PROJET 6 – « CLASSIFIEZ AUTOMATIQUEMENT DES BIENS DE CONSOMMATION »

Sommaire

1 - PROBLÉMATIQUE

- Rappel de la problématique
- Présentation du jeu de données

Moteur de classification des articles

• **Contexte** : Ce projet s'inscrit dans le cadre d'une "Place de marché" ou plateforme de commerce électronique.

• **Moyen** : Nous proposons d'automatiser l'attribution des catégories aux articles.

• **Objectif** : Améliorer l'expérience utilisateur et renforcer la fiabilité de la catégorisation.

• **Objectif du projet :** Notre but principal est d'étudier la faisabilité de cette catégorisation en suivant les étapes suivantes :

- Extraction des données à partir d'une API.
- Analyse et prétraitement du jeu de données, à la fois visuelles et textuelles.
- Application de techniques de regroupement (clustering) pour regrouper les articles dans des catégories pertinentes.

Notre étude de faisabilité repose sur un processus en plusieurs étapes :

1. **Prétraitement des données** :
   - Nous réalisons le prétraitement des données en les divisant en deux catégories : les données textuelles et les données visuelles.
2. **Essais de classification supervisée** :
   - Nous effectuons des essais de classification supervisée en utilisant à la fois les données textuelles et les données visuelles.
3. **Essais de classification non supervisée :**
   - Nous menons également des essais de classification non supervisée, en utilisant toujours les deux types de données : textuelles et visuelles.
4. **Assemblage des données et Essais de classification non supervisée :**
   - Nous combinons les données textuelles et visuelles et réalisons de nouveaux essais de classification non supervisée sur cet ensemble combiné.

Jeu de données

```
1 print ("Le dataset compte {} lignes et {} variables".format(data.shape[0], data.shape[1]))
```

```
Le dataset compte 1050 lignes et 15 variables
```

```
1 data.isna().sum()
```

```
uniq_id                    0
crawl_timestamp            0
product_url                0
product_name               0
product_category_tree      0
pid                        0
retail_price               1
discounted_price           1
image                      0
is_FK_Advantage_product    0
description                0
product_rating             0
overall_rating             0
brand                    338
product_specifications     1
dtype: int64
```

Exemple:

Répéteur Wi-Fi
Déodorant
Montres
Ventilateurs

II – API, PRETRAITEMENTS ET CLUSTERING

- Données textuelles
- Données Visuelles
- Modélisations effectuées

Données complémentaires : API Adamam

- **Exemple :**

extraction de données pour un type d'article absent de la base de données : *Champagne*

Requete d’extraction :

```python
url = "https://edamam-food-and-grocery-database.p.rapidapi.com/api/food-database/v2/parser"

querystring = {"nutrition-type":"cooking","category[0]":"generic-foods","health[0]":"alcohol-free"}

headers = {
    "X-RapidAPI-Key": "43c5049df4msh4f79b67cf3fced3p18ee6djsndd7ec5cd2f83",
    "X-RapidAPI-Host": "edamam-food-and-grocery-database.p.rapidapi.com"
}

response = requests.get(url, headers=headers, params=querystring)


# Extraction de la réponse JSON de la requête
data_API = response.json()

# Extraction de la liste des suggestions d'aliments à partir des résultats JSON
hints = data_API.get('hints', [])

# Liste pour stocker les informations des suggestions d'aliments foodId, label, category, foodContentsLabel, image.
hint_data = []
for hint in hints:
    food = hint['food']
    food_id = food.get('foodId')
    label = food.get('label')
    category = food.get('category')
    food_contents_label = food.get('foodContentsLabel')
    image = food.get('image')

    hint_data.append({
        'foodId': food_id,
        'label': label,
        'category': category,
        'foodContentsLabel': food_contents_label,
        'image': image
    })
```

Données textuelles : prétraitement

	cat_lvl_1	product_name	description	name_desc	remove_punc	remove_emoj	tokens	net_tokens	stem_words	lem_words	clean_text
0	Home Furnishing	Elegance Polyester Multicolor Abstract Eyelet Door Curtain	Key Features of Elegance Polyester Multicolor Abstract Eyelet Door Curtain F...	Elegance Polyester Multicolor Abstract Eyelet Door Curtain Key Features of E...	Elegance Polyester Multicolor Abstract Eyelet Door Curtain Key Features of E...	Elegance Polyester Multicolor Abstract Eyelet Door Curtain Key Features of E...	[Elegance, Polyester, Multicolor, Abstract, Eyelet, Door, Curtain, Key, Feat...	[Elegance, Polyester, Multicolor, Abstract, Eyelet, Door, Curtain, Key, Feat...	[eleg, polyest, multicolor, abstract, eyelet, door, curtain, key, featur, el...	[eleg, polyest, multicolor, abstract, eyelet, door, curtain, key, featur, el...	eleg polyest multicolor abstract eyelet door curtain key featur eleg polyest...
1	Baby Care	Sathiyas Cotton Bath Towel	Specifications of Sathiyas Cotton Bath Towel (3 Bath Towel, Red, Yellow, Blu...	Sathiyas Cotton Bath Towel Specifications of Sathiyas Cotton Bath Towel (3 B...	Sathiyas Cotton Bath Towel Specifications of Sathiyas Cotton Bath Towel (Ba...	Sathiyas Cotton Bath Towel Specifications of Sathiyas Cotton Bath Towel (3 B...	[Sathiyas, Cotton, Bath, Towel, Specifications, Sathiyas, Cotton, Bath, Towe...	[Sathiyas, Cotton, Bath, Towel, Specifications, Sathiyas, Cotton, Bath, Towe...	[sathiya, cotton, bath, towel, specif, sathiya, cotton, bath, towel, bath, t...	[sathiya, cotton, bath, towel, specif, sathiya, cotton, bath, towel, bath, t...	sathiya cotton bath towel specif sathiya cotton bath towel bath towel red ye...
2	Baby Care	Eurospa Cotton Terry Face Towel Set	Key Features of Eurospa Cotton Terry Face Towel Set Size: small Height: 9 in...	Eurospa Cotton Terry Face Towel Set Key Features of Eurospa Cotton Terry Fac...	Eurospa Cotton Terry Face Towel Set Key Features of Eurospa Cotton Terry Fac...	Eurospa Cotton Terry Face Towel Set Key Features of Eurospa Cotton Terry Fac...	[Eurospa, Cotton, Terry, Face, Towel, Set, Key, Features, Eurospa, Cotton, T...	[Eurospa, Cotton, Terry, Face, Towel, Set, Key, Features, Eurospa, Cotton, T...	[eurospa, cotton, terri, face, towel, set, key, featur, eurospa, cotton, ter...	[eurospa, cotton, terri, face, towel, set, key, featur, eurospa, cotton, ter...	eurospa cotton terri face towel set key featur eurospa cotton terri face tow...

Conclusion

```
1  # Affichage des resultat
2
3  df_resultats
```

	model	ARI
0	BOW	0.429
1	TF-IDF	0.503
2	Word2Vec	0.252
3	BERT (bert-base-uncased)	0.376
4	BERT(hub Tensorflow)	0.359
5	USE	0.451

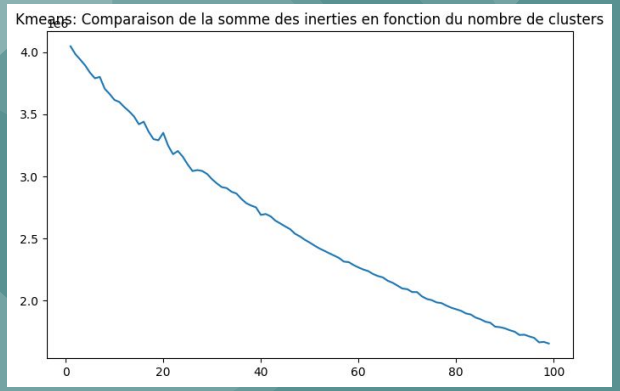
Voici une analyse des différents modèles et de leurs valeurs respectives d'ARI (Indice de Rand Ajusté) qui ont été utilisés pour l'analyse :

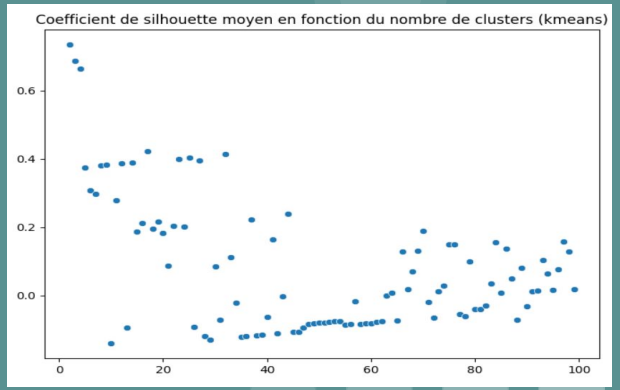
1. **BOW** (Sacs de mots) : A obtenu un ARI de 0,429. Cela signifie que le modèle de Sacs de mots a réussi à obtenir une certaine mesure de regroupement approprié, bien qu'il puisse y avoir une marge d'amélioration.
2. **TF-IDF** : A obtenu un ARI de 0,502. Ce modèle a eu légèrement de meilleurs résultats que le modèle de Sacs de mots, ce qui indique que l'utilisation de la pondération TF-IDF pourrait avoir amélioré le regroupement des données.

Données textuelles : catégorisation

Classifieurs non supervisés
Réduction de dimension :
Latent Dirichlet Allocation avec 7 catégories
Affichage des mots les plus importants de chaque topic

```
Topic #0: warranty laptop combo adapter skin set replacement type product print
Topic #1: baby 's detail girl fabric cotton specification boy battery general
Topic #2: mug ceramic coffee perfect gift bring '' price home ``
Topic #3: watch analog product online men day buy guarantee delivery 30
Topic #4: free product cash shipping genuine delivery buy flipkart.com 30 day
Topic #5: cm 1 feature color box material pack type inch specification
Topic #6: '' ceramic rockmantra mug 3.5 4 specification gift feature exclusive
```





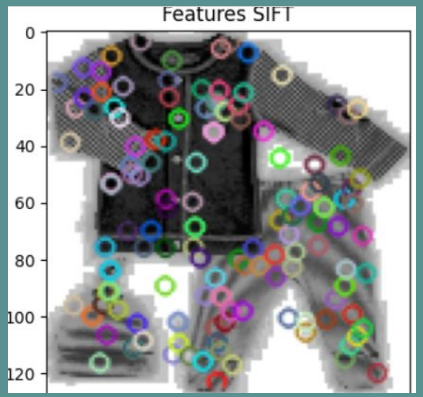
Classifieur supervisé (Essai avec SVC) :
• Création d'une nouvelle feature « catégorie niveau 2 » à partir des données (62 catégories)
• Accuracy sur jeu test : 79 %

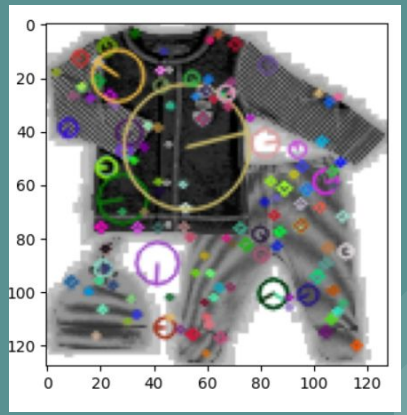
Données visuelles : extraction de features

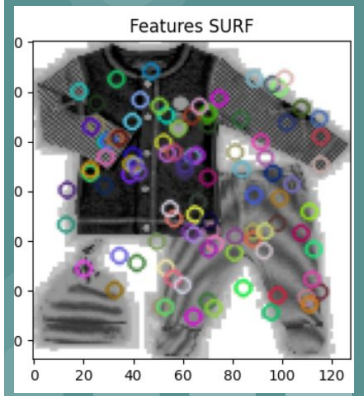



Pré-traitement
1. Noir et Blanc
2. Réduction bruit (flou gaussien)
3. Egaliseur
4. Redimensionnement

• Extraction de features (Exemple avec SIFT/SURF







Données visuelles : extraction de features

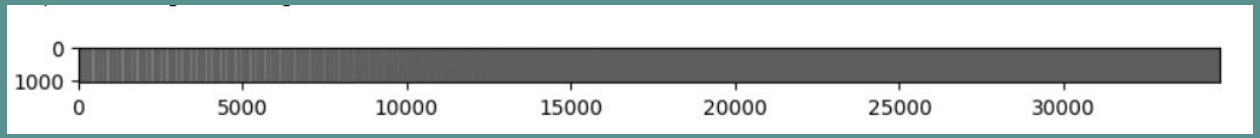
- **Création de features à partir des informations**

N descripteurs SURF par image (64px / 1 dim)

Concaténation des features d'une image sur une ligne de tableau
Remplissage des features « absentes » par des 0

Concaténation des features de toutes les images dans un même tableau

- **Obtention d'un array « creux » :**

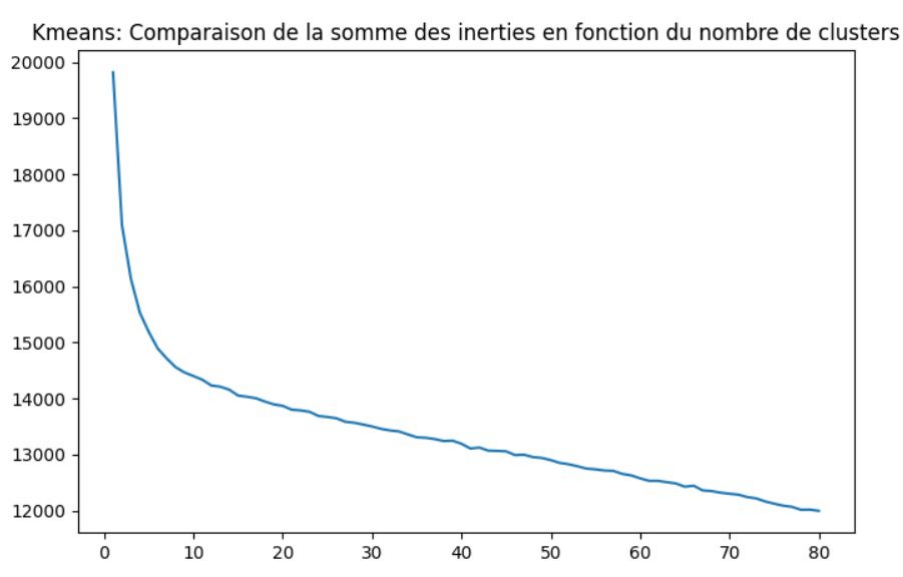


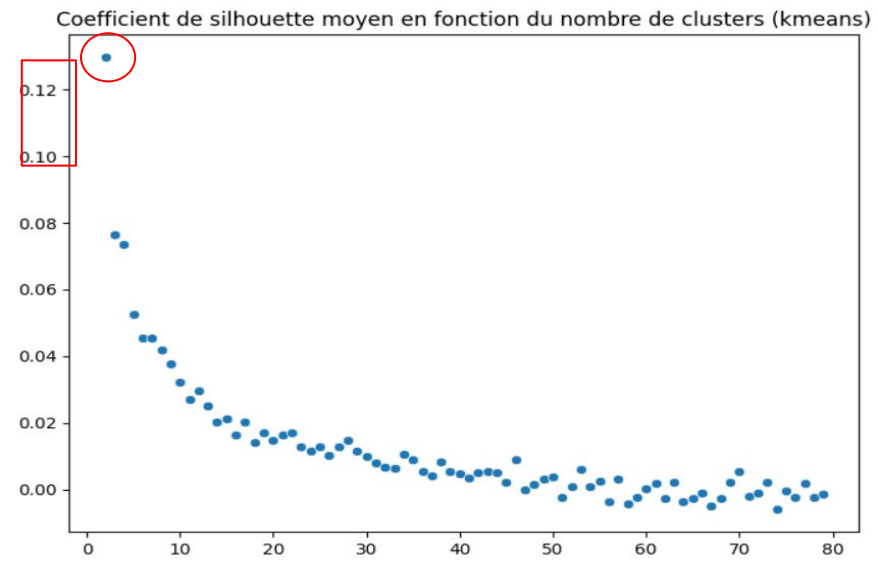
- **Création de nouvelles colonnes à partir des descripteurs:**
  - **Min, max, médiane, variance, moments d'ordre 3 et 4**

Données visuelles : classification

Classification après extraction des features •

Kmeans après ACP : Non concluant





Données visuelles : Réseaux de neurones

- **Construction d'un réseau de neurone convolutif simple**

```
model = Sequential()
model.add(Conv2D(32, kernel_size=(3,3), padding='same', activation='relu',
model.add(MaxPooling2D(pool_size=(2,2)))
model.add(Conv2D(32, kernel_size=(3,3), padding='same', activation='relu'))
model.add(MaxPooling2D(pool_size=(2,2)))
model.add(Flatten())
model.add(Dense(ohe.categories_[0].shape[0], activation='softmax'))
model.compile(loss='mean_squared_error', optimizer='sgd')
```

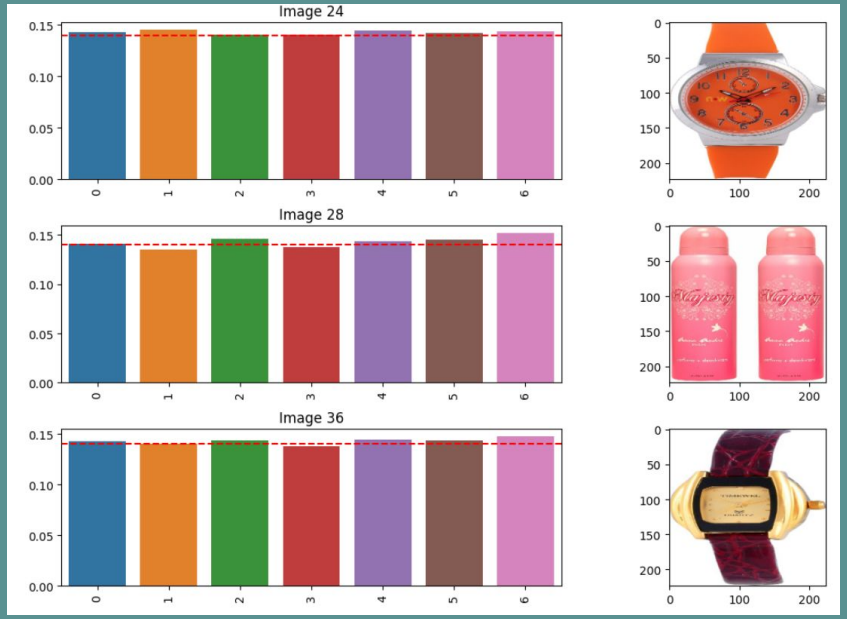
- Entraînement supervisé
  - Accuracy score : 8 %
  - Classification de tout le jeu de test dans la catégorie la plus représentée : **non concluant**

```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 conv2d (Conv2D)             (None, 128, 128, 32)      896

 max_pooling2d (MaxPooling2D  (None, 64, 64, 32)       0
 )

 conv2d_1 (Conv2D)           (None, 64, 64, 32)        9248

 max_pooling2d_1 (MaxPooling  (None, 32, 32, 32)       0
 2D)

 flatten (Flatten)           (None, 32768)             0

 dense (Dense)               (None, 57)                1867833

_________________________________________________________________
```

```
model_info = model.fit(train_array_cnn, train_arra
Epoch 1/3
20/20 - 7s - loss: 0.0316 - 7s/epoch - 348ms/step
Epoch 2/3
20/20 - 5s - loss: 0.0301 - 5s/epoch - 239ms/step
Epoch 3/3
20/20 - 5s - loss: 0.0301 - 5s/epoch - 254ms/step
```

Données visuelles : Réseaux de neurones

- **Transfer Learning (ImageNet VGG16):**
- **Substitition dernières couches par couche Dense**
- **Préparation des données:**
- **Redimensionnement 224 x 224**



Analyses image

Voici une analyse des différents modèles et de leurs valeurs respectives d'ARI (Indice de Rand Ajusté) qui ont été utilisés pour le traitement d'images :

VGG16 : A obtenu un ARI de 0,15248. Cela suggère que le modèle VGG16 n'a pas réussi à capturer des motifs significatifs dans les images pour le regroupement. Il pourrait y avoir des problèmes liés à la complexité du modèle ou à la nature des données.

PCA (Analyse en composantes principales) : A obtenu un ARI de 0,21414. Bien que ce modèle ait obtenu une meilleure performance que VGG16, il reste encore de la marge pour améliorer la capacité de regroupement. Cela pourrait être dû à une réduction de la dimensionnalité insuffisante ou à des problèmes liés à la représentation des données.

EfficientNetV2M : A obtenu un ARI de 0,22897. Ce modèle a montré une amélioration par rapport aux modèles précédents, mais il peut y avoir des aspects tels que les paramètres de l'architecture, la taille du jeu de données ou la préparation des images qui pourraient être optimisés pour obtenir de meilleurs résultats.

ConvNeXtBase : A obtenu un ARI de 0,53229. Ce modèle a obtenu le meilleur ARI parmi les modèles que vous avez testés jusqu'à présent. Cela suggère que ConvNeXtBase a réussi à extraire des caractéristiques significatives des images pour le regroupement. Il est important de noter que cet ARI élevé peut indiquer une bonne capacité de regroupement, mais il pourrait être utile d'explorer davantage les paramètres du modèle ou les méthodes de prétraitement des images pour une performance encore meilleure.

En comparant ces résultats, il est clair que **ConvNeXtBase** a obtenu le **meilleur ARI** parmi les modèles que vous avez testés pour le traitement d'images. Cependant, il convient de continuer à explorer et à expérimenter avec différents modèles, architectures et paramètres pour maximiser la qualité du regroupement et obtenir des résultats encore plus précis.

```
In [364]:
1  # Mostrar el DataFrame
2  #print(df_resultats)
3  df_resultats
```

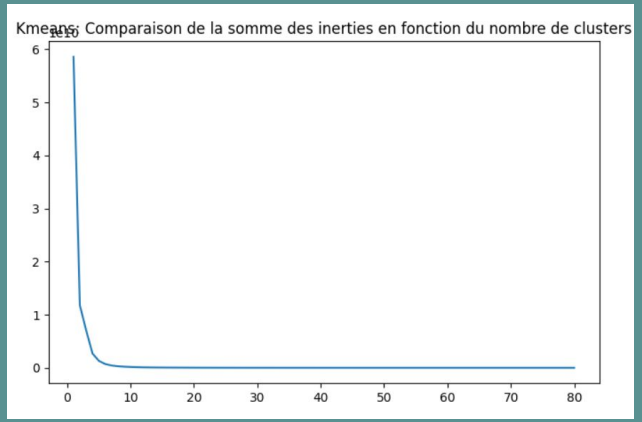
Out[364]:

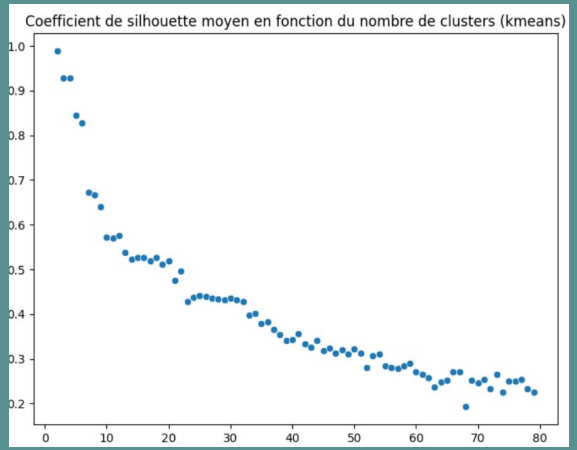
	model	ARI
0	VGG16	0.06417
1	PCA	0.06301
2	EfficientNetV2M	0.14799
3	ConvNeXtBase	0.54825

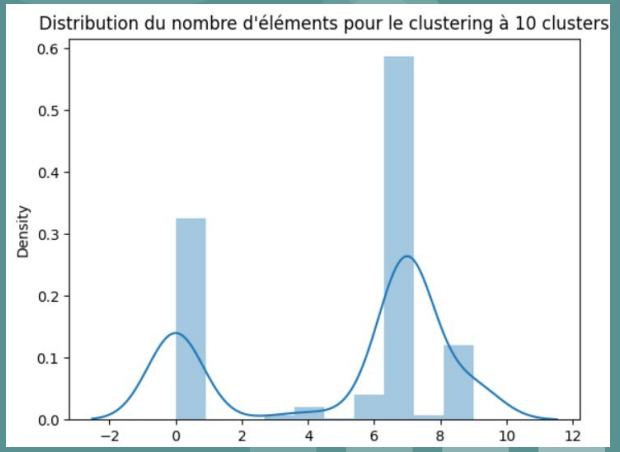
Assemblage données visuelles et textuelles

```
Tableau données textuelles NLP :  (1050, 340)
Nouvelles features Descripteurs: Descripteurs :  (1050, 6)
Réseau de neurone Imagenet :CNN :  (1050, 7)
```

- Assemblage: obtention d'un array de 1050 lignes X 341 colonnes





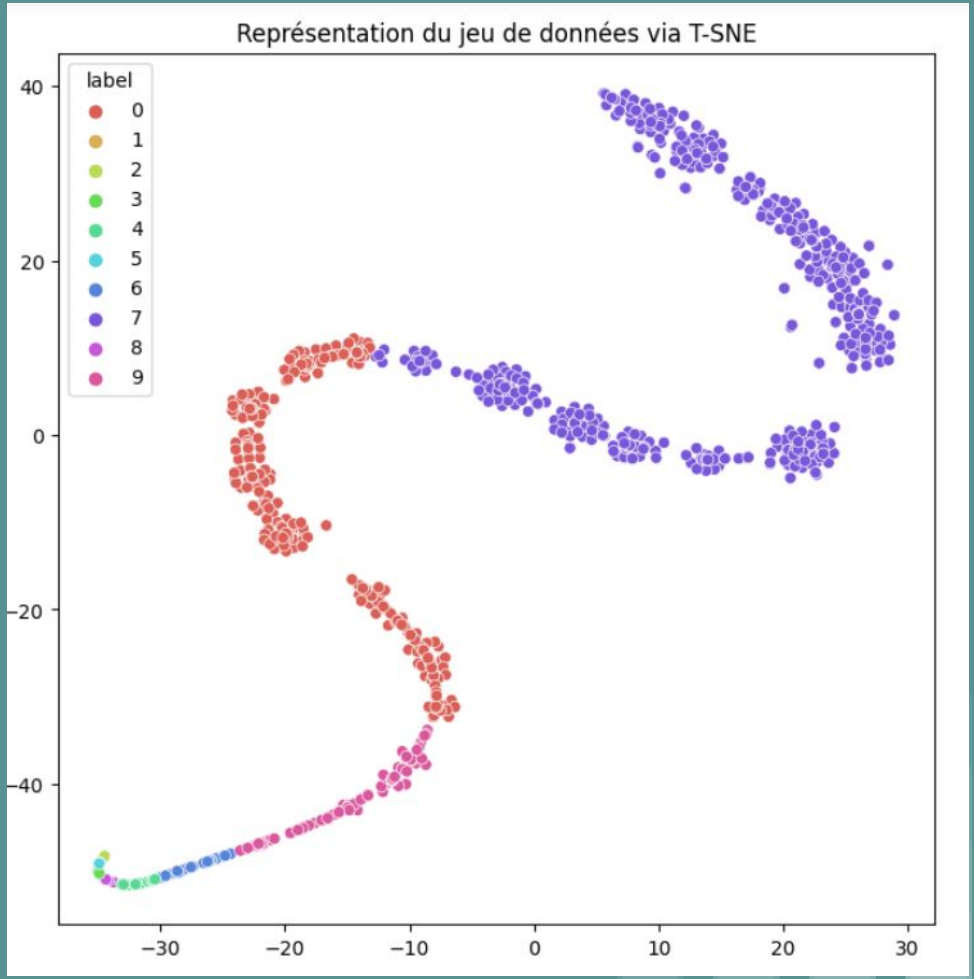


3 – CONCLUSIONS ET RECOMMANDATIONS

Résultats du clustering T-SNE pour visualisation

- Clustering non supervisé : résultat non concluant

- Alternative envisageable : apprentissage supervisé



MERCI DE VOTRE ATTENTION