

Projet 06 :

# **Ameliorer le produit IA de votre start-up**

Mohamed A.



StartUp IA qui met en relation des clients et des restaurants.

**Votre entreprise souhaite améliorer sa plateforme avec une nouvelle fonctionnalité de collaboration. Les utilisateurs pourront par exemple poster des avis et des photos sur leur restaurant préféré. Ce sera aussi l'occasion, pour l'entreprise, de mieux comprendre les avis postés par les utilisateurs.**

Use Case	<p>En tant qu'utilisateur de Avis Restau, je peux :</p> <ul style="list-style-type: none"><li>• poster des avis sous forme de commentaires.</li><li>• poster des photos prises dans le restaurant.</li></ul> <p>En tant qu'Avis Restau, je souhaite :</p> <ul style="list-style-type: none"><li>• Détecter les sujets d'insatisfaction présents dans les commentaires postés sur la plateforme.</li><li>• Labelliser automatiquement les photos postées sur la plateforme. Par exemple, identifier les photos relatives à la nourriture, au décor dans le restaurant ou à l'extérieur du restaurant.</li></ul>	Jeu de données	<ul style="list-style-type: none"><li>• Problème : Pas assez de données sur la plateforme Avis Restau.</li><li>• Solution : utiliser un jeu de données existant.</li><li>• Lien vers le jeu de données : <a href="https://www.yelp.com/dataset">https://www.yelp.com/dataset</a></li><li>• Contient des informations générales (par exemple type de cuisine) et les avis des consommateurs sur les différents restaurants.</li><li>• Au vu du volume du jeu de données, pour pouvoir le charger entièrement, s'aider de <a href="#">cet article</a>.</li></ul>
Scope du projet	<p>Étude préliminaire fonctionnalité "Détecter les sujets d'insatisfaction" et "Labelliser automatiquement les photos postées"</p>	Collecte des données	<ul style="list-style-type: none"><li>• Problème : s'assurer de la possibilité de collecter de nouvelles données.</li><li>• Solution : collecter de nouvelles données via l'API Yelp. Valider la faisabilité de la solution en collectant les informations relatives à environ 200 restaurants pour une ville en utilisant l'API.</li><li>• Lien vers la <a href="#">documentation</a> de l'API Yelp.</li></ul>
		Outils	<ul style="list-style-type: none"><li>• Python et librairies spécialisées NLP/CV.</li><li>• Jupyter Notebook et package Voilà</li></ul>

Bonjour Ider,

J'ai bien reçu et pris en compte ce cahier des charges. Je t'en remercie.

Voici les différentes étapes que je vais réaliser :

- analyser les commentaires négatifs pour détecter les différents sujets d'insatisfaction :
  - sélection de quelques milliers de commentaires négatifs,
  - prétraitement des données textuelles,
  - utilisation de techniques de réduction de dimension,
  - visualisation des données de grandes dimensions afin de détecter des mots clés et sujets d'insatisfaction ;
- analyser les photos pour déterminer les catégories des photos :
  - sélection de 100 à 200 photos par catégorie,
  - prétraitement des images. Je vais tester deux approches, une par extraction de descripteurs (SIFT, ORB ou SURF) et une par Transfer Learning d'un réseau de neurones de type CNN,
  - utilisation de techniques de réduction de dimension,
  - visualisation des données de grandes dimensions en mettant en évidence les catégories des images,
  - vérification que les images sont correctement regroupées selon les catégories en réalisant un clustering, puis une comparaison des clusters avec les catégories des images, via un graphique et une mesure. Je vais analyser également quelles sont les catégories les mieux regroupées,
  - cette vérification me permettra de conclure sur la faisabilité de réaliser ultérieurement une classification supervisée, j'ai bien compris qu'il n'était pas nécessaire à ce stade de réaliser cette classification supervisée ;
- collecter un échantillon de données (environ 200 restaurants et leurs revues) via l'API Yelp :
  - récupérer uniquement les champs nécessaires,
  - stocker les résultats dans un fichier exploitable (par exemple CSV).

Je te présenterai mon travail en illustrant au maximum pour le rendre facilement compréhensible lors de notre prochain point.

A bientôt

Projet :

**VOLET I**

**VOLET II**

**VOLET III**

## VOLET I : TOPIC MODELING

- 1- Chargement, selection et preparation des données a utiliser pour le Topic modeling.
- 2- Pre-processing des donnees selon les etapes suivantes :
  - 2-1 Lower case of all documents and special characters removal
  - 2-2 Tokenization & stopwords removal
  - 2-3 Lemmatization en utilisant le referant 'english'
- 3- Vectorization du corpus pour l'extraction des features et Constrcution du model NMF
- 4- Choix du nombre de topics et visualisation des resultats
- 5- Interpretation de chaque topic principal

## VOLET II : IMAGE CLASSIFICATION

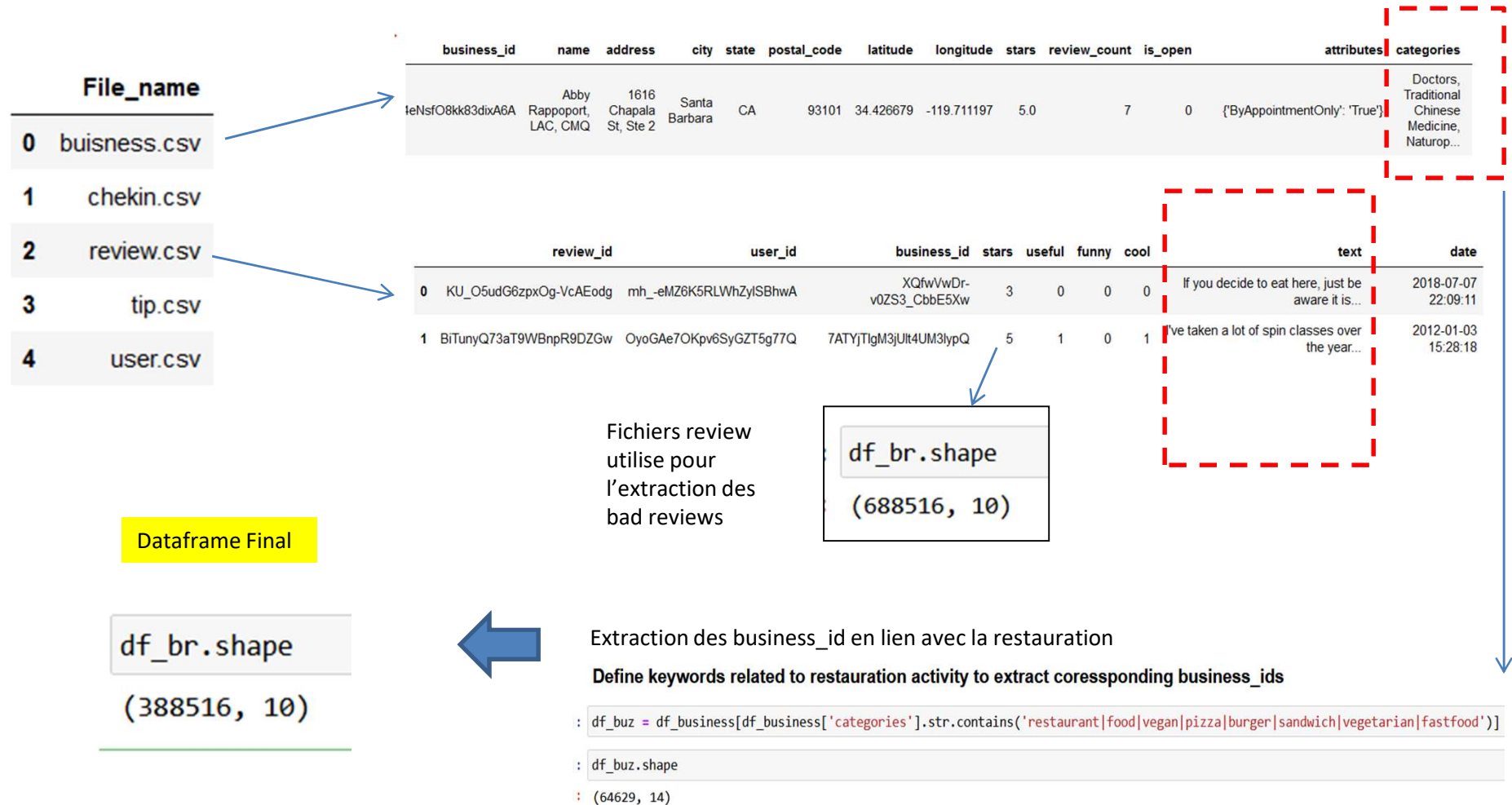
- 1- Chargement, selection et preparation des données a utiliser pour la classification d'image
- 2- classification par l'approche SIFT :
  - 2-1 Extraction des descripteurs et keypoints
  - 2-2 Creation de clusters de descripteurs
  - 2-3 Creation d'histogrammes par image
  - 2-4 Reduction de dimension PCA
  - 2-5 Reduction de dimension T-SNE pour visualtion en 2D
  - 2-6 Visualisation des resultat en mettant en evidence les categories d'image
  - 2-7 Klustering via kmeans de la matrice reduite T-SNE et calcul de l'accuracy\_score, f1-score et matrice de confusion
- 3- classification par l'approche CNN :
  - 3-1 Definition du modele VGG16 et elimination des layers OUTPUT
  - 3-2 Extraction des descripteurs et keypoints
  - 3-3 Reduction de dimension PCA
  - 3-4 Reduction de dimension T-SNE
  - 3-5 Visualisation des resultat en mettant en evidence les categories d'image
  - 3-6 Klustering via kmeans de la matrice reduite T-SNE et calcul de l'accuracy\_score, f1-score et matrice de confusion

## VOLET II : YELP DATA EXTRACTION VIA API

- 1- keys and parameters setting
- 2- Get response from YELP database
- 3-response treatment and export to csv format

# VOLET I : TOPIC MODELING

1- Chargement, selection et preparation des données a utiliser pour le Topic modeling.



# VOLET I : TOPIC MODELING

## 2- Pre-processing des donnees : .

Lower case of all documents and special characters removal



```
: # Apply lower case to all words
df_br['text'] = df_br['text'].str.lower()
# Clean text from special characters
tt= []
for x in df_br['text']:
    x = re.sub(r'^\w\s', '', x)
    #x = re.sub('[. *? \]', ' ', x)
    #x = re.sub('[%s]' % re.escape(string.
    #x = re.sub(r'\w*\d\w*', ' ', x)
    #x = re.sub('?', ' ', x)
    tt.append(x)

df_br['text'] = tt
print('*****Cleaning ok')
print(df_br['text'].head(2))
del tt
```

```
*****Cleaning ok
0    i am a long term frequent customer of
1    if you want to pay for everything a
Name: text, dtype: object
```

Tokenization & stopwords removal



```
## Creating new specific list of stopwords
stop_words = stopwords.words('english')
new_stopwords = ["came", "didn't", "dont", "ordered", "would"]
stop_words.extend(new_stopwords)

## tokenization and Removing stop words
tt= []
for x in df_br['text']:
    x=nlk.word_tokenize(x)
    x= [w for w in x if not w in stop_words]
    tt.append(x)

df_br['text'] = tt
print('*****Tokenization & Stop-words removed')
print(df_br['text'].head(2))
del tt
```

```
*****Tokenization & Stop-words removed ok
0    [long, term, frequent, customer, establishment...
1    [want, pay, everything, la, carte, place, food...
Name: text, dtype: object
```

Lemmaatization en utilisant le referant 'english'



```
## Apply Lemmatization according to english language
tt=[]
for x in df_br['text']:
    ee=[]
    for word in x :
        ee.append(lemmatizer.lemmatize(word))

    tt.append(ee)
df_br['text'] = tt
print('*****Lemmatization ok')
print(df_br['text'].head(2))
del tt
```

```
*****Lemmatization ok
0    [long, term, frequent, customer, establishment
1    [want, pay, everything, la, carte, place, food
Name: text, dtype: object
```

# VOLET I : TOPIC MODELING

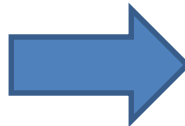
3/4- Vectorization du corpus pour l'extraction des features et Constrcution du model NMF.

Corpus :

df\_br['text']

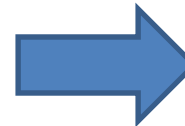


TF-IDF  
vectorization



Model NMF  
5 components/  
Fit et  
transform(X)  
pour  
nmf\_features

avoir:



*#check the dimensions of the 3 tables*

```
X.shape, nmf_features.shape, model.components_.shape
```

```
((388516, 13046), (388516, 6), (6, 13046))
```

```
X.shape
```

```
(388516, 13046)
```



```
components_df.idxmax(axis = 1)
```

```
0      time
1      food
2      order
3      pizza
4  service
5      table
dtype: object
```

# VOLET I : TOPIC MODELING

5- Interpretation des resultats pour le Topic modeling.

Main words/Topics

0      time  
1      food  
2      order  
3      pizza  
4      service

Topic 0

- ☐ Delais de service
- ☐ Nombre de fois ou le service n'a pas ete bon

Topic 1

- ☐ Mauvaise qualite de la nourriture
- ☐ Mauvaise cuisson, nourriture quasi perimee....

Topic 2

- ☐ Problemes rencontres lors des commandes
- ☐ Impolitesses lors de prise de commande ou remplacement
- ☐ Erreur de commandes....

Topic 3

- ☐ Problemes rencontres avec le service pizza
- ☐ Pizza non cuite, peu de fromage, mauvaise qualite

Topic 4

- ☐ Problemes observes lors de diverses etapes du service
- ☐ Les client pratiquement decredibilisent le service en entier



# VOLET II : IMAGE CLASSIFICATION

## VOLET II : IMAGE CLASSIFICATION

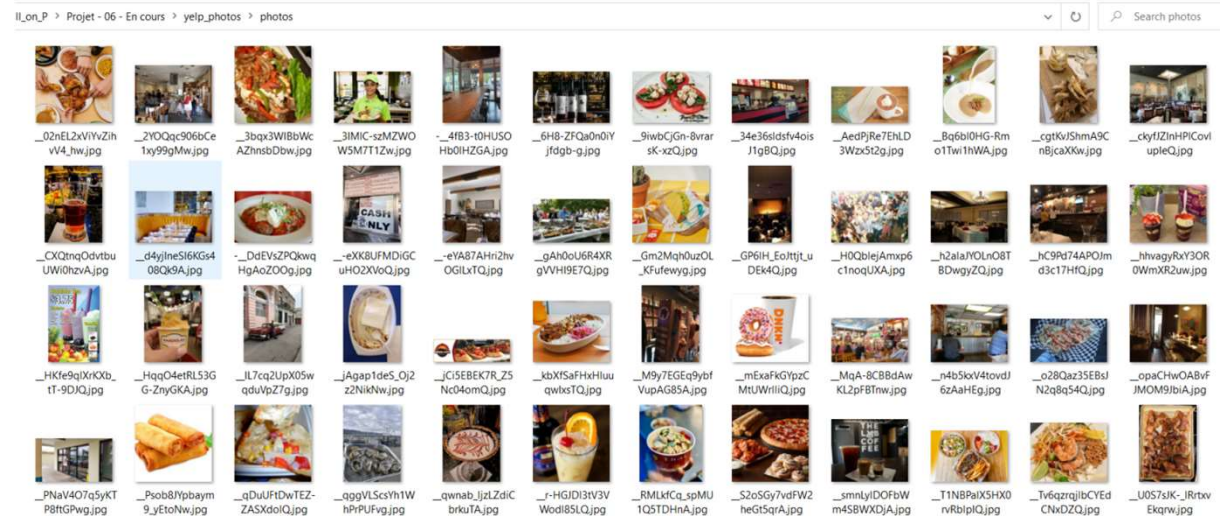
- 1- Chargement, selection et preparation des données a utiliser pour la classification d'image
- 2- classification par l'approche SIFT :
  - 2-1 Extraction des descripteurs et keypoints
  - 2-2 Creation de clusters de descripteurs
  - 2-3 Creation d'histogrammes par image
  - 2-4 Reduction de dimension PCA
  - 2-5 Reduction de dimension T-SNE pour visualtion en 2D
  - 2-6 Visualisation des resultat en mettant en evidence les categories d'image
  - 2-7 Klustering via kmeans de la matrice reduite T-SNE et calcul de l'accuracy\_score, f1-score et matrice de confusion
- 3- classification par l'approche CNN :
  - 3-1 Definition du modele VGG16 et elimination des layers OUTPUT
  - 3-2 Extraction des descripteurs et keypoints
  - 3-3 Reduction de dimension PCA
  - 3-4 Reduction de dimension T-SNE
  - 3-5 Visualisation des resultat en mettant en evidence les categories d'image
  - 3-6 Klustering via kmeans de la matrice reduite T-SNE et calcul de l'accuracy\_score, f1-score et matrice de confusion

# VOLET II : IMAGE CLASSIFICATION

1- Chargement, selection et preparation des données a utiliser pour la classification d'image :

❑ Source des photos : YELP

❑ Nombre de photos : 200098



Fichiers json fourni avec le dataset

```
photos.head(3)
```

	photo_id	business_id	caption	label
0	zsvj7vIoL4L5jhYyPluVwg	Nk-SJhPIDBkAZvfsADtccA	Nice rock artwork everywhere and craploads of ...	inside
1	HCUdRJJHm_e0OCTIZetGLg	yVZtL5MmrpiivyClrVKGgA		NaN outside
2	vkr8T0scuJmGVvN2HJeIEA	_ab50qdWOk0DdB6XOrBitw		oyster shooter drink

❑ 1000 IMAGES SOIT 200 PAR TYPE DE LABELS ONT ETES UTILISES POUR LA CLASSIFICATION

❑ 5 LABELS : FOOD,INSIDE,OUTSIDE,DRINK,MENU

# VOLET II : IMAGE CLASSIFICATION

## 2- classification par l'approche SIFT :

- 2-1 Extraction des descripteurs et keypoints
- 2-2 Creation de clusters de descripteurs
- 2-3 Creation d'histogrammes par image
- 2-4 Reduction de dimension PCA
- 2-5 Reduction de dimension T-SNE pour visualtion en 2D

Nombre de descripteurs : (488573, 128)

Nombre de clusters estimés : 699

```
sift_keypoints=[]
i=0
for x in global_list:
    link = os.path.join(parent_dir, x)
    yy=cv2.imread(link,cv2.IMREAD_GRAYSCALE)
    res = cv2.equalizeHist(yy) # equalize image histogram
    kp, des = sift.detectAndCompute(res,None)
    i=i+1
    sift_keypoints.append(des)

sift_kp_by_img = np.asarray(sift_keypoints)
sift_kp_all = np.concatenate(sift_kp_by_img,axis=0)
```

```
from sklearn import cluster, metrics

# Determination number of clusters
temps1=time.time()

k = int(round(np.sqrt(len(sift_kp_all)),0))
print("Nombre de clusters estimés : ", k)
print("Création de",k, "clusters de descripteurs ...")

# Clustering
kmeans = cluster.MiniBatchKMeans(n_clusters=k, init_size=3*k, random_state=0)
kmeans.fit(sift_kp_all)
```


```
def build_histogram(kmeans, des, image_num):
    res = kmeans.predict(des)
    hist = np.zeros(len(kmeans.cluster_centers_))
    nb_des=len(des)
    if nb_des==0 : print("problème histogramme image : ", image_num)
    for i in res:
        hist[i] += 1.0/nb_des # normalisation des histograms

    return hist

# Creation of a matrix of histograms
hist_vectors=[]

for i, image_desc in enumerate(sift_kp_by_img):
    if i%100 == 0 : print(i)
    hist = build_histogram(kmeans, image_desc, i) #calculates the histogram
    hist_vectors.append(hist) #histogram is the feature vector

im_features = np.asarray(hist_vectors)
```

Reduction PCA :  Dimensions dataset avant réduction PCA : (1000, 699)  
Dimensions dataset après réduction PCA : (1000, 543)

Reduction T-SNE :  (1000, 3)

# VOLET II : IMAGE CLASSIFICATION

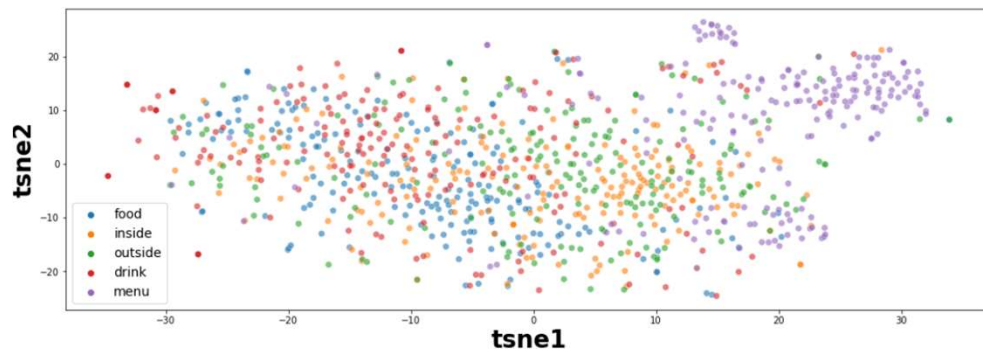
2- classification par l'approche SIFT :

2-6 Visualisation des resultat en mettant en evidence les categories d'image

2-7 Klustering via kmeans de la matrice reduite T-SNE et calcul de l'accuracy\_score, f1-score et matrice de confusion

- ❑ On observe une certaine distinction de la classe 'menu' seulement pour le premier graphique
- ❑ Le score accuracy apres clustering est faible et la matrice de confusion est disparate, ce qui ne favorise pas l'utilisation de cette approche pour ce projet de classification.

**TSNE selon les vraies classes**

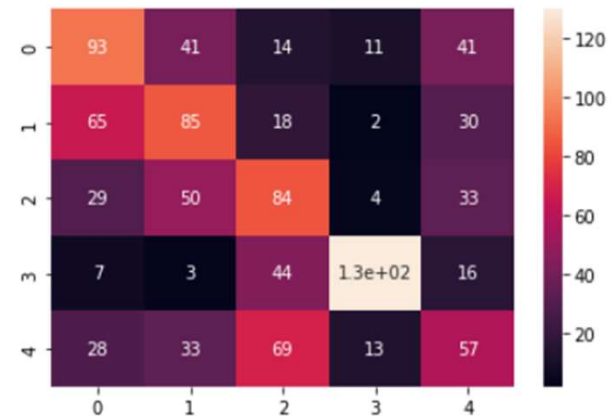
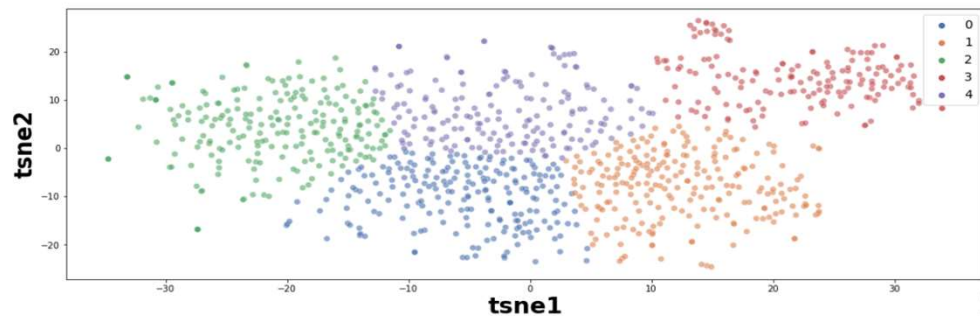


ARI : 0.15178503874306556

Accuracy = 0.449

f1-score = 0.4539195069594573

**TSNE selon les clusters**



# VOLET II : IMAGE CLASSIFICATION

3- classification par l'approche CNN :

3-1 Definition du modele VGG16 et elimination des layers OUTPUT

3-2 Extraction des descripteurs et keypoints

3-3 Reduction de dimension PCA

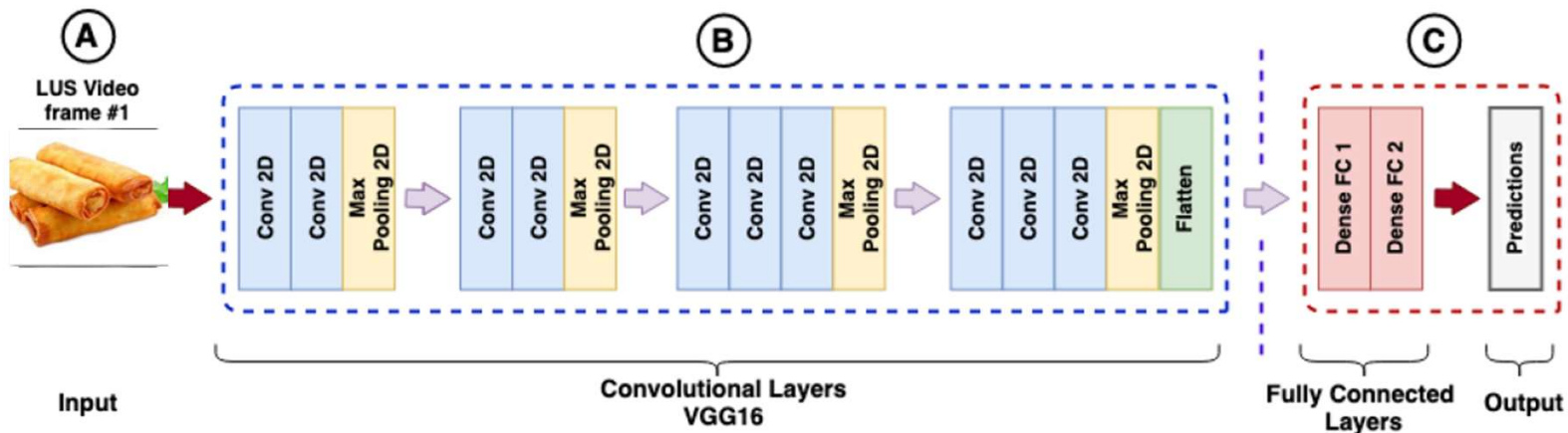
3-4 Reduction de dimension T-SNE

3-5 Visualisation des resultat en mettant en evidence les categories d'image

3-6 Klustering via kmeans de la matrice reduite T-SNE et calcul de l'accuracy\_score, f1-score et matrice de confusion

Architecture du modele VGG16 :

- ❑ VGG16 est un 'convolution neural network modele' tres utilise dans le traitement d'image .
- ❑ Layers "C" sont a eliminer pour travailler avec le modele comme feature extractor.





# VOLET II : IMAGE CLASSIFICATION

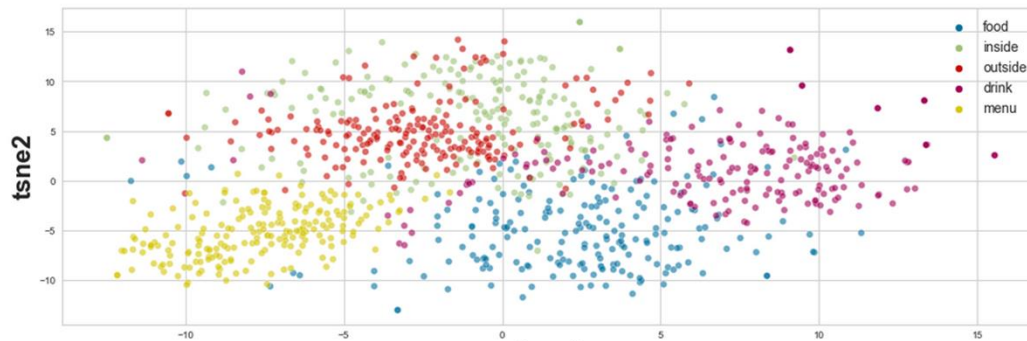
3- classification par l'approche CNN :

3-5 Visualisation des resultat en mettant en evidence les categories d'image

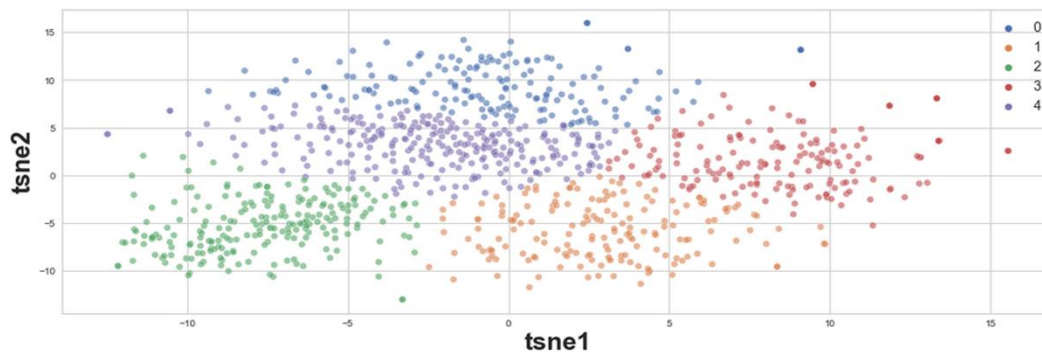
3-6 Klustering via kmeans de la matrice reduite T-SNE et calcul de l'accuracy\_score, f1-score et matrice de confusion

- ❑ On observe une meilleure distinction entre 'menu', 'drink' et plus ou moins 'food' pour le premier graphique
- ❑ Le score accuracy apres clustering est meilleur ainsi que la matrice de confusion qui semble assez reguliere, ce qui favorise l'utilisation d'un CNN pour adresser notre classification>

TSNE selon les vraies classes



TSNE selon les clusters



ARI : 0.5482373766061898

Accuracy = 0.773

f1-score = 0.7746161668687771



# VOLET III : YELP DATA EXTRACTION VIA API

- 1- keys and parameters setting
- 2- Get response from YELP database
- 3-response treatment and export to csv format

- ❑ Les data sont telechargees a partir du site YELP via l'API (yelp\_api) a travers laquelle on envoit une requete pour recevoir un fichier dict.
- ❑ Les 2 fichiers ont etes exportes sur format csv

Fichier business\_id :

df																	
		id	alias	name	image_url	is_closed	url	review_count	categories	rating	coordinates	transactions	price	location	phone	display_phone	distance
0	W0sXHSSpkiMEDJiBmSLNYQ	trestle-astoria	Trestle	media2.fl.yelpcdn.com/bphoto/JOqbgzUDibw5hLL5M2H24/0.jpg	https://s3-media2.fl.yelpcdn.com/bphoto/JOqbgzUDibw5hLL5M2H24/0.jpg	False	https://www.yelp.com/biz/trestle-astoria?adjust_creative=A57LwZ5W7bZbVIG9Tnw&utm_campaign=yelp_api_v3&utm_medium=api_v3_business_search&utm_source=A57LwZ5W7bZbVIG9Tnw	477	{'alias': 'cocktailbars', 'title': 'Cocktail Bars', 'alias': 'newamerican', 'title': 'American (New)', 'alias': 'breakfast_brunch', 'title': 'Breakfast & Brunch']}	4.5	{'latitude': 40.7606154, 'longitude': -73.9229562}	[delivery, restaurant_reservation, pickup]	\$\$	{'address1': '34-02 Broadway', 'address2': '', 'address3': None, 'city': 'Astoria', 'zip_code': '11106', 'country': 'US', 'state': 'NY', 'display_address': ['34-02 Broadway', 'Astoria, NY 11106']}	+13478080290	(347) 808-0290	700.500655

Fichier reviews :

	id	url	text	rating	time_created	user	business_id
0	8vxcG-frKT45XxKDdyRTjQ	https://www.yelp.com/biz/trestle-astoria?adjust...	This was my first time here at Trestle with my...	5	2022-07-08 21:55:51	{'id': 'wARPT6TBII02szkhhjufA', 'profile_url': ...}	W0sXHSSpkiMEDJiBmSLNYQ
1	gaOle31H1e3GL0Kpyf6Yaw	https://www.yelp.com/biz/trestle-astoria?adjust...	Nice vibes in here. I came on a humble. Alone ...	5	2022-07-24 04:35:49	{'id': 'QuWYxm3ij1Qsa9mUH371Xw', 'profile_url': ...}	W0sXHSSpkiMEDJiBmSLNYQ
2	vwHOi0_1T7EXZV1tDpa_uQ	https://www.yelp.com/biz/trestle-astoria?adjust...	Food was good and the seat booth was solid sea...	4	2022-07-23 18:08:53	{'id': 'PewlvbzvPSzVj2WnVZ7s4w', 'profile_url': ...}	W0sXHSSpkiMEDJiBmSLNYQ

Merci pour votre attention