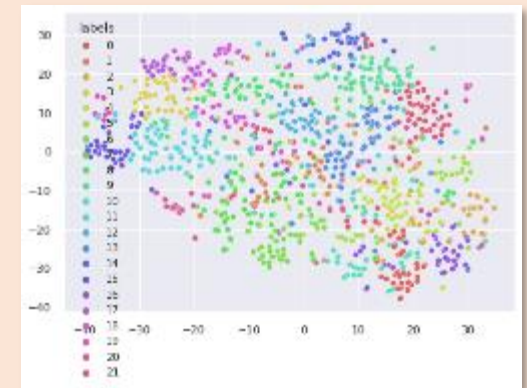
uniq_id	descriptor_id	keypoints_x	keypoints_y	labels
280	1.0	310.00000	519.00000	14
280	2.0	519.60004	304.80002	13
280	3.0	302.40002	613.44000	14
280	4.0	172.80002	559.87207	14
86	1.0	889.00000	182.00000	5
86	2.0	172.80000	687.60004	14
86	3.0	732.96002	694.08002	15
86	4.0	412.99203	765.50409	21



	0	1	2	3	4	...	20	21
index								
820	2.0	0.0	0.0	8.0	0.0	...	0.0	0.0
1003	0.0	0.0	0.0	8.0	0.0	...	0.0	0.0
162	0.0	0.0	0.0	7.0	0.0	...	1.0	0.0
200	1.0	0.0	0.0	5.0	0.0	...	0.0	0.0
137	0.0	0.0	0.0	4.0	0.0	...	2.0	0.0

Avec le descripteur

uniq_id	descriptor_id	0	1	2	3	4	...	labels
280	1.0	165	252	156	97	56	...	17
280	2.0	88	32	19	32	1	...	11
280	3.0	43	173	2	186	89	...	20
280	4.0	96	48	40	104	40	...	19
86	1.0	64	50	128	101	96	...	10
86	2.0	48	32	26	48	9	...	11
86	3.0	65	20	20	69	48	...	13
86	4.0	36	96	177	64	128	...	6



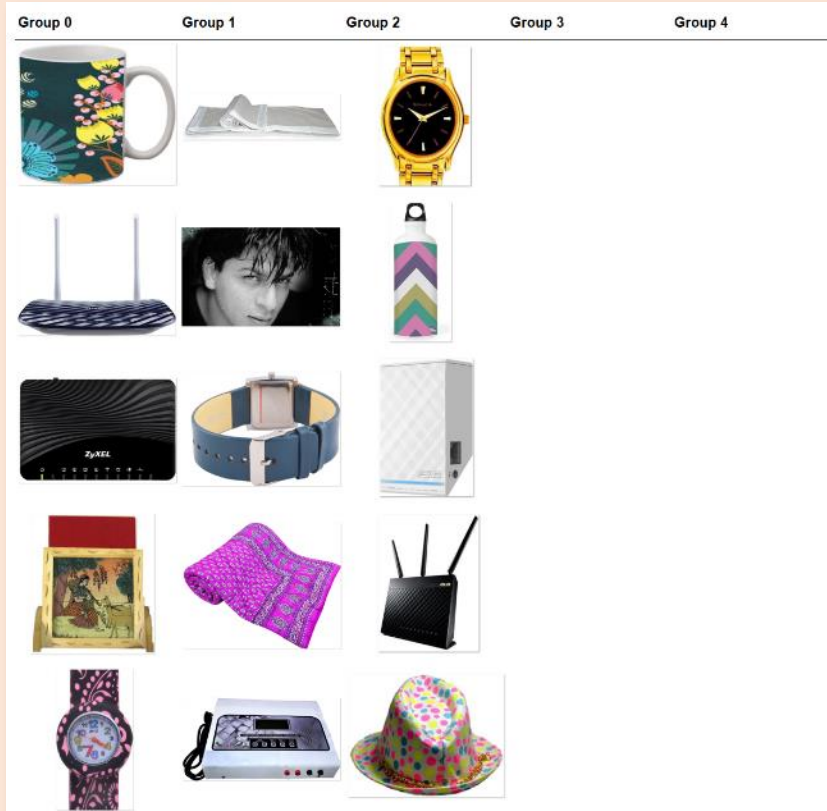
	0	1	2	3	4	...	20	21
index								
78	0.0	0.0	0.0	3.0	0.0	...	0.0	1.0
471	1.0	1.0	0.0	2.0	0.0	...	0.0	3.0
244	0.0	0.0	1.0	1.0	1.0	...	0.0	0.0
955	2.0	0.0	0.0	1.0	0.0	...	2.0	1.0
137	0.0	1.0	0.0	1.0	0.0	...	0.0	0.0

→ t-SNE distingue mieux les points remarquables via leurs **coordonnées**.

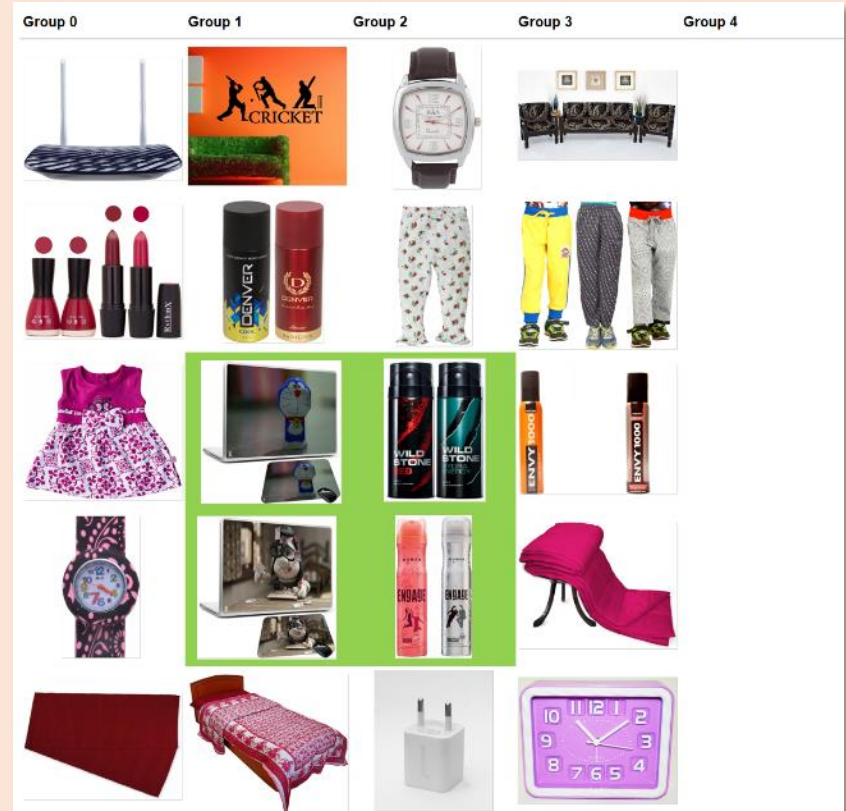
II. Traitement des images

5. Aperçu des groupes formés

1. Coordonnées



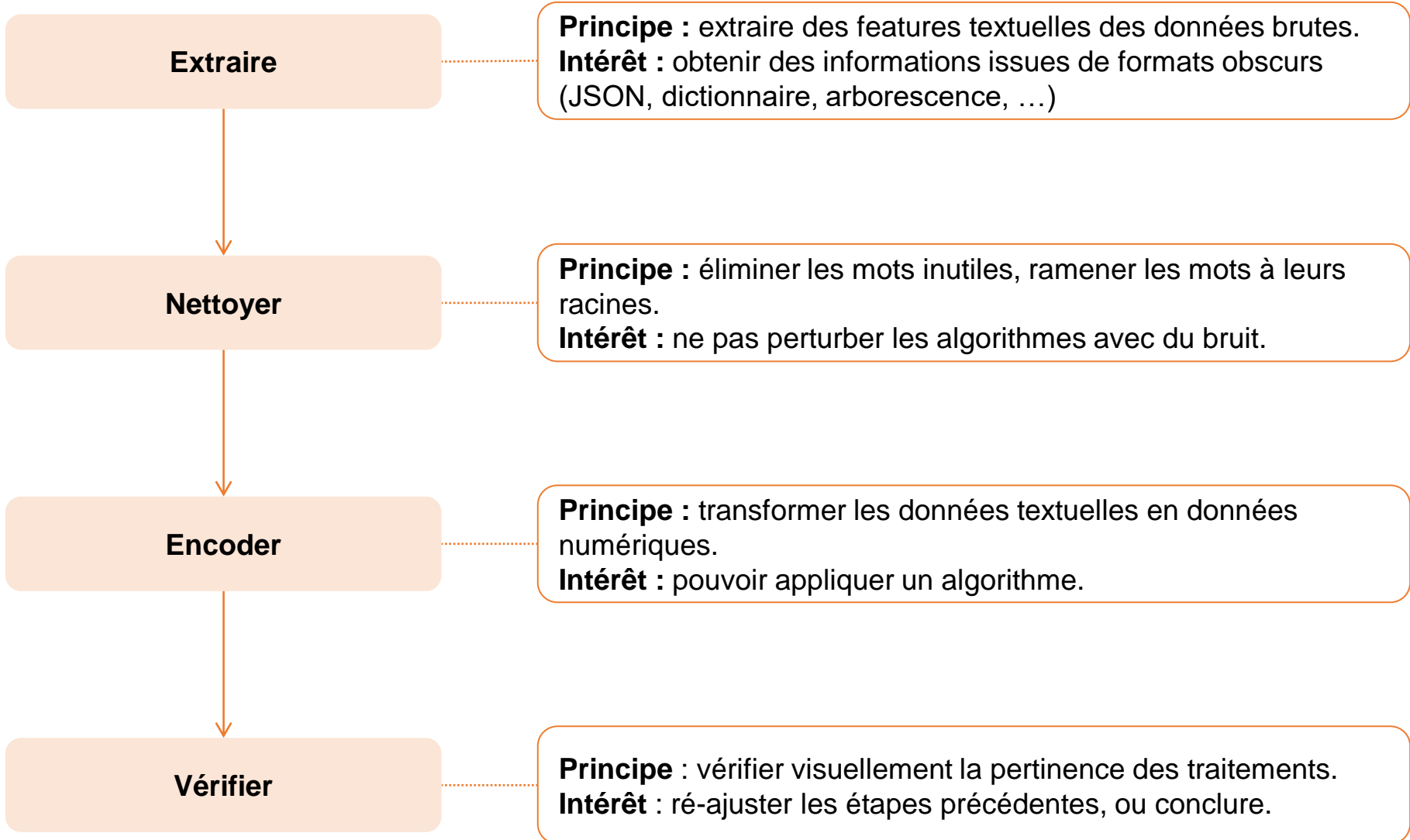
2. Descripteur



→ Groupes peu cohérents

III. Traitement des textes

1. Les étapes de traitement



III. Traitement des textes

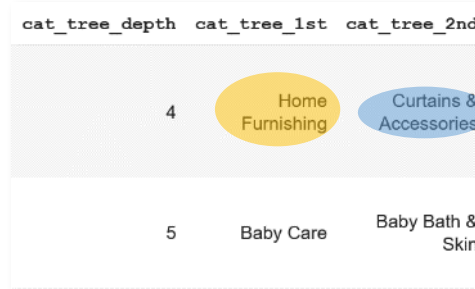
2. Extraire

→ Arborescence

product_category_tree

```
[{"Home Furnishing >> Curtains & Accessories >> Curtains >> ...
```

2 premières branches



→ Dictionnaire (≈ json)

product_specifications

```
{"product_specification"=>[{"key"=>"Brand", "value"=>"Elegance"}, ...  
{"key"=>"Type", "value"=>"Eyelet"},  
...
```

> 2% du jeu de données



description

```
Cartier W6701005 Analog Watch -  
For Boys, Men - Buy Cartier  
W6701005 Analog Watch - For  
...
```

product_name

```
Cartier W6701005 Analog Watch - For  
Boys, Men
```

Brand

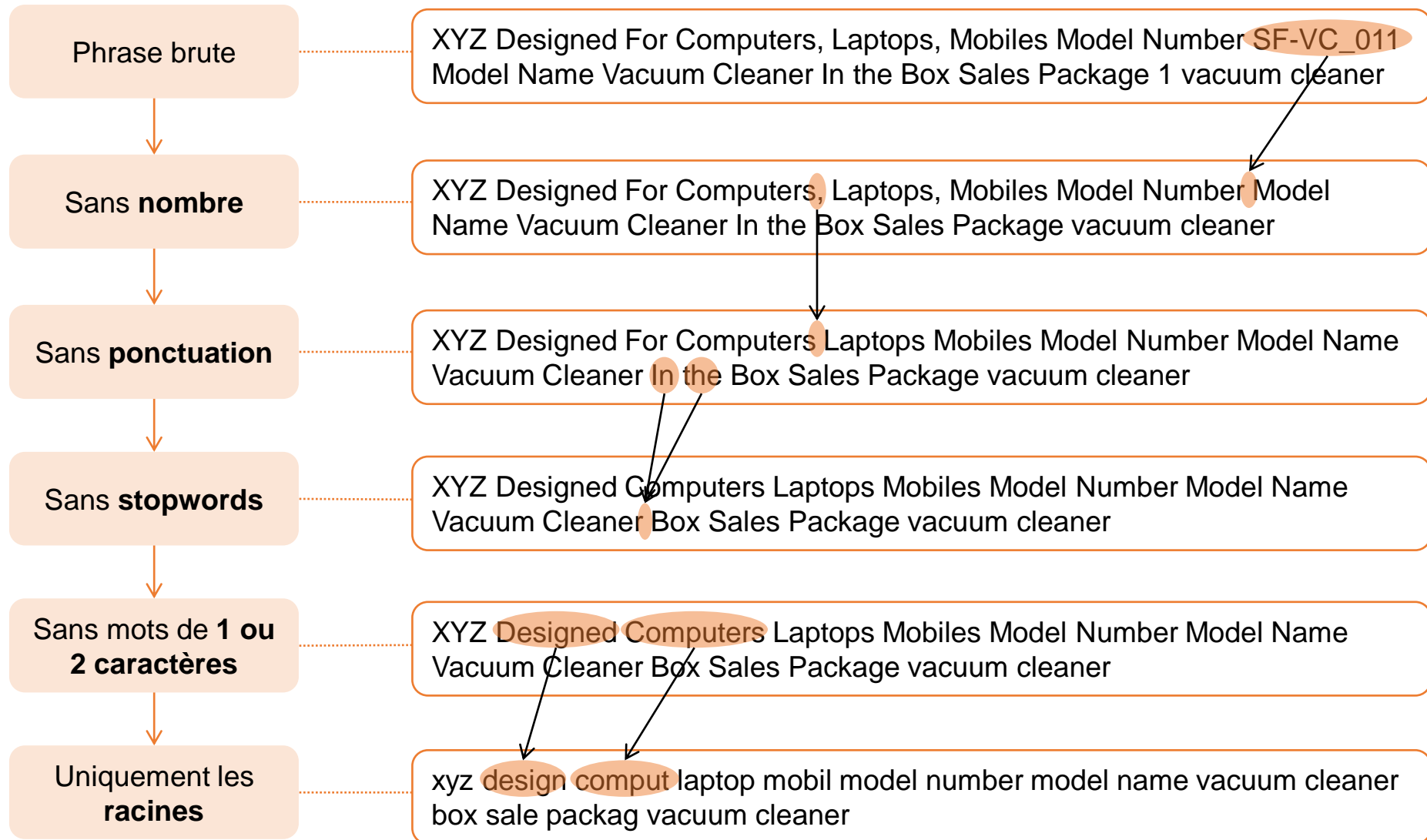
```
Cartier
```

Supprimer (trop de valeurs uniques, inutile pour l'algorithme)

Nettoyer

III. Traitement des textes

3. Nettoyer



III. Traitement des textes

4. Encoder

OneHotEncoding

→ Une **catégorie** par cellule

cat_tree_1st
home furnish
babi care

cat1_home_furnish	cat1_babi_care
0.0	0.0
1.0	0.0

Specific encoding

→ **Quelques** mots distincts par cellule

spec_type	spec_brand	spec_sales_package
eyelet	eleg	curtain
bath towel	sathiya	bath towel
flat	jaipur print	bed sheet pillow cover

spec_type_eyelet	spec_type_towel	spec_type_bath
1	0	0
0	1	1

Poids tf-idf

→ Un **nombre important** de mots distincts par cellule

description
key featur eleg
polyest
multicolor
abstract ey...
specif sathiya
cotton bath
towel bath
towel re...

desc_key	desc_featur	desc_eleg
0.0	0.0	0.0
0.0	0.0	0.0
0.0	0.0	0.0

product_name
eleg polyest
multicolor
abstract eyelet
door c...
sathiya cotton
bath towel

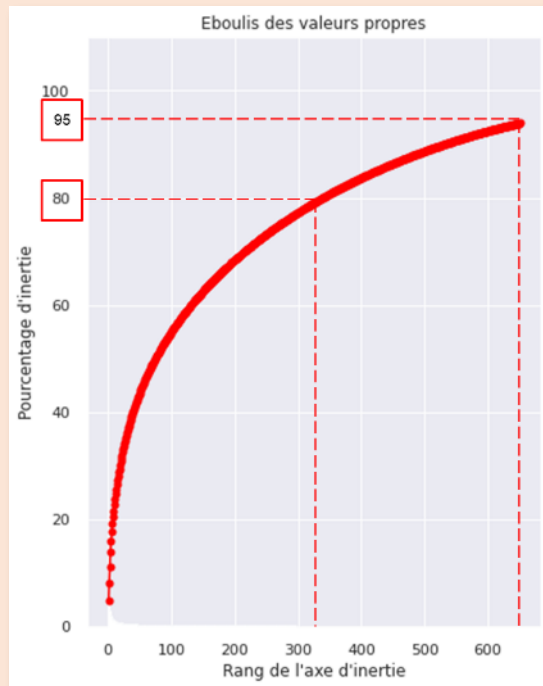
name_eleg	name_polyest	name_multicolor
0.0	0.0	0.0
0.0	0.0	0.0
0.0	0.0	0.0

IV. Segmentation

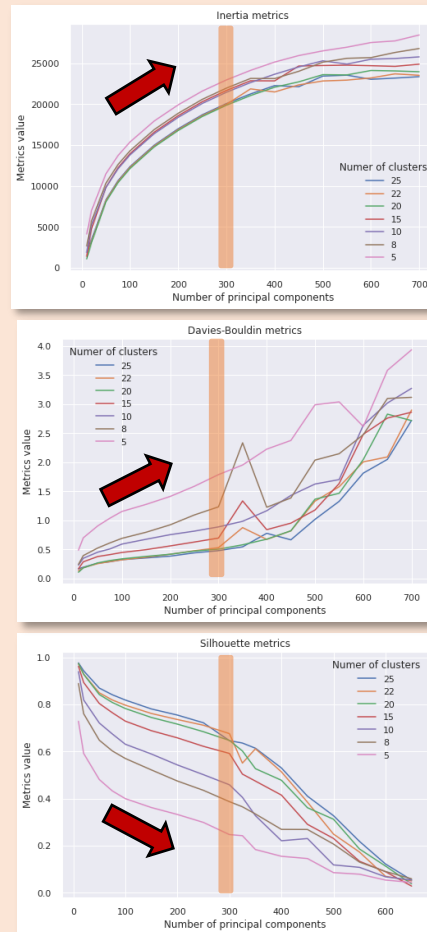
1. La réduction dimensionnelle : choisir en fonction de l'algorithme

Approche habituelle

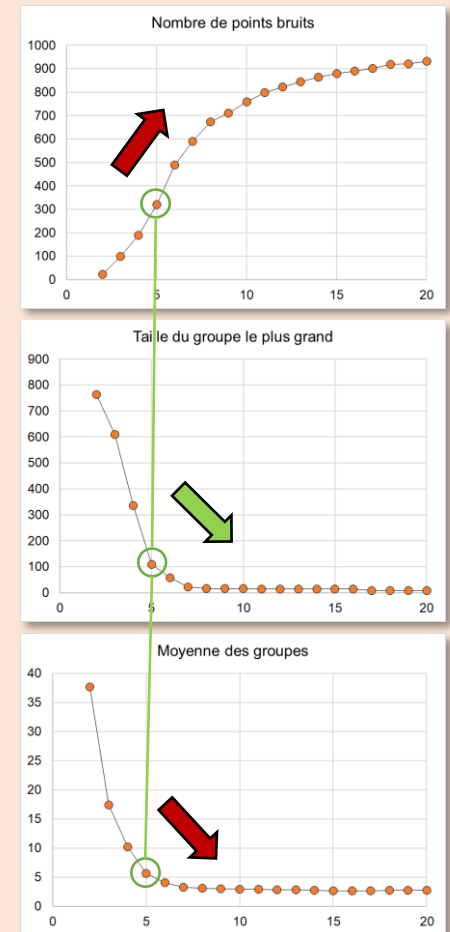
Garder les 80% ou 95% des composantes principales les plus importantes:



Fonction de k-means



Fonction de DBSCAN



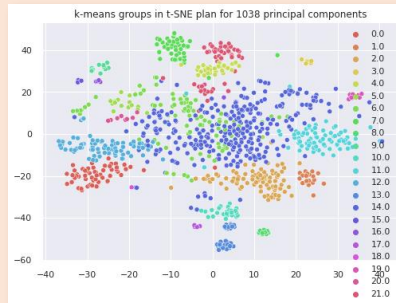
- Pour k-means : 300 composantes principales
- Pour DBSCAN, le meilleur compromis : 5 composantes principales

IV. Segmentation

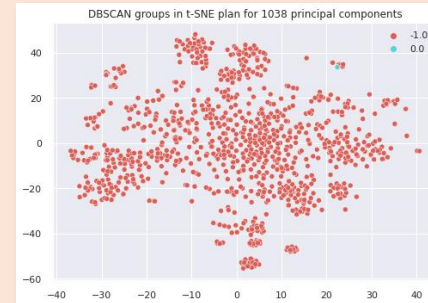
2. La visualisation dans t-SNE

1038 composantes
(nombre de produits)

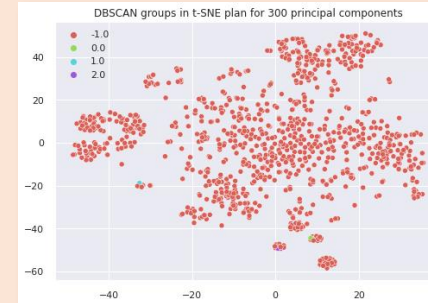
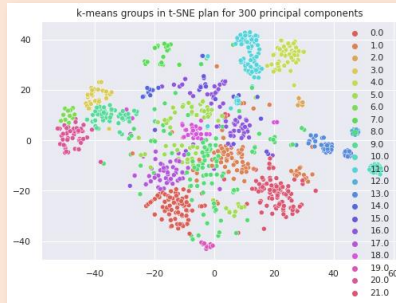
k-means



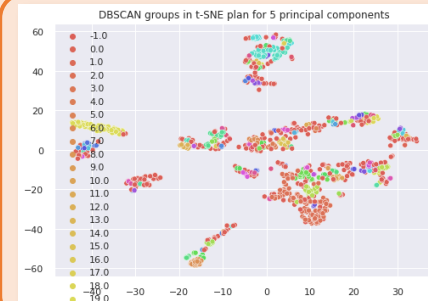
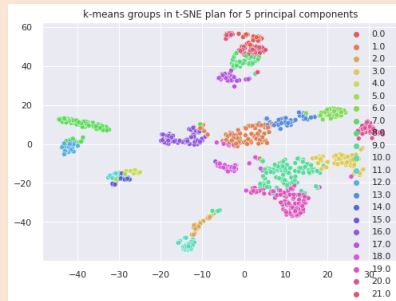
DBSCAN



300 composantes



5 composantes



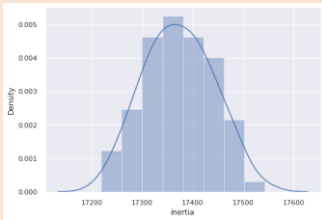
- Cohérence entre t-SNE et k-means.
- Les performances de DBSCAN sont insuffisantes (≈ 100 groupes)

IV. Segmentation

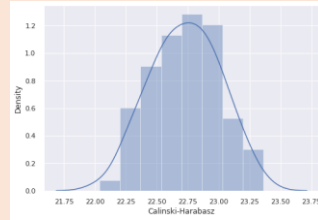
3. L'optimisation : rechercher les meilleures performances

k-means

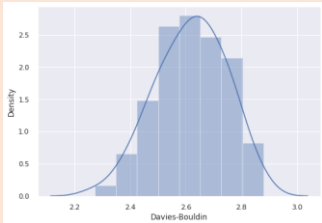
```
# Hyperparameters
n_init_list = [10, 15, 20, 30]
max_iter_list = [300, 400, 500, 600]
tol_list = [0.0001, 0.0003, 0.0005, 0.0008, 0.001]
```



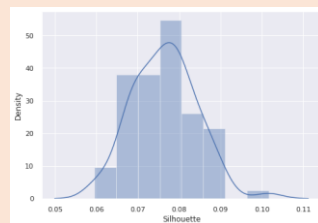
Inertie



Calinski-Harabasz



Davies-Bouldin



Silhouette

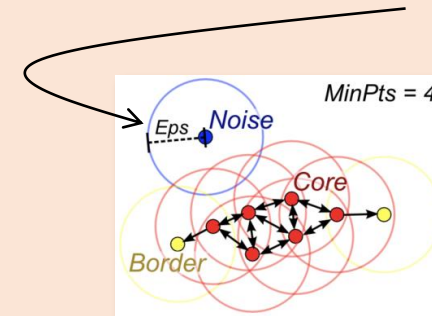
→ Métriques semblables d'une combinaison à l'autre.

→ Aucune combinaison ne se démarque par rapport aux autres.

DBSCAN

```
# Hyperparameters
eps_list = [0.02, 0.05, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7]
min_samples_list = [2, 3, 4, 5]
leaf_size_list = [10, 20, 30]
```

→ Meilleurs scores obtenus pour $eps = 0,1$.



eps est habituellement de **0,2**.

Pour $eps = 0,1$, les groupes formés seront plus **compacts**.

Mais il y aura plus de points considérés comme du **bruit**.

→ paramètres de référence pour k-means

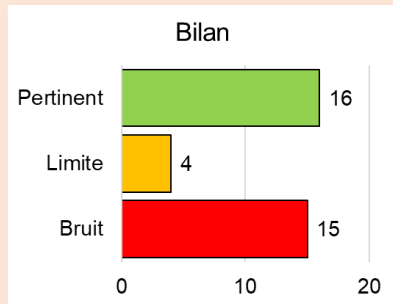
→ $eps = 0,1$ pour DBSCAN

IV. Segmentation

4. Le livrable

→ Etude d'un échantillon de sept groupes, à raison de cinq photos par groupe.

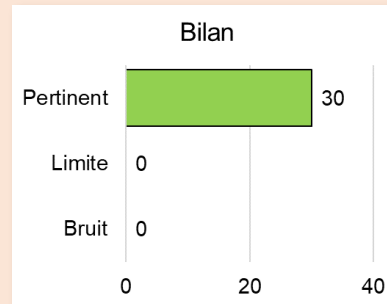
k-means



→ Niveau de précision : 46%

→ Quelques mélanges entre groupes (par ex. groupe 0 et groupe 6).

DBSCAN





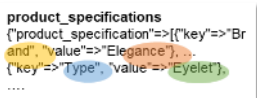


→ Niveau de précision : 100%

Mais :

1. 95,6 % du jeu de données considéré comme du bruit : inutilisable donc.
2. Groupes similaires.

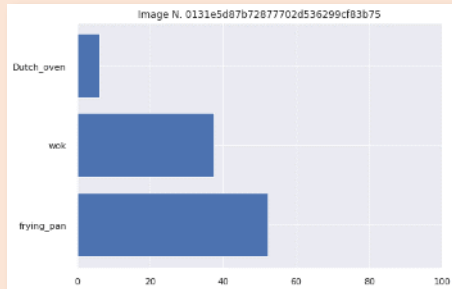
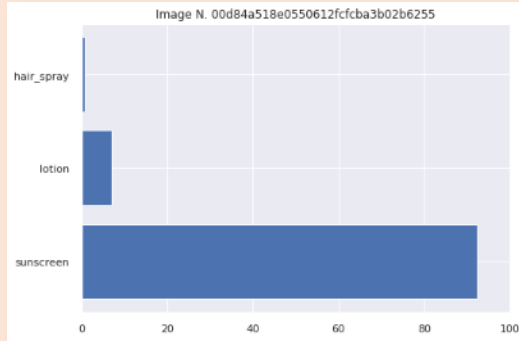
→ k-means, bien qu'imparfait, est plus adapté que DBSCAN pour ce jeu de données.

V. Bilan et perspectives

Sujet	Commentaire
	<p>Le jeu de données</p> <ul style="list-style-type: none">• Peu de données (1050 produits), mais les caractéristiques sont exploitables dans l'ensemble.• Le jeu de donnée est petit et limite l'efficacité des algorithmes.
	<p>Les images</p> <ul style="list-style-type: none">• Pour certaines images, l'algorithme ORB trouve des keypoints pertinents.• Pour d'autres images, ORB semble perturbé par du bruit difficilement gommé par les filtres.
	<p>Les textes</p> <ul style="list-style-type: none">• Les données textuelles sont exploitables.• Le résultat du nettoyage est pertinent.
	<p>La segmentation</p> <ul style="list-style-type: none">• La réduction dimensionnelle diffère d'un algorithme à l'autre.• k-means fournit les meilleurs résultats, avec 46% sur un échantillon.• DBSCAN ne semble pas adapté pour le jeu de données.
	<p>Perspectives</p> <ul style="list-style-type: none">• Enrichir le jeu de données avec d'autre produits.• SIFT n'a pas pu être utilisé, mais semble plus accessible dans sa version payante (opencv_contrib).

Perspectives d'améliorations

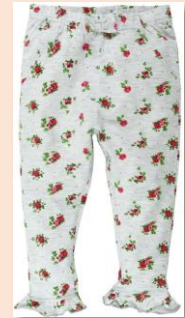
Réseaux de neurones VGG16 (Keras)



Traitement spécifique aux images difficiles



Abondance de détails

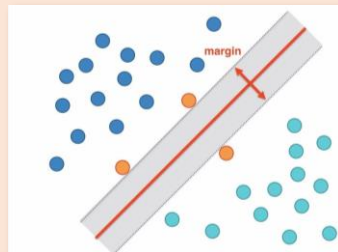


Motifs floraux

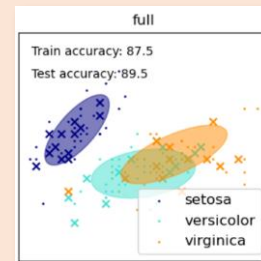


Objets brillants

Autres algorithmes de segmentation



SVM



Gaussian Mixture

Fin de la présentation



Merci pour votre attention

