



# Food2vec

## Machine learning for NLP

Quentin Navarre D.I.A. 1  
Groupe 11

# **Contexte du projet**

# Contexte du projet

- Le dataset utilisé pour ce projet provient du site [OpenFoodFacts](#)
  - Ce site peut être considéré comme un Wikipédia des produits alimentaires.
  - Son objectif est de partager avec les utilisateurs un maximum d'informations sur les compositions et catégories des produits.
- Le dataset fourni provient du lien suivant : [dataset](#)
  - Il est composé de :
    - 2,5 millions de lignes, chaque ligne correspond à un produit.
    - 196 colonnes, chaque colonne correspond à des caractéristiques du produit.
- Le code réalisé pour ce projet est disponible sur le lien : [Github](#)

# Objectifs

- L'objectif final du projet est de réaliser un clustering des produits en fonction de leurs similarités puis de les représenter sur une carte en 2 dimensions.
- Pour cela nous décomposons le travail nécessaire en **3 missions distinctes** :
  - 1.Réaliser une vectorisation des ingrédients qui composent les produits.
  - 2.Déterminer une méthode pour calculer la similarité des produits à partir de leurs ingrédients.
  - 3.Réaliser un clustering des produits puis les afficher sur une carte en 2 dimensions.



01

## Exploration des données

02

## Preprocessing

03

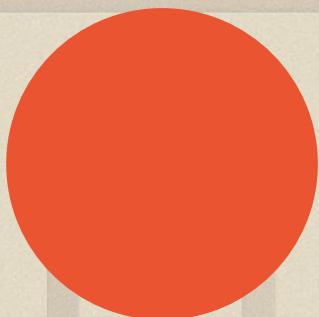
### Partie A

Vectorisation des ingrédients

04

### Partie B

Similarité des produits



05

### Partie C

Clustering et carte des produits

06

### Conclusion



01

# Exploration des données

# Sélection des colonnes

df.shape

(2625589, 196)

- Le dataset d'origine est composé de 2 6525 589 lignes et 196 colonnes.
- Chaque ligne correspond à un produit et ses caractéristiques.
- Nous retirons dans un premier temps en local les colonnes pour lesquelles nous sommes sûrs qu'elles ne nous serviront pas afin de ne garder que les suivantes :

	Shape du dataset : (2625589, 8)							
	product_name	brands	categories_en	ingredients_text	ingredients_tags	food_groups_en	main_category_en	image_url
0	jeunes pousses	endives		NaN	NaN	NaN	NaN	NaN
1	Andrè	Nan		NaN	NaN	NaN	NaN	NaN
2	L.casei	Nan		NaN Leche semidesnatada, azucar 6.9%, leche desnat...	en:semi-skimmed- milk,en:dairy,en:milk,en:sugar...	NaN	NaN	https://images.openfoodfacts.org/images/produc...
3	Skyr	Danone	Dairies,Desserts,Fermented foods,Fermented mil...	NaN	NaN	Milk and dairy products,Dairy desserts	Cream cheeses	https://images.openfoodfacts.org/images/produc...
4	Vitória crackers	Nan		NaN	NaN	NaN	NaN	https://images.openfoodfacts.org/images/produc...

# Sélection des colonnes

- Les colonnes qui vont nous intéresser pour la suite sont

```
check_nan(df)
```

```
Valeurs nan dans product_name : 93231
Valeurs nan dans brands : 1337721
Valeurs nan dans categories_en : 1537615
Valeurs nan dans ingredients_text : 1827462
Valeurs nan dans ingredients_tags : 1829055
Valeurs nan dans food_groups_en : 1746150
Valeurs nan dans main_category_en : 1537615
Valeurs nan dans image_url : 511421
```

```
check_unique(df)
```

```
Valeurs uniques dans product_name : 1595035
Valeurs uniques dans brands : 220324
Valeurs uniques dans categories_en : 93977
Valeurs uniques dans ingredients_text : 677599
Valeurs uniques dans ingredients_tags : 599501
Valeurs uniques dans food_groups_en : 46
Valeurs uniques dans main_category_en : 39222
Valeurs uniques dans image_url : 2113998
```

- **product\_name** : le nom du produit
- **Ingredients\_text** : la liste des ingrédients sous forme non normalisée
- **Ingredients\_tags** : la liste des ingrédients dans un format normalisé

La colonne ingredients\_tags semble à peu près autant remplie que ingredients\_text.

Il y a environ 1,6 millions de produits avec des noms différents.



# Observations

En comptant les ingrédients uniques à l'intérieur de la colonne ingredients\_text nous obtenons environ 1,9 millions de produits :

Nombre d'ingrédients uniques : 1903287

Si l'on décide d'en afficher certains voici ce que l'on obtient :

```
df_temp["ingredients_text_list"].unique()  
  
array(['nan', 'Leche semidesnatada', 'azucar 69%', ...,  
      'MAGERMLCHPULVER', 'VIANDE HACHEE PUR BOEUF 5% M',  
      'POURCENTAGE DE MATIERE GRASSE INFERIEUR A 5% RAPPORT COLLAGENE/PROTEINE VIANDE INFERIEUR A 12% VIANDE HACHEE DE BOEUF(*)'],  
      dtype=object)
```

Vu qu'il y a des valeurs nan, des majuscules, des minuscules, des nombres , des noms d'ingrédients très longs...

Il semble difficile d'exploiter cette colonne pour traiter les ingrédients des produits.  
Nous nous intéressons donc à la colonne ingredients\_tags



# Observations

Voici le format des données inscrites dans la colonne ingredients\_tags :

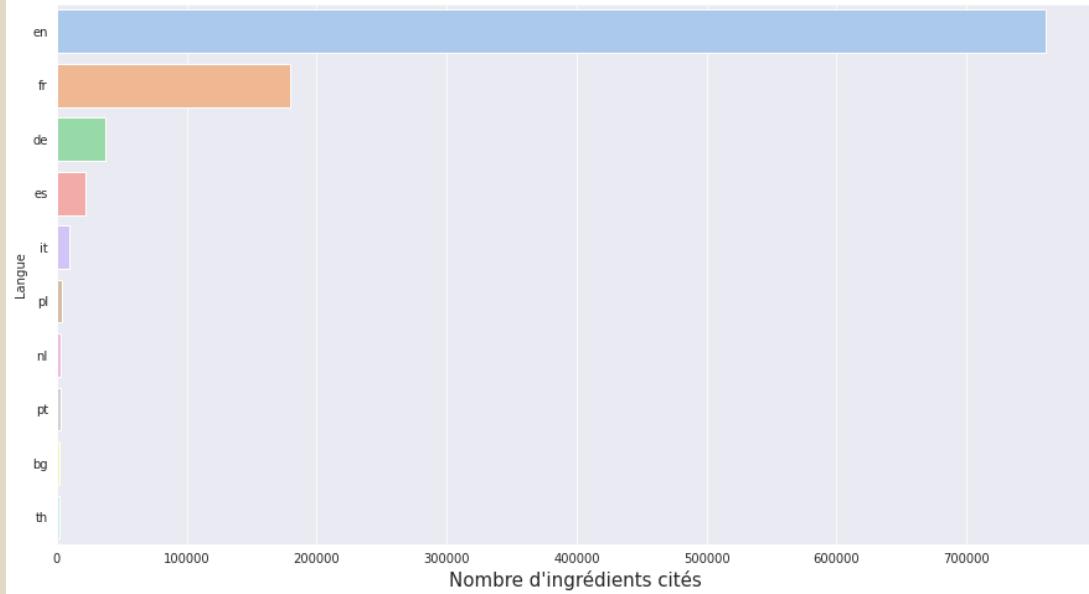
```
'en:semi-skimmed-milk,en:dairy,en:milk,en:sugar,en:added-sugar,en:disaccharide,es:leche-desnatada-en-polva,en:lactic-ferments,en:ferment,en:microbial-culture,en:vitamins,en:vitamin-b6,en:vitamin-d,es:contiene-lactobacidus-casei',  
'en:beta-alanine,en:creatine-hcl,en:ancient-peat-and-apple-extract,en:l-leucine,en:l-isoleucine,en:l-valine,en:betaine-anhydrous,en:arginine-silicate-inositol,en:cordyceps-militaris,en:ganoderma-lucidum,en:pleurotus-eryngii,en:shittake,en:mushroom,en:hericium-erinaceus,en:and-trametes-veriscolor,en:phyllanthus-embllica,en:extract,en:nattokinase,en:1000-fu,en:of-enzyme-activity,en:aframomum-melegueta,en:caffeine-anhydrous,en:methyliberine,en:theacrine,en:e330,en:natural-and-artificial-flavouring,en:flavouring,en:natural-flavouring,en:artificial-flavouring,en:e296,en:soluble-corn-fiber,en:cereal,en:corn,en:corn-fiber,en:e955,en:e950,en:e552,en:e51,en:fd-c-blue-lake-1,en:as-elevatp,en:as-nitrosigine,en:cordyceps,en:reishi,en:king-trumpet,en:shitake,en:lion-s-mane,en:tail,en:as-peako2,en:fruit,en:as-capros,en:as-nsk-sd40,en:fibronolytic-units,en:6-paradol,en:as-caloriburn-gp,en:as-dynamine,en:as-teacrine',
```

- Les données semblent plus exploitables.
- Chaque ingrédient est renseigné sous le format :
  - [Abréviation langue] : [nom de l'ingrédient dans la langue]
- Il peut y avoir plusieurs langues pour les ingrédients d'un seul produit.
- Les ingrédients peuvent apparaître plusieurs fois dans des langues différentes.



# Observations

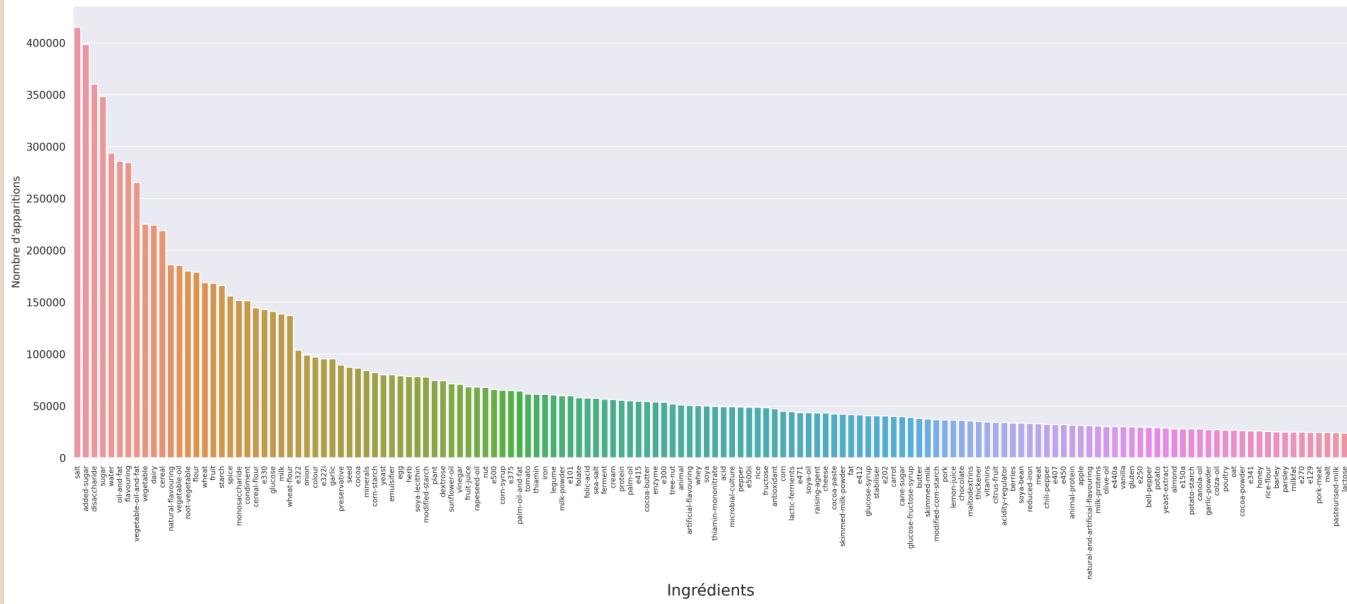
Nombre d'ingrédients cités par langue (Top 10)



Il y a une très forte majorité  
d'ingrédients renseignés en anglais.

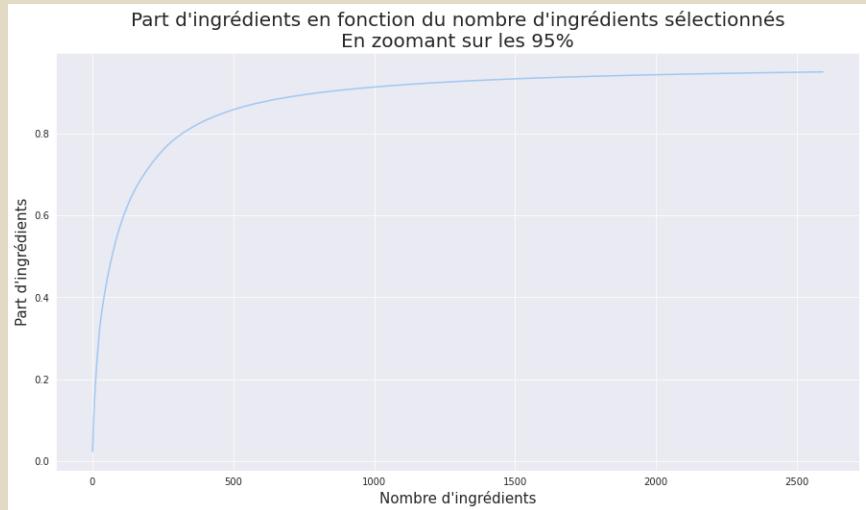
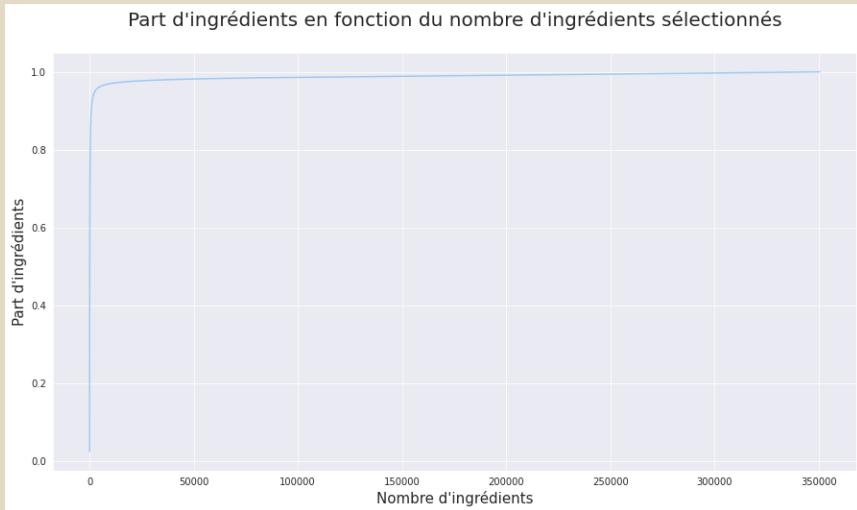
# Observations

Ingrédients avec le plus d'apparitions dans le dataset (Top 150)

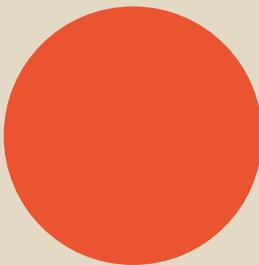


Certains ingrédients tels que le sel, le sucre et l'eau apparaissent un grand nombre de fois.

# Observations



Si l'on regroupe toutes les occurrences des ingrédients dans une seule liste, nous remarquons qu'en en sélectionnant un petit nombre nous pouvons couvrir une grande part de ceux utilisés dans les produits.



02

# Preprocessing

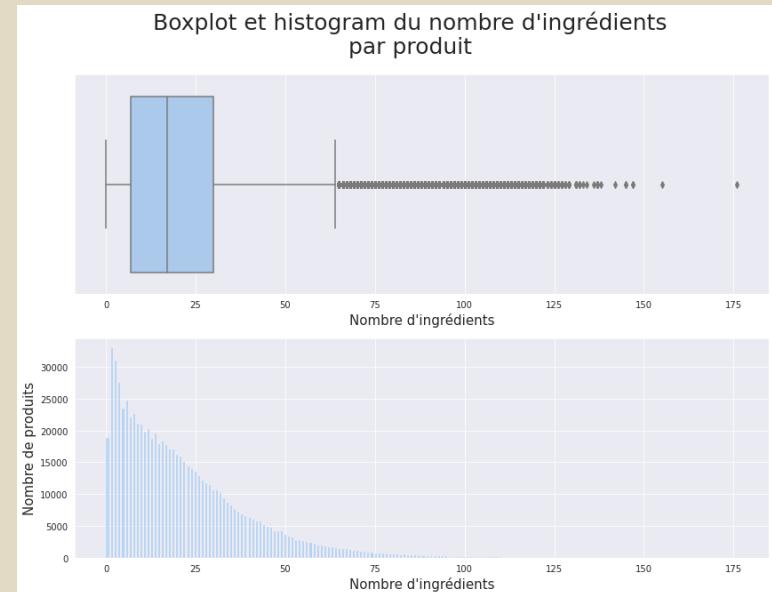
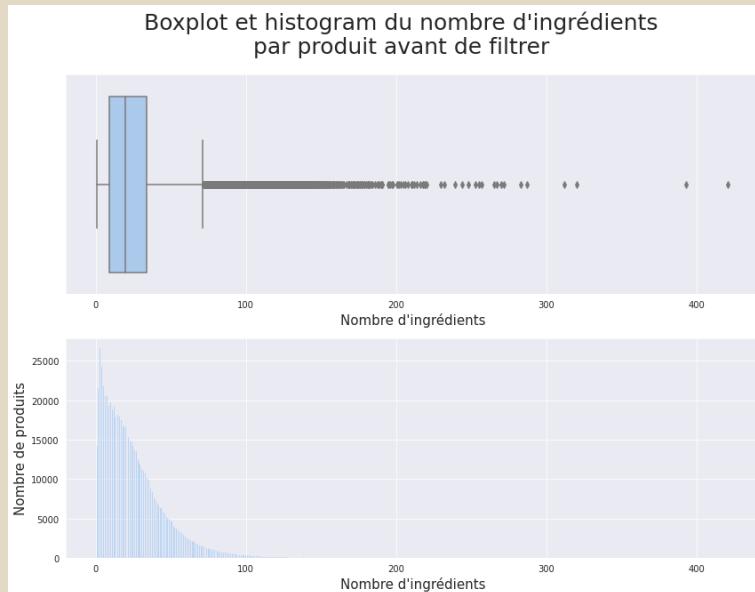
# Nettoyage et mise en forme

- Nous ne gardons finalement que les colonnes product\_name et ingredients\_tags
- Nous transformons ingredients\_tags dans un format exploitable puis nous filtrons sur les ingrédients anglais afin de ne pas avoir de redondance entre plusieurs langues.

	product_name	ingredients_tags	liste_ingredients
2	L.casei	{'en': ['semi-skimmed-milk', 'dairy', 'milk', ...}	[semi-skimmed-milk, dairy, milk, sugar, added-sugar, ...]
9	hyde icon	{'en': ['beta-alanine', 'creatine-hcl', 'ancient-peat-and...']}	[beta-alanine, creatine-hcl, ancient-peat-and-peat, ...]
28	Solène céréales poulet	{'en': ['antioxidant', 'colour', 'tomato', 've...']}	[antioxidant, colour, tomato, vegetable, mayonaise, ...]
36	Crème dessert chocolat	{'en': ['whole-milk', 'dairy', 'milk', 'sugar', ...]}	[whole-milk, dairy, milk, sugar, added-sugar, ...]
45	Baguette Poitevin	{'en': [None, 'water', 'salt', 'yeast', 'glute...']}	[None, water, salt, yeast, gluten, None, deact...]

- Nous gardons tous les ingrédients uniques nous permettant de couvrir 90 % de ceux utilisés, ce qui correspond à **812 ingrédients**.
- En procédant de la sorte, chaque ingrédient à au moins **1 500 occurrences** ce qui permet de retirer tous les ingrédients mal orthographiés.
- Nous filtrons sur les produits ayant au moins un ingrédient en anglais.

# Observations sur les ingrédients filtrés

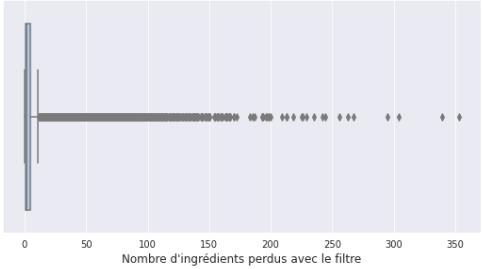


- La distribution des ingrédients par produits est moins étendue après avoir filtré les ingrédients.

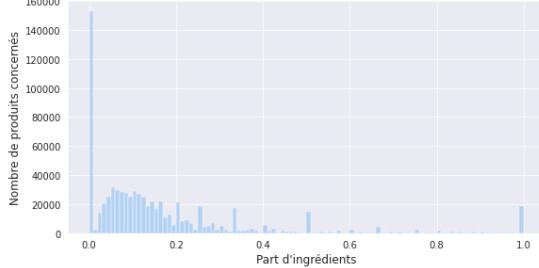
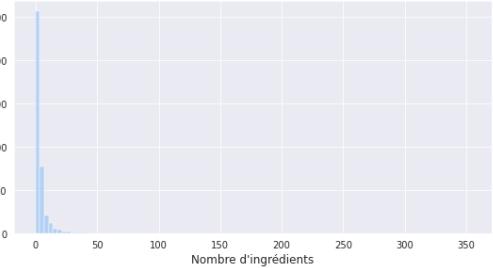
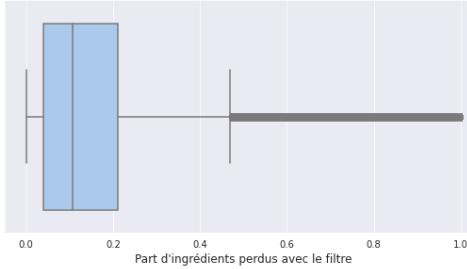


# Observations sur les ingrédients filtrés

Boxplot du nombre d'ingrédients perdus par produit



Boxplot de la part d'ingrédients perdus par produit



- Une très grande partie des produits n'a perdue aucun ingrédient avec le filtre mais nous devons garder en tête pour la suite que d'autres auront des compositions incomplètes ce qui pourrait fausser les mesures de similarités.
- Nous décidons de retirer les produits ayant perdu plus de 50% de leurs ingrédients et ceux qui ont perdu 0 ou 1 seul ingrédient.
- Sur les 2 600 000 produits initiaux il nous en reste 691 737.

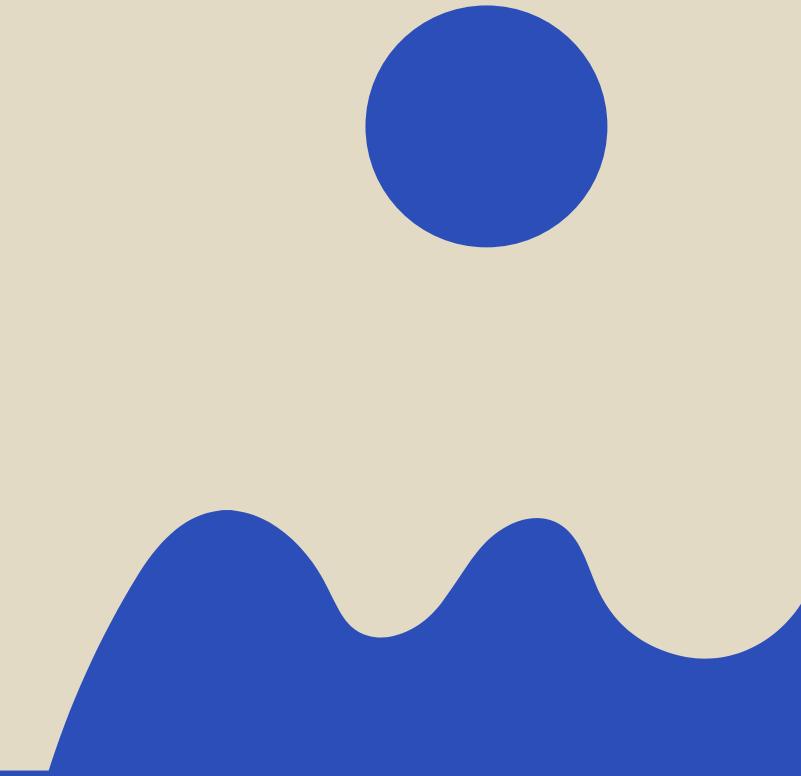
```
df_size_bf= len(df)
df=df[(df["part_ingredients_perdus"]<=0.5) & (df["nombre_ingredients"]!=0) & (df["nombre_ingredients"]!=1)]
print("Nous avons retiré "+str(df_size_bf - len(df))+" produits sur "+str(df_size_bf)+", il nous en reste donc "+str(len(df))+".")
```

Nous avons retiré 69123 produits sur 760860, il nous en reste donc 691737.

# 03

## Partie A

Vectorisation des ingrédients et  
représentation en 2 dimensions



# Vectorisation des ingrédients

- Entraînement d'un modèle Word2Vec :

```
#Entraînement du model
model = Word2Vec(sentences=df["liste_ingredients"], size=15, workers=4, seed =42, sg=1)
model.wv.save(model_path+model_name+".wordvectors")
```

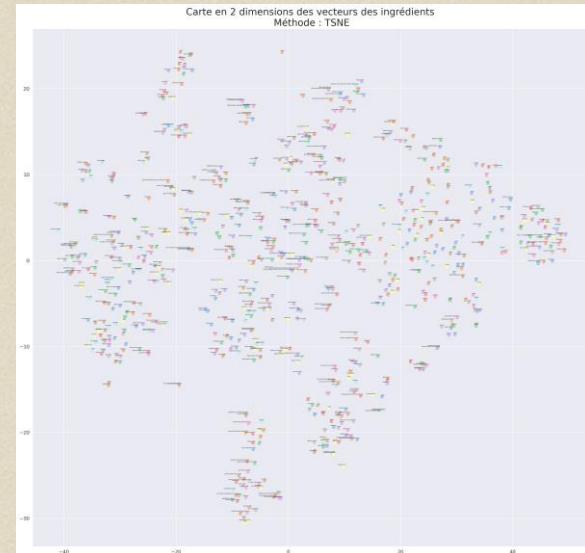
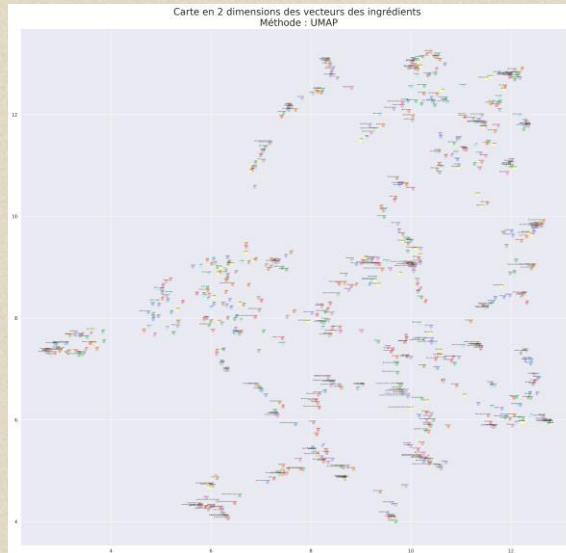
- Nous avons 812 ingrédients ce qui est peu par rapport à la taille d'un vocabulaire de texte usuel pouvant faire plusieurs dizaines de milliers de mots uniques.
- Nous choisissons donc arbitrairement de prendre une taille d'embedding plutôt petite d'une quinzaine de dimensions.
- Après plusieurs essais (notamment du modèle CBOW ) nous gardons le modèle skip-gram sans préciser de distance de fenêtre maximale car l'ordre des tokens n'a pas d'importance dans notre cas.

ingredient	dim_1	dim_2	dim_3	dim_4	dim_5	dim_6	dim_7	dim_8	dim_9	dim_10	dim_11	dim_12	dim_13	dim_14	dim_15
semi-skimmed-milk	-1.356199	-0.269292	0.585668	0.283947	0.373141	-0.232849	0.566497	0.087994	0.008522	-0.107520	0.637058	-1.027239	0.486392	-0.571366	0.157028
dairy	-1.003779	-0.649851	0.203489	0.508861	0.582758	0.403989	0.434332	-0.595374	-0.214006	-0.183570	0.393287	-0.883933	0.230411	-0.335802	0.419919
milk	-1.130293	-0.664711	0.288226	0.490643	0.417961	0.113500	0.415383	-0.514224	0.177608	0.098012	0.270121	-0.669801	0.177212	-0.451612	0.011921
sugar	0.166919	-0.380515	-0.082900	0.435718	0.588095	-0.178151	0.520020	-0.368368	-0.041126	-0.556826	0.283092	-0.363201	-0.078687	-0.352442	0.648472
added-sugar	0.198725	-0.377257	-0.070209	0.302833	0.552530	-0.235795	0.601683	-0.458263	-0.052005	-0.627520	0.307685	-0.325914	-0.070757	-0.317077	0.725126

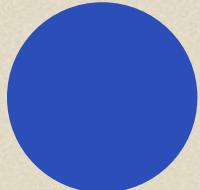
# Réduction en 2 dimensions :

- Nous essayons trois techniques différentes pour réduire les vecteurs en 2 dimensions :
  - Analyse en composante principale
  - UMAP
  - T-SNE

Voici les résultats :



Nous décidons d'utiliser la réduction UMAP pour la carte finale car je trouve que ses clusters plus espacés la rendent plus lisible.



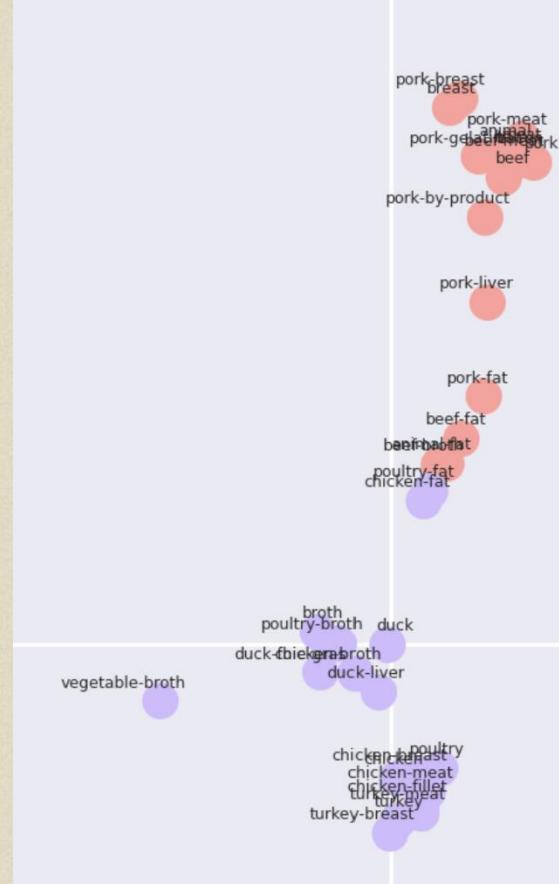
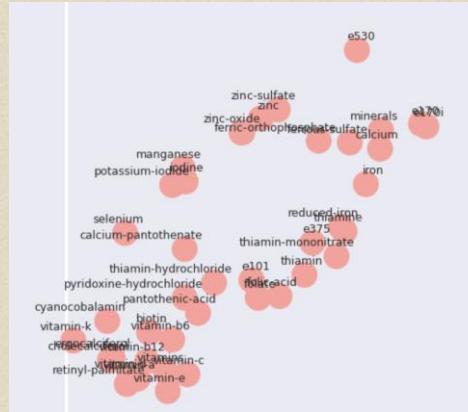
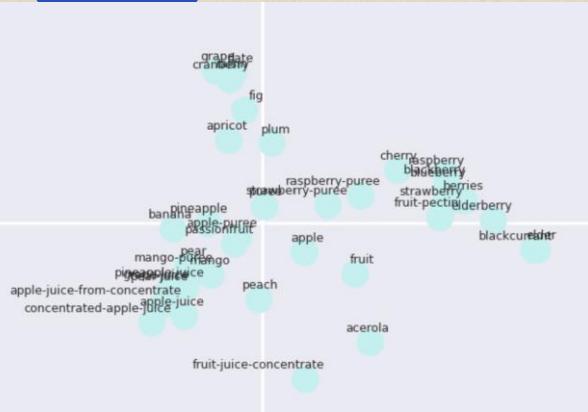
# Résultats

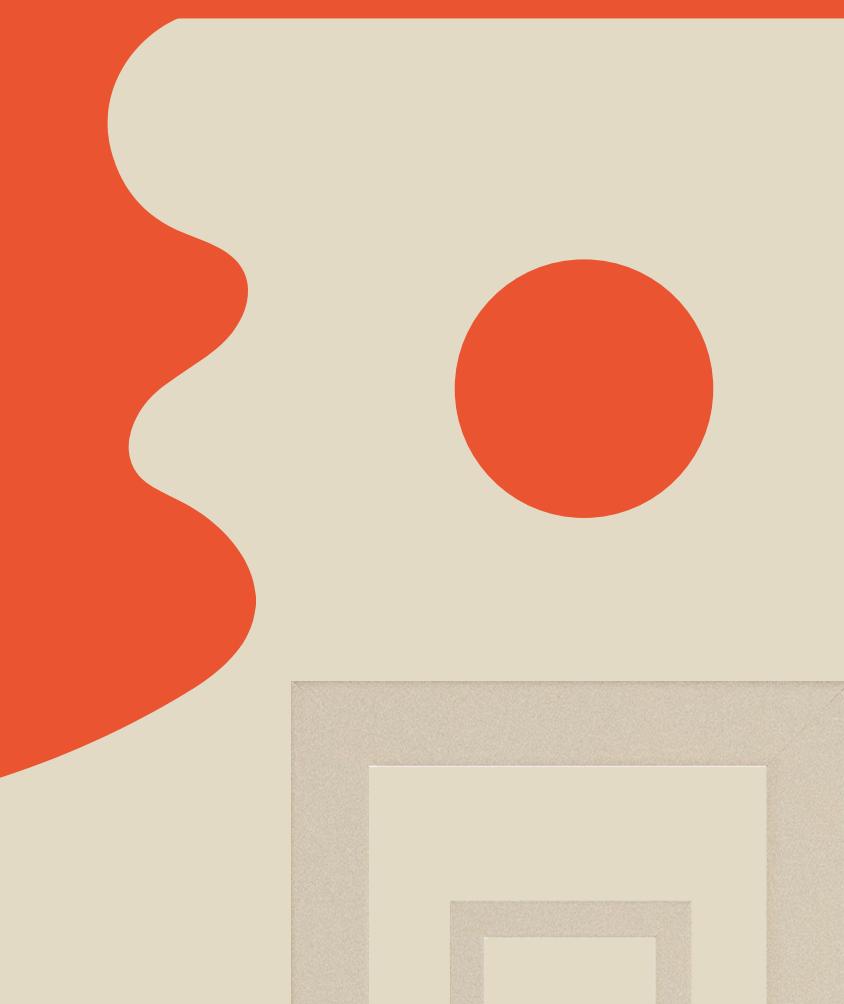
Nous réalisons un clustering Kmeans au dessus de la réduction en 2 dimensions pour l'aspect esthétique.

Carte en 2 dimensions des vecteurs des ingrédients avec leurs clusters  
Méthodes : UMAP, Kmeans(30 clusters)



# Zoom sur certains clusters





04

## Partie B

Comparaison des produits

# Problématique

- Maintenant que nous avons réalisé la vectorisation des ingrédients, nous souhaitons pouvoir mesurer la similarité des produits entre eux.
- Pour mesurer la similarité entre 2 vecteurs nous pouvons utiliser la similarité cosinus :

$$\text{similarity}(A, B) = \frac{A \cdot B}{\|A\| \times \|B\|} = \frac{\sum_{i=1}^n A_i \times B_i}{\sqrt{\sum_{i=1}^n A_i^2} \times \sqrt{\sum_{i=1}^n B_i^2}}$$

- Chaque produit peut être composé d'un nombre d'ingrédients différent, donc d'un nombre de vecteurs différent.
- Nous n'avons donc pas les mêmes dimensions pour tous les produits.



# Moyenne des vecteurs des produits

- Pour ramener tous les produits aux mêmes dimensions nous pourrions réaliser la moyenne des vecteurs des ingrédients qui les composent et par la suite calculer la similarité cosinus entre ces vecteurs moyens.
- Le problème avec cette méthode est que l'on perd beaucoup d'informations lorsque l'on fait la moyenne.
- Nous pouvons l'illustrer ce problème avec l'exemple suivant :
- **Composition du produit A** : [[ -100,-100] , [100, 100]],
  - moyenne de A : [0,0]
- **Composition du produit B** : [[0,0]] ,
  - moyenne de B : [0,0]
- **Similarité(moyenne\_A, moyenne\_B) = 1**
- Nous obtenons une similarité de 1 alors que l'on voit bien que les compositions des 2 produits ne sont pas du tout semblables.



# Première solution envisagée

- Pour comparer les produits entre eux nous décidons donc de comparer les ingrédients les composants 1 à 1.
- Algorithme utilisé :
  1. Pour chaque ingrédient du produit A , nous calculons sa similarité avec tous les ingrédients du produit B.
  2. Nous gardons ensuite la similarité maximale trouvée pour chaque ingrédient du produit A.
  3. Puis nous faisons la moyenne de ses similarités pour obtenir la similarité de A et B.

# Première solution envisagée

- Cette méthode peut s'apparenter à une moyenne de max poolings :
  - Composition du produit A = [a,b,c,d]
  - Composition du produit B = [e,f,g,h]

	e	f	g	h
a	Sim(a,e)	Sim(a,f)	Sim(a,g)	Sim(a,h)
b	Sim(b,e)	Sim(b,f)	Sim(b,g)	Sim(b,h)
c	Sim(c,e)	Sim(c,f)	Sim(c,g)	Sim(c,h)
d	Sim(d,e)	Sim(d,f)	Sim(d,g)	Sim(d,h)

Calcul des similarités

0	0.5	<b>0.7</b>	0.2
0.3	0	<b>0.9</b>	0.5
0.2	0	<b>1</b>	0.5
<b>0.5</b>	0.3	0	0.2

Max Pooling

0.7
0.9
1
0.5

Moyenne

0.775

# Implémentation

- Pour ne pas avoir à calculer les similarités des ingrédients à chaque fois et ainsi améliorer la vitesse d'exécution du programme, nous calculons à l'avance la matrice des similarités par paire d'ingrédients des produits (nommée **df\_similarities** dans le code):

Matrice des similarités des ingrédients

semi-skimmed-milk	dairy	milk	sugar	added-sugar	disaccharide	lactic-ferments	ferment	microbial-culture	vitamins	...
semi-skimmed-milk	1.000000	0.812481	0.856659	0.408533	0.374410	0.395696	0.825005	0.791201	0.761323	0.399250
dairy	0.812481	1.000000	0.922171	0.631504	0.590832	0.624127	0.630645	0.598705	0.574923	0.252504
milk	0.856659	0.922171	1.000000	0.466385	0.417522	0.456121	0.755143	0.738621	0.719190	0.341347
sugar	0.408533	0.631504	0.466385	1.000000	0.989781	0.998432	0.284654	0.287253	0.249729	0.139359
added-sugar	0.374410	0.590832	0.417522	0.989781	1.000000	0.989594	0.253843	0.269749	0.223360	0.150211

Similarité du produit A avec B

```
def similarity(A,B):
    return df_similarities.loc[A, B].max(axis=1).mean()
```

A et B sont les listes d'ingrédients qui composent les produits pour lesquels on calcule la similarité.

# Résultats obtenus

Résultats obtenus lorsque l'on calcule les similarités du produit « Crème dessert chocolat » avec les autres produits :

```
df_test = find_similar_products_max_pooling(df,"Crème dessert chocolat")
df_test.sort_values("similarity",ascending=False).head(15)
```

	product_name	liste_ingredients	similarity
200535	Profiteroles	[dairy, cream, sugar, added-sugar, disaccharid...	1.000000
617940	Génoise au cacao fourrée au praliné	[sugar, added-sugar, disaccharide, whole-milk,...	1.000000
621829	Tourte nougat	[sugar, added-sugar, disaccharide, whole-milk,...	1.000000
268399	Gourmet cupcakes	[icing-sugar, added-sugar, disaccharide, sugar...	1.000000
629896	La recova	[dairy, milk, condensed-milk, sweetened-conden...	1.000000
992	Chocolate Bites	[sugar, added-sugar, disaccharide, wheat-flour...	1.000000
621850	Tourte au Nougat	[whole-milk, dairy, milk, sugar, added-sugar, ...	1.000000
44411	Frosted Cookies	[flour, vegetable-fat, oil-and-fat, vegetable-...	1.000000
468910	6 Macarons gourmands	[sugar, added-sugar, disaccharide, nut, tree-n...	1.000000
24056	Lunds & byerlys, shortcake, lemon & cream	[lemon, fruit, citrus-fruit, sugar, added-suga...	1.000000
24028	Lunds & byerlys, yule log	[sugar, added-sugar, disaccharide, dark-chocol...	1.000000
46309	Moist Chocolate Cupcakes Fillet With Creamy Fu...	[sugar, added-sugar, disaccharide, icing-sugar...	1.000000
654445	profiteroles	[chocolate, skimmed-milk, dairy, milk, sugar, ...	0.998821
629897	Junior Blanco	[dairy, milk, condensed-milk, sweetened-conden...	0.998821
629880	Alfajor de chocolate blanco	[dairy, milk, condensed-milk, sweetened-conden...	0.998821

- Nous remarquons que les résultats ne sont pas concluants. En effet les produits avec beaucoup d'ingrédients sont très avantagés avec cette méthode.
- Comme on ne garde que la similarité maximale il n'y a pas de pénalisation pour les ingrédients de B n'étant pas dans A.
- Ce problème vient du fait que les produits n'ont pas le même nombre d'ingrédients.
- Un autre inconvénient est que **la similarité de A dans B n'est pas la même que celle de B dans A**, le calcul de cette mesure perd donc tout son sens.

```
print('Similarité Max pooling des ingrédients de A dans B : ',similarity_max_A_dans_B(A,B))
print('Similarité Max pooling des ingrédients de B dans A : ',similarity_max_A_dans_B(B,A))

Similarité Max pooling des ingrédients de A dans B : 1.0
Similarité Max pooling des ingrédients de B dans A : 0.6262173263633892
```

# Solution finale

- Pour remédier à ce problème nous gardons la même fonction mais cette fois nous calculons également la similarité de B dans A. Nous faisons ensuite la moyenne des 2 similarités calculées :

## Calcul des similarités

	e	f	g	h	i
a	0.5	0	0.5	0.7	0.2
b	0.1	0.3	0	0.9	0.5
c	0.6	0.2	0	1	0.5
d	0.4	0.5	0.3	0	0.2

## Max Pooling

0.7
0.9
1
0.5

## Moyenne

0.775

## Max Pooling

0.6	0.5	0.5	1	0.5
-----	-----	-----	---	-----

## Moyenne

0.698

## Moyenne

0.62

- $A = [a,b,c,d]$
- $B = [e,f,g,h,i]$
- $\text{Sim}(A,B) = 0.698$

= Similarité finale

# Résultats obtenus

Résultats obtenus lorsque l'on calcule les similarités du produit « Crème dessert chocolat » avec les autres produits :

```
%%time
df_test = find_similar_products_max_pooling_both_ways(df, "Crème dessert chocolat")
df_test.head(10)
```

```
100%|██████████| 701084/701084 [36:06<00:00, 323.64it/s]
CPU times: user 34min 9s, sys: 49.1 s, total: 34min 58s
Wall time: 36min 7s
```

	product_name	liste_ingredients	similarity1	similarity2	mean_similarity
502997	Creme dessert artisanale	[dairy, milk, whole-milk, sugar, added-sugar, ...]	0.975463	0.966879	0.971171
355916	Crème dessert	[dairy, milk, pasteurised-milk, whole-milk, ca...	0.978254	0.958944	0.968599
536327	Délices de lait - Crème dessert Cacao	[whole-milk, dairy, milk, sugar, added-sugar, ...]	0.980800	0.952819	0.966810
440923	Crème dessert au chocolat	[whole-milk, dairy, milk, sugar, added-sugar, ...]	0.975463	0.957325	0.966394
528512	Landliebe Sahne Pudding Dunkle Schokolade	[whole-milk, dairy, milk, modified-starch, sta...	0.945702	0.984950	0.965326
677269	Ovocný košík s kousky jahod	[milk, dairy, milk-powder, sugar, added-sugar, ...]	0.962020	0.966869	0.964444
447631	Lactel max bio chocolat	[semi-skimmed-milk, dairy, milk, rice-starch, ...]	0.969071	0.959748	0.964410
500772	Crèmes dessert chocolat	[dairy, milk, pasteurised-milk, whole-milk, su...	0.945702	0.976264	0.960983
425199	P'tit Goûter au lait cacao	[whole-milk, dairy, milk, cane-sugar, added-su...	0.951536	0.966104	0.958820
424994	P'tit Gouter Au Lait chocolat	[whole-milk, dairy, milk, cane-sugar, added-su...	0.951536	0.966104	0.958820

- Les résultats affichés sont triés par ordre décroissant.
- Ils semblent bien plus cohérents avec cette nouvelle méthode.
- Le dataset comprenant 700 000 produits le temps d'exécution est très lent (36 minutes).

# Optimisation de la fonction

- Pour améliorer l'efficacité de la fonction nous n'utilisons plus un dataframe pour récupérer les coefficients de similarité entre les ingrédients mais une liste de dictionnaires où chaque dictionnaire contient les similarités d'un ingrédient de A avec les 812 ingrédients du dataset.
- Nous exécutons également cette fonction sur tous les produits du dataframe en multiprocessing afin de réduire le temps d'exécution.

```
def similarities_both_ways(A,B, dicts_A) :  
    list_vects=[]  
    for dico in dicts_A :  
        list_vects.append(np.array(operator.itemgetter(*B)(dico)))  
    return np.array(list_vects).max(axis=1).mean(), np.array(list_vects).max(axis=0).mean()# similarité A dans B , B dans A  
  
def find_similar_products_max_pooling_both_waysV3_multiprocessing(df_products, product_name):  
    liste_ingredients = df_products.loc[df_products["product_name"]==product_name, "liste_ingredients"].iloc[0]  
  
    df_temp = df_products[df_products["product_name"]!=product_name]# On retire le produit testé de la liste  
    list_dicts = [df_similarities[ingredient].to_dict() for ingredient in liste_ingredients]  
  
    with mp.Pool(mp.cpu_count()) as pool :  
        df_temp[["similarity1","similarity2"]] = pool.starmap(similarities_both_ways, zip(repeat(liste_ingredients),df_temp["liste_ingredients"], repeat(list_dicts)))  
  
    df_temp["mean_similarity"] = (df_temp["similarity1"] + df_temp["similarity2"])/2  
  
    return df_temp[["product_name","liste_ingredients","similarity1","similarity2","mean_similarity"]].sort_values("mean_similarity",ascending=False)
```

# Optimisation de la fonction

- Cela nous permet de réduire le temps d'exécution de 36 minutes à 1 minute et 15 secondes pour le même produit :

<pre>%%time df_test =find_similar_products_max_pooling_both_waysV3_multiprocessing(df, "Crème dessert chocolat") df_test.head(10)</pre>					
CPU times: user 15.1 s, sys: 2.88 s, total: 18 s Wall time: 1min 14s					
	product_name	liste_ingredients	similarity1	similarity2	mean_similarity
497444	Creme dessert artisanale	[dairy, milk, whole-milk, sugar, added-sugar, ...]	0.978151	0.968251	0.973201
352550	Crème dessert	[dairy, milk, pasteurised-milk, whole-milk, ca...	0.980159	0.960495	0.970327
436326	Crème dessert au chocolat	[whole-milk, dairy, milk, sugar, added-sugar, ...]	0.978151	0.960278	0.969215
530200	Délices de lait - Crème dessert Cacao	[whole-milk, dairy, milk, sugar, added-sugar, ...]	0.983300	0.953759	0.968530
442920	Lactel max bio chocolat	[semi-skimmed-milk, dairy, milk, rice-starch, ...]	0.972582	0.963024	0.967803
522517	Landliebe Sahne Pudding Dunkle Schokolade	[whole-milk, dairy, milk, modified-starch, sta...]	0.945112	0.985059	0.965086
668393	Ovocný košík s kousky jahod	[milk, dairy, milk-powder, sugar, added-sugar,...]	0.961873	0.965706	0.963790
495263	Crèmes dessert chocolat	[dairy, milk, pasteurised-milk, whole-milk, su...	0.945112	0.975642	0.960377
535656	Schoko Pudding	[whole-milk, dairy, milk, added-sugar, disacch...	0.957248	0.962144	0.959696
420849	P'tit Goûter au lait cacao	[whole-milk, dairy, milk, cane-sugar, added-su...	0.953359	0.965138	0.959248

# Exemples sur d'autres produits

- Les exemples suivants ont été calculés sur le dataset contenant 700 000 produits:

Produits similaires au produit : Lowes foods, macaroni & cheese					
	product_name	liste_ingredients	similarity1	similarity2	mean_similarity
11169	Macaroni & cheese dinner	[wheat-flour, cereal, flour, wheat, cereal-flo...	1.000000	0.995470	0.997735
299516	Lowes foods, macaroni spirals & cheese	[wheat-flour, cereal, flour, wheat, cereal-flo...	1.000000	0.995470	0.997735
10993	Dinosaurs macaroni & cheese dinner'	[wheat-flour, cereal, flour, wheat, cereal-flo...	1.000000	0.995470	0.997735
11172	Spiral dinner macaroni & cheese	[wheat-flour, cereal, flour, wheat, cereal-flo...	1.000000	0.995470	0.997735
10580	Macaroni & cheese dinner	[wheat-flour, cereal, flour, wheat, cereal-flo...	1.000000	0.995470	0.997735
247591	Macaroni and cheese dinner	[wheat-flour, cereal, flour, wheat, cereal-flo...	0.998796	0.995323	0.997060
246448	Macaroni and cheese dinner	[wheat-flour, cereal, flour, wheat, cereal-flo...	0.998796	0.995323	0.997060
195986	Market pantry, macaroni & cheese dinner	[wheat-flour, cereal, flour, wheat, cereal-flo...	0.998796	0.995323	0.997060
6457	Macaroni & cheese dinner	[wheat-flour, cereal, flour, wheat, cereal-flo...	0.998796	0.994027	0.996411
5921	Mac and cheese	[wheat-flour, cereal, flour, wheat, cereal-flo...	0.998796	0.994027	0.996411
5924	Kroger, macaroni & cheese	[e375, reduced-iron, minerals, iron, thiamin-m...	0.989126	0.993840	0.991483
315181	Macaroni and cheese	[dairy, flour, cheddar, cheese, salt, modified...	0.984869	0.979550	0.982209

Produits similaires au produit : Lempari					
	product_name	liste_ingredients	similarity1	similarity2	mean_similarity
596836	Flatbread Street Food	[wheat-flour, cereal, flour, wheat, cereal-flo...	0.976443	0.961098	0.968771
346115	Stenovns Ciabatta Stykker	[wheat-flour, cereal, flour, wheat, cereal-flo...	0.924376	0.998435	0.961405
596688	Relyt taysjyvä	[whole-wheat-flour, cereal, flour, wheat, cere...	0.943876	0.976220	0.960048
322985	pain spécial	[wheat-flour, cereal, flour, wheat, cereal-flo...	0.951694	0.965241	0.958467
450540	Pain spécial	[wheat-flour, cereal, flour, wheat, cereal-flo...	0.951694	0.965241	0.958467
596689	Reili	[wheat-flour, cereal, flour, wheat, cereal-flo...	0.921169	0.995370	0.958269
408430	Boule Tranchée Complète	[wheat-flour, cereal, flour, wheat, cereal-flo...	0.948029	0.964870	0.956449
391117	Demi baguettes précuites complètes	[wheat-flour, cereal, flour, wheat, cereal-flo...	0.943125	0.958853	0.950989
653834	Mollete esfío andaluz	[wheat-flour, cereal, flour, wheat, cereal-flo...	0.930122	0.970040	0.950081
596817	Vehna pahto	[wheat-flour, cereal, flour, wheat, cereal-flo...	0.921169	0.978860	0.950014
482122	Le petit paillasse du larzac la pièce de 270 gr	[wheat-flour, cereal, flour, wheat, cereal-flo...	0.933377	0.962435	0.947906
482117	Galzin Le pavé du Larzac	[wheat-flour, cereal, flour, wheat, cereal-flo...	0.933377	0.962435	0.947906

Produits similaires au produit : Tomato Ketchup					
	product_name	liste_ingredients	similarity1	similarity2	mean_similarity
678646	Ketchup à la tomate	[tomato, vegetable, vinegar, sugar, added-suga...	1.0	1.0000	1.000000
875766	Hoi Ketchup	[tomato, vegetable, vinegar, sugar, added-suga...	1.0	1.0000	1.000000
478925	Ketchup à la tomate	[tomato, vegetable, vinegar, sugar, added-suga...	1.0	1.0000	1.000000
478937	Tomato ketchup	[tomato, vegetable, vinegar, sugar, added-suga...	1.0	1.0000	1.000000
220	Tomato Ketchup Heinz Ouverture En Bas	[tomato, vegetable, vinegar, sugar, added-suga...	1.0	1.0000	1.000000
675564	Ketchup à la tomate	[tomato, vegetable, vinegar, sugar, added-suga...	1.0	1.0000	1.000000
675701	Heinz Tomato Ketchup	[tomato, vegetable, vinegar, sugar, added-suga...	1.0	1.0000	1.000000
560602	Emblématique ketchup HEINZ	[tomato, vegetable, vinegar, sugar, added-suga...	1.0	1.0000	1.000000
647777	Tomato Ketchup (offre Découverte)	[tomato, vegetable, vinegar, sugar, added-suga...	1.0	1.0000	1.000000
675645	Tomato ketchup	[tomato, vegetable, vinegar, sugar, added-suga...	1.0	1.0000	1.000000
563248	Tomato Ketchup (20% extra free)	[tomato, vegetable, alcohol-vinegar, vinegar, ...	1.0	0.9983	0.99915
675629	Tomato ketchup	[tomato, vegetable, alcohol-vinegar, vinegar, ...	1.0	0.9983	0.99915

Produits similaires au produit : Mor Braz Bio Blonde (5%)					
	product_name	liste_ingredients	similarity1	similarity2	mean_similarity
578098	Bière belge Victoria blonde	[water, hops, plant, cereal, yeast]	1.0	1.000000	1.000000
489739	Briarde - Ambrée	[cereal, water, hops, plant, yeast]	1.0	1.000000	1.000000
501947	On the top	[water, malt, cereal, wheat, hops, plant, yeast]	1.0	0.969785	0.984892
568604	Hoppel Hammer	[malt, cereal, wheat, hops, plant, water, yeast]	1.0	0.969785	0.984892
539490	Urstrom	[water, malt, cereal, yeast, hops, plant]	1.0	0.967949	0.983974
500916	L'Eurélienne Blanche	[water, malt, cereal, hops, plant, yeast]	1.0	0.967949	0.983974
498236	Lager des étoiles	[water, malt, cereal, hops, plant, yeast]	1.0	0.967949	0.983974
498237	Gens de la Lune	[water, malt, cereal, hops, plant, yeast]	1.0	0.967949	0.983974
597438	Terapia Platin - Bere albă nefiltrată	[water, hops, plant, malt, cereal, yeast]	1.0	0.967949	0.983974
485111	Gallia Brut IPA	[water, malt, cereal, hops, plant, yeast]	1.0	0.967949	0.983974
500913	L'Eurélienne Blonde	[water, malt, cereal, hops, plant, yeast]	1.0	0.967949	0.983974
498311	Hegoa	[water, malt, cereal, hops, plant, yeast]	1.0	0.967949	0.983974

# Exemples sur d'autres produits

- Les exemples suivants ont été calculés sur le dataset contenant 700 000 produits:

Produits similaires au produit : Légumes vapeur assaisonnés - Trio de haricots et poivrons					
	product_name	liste_ingredients	similarity1	similarity2	mean_similarity
472770	Légumes vapeur assaisonnés - Sélection de 4 légumes	[garden-peas, legume, pea, green-peas, green-bean, vegetable, bell-pepper, ...]	0.977088	0.983856	0.980472
394991	Légumes Vapeur haricots beurre et plat et poivron	[legume, green-bean, vegetable, bell-pepper, ...]	0.924962	1.000000	0.962481
457723	Légumes Méditerranéens	[water, extra-virgin-olive-oil, oil-and-fat, vegetable, ...]	0.915054	1.000000	0.957527
394992	Légumes Vapeur petits pois, haricots verts et poivron	[green-bean, legume, pea, water, extra-virgin-olive-oil, oil-and-fat, ...]	0.914918	0.998499	0.956708
348052	Mélange de légumes et pommes de terre surgelé	[vegetable, root-vegetable, carrot, legume, green-peas, ...]	0.939927	0.955844	0.947885
404714	Purée de Céleri - Surgelé	[celery, vegetable, water, butterfat, dairy, oil-and-fat, ...]	0.895610	1.000000	0.947805
374229	Les légumes à la Printanière	[vegetable, root-vegetable, onion, butterfat, ...]	0.927186	0.966873	0.947030
394990	Légumes Vapeur chou fleur, chou romanesco, brocoli et poivron	[vegetable, cauliflower, root-vegetable, carrots, ...]	0.901705	0.988765	0.944235
363015	Poêlée ratatouille cuisinée	[vegetable, water, tomato-concentrate, tomato-sauce, ...]	0.901893	0.981897	0.941895
388179	Petits Mélanges vapeur	[vegetable, broccoli, cauliflower, water, extract, ...]	0.894259	0.987462	0.940861
568783	nan	[tomato, vegetable, water, olive-oil, oil-and-fat, ...]	0.919637	0.955271	0.937454
307091	Fine green beans	[green-peas, legume, pea, red-bell-pepper, vegetable, ...]	0.937566	0.936276	0.936921

Produits similaires au produit : Procacci Brothers, Italian Chestnuts					
	product_name	liste_ingredients	similarity1	similarity2	mean_similarity
379585	Châtaignes entières	[nut, tree-nut]	1.0	1.0	1.0
354744	Marrons Entiers Sous Vide, à L'étoffée	[nut, tree-nut]	1.0	1.0	1.0
381317	Marronen	[nut, tree-nut]	1.0	1.0	1.0
583067	Erdmandel Mehl	[nut, tree-nut]	1.0	1.0	1.0
230027	Elizabeth's naturals, raw macadamia nuts	[nut, tree-nut]	1.0	1.0	1.0
381316	Ponthier : Gekochte Maronen	[nut, tree-nut]	1.0	1.0	1.0
381315	Marrons cuits	[nut, tree-nut]	1.0	1.0	1.0
502610	Châtaignes entières bio	[nut, tree-nut]	1.0	1.0	1.0
155675	Mauna loa, dry roasted macadamia	[nut, tree-nut]	1.0	1.0	1.0
326224	Châtaigne Bouche Rouge G2 CAT 2 BIO France ~5kg	[nut, tree-nut]	1.0	1.0	1.0
507036	Mandeln gemahlen	[nut, tree-nut]	1.0	1.0	1.0
502609	Châtaignes entières bio	[nut, tree-nut]	1.0	1.0	1.0

Produits similaires au produit : Hot Cocoa With Natural & Artificial Flavors Of Peanut Butter Cup & Fugge					
	product_name	liste_ingredients	similarity1	similarity2	mean_similarity
126839	Rich & Creamy Hot Cocoa Beverage Mix	[sugar, added-sugar, disaccharide, corn-syrup, ...]	1.000000	0.989084	0.994542
295781	Duck Commander, Duck-Cups Cocoa Coffee, Uncle ...	[sugar, added-sugar, disaccharide, cocoa, salt, ...]	0.994241	0.984349	0.989295
80851	Milk chocolate hot cocoa drink mix	[sugar, added-sugar, disaccharide, corn-syrup, ...]	0.994241	0.983445	0.988843
32483	Hot Cocoa Drink Mix	[sugar, added-sugar, disaccharide, corn-syrup, ...]	0.994241	0.983445	0.988843
84146	Milk chocolate flavor hot cocoa drink mix, milk choco...	[sugar, added-sugar, disaccharide, corn-syrup, ...]	0.994241	0.983445	0.988843
34508	Milk chocolate hot cocoa drink mix, milk choco...	[sugar, added-sugar, disaccharide, corn-syrup, ...]	0.994241	0.983445	0.988843
314864	Hot Cocoa	[sugar, added-sugar, disaccharide, cocoa, skimmed ...]	0.994241	0.978067	0.986154
58564	French vanilla cappuccino mix	[corn-syrup-solids, added-sugar, disaccharide, ...]	1.000000	0.972197	0.986098
132962	Mild coffee and hazelnut flavor cappuccino mix...	[corn-syrup-solids, added-sugar, disaccharide, ...]	1.000000	0.972197	0.986098
132961	Mild coffee and french vanilla flavor cappuccino ...	[corn-syrup-solids, added-sugar, disaccharide, ...]	1.000000	0.972197	0.986098
58563	Milk chocolate flavored hot cocoa mix, milk choco...	[corn-syrup-solids, added-sugar, disaccharide, ...]	1.000000	0.971087	0.985544
132965	Sweet salty caramel chocolate flavor hot cocoa...	[corn-syrup-solids, added-sugar, disaccharide, ...]	1.000000	0.970929	0.985465

Produits similaires au produit : Chicken Breast Nuggets With Rib Meat					
	product_name	liste_ingredients	similarity1	similarity2	mean_similarity
288073	Chicken Breast Patties	[water, modified-starch, starch, salt, sugar, ...]	0.992367	0.999187	0.995777
288074	Breaded full cooked chicken breast tenders wit...	[water, modified-starch, starch, salt, sodium, ...]	0.981804	0.999140	0.990472
288072	Breaded Fully Cooked Chicken Breast Rings With...	[water, modified-starch, starch, salt, sodium, ...]	0.981804	0.999140	0.990472
315681	Fully cooked portioned chicken fillet white me...	[chicken-breast, poultry, chicken, chicken-meat, ...]	0.967513	0.957202	0.962357
210670	Corn veggie tots, corn	[soya-oil, oil-and-fat, vegetable-oil-and-fat, ...]	0.967539	0.951326	0.959432
210655	Cauliflower veggie tots, cauliflower	[cauliflower, vegetable, soya-oil, oil-and-fat, ...]	0.967539	0.949927	0.958733
210656	Broccoli veggie tots, broccoli	[broccoli, vegetable, soya-oil, oil-and-fat, ...]	0.967539	0.948916	0.958227
210671	Sweet potato & cauliflower veggie tots, sweet ...	[sweet-potato, vegetable, root-vegetable, cauli...	0.969926	0.946115	0.958021
158454	Chicken Fried Steak Breading Mix	[wheat-flour, cereal, flour, wheat, cereal-flo...	0.921436	0.988562	0.953999
316071	Chicken Rings	[water, coating, reduced-iron, minerals, iron, ...]	0.967613	0.937969	0.952791
6413	Chicken & cheese cornbread sandwiches	[barley-malt-flour, cereal, flour, cereal-flo...	0.981209	0.922682	0.951945
63453	Breaded mushrooms, breaded	[mushroom, water, wheat-flour, cereal, flour, ...]	0.966107	0.936523	0.951315

# Exemples sur d'autres produits

- Les exemples suivants ont été calculés sur un échantillon aléatoire de 50 000 produits :

Produits similaires au produit : Creamy peanut butter					
	product_name	liste_ingredients	similarity1	similarity2	mean_similarity
112128	Skippy, peanut butter	[roasted-peanuts, nut, peanut, sugar, added-su...	1.000000	1.000000	1.000000
271659	Crunchy peanut butter, crunchy	[roasted-peanuts, nut, peanut, sugar, added-su...	1.000000	1.000000	1.000000
252970	Peanut butter	[roasted-peanuts, nut, peanut, sugar, added-su...	1.000000	1.000000	1.000000
220845	Peanut butter	[peanut, nut, sugar, added-sugar, disaccharide...	0.996382	1.000000	0.998191
110037	Creamy peanut butter, creamy	[peanut, nut, sugar, added-sugar, disaccharide...	0.996382	1.000000	0.998191
109736	Crunchy	[peanut, nut, sugar, added-sugar, disaccharide...	0.996382	1.000000	0.998191
8001	Roundy's, peanut butter	[peanut, nut, sugar, added-sugar, disaccharide...	0.988989	1.000000	0.994495
605693	skippy	[peanut, nut, peanut-oil, oil-and-fat, vegetab...	0.996382	0.992438	0.994410
10566	Peanut butter	[peanut, nut, sugar, added-sugar, disaccharide...	0.996382	0.983148	0.989765
53010	Peanut butter	[peanut, nut, sugar, added-sugar, disaccharide...	0.996382	0.983148	0.989765
134417	Crunchy peanut butter	[peanut, nut, sugar, added-sugar, disaccharide...	0.996382	0.983148	0.989765

Produits similaires au produit : Restaurant style white corn tortillas chips					
	product_name	liste_ingredients	similarity1	similarity2	mean_similarity
301741	Tortilla Chips	[corn, cereal, vegetable-oil, oil-and-fat, veg...	1.000000	0.962887	0.981444
653949	Galette de Mais Integrale Biologiche	[corn, cereal, corn-oil, oil-and-fat, vegetabl...	0.938672	1.000000	0.969336
44849	Fritos Lightly Salted Corn Chips 9.75 Ounce Pl...	[corn, cereal, corn-oil, oil-and-fat, vegetabl...	0.910829	1.000000	0.955415
553853	Maiswaffeln mit Meersalz	[corn, cereal, corn-oil, oil-and-fat, vegetabl...	0.938672	0.971154	0.954913
516816	Maiswaffeln Meersalz	[corn, cereal, sea-salt, salt, corn-oil, oil-a...	0.938672	0.971154	0.954913
652360	Galette Mais e Quinoa	[corn, cereal, corn-oil, oil-and-fat, vegetabl...	0.938672	0.971154	0.954913
49397	Golden fluff, popcorn	[vegetable-oil, oil-and-fat, vegetable-oil-and...	0.972848	0.926619	0.949734
85419	Schnucks, authentic restaurant style tortilla ...	[cereal, corn, vegetable-oil, oil-and-fat, veg...	0.887696	1.000000	0.943848
674950	Maiz gigante sabor BBQ	[corn, cereal, vegetable-oil, oil-and-fat, veg...	0.887696	1.000000	0.943848
466426	Pop corn salé	[corn, cereal, sunflower-oil, oil-and-fat, veg...	0.887886	0.997269	0.942578
292708	Popcorn, Indiana, Fit Popcorn Chips, Himalayan...	[corn, cereal, sunflower-oil, oil-and-fat, veg...	0.887886	0.997269	0.942578
548221	Tortilla Chips Natur	[cereal, corn, sunflower-oil, oil-and-fat, veg...	0.887886	0.997269	0.942578

Produits similaires au produit : Chocolate sandwich cookies					
	product_name	liste_ingredients	similarity1	similarity2	mean_similarity
105714	Oreo cookies 12x10 700 oz	[sugar, added-sugar, disaccharide, flour, palm...	0.991382	0.978811	0.985096
105797	Oreo cookies hot cocoa 1x10.7 oz	[sugar, added-sugar, disaccharide, flour, palm...	0.991382	0.978811	0.985096
106027	Chocolate creme sandwich cookies	[sugar, added-sugar, disaccharide, flour, palm...	0.990543	0.978318	0.984430
105764	Oreo cookies 1x20 000 oz	[sugar, added-sugar, disaccharide, flour, palm...	0.990543	0.971891	0.981217
62369	Original chocolate chip cookies, original choc...	[flour, sugar, added-sugar, disaccharide, palm...	0.989450	0.970196	0.979823
20085	Original chocolate chip cookies, original choc...	[flour, sugar, added-sugar, disaccharide, palm...	0.989450	0.970196	0.979823
74687	Chocolate Chip Cookies	[wheat-flour, cereal, flour, wheat, cereal-flo...	0.968883	0.990347	0.979615
93864	Chocolate Sandwich Cookies	[sugar, added-sugar, disaccharide, flour, vege...	0.991406	0.967362	0.979384
118070	Chocolate Sandwich Cookies	[sugar, added-sugar, disaccharide, flour, palm...	0.991406	0.966565	0.978985
133370	Original chocolate sandwich cookie cremes	[flour, sugar, added-sugar, disaccharide, palm...	0.991406	0.966565	0.978985
106086	Chocolage confetti cake	[sugar, added-sugar, disaccharide, flour, palm...	0.991382	0.963738	0.977560
136193	Crispy Wheat Snack Crackers	[flour, whole-wheat-flour, cereal, wheat, cere...	0.971420	0.983107	0.977263

Produits similaires au produit : Nature's rancher, ground organic chicken					
	product_name	liste_ingredients	similarity1	similarity2	mean_similarity
321164	Ground chicken	[chicken, poultry, e392]	1.000000	1.000000	1.000000
102290	85% lean 15% fat ground turkey	[turkey, poultry, e392]	0.998805	0.983604	0.991204
69984	Ground Turkey	[turkey, poultry, e392]	0.998805	0.983604	0.991204
36586	Ground turkey	[turkey, poultry, e392]	0.998805	0.983604	0.991204
102599	Jennie-o, ground turkey	[turkey, poultry, e392]	0.998805	0.983604	0.991204
102484	Lean ground turkey	[turkey, poultry, e392]	0.998805	0.983604	0.991204
40517	Oven roasted chicken breast, oven roasted	[chicken-breast, poultry, chicken, chicken-meat]	1.000000	0.807623	0.903812
225503	Pine Manor Farms, Extra Lean Ground Chicken	[chicken-breast, poultry, chicken, chicken-meat]	0.796170	0.986675	0.891422
217391	Organic boneless and skinless chicken breast	[chicken-breast, poultry, chicken, chicken-meat]	0.796170	0.986675	0.891422
196494	No salt added chicken bone broth, chicken	[chicken-broth, poultry, broth, chicken, poult...	0.891577	0.890993	0.891285
327560	Emincé De Dinde Kebab Cuit ~1kgx4	[turkey, poultry]	0.806598	0.975406	0.891002
317425	Organic ground turkey	[turkey, poultry]	0.806598	0.975406	0.891002

# Exemples sur d'autres produits

- Les exemples suivants ont été calculés sur un échantillon aléatoire de 50 000 produits :

Produits similaires au produit : Peanut Butter Spread					
	product_name	liste_ingredients	similarity1	similarity2	mean_similarity
216167	Peanut butter, crunchy	[peanut, nut, sugar, added-sugar, disaccharide...]	0.992249	1.000000	0.996125
249503	Creamy Peanut Butter Spread With Honey	[peanut, nut, sugar, added-sugar, disaccharide...]	0.992249	1.000000	0.996125
10875	Peanut butter	[peanut, nut, sugar, added-sugar, disaccharide...]	0.992249	1.000000	0.996125
133371	Crunchy peanut butter	[peanut, nut, sugar, added-sugar, disaccharide...]	0.992249	1.000000	0.996125
63520	Crunchy Peanut Butter	[peanut, nut, sugar, added-sugar, disaccharide...]	0.992249	1.000000	0.996125
168878	Creamy peanut butter	[roasted-peanuts, nut, peanut, sugar, added-su...]	0.992249	0.995883	0.994066
108859	Crunchy peanut butter	[peanut, nut, sugar, added-sugar, disaccharide...]	0.973765	1.000000	0.986882
176860	Crunchy peanut butter	[roasted-peanuts, nut, peanut, sugar, added-su...]	0.976268	0.995686	0.985977
176767	Peanut butter creamy	[roasted-peanuts, nut, peanut, sugar, added-su...]	0.976268	0.995686	0.985977
7916	Roundy's, peanut butter	[peanut, nut, sugar, added-sugar, disaccharide...]	0.965563	1.000000	0.982782
7918	Roundy's, creamy peanut butter	[peanut, nut, sugar, added-sugar, disaccharide...]	0.965563	1.000000	0.982782
57714	Crunchy Peanut Butter	[roasted-peanuts, nut, peanut, sugar, added-su...]	0.992249	0.972820	0.982535

Produits similaires au produit : Nonfat greek yogurt					
	product_name	liste_ingredients	similarity1	similarity2	mean_similarity
239073	Greek strained yogurt with cherry	[cane-sugar, added-sugar, disaccharide, sugar,...]	0.993806	0.994424	0.994115
239070	Fage, total greek strained yogurt, cherry	[cane-sugar, added-sugar, disaccharide, sugar,...]	0.993806	0.971998	0.982902
239096	Classic greek strained yogurt	[cane-sugar, added-sugar, disaccharide, sugar,...]	0.991402	0.967126	0.979264
239056	Nonfat greek strained yogurt	[cane-sugar, added-sugar, disaccharide, sugar,...]	0.952307	0.993893	0.973100
89176	Premium blended black cherry authentic greek l...	[cherry, fruit, cane-sugar, added-sugar, disac...]	0.972279	0.970780	0.971529
239053	Greek strained yogurt	[cane-sugar, added-sugar, disaccharide, sugar,...]	0.951567	0.984592	0.968080
486014	Yaourt brassé sur lit de myrtilles	[blueberry, fruit, berries, water, cane-sugar,...]	0.952265	0.981168	0.966717
5327	2% milkfat lowfat greek yogurt	[sugar, added-sugar, disaccharide, corn-starch...]	0.956619	0.989798	0.963204
689726	Greek Nonfat Yogurt With Fruit On The Bottom	[sugar, added-sugar, disaccharide, orange, fru...]	0.962068	0.956451	0.956259
239082	Greek Strained Yogurt	[cane-sugar, added-sugar, disaccharide, sugar,...]	0.951567	0.961618	0.956593
291796	Harmless harvest strawberry	[cane-sugar, added-sugar, disaccharide, sugar,...]	0.943401	0.963125	0.953263
59270	Smoothie	[cane-sugar, added-sugar, disaccharide, sugar,...]	0.967467	0.937780	0.952623

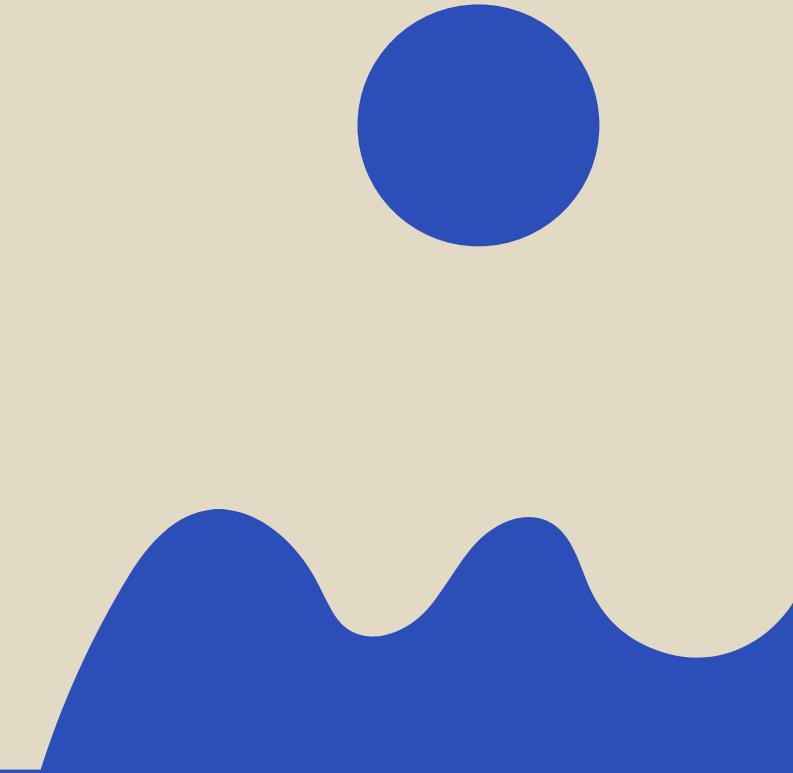
Produits similaires au produit : Wok st shrimp veggie lo mein p					
	product_name	liste_ingredients	similarity1	similarity2	mean_similarity
270363	Shrimp wonton soup	[wheat-flour, cereal, flour, wheat, cereal-flo...]	0.934494	0.951751	0.943123
168336	Steamable asian vegetables in a premium citrus...	[broccoli, vegetable, legume, pea, carrot, roo...]	0.933959	0.934581	0.934270
200022	Soy soy, marinade & dip, spicy'n sweet chili h...	[soy-sauce, sauce, sugar, added-sugar, disacch...]	0.909885	0.958281	0.934073
292996	Palcha, marinade & sauce	[legume, soya, soya-bean, sugar, added-sugar, ...]	0.992526	0.973113	0.932820
236287	Egg rolls pork	[filling, shrimp, shellfish, crustacean, celer...]	0.913953	0.950307	0.932130
560473	Tako Yaki Octopus balls	[wheat-flour, cereal, flour, wheat, cereal-flo...]	0.912343	0.951146	0.931745
186309	Hoisin Sauce	[sugar, added-sugar, disaccharide, water, salt...]	0.892412	0.969312	0.930862
597934	White corn, red peppers & edamame in a coconut...	[cereal, corn, red-bell-pepper, vegetable, bel...]	0.931152	0.929467	0.930309
232886	Pulo, Marinade, Mango Chili	[mango, fruit, water, sugar, added-sugar, disa...]	0.901198	0.959129	0.930164
376982	Saveurs d'Ailleurs - Beignets de crevettes ave...	[water, potato-starch, starch, wheat-flour, ce...]	0.939875	0.918776	0.929325
148652	Ht traders, noodle bowl, sesame teriyaki	[noodle, dough, sauce, vegetable, wheat-flour,...]	0.923511	0.934208	0.928860
683482	Huhn Teriyaki Pita Roll	[bread, carrot, vegetable, root-vegetable, wat...]	0.930499	0.926717	0.928608

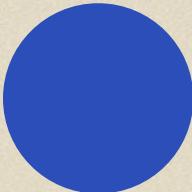
Produits similaires au produit : Super premium ice cream bars					
	product_name	liste_ingredients	similarity1	similarity2	mean_similarity
248712	Vanilla caramel ice cream bars	[milk, dairy, cream, skimmed-milk, sugar, add...]	1.000000	0.960304	0.980152
137482	Cheesecake ice cream bites	[milk, dairy, cream, skimmed-milk, sugar, add...]	0.995585	0.905692	0.950639
295442	Truffle Basket, French Chocolate Truffle	[vegetable-oil, oil-and-fat, vegetable-oil-and...]	0.914708	0.978926	0.946817
13996	Milk Chocolate	[milk-chocolate, chocolate, sugar, added-sugar...]	0.959967	0.921848	0.940907
223922	Cape Harvest, Milk Chocolate Covered Cherries	[milk-chocolate, chocolate, cherry, fruit, e41...]	0.960625	0.972315	0.939170
258199	Coconut Caramel Bites	[e150, milk-chocolate, chocolate, coconut, fru...]	0.978024	0.897009	0.937516
162955	Haagen-dazs,gelato, italian style frozen dess...	[cream, dairy, corn-syrup, added-sugar, disac...]	0.922809	0.949855	0.936332
150023	mint chocolate chip	[milk, dairy, cream, sugar, added-sugar, disac...]	0.978998	0.892948	0.935817
507415	LINDOR	[sugar, added-sugar, disaccharide, cocoa-butter...]	0.899137	0.972096	0.935616
174338	The original ice cream treat	[coconut-oil, oil-and-fat, vegetable-oil-and-f...]	0.883334	0.987605	0.935469
174342	Ice Cream Sandwiches	[coconut-oil, oil-and-fat, vegetable-oil-and-f...]	0.883334	0.987605	0.935469
137077	Ice cream, chocolate chip	[milk, dairy, cream, skimmed-milk, buttermilk,...]	0.964202	0.906556	0.935379

# 05

## Partie C

Carte et clustering des produits





# Vectorisation des ingrédients

- L'objectif de cette partie est de réaliser un clustering des produits et d'ensuite les afficher sur une carte en 2 dimensions.
- La difficulté est la même que pour la partie précédente => Les produits n'ont pas le même nombre d'ingrédients et donc ne sont pas naturellement de la même dimension.
- Pour remédier à ce problème nous pouvons utiliser la méthode de calcul du score de similarité.
- Nous allons tester 3 méthodes :
  1. Calculer la matrice des distances par paire entre les produits afin de reconstruire les coordonnées en 2 dimensions.
  2. Calculer la matrice des similarités par paire entre les produits et réduire les résultats à 2 dimensions.
  3. Sélectionner quelques dizaines de produits aléatoirement afin de calculer les similarités entre tous les produits du dataset et ceux choisis pour s'en servir comme coordonnées de vecteur.

# Première méthode

- Nous souhaitons calculer les distances entre les produits par paire et les utiliser pour reconstruire une carte des produits en 2 dimensions.
- Pour calculer les distances entre les produits nous faisons le calcul :
  - $\text{Distance}(A,B) = 1 - \text{Similiarité}(A,B)$

- Comme le calcul des similarités est plutôt long nous sélectionnons aléatoirement 5 000 produits parmi ceux qui n'ont pas perdu d'ingrédients lors que preprocessing, qui sont composés d'au moins 5 ingrédients et qui ont des noms différents.
- Nous calculons ensuite leur matrice des similarités :

	La Framboise Intense	Easter Gummy Rings	Original Tapitas Serrano Schinken	Crème Anglaise	Goût original	Jambon De Campagne	Saint agur offre gourmande	semola di grano duro con germe di grano	Le Gâteau Délice d'Or Nature	Brochettes de porc au paprika	... ...	Caffe latte coffee with milk
ingredients												
La Framboise Intense	1.000000	0.727130	0.667388	0.669098	0.614450	0.630391	0.540836	0.563380	0.654759	0.667502	...	0.646316
Easter Gummy Rings	0.727130	1.000000	0.710219	0.790429	0.784077	0.689682	0.508566	0.551688	0.773039	0.750619	...	0.703133
Original Tapitas Serrano Schinken	0.667388	0.710219	1.000000	0.802189	0.687396	0.967835	0.651195	0.604054	0.780632	0.866182	...	0.691280
Crème Anglaise	0.669098	0.790429	0.802189	1.000000	0.754844	0.789592	0.791588	0.589368	0.870901	0.818823	...	0.801857
Goût original	0.614450	0.784077	0.687396	0.754844	1.000000	0.683569	0.468965	0.590292	0.696357	0.690222	...	0.616188

# Première méthode

- Pour ne pas avoir d'erreur, nous retirons tous les produits qui ont une similarité négative (4 produits).
- Nous calculons ensuite la matrice des distances :

	La Framboise Intense	Easter Gummy Rings	Original Tapitas Serrano Schinken	Crème Anglaise	Goût original	Jambon De Campagne	Saint agur offre gourmande	semola di grano duro con germe di grano	Le Gâteau Délice d'Or Nature	Brochettes de porc au paprika	...
<b>ingredients</b>											
La Framboise Intense	0.000000	0.272870	0.332612	0.330902	0.385550	0.369609	0.459164	0.436620	0.345241	0.332498	...
Easter Gummy Rings	0.272870	0.000000	0.289781	0.209571	0.215923	0.310318	0.491434	0.448312	0.226961	0.249381	...
Original Tapitas Serrano Schinken	0.332612	0.289781	0.000000	0.197811	0.312604	0.032165	0.348805	0.395946	0.219368	0.133818	...
Crème Anglaise	0.330902	0.209571	0.197811	0.000000	0.245156	0.210408	0.208412	0.410632	0.129099	0.181177	...
Goût original	0.385550	0.215923	0.312604	0.245156	0.000000	0.316431	0.531035	0.409708	0.303643	0.309778	...

# Première méthode : Implémentation

- Source de la théorie : <https://math.stackexchange.com/questions/156161/finding-the-coordinates-of-points-from-distance-matrix>
- L'idée est de prendre les 2 premiers produits de la matrice comme repère, et de calculer les coordonnées des autres produits à partir du repère.

```
def x_coord_of_point(D, j):
    return ( D[0,j]**2 + D[0,1]**2 - D[1,j]**2 ) / ( 2*D[0,1] )

def coords_of_point(D, j):
    x = x_coord_of_point(D, j)
    #print("D[0,j]**2 : "+str( D[0,j]**2)+ " x**2 : "+str(x**2))
    return np.array([x, math.sqrt( (D[0,j]**2 - x**2) if (D[0,j]**2 - x**2)>0 else 0 )])

def calculate_positions(D):
    (m, n) = D.shape
    P = np.zeros( (n, 2) )
    tr = ( min(min(D[2,0:2]), min(D[2,3:n])) / 2)**2
    P[1,0] = D[0,1]
    P[2,:] = coords_of_point(D, 2)
    for j in range(3,n):
        P[j,:] = coords_of_point(D, j)
        if abs( np.dot(P[j,:]-P[2,:], P[j,:]-P[2,:]) - D[2,j]**2 ) > tr:
            P[j,1] = - P[j,1]
    return P
```

# Première méthode : Résultats

- Voici les résultats que nous avons obtenus :

Répartition des produits en 2 dimensions



	X	Y
ingredients		
La Framboise Intense	0.000000	0.000000
Easter Gummy Rings	0.272870	0.000000
Original Tapitas Serrano Schinken	0.185282	0.276227
Crème Anglaise	0.256595	-0.208938
Goût original	0.323385	-0.209931

- La répartition des points n'est pas uniforme, seulement 4 produits ont une ordonnées positive et beaucoup de produits sont alignés sur l'axe des abscisses.
- Sans comprendre exactement pourquoi nous obtenons ces résultats, il semblerait que cela puisse venir du fait que la matrice des distances n'est pas positive semi-définie.

```
print("La matrice est positive semidéfinie : ",is_hermitian_positive_semidefinite(df_distances.values))  
La matrice est positive semidéfinie : False
```

# Deuxième méthode:

- Pour la deuxième méthode, nous repartons de la matrice des similarités par paire.
- Chaque produit est donc défini par un vecteur de dimension 5000 sur lesquels nous effectuons un clustering Kmeans, puis nous les réduisons à 2 dimensions avec la méthode UMAP afin de pouvoir les afficher sur la carte suivante :
- Nous générerons 24 clusters.



# Exemples de clusters deuxième méthode:

```
cluster= 0
df_clusters_products[df_clusters_products["cluster"]==cluster].head()
```

	product	2Dvecteur_dim1	2Dvecteur_dim2	cluster
3	Crème Anglaise	13.418544	2.070119	0
54	Flan Entremet Praliné Ancel	16.433201	3.204122	0
57	Petits Filous Tub's Goûts Myrtille et Abricot ...	13.553294	2.068043	0
82	Riz au lait entier façon grand-mère	14.379497	2.242316	0
113	La teurgoule	13.643574	1.663870	0

```
cluster= 1
df_clusters_products[df_clusters_products["cluster"]==cluster].head()
```

	product	2Dvecteur_dim1	2Dvecteur_dim2	cluster
0	La Framboise Intense	11.526351	12.318079	1
36	Confiture Poires Vanille	12.063567	11.662287	1
49	Confiture allégée mangue ananas passion	9.752068	12.863735	1
51	Premier pink lady 100% apple juice from concen...	6.760190	12.211227	1
77	Confiture de mangue extra	12.413439	11.760283	1

```
cluster= 2
df_clusters_products[df_clusters_products["cluster"]==cluster].head()
```

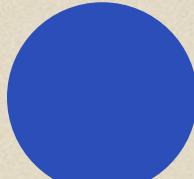
	product	2Dvecteur_dim1	2Dvecteur_dim2	cluster
16	Sauce tomate aubergines	18.096264	-8.125980	2
25	Adobo, All Purpose Seasoning With Pepper	5.666723	6.735769	2
38	Olives cassées catalane	18.104156	-8.136171	2
125	Black beans	4.624235	6.099005	2
135	Huile spéciale pizza pimentée	7.179883	2.108928	2

```
cluster= 3
df_clusters_products[df_clusters_products["cluster"]==cluster].head()
```

	product	2Dvecteur_dim1	2Dvecteur_dim2	cluster
6	Saint agur offre gourmande	15.218687	20.078260	3
10	100% natural wisconsin pre-sliced mild cheddar...	16.753920	16.723969	3
24	l'emmental français extra fin	14.837915	20.721762	3
31	Queso havarti	26.969465	1.188625	3
44	Ser Rycki Edam kl.I	16.699915	16.819681	3

# Nuages de mots deuxième méthode:





# Nuages de mots deuxième méthode:



- Les clusters semblent plutôt cohérents.
- Le problème de cette technique reste cependant le coût de calcul, en effet il faut calculer  $n^2$  similarités ( $n$  le nombre de produits), ce qui correspond à 25 millions de similarités pour 5 000 produits.
- Il parait donc inenvisageable de l'utiliser pour plusieurs dizaines de milliers de produits.

# Troisième méthode

- Dans la deuxième méthode chaque produit était défini par un vecteur de dimension 5000 où chaque dimension correspond à la similarité avec un autre produit.
- Pour réduire le nombre de similarités à calculer par produit nous pouvons réduire le nombre de dimensions du vecteur permettant de définir un produit.
- Au lieu de calculer les similarités entre tous les produits du dataset, nous sélectionnons un échantillon de produits et nous calculons les similarités de tous les produits avec cet échantillon. Pour un échantillon de  $n$  produits :

	Produit1	Produit2	Produit3	Produit4	...	Produitn
ProduitA	Sim(A,1)	Sim(A,2)	Sim(A,3)	Sim(A,5)	...	Sim(A,n)

Vecteur du produit A

- Pour que chaque produit soit bien représenté par son vecteur il faut avoir un bon équilibre entre le nombre de produits dans l'échantillon et la similarité entre les produits au sein de l'échantillon.

# Génération de l'échantillon de produits :

```
def selection_vecteurs_representation(df, seuil) :  
    df=df.set_index("product_name")  
  
    #Initialisation des variables :  
    product_init = df.sample().index[0]  
    list_products_names =[product_init]  
    list_ingredients_list= [df.loc[df.index==product_init,"liste_ingredients"].iloc[0]]#list of list of ingredient  
    list_dicts_list =[[df_similarities[ingredient].to_dict() for ingredient in list_ingredients_list[0]]]  
  
    df_temp= shuffle(df)  
  
    for i in tqdm(range(len(df_temp))) : # Pour chaque produit dans le dataframe  
        #product_name = df_temp.index[i]  
        product_ingredients= df_temp["liste_ingredients"].iloc[i]  
  
        add=True  
  
        for j in range(len(list_ingredients_list)) : #Pour chaque produit dans le vecteur  
            sim = similarities_both_ways_mean(list_ingredients_list[j], product_ingredients,list_dicts_list[j])  
  
            if sim >seuil :  
                add = False  
                break  
  
    if add :  
        list_products_names.append(df_temp.index[i])  
        list_ingredients_list.append(product_ingredients)  
        list_dicts_list.append([df_similarities[ingredient].to_dict() for ingredient in product_ingredients])  
  
    return list_products_names, list_ingredients_list
```

## Fonctionnement :

- On mélange le dataframe des produits
- On ajoute le premier produit à l'échantillon
- Pour chaque produit dans le dataset :
- Si la similarité maximale entre le produit et les produits de l'échantillon est inférieure au seuil (renseigné en input) on ajoute le produit à l'échantillon.

Plus le seuil en input de la fonction est bas plus l'échantillon de produits renvoyé sera petit et inversement.

# Génération de l'échantillon de produits :

## Taille des échantillons obtenus en fonction du seuil en input :

```
[ ] list_products_name6, list_ingredients_list6 = selection_vecteurs_representation(df.drop_duplicates(["product_name"]).dropna(), 0.6)
print(len(list_products_name6))

100%|██████████| 506467/506467 [02:10<00:00, 3883.94it/s]
21

[ ] list_products_name, list_ingredients_list = selection_vecteurs_representation(df.drop_duplicates(["product_name"]).dropna(), 0.7)
print(len(list_products_name))

100%|██████████| 506467/506467 [03:58<00:00, 2126.60it/s]
56

[ ] list_products_name8, list_ingredients_list8 = selection_vecteurs_representation(df.drop_duplicates(["product_name"]).dropna(), 0.8)
print(len(list_products_name8))

100%|██████████| 506467/506467 [15:07<00:00, 558.30it/s]388

Trop long pour être exécuté:

[ ] list_products_name9, list_ingredients_list9 = selection_vecteurs_representation(df.drop_duplicates(["product_name"]).dropna(), 0.9)
print(len(list_products_name9))

1%| | 5108/506467 [05:08<8:24:30, 16.56it/s]
```

La taille de l'échantillon et le temps d'exécution augmentent de manière exponentielle lorsque la valeur de seuil augmente.

Nous choisissons pour la suite un seuil de 0.8 de similarité avec pour logique qu'un vecteur avec beaucoup de dimensions contiendra plus d'information pour chaque produits.

# Troisième méthode : vectorisation des produits

Seuil de similarité : 0.8

Taille des échantillons obtenus en fonction du seuil en input :

```
Nombre de produits après avoir retiré ceux avec des noms similaires et ceux sans nom : 506467  
Nombre de produits après avoir également retiré ceux qui ont perdu des ingrédients lors du préprocessing : 95728  
Nombre de produits final : 95728
```

product_name	liste_ingredients	FruitSations	Tastykrisps, Creme Filled Wafers	Bacon n eggs	Galettes sarrasin bio	Chunky Vegetable Pasta Sauce	Chipotle & habanero cheddar cheese	Lait bébé en poudre 2ème âge	Pure Parówki z Szynki	DOONUTS NAPPES CHOCOLAT X12	... seleccion - Lobos	Sal de borage oil	Fish, flax & borage oil	Doritos bbq	Pistach flav sug fr puddi p filli
Solène céréales poulet	[antioxidant, colour, tomato, vegetable, mayon...	0.790596	0.771569	0.808505	0.675141	0.844268	0.777894	0.736789	0.884753	0.750685	...	0.566398	0.691746	0.663914	0.7313
Crème dessert chocolat	[whole-milk, dairy, milk, sugar, added-sugar, ...	0.721975	0.684552	0.680971	0.604722	0.661311	0.699271	0.660229	0.679323	0.810186	...	0.321820	0.456348	0.290248	0.6920
Peanuts	[peanut, nut, wheat-flour, cereal, flour, whea...	0.657928	0.849548	0.679794	0.806511	0.711891	0.590332	0.675658	0.711194	0.813660	...	0.395880	0.593831	0.376113	0.7204
Organic Hazelnuts	[hazelnut, nut, tree-nut]	0.415945	0.764125	0.417691	0.596679	0.452293	0.445941	0.475246	0.471253	0.688729	...	0.018930	0.330873	-0.163923	0.6502
Organic Sweetened Banana Chips	[banana, fruit, coconut-oil, oil-and-fat, vege...	0.807730	0.812906	0.656390	0.653296	0.764967	0.534395	0.648041	0.717185	0.687311	...	0.202217	0.541048	0.112020	0.5314

- Voici le dataframe que l'on obtient après avoir calculé les similarités des 95 728 produits gardés et les produits de l'échantillon.
- Il y a 388 produits dans l'échantillon.
- Donc chaque produit est défini sur 388 dimensions.

# Troisième méthode : carte en 2 dimensions

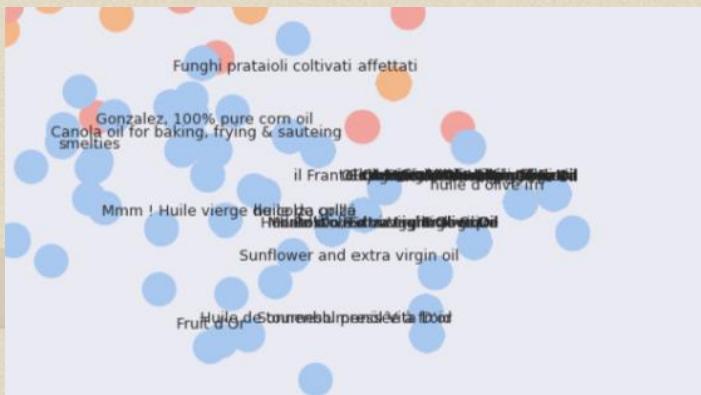
Seuil de similarité : 0.8

Après avoir réalisé un kmeans de 30 clusters sur les produits nous effectuons une réduction de dimensions par analyse de composantes principales, pour générer la carte suivante en 2 dimensions.



# Troisième méthode : Zoom sur des clusters

Seuil de similarité : 0.8



Pour ne pas surcharger la carte nous n'affichons qu'un nom de produit sur 50 mais cela reste peu lisible.

# Troisième méthode : Exemples de clusters

Seuil de similarité : 0.8

	product	2Dvecteur_dim1	2Dvecteur_dim2	cluster
9	Divinely Organic Granola	0.060734	-0.144990	0
18	Organic Trail Mix	0.234021	-0.012762	0
46	Belgian Vanilla Waffle	-0.271880	0.013763	0
89	Dried Cranberries, Pomegranate Flavor	0.011956	-0.018088	0
90	Welch's, dried cranberries, blueberry	0.011956	-0.018088	0

	product	2Dvecteur_dim1	2Dvecteur_dim2	cluster
61	Monterey Jack	0.344140	0.292756	1
108	Provola Typical Crotonese	0.442731	-0.034573	1
582	Salted butter half sticks, salted	0.457578	-0.626415	1
604	Private selection, baby swiss sliced cheese	0.419276	0.296039	1
605	Private selection, mozzarella sliced cheese	0.467601	0.274274	1

	product	2Dvecteur_dim1	2Dvecteur_dim2	cluster
6	Organic Medium Shredded Coconut	2.023834	1.623453	2
7	Organic Coconut Chips	2.023834	1.623453	2
30	Turkish Apricots	1.689984	1.835891	2
32	Organic Dried Turkish Apricots, Bin # 5801	2.603899	1.907965	2
34	Organic Pitted Prunes	2.613868	1.866310	2

	product	2Dvecteur_dim1	2Dvecteur_dim2	cluster
41	Flute	2.188738	-0.991155	3
106	Creamy wheat cereal	2.188738	-0.991155	3
113	Lumaconi Giant Pasta Shells Italian Macaroni P...	1.131611	-0.766182	3
130	POP TARTS FROSTED STRAWBERRY	1.168541	-0.774448	3
139	Spaghetti	1.973198	-0.907886	3

# Troisième méthode : Exemples de clusters

Seuil de similarité : 0.8

	product	2Dvecteur_dim1	2Dvecteur_dim2	cluster
1	Crème dessert chocolat	-0.186629	0.217859	5
65	Lactaid, ice cream, vanilla	-0.171415	0.468500	5
66	Natural ice cream	-0.153585	0.447448	5
67	Lactaid, ice cream, butter pecan	-0.311655	0.257876	5
174	Butter Quarters Unsalted	-0.100050	0.291625	5

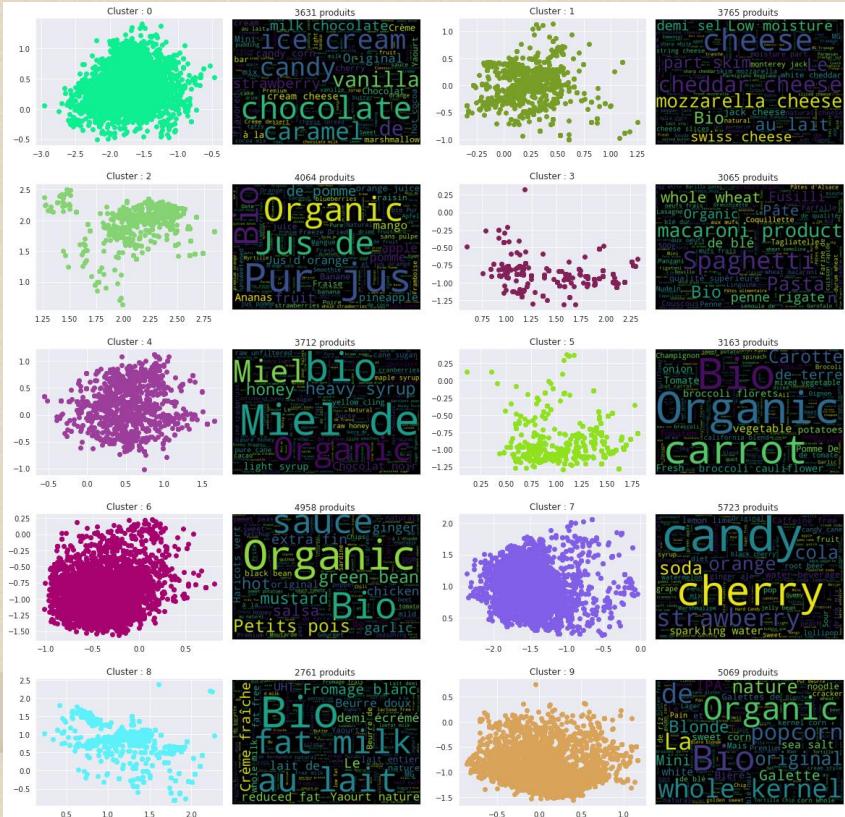
	product	2Dvecteur_dim1	2Dvecteur_dim2	cluster
3	Organic Hazelnuts	0.447938	0.208971	6
13	Organic Large Raw Whole Cashews	0.434128	0.087337	6
16	Organic Raw Walnuts	0.452024	0.202009	6
33	Organic Unsalted Pistachios	0.412270	0.182280	6
36	Organic U.S. Peanuts	0.475852	0.159926	6

	product	2Dvecteur_dim1	2Dvecteur_dim2	cluster
0	Solène céréales poulet	-0.462076	0.033424	7
47	Freshly Baked Apple Pie	-0.209925	0.053064	7
111	Coleslaw	-0.348527	-0.126471	7
184	Sauce Mix	-0.300140	-0.075228	7
185	Golden Curry, Sauce Mix	-0.300140	-0.075228	7

Tous les produits dans les clusters ne semblent pas toujours logiques mais cela vient du fait qu'il est difficile d'avoir une vision d'ensemble d'un cluster en affichant seulement 5 produits sur plusieurs milliers. En observant les nuages de mots il est plus simple de comprendre les types de produits associés aux clusters.

# Troisième méthode : Nuages de mots

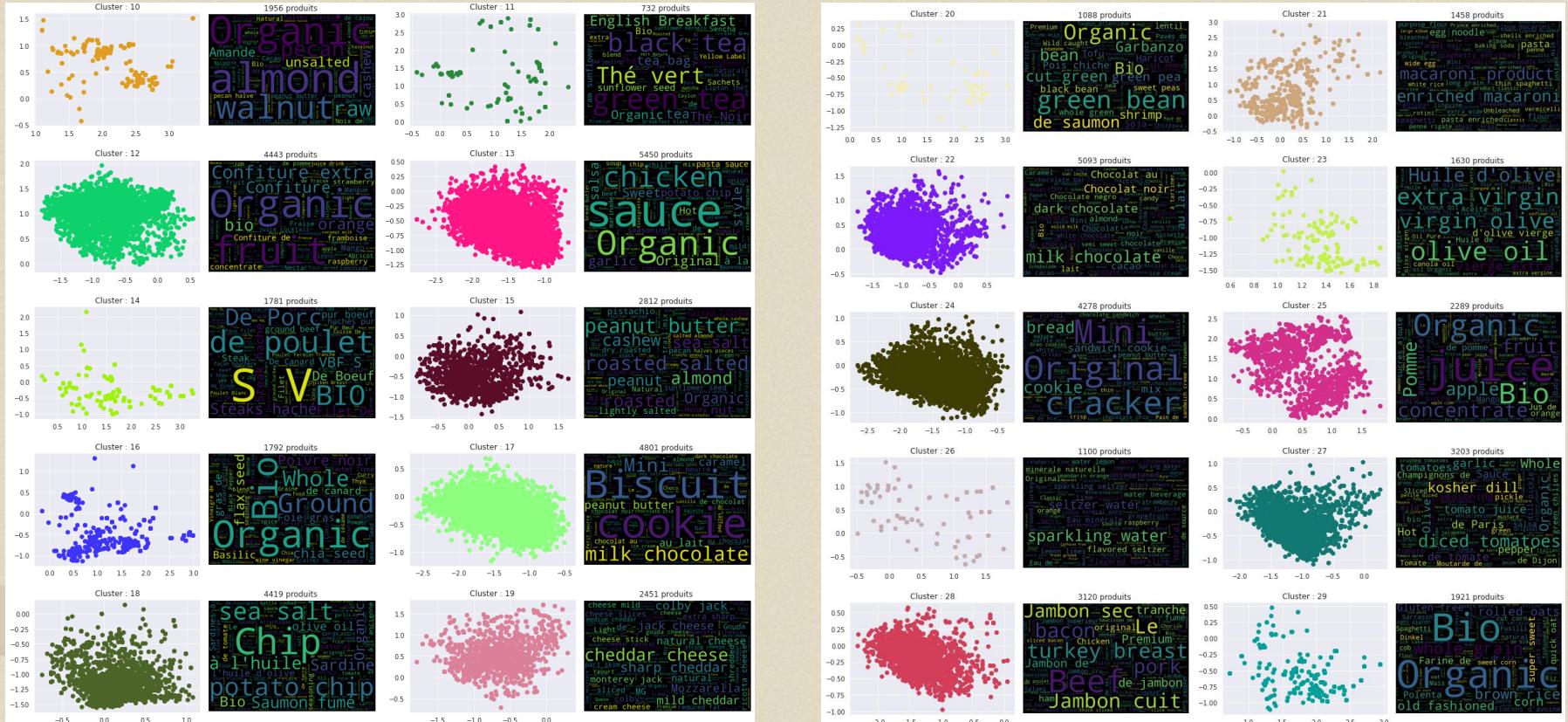
Seuil de similarité : 0.8



Une certaine cohérence semble se dégager des clusters, nous pouvons constater quelques valeurs étranges qui peuvent être dues au choix de leur nombre lors de l'entraînement du Kmeans.

# Troisième méthode : Nuages de mots

Seuil de similarité : 0.8



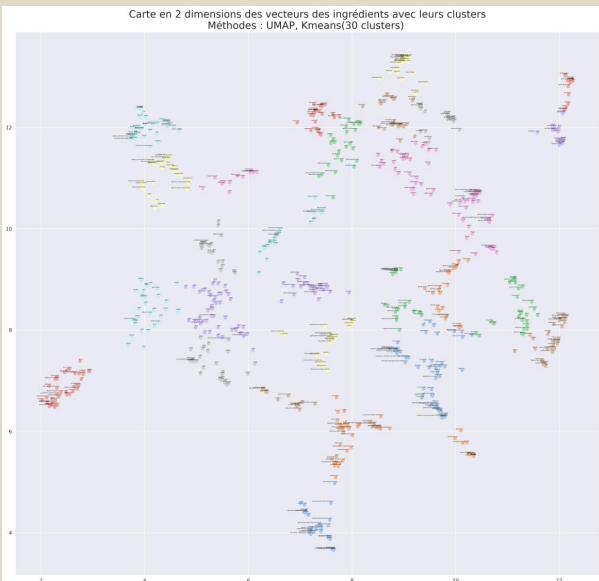


06

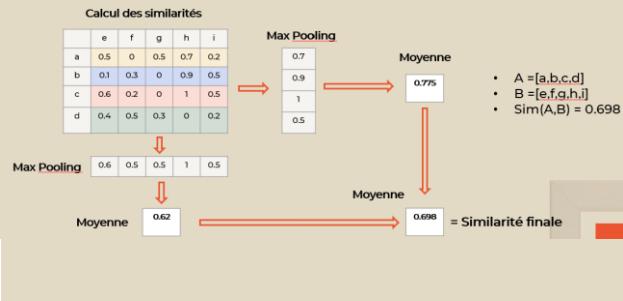
# Conclusion

# Résumé des méthodes et résultats

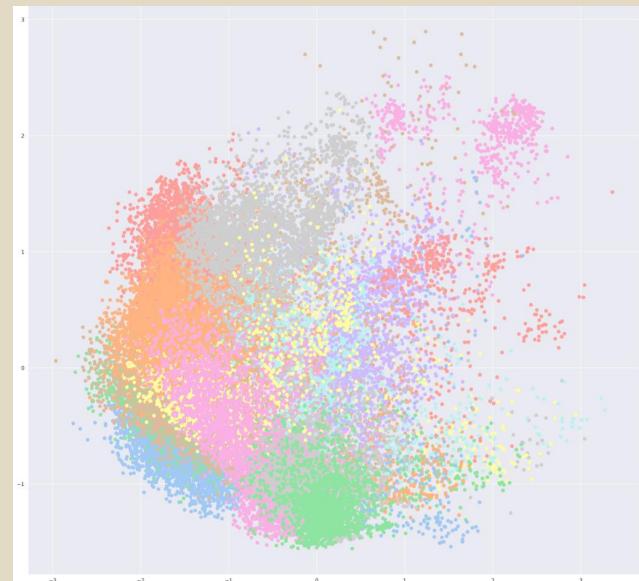
## Partie A :



## Partie B :



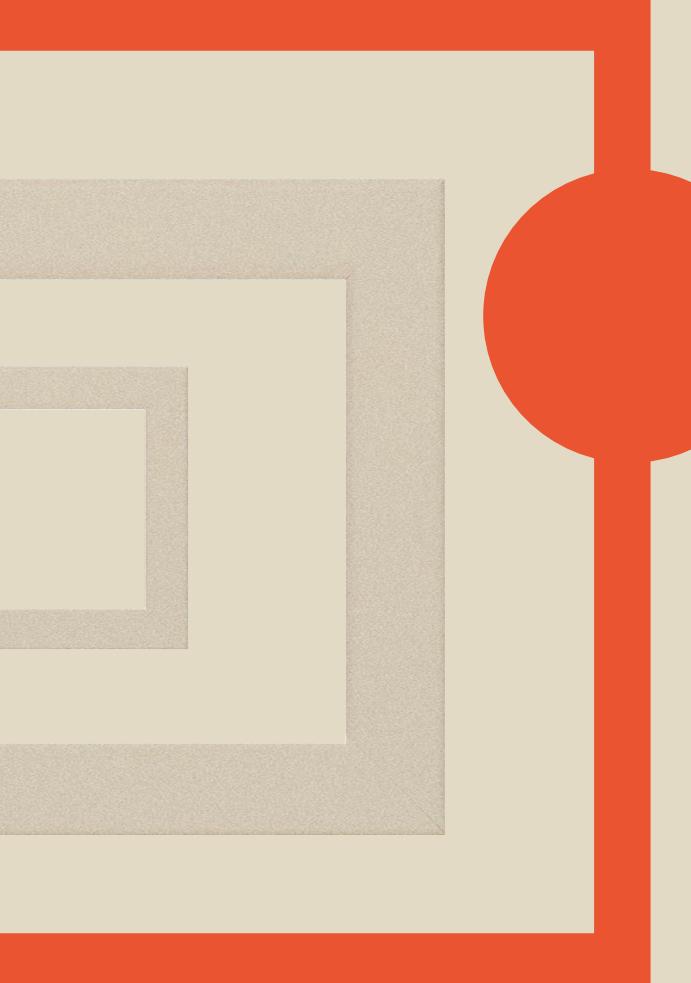
## Partie C :





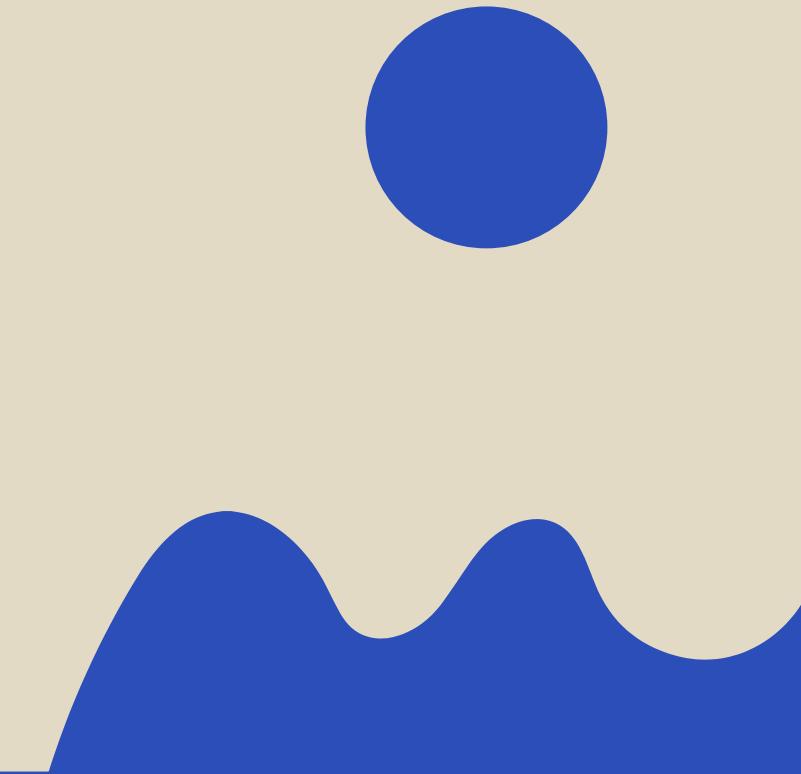
# Conclusion :

- Globalement les résultats obtenus pour les cartes et les clusters semblent corrects.
- En essayant d'entraîner les Kmeans avec des nombres de clusters différents nous pourrions peut être obtenir des résultats plus cohérents.
- Une autre piste d'amélioration pourrait être de tester différentes tailles de vecteur pour la vectorisation des ingrédients.

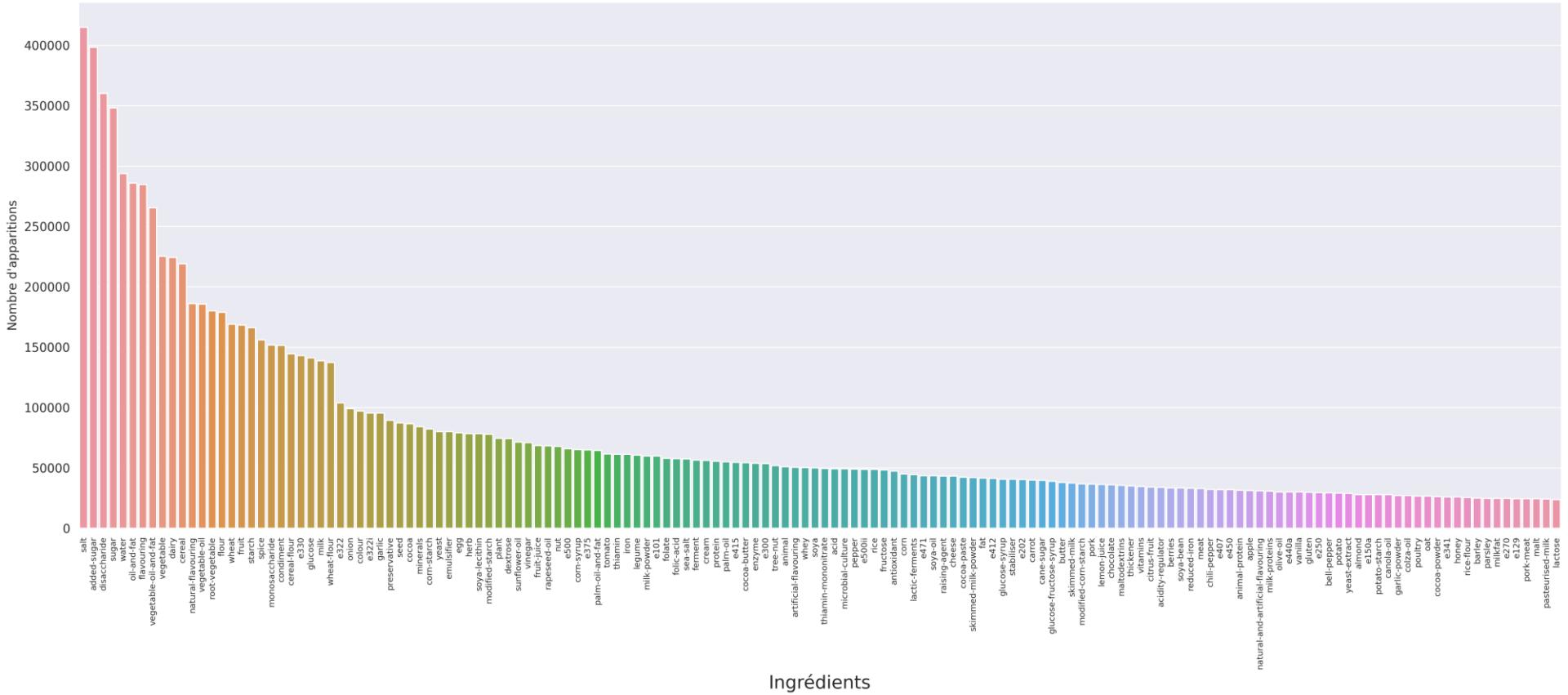


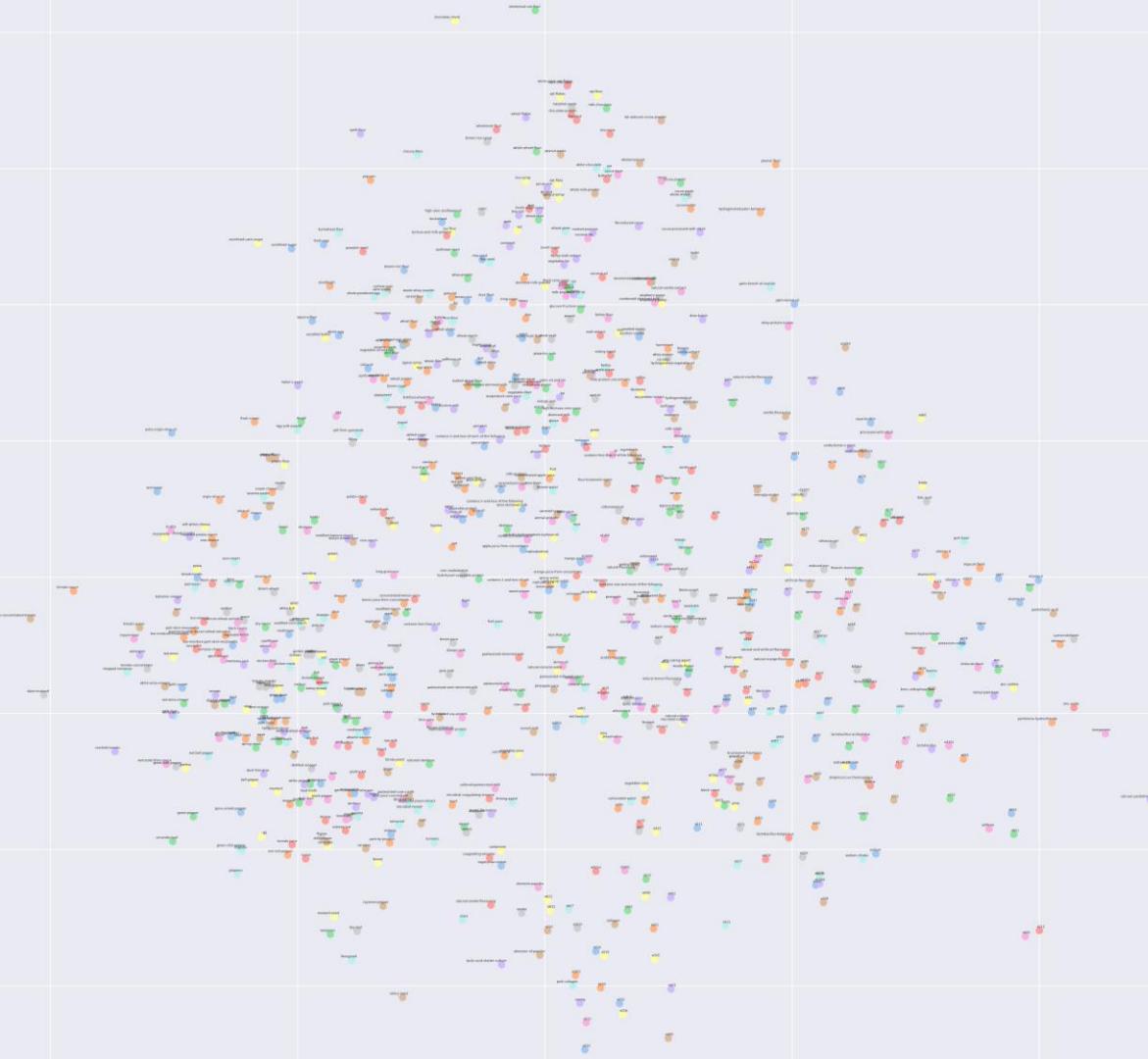
Merci

# Annexes



## Ingrédients avec le plus d'apparitions dans le dataset (Top 150)





## Carte des ingrédients ACP

(On peut zoomer sur le slide pour apercevoir le nom des ingrédients)



## Carte des ingrédients UMAP

(On peut zoomer sur le slide pour apercevoir le nom des ingrédients)



## Carte des ingrédients TSNE

(On peut zoomer sur le slide pour apercevoir le nom des ingrédients)



## Carte des ingrédients UMAP finale (30 clusters)

(On peut zoomer sur le slide pour apercevoir le nom des ingrédients)

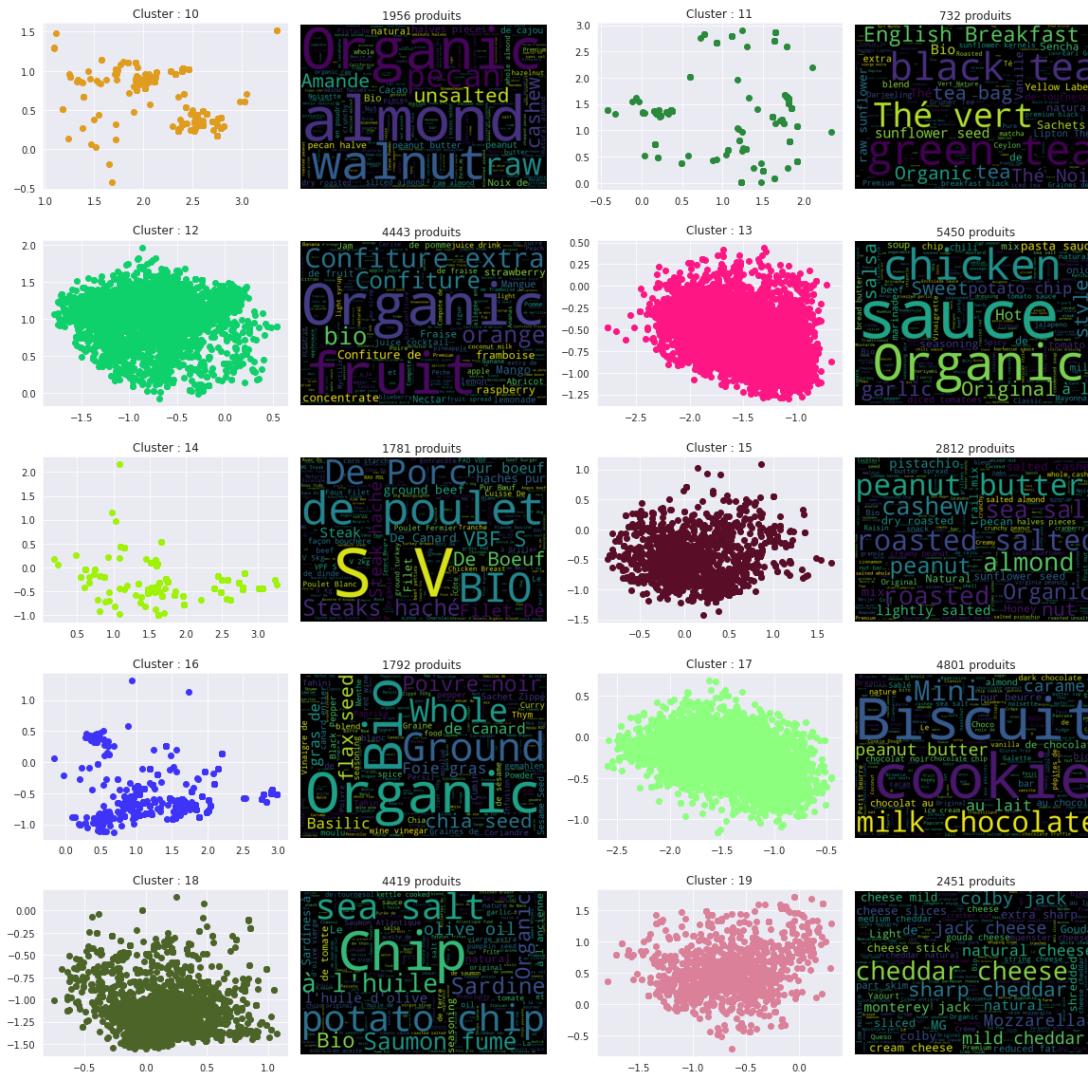


## Carte des produits ACP finale (30 clusters)

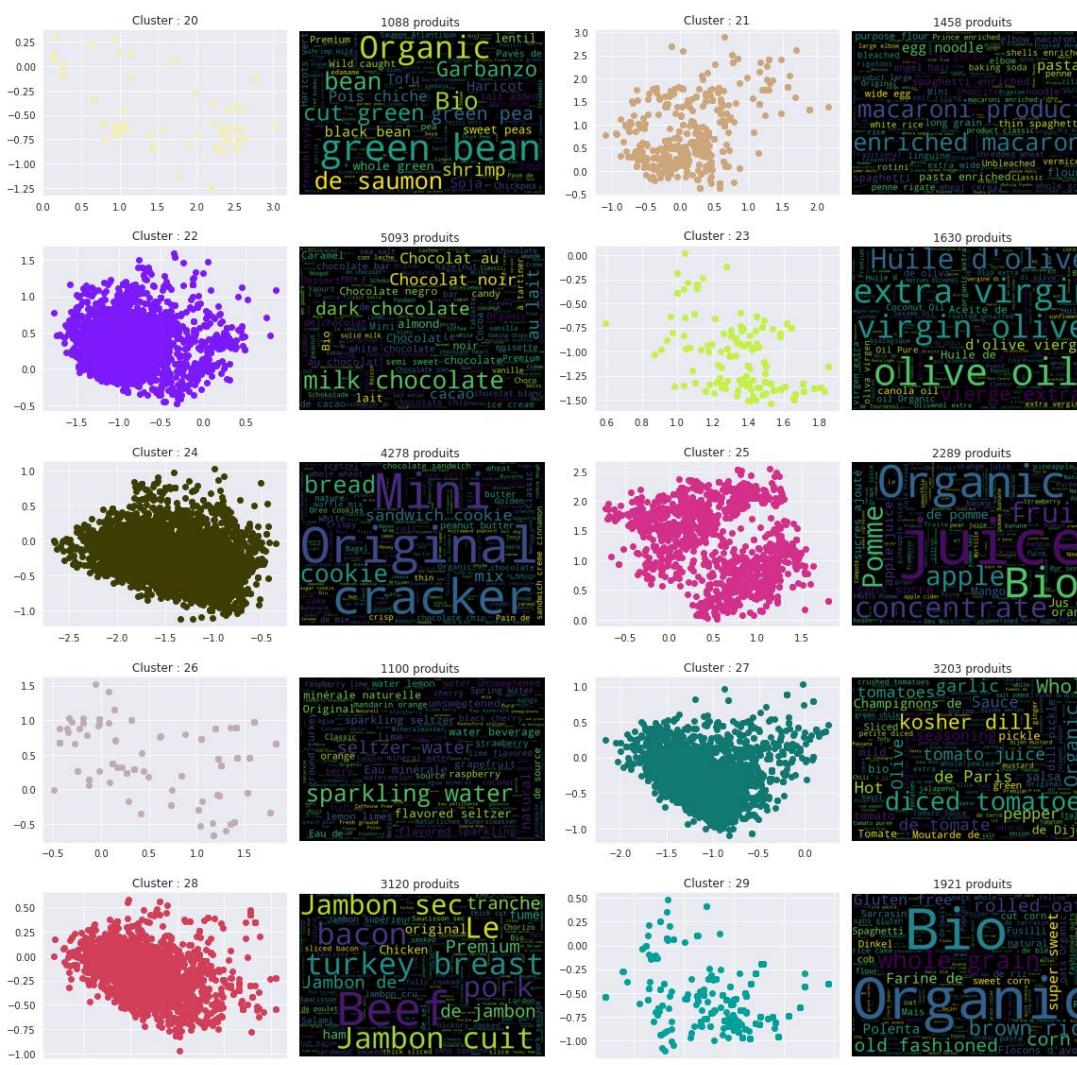
(On peut zoomer sur le slide pour apercevoir le nom des ingrédients)



## Nuage de mots des clusters produits finaux (0 à 9)



## Nuage de mots des clusters produits finaux (10 à 19)



Nuage de mots des clusters  
produits finaux (20 à 29)