ESILV
ENGINEERING SCHOOL
DE VINCI PARIS

OPEN FOOD FACTS

NLP Food Facts projet

NLP projet : Open Food Facts

Binôme :
Nassar Ibrahim
Trang Thomas

Encadrant :
M. Ellul Aline

Remerciements

Nous remercions Madame ELLUL Aline pour avoir assuré les cours de travaux dirigés, ses conseils, sa bienveillance et pour nous avoir encadré tout au long du projet.

Table des matières

1 Présentation Open Food Facts

Open Food Facts est une base de données collaborative contenant les produits alimentaires qui a été créée et alimentée par des volontaires à travers le monde. Tout le monde peut contribuer à l'enrichissement de cette base de données en la renseignant soi même. Le but de Open Food Facts est de partager avec tout le monde un maximum d'informations sur les produits alimentaires.

Il contient plus de 800 000 produits mais qui ne sont pas tous parfaitement décrits. En effet un inconvénient de cette plate-forme pourrait être que les informations ne soient pas totalement exactes ni au bon format. Ce qui engendrerait qu'elles ne soient pas comparables car elles ne sont pas normalisées.

La plateforme a connu un grand développement au niveau de la richesse de sa base de données, en effet le nombre de produits est passé de 250 000 en 2018 à 750 000 en fin 2020 ce qui représente une évolution de plus de plus de +300% que la quantité en 2018.

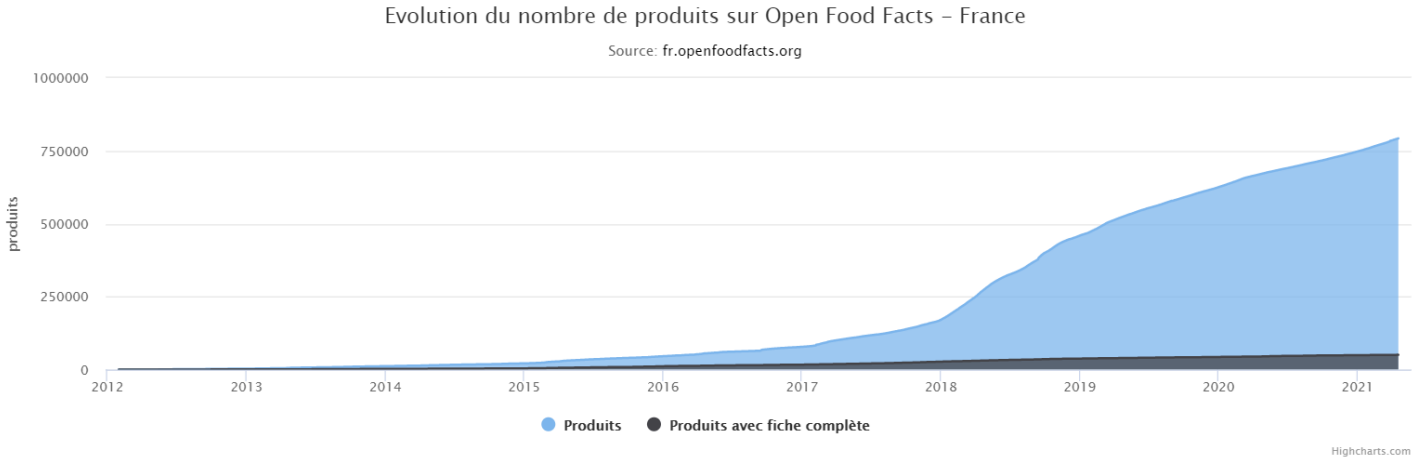

FIGURE 1 – Evolution de la base de données sur Open Food Facts

Parmi les informations les plus importantes sur les produits, nous avons :
— Le nutri-score : la qualité nutritionnelle
— L'éco-score : l'impact environnemental
— La masse du produit
— La catégorie de l'aliment (biscuit, viande, boisson ...)
— Les pays de vente
— Les ingrédients

Ce projet est composé de 3 questions autour desquelles s'articule l'avancement de l'étude de cette base de données.

Nous distinguons plusieurs grandes étapes pour mener à bien ce projet, tout d'abord un travail de **data cleaning et de data processing** qui sera le travail le plus complexe et conséquent étant donné que la data est désordonnée et faussée à certains moments, ces étapes rejoignent en partie la **question 1** étant donné qu'elle concerne la colonne des ingrédients.

Puis nous devrons proposer des approches de **modélisation** notamment en clustering des aliments selon toutes les informations de chaque produit que nous avons à notre disposition, cette étape correspond à la **question 2**.

Enfin nous devrons proposer une **perspective technique** à appliquer sur cette base de données qui serait intéressante à implémenter sur ce projet pour lui ajouter une bonne valeur ajoutée et de l'innovation, ceci répondra à la **question 3**.

2 Data Importation & Exploration

Le dataset présente initialement environ 1,9 million de lignes et 186 colonnes. Pour faciliter la récupération des données qui est très coûteuse en terme de mémoire pour l'ordinateur et en terme de complexité temporelle, le dataset a été importé 50 000 lignes par 50 000 lignes, ce qui fait plusieurs échantillons, il y a au total 40 fichiers csv. Nous pourrons ensuite travailler sur plusieurs fichiers à notre guise.

Nous avons décidé de travailler sur 850 000 produits soit presque la moitié des données. Cela demanderait une ressource importante en terme de calcul pour travailler sur l'ensemble des données.

Parmi les produits du dataset, il n'y pas uniquement des produits alimentaires pour humains car on peut également y retrouver des produits pour animaux, des produits cosmétiques et même des médicaments. Nous trouvons donc tout type de produits.

De plus, il n'y a pas qu'une seule langue dans le dataset. En effet, ce dataset est alimenté à travers le monde, c'est pourquoi nous pouvons y retrouver d'autres langues comme par exemple de l'espagnol, de l'anglais et même du russe, on en revient donc au fait que les données ne sont pas normalisées car nous ne pouvons les comparer entre elles.

Nous avons décidé d'explorer la colonne 'main_category_en' qui représente la catégorie principale traduite en anglais d'un produit. Nous avons réalisé un nuage de mots représentant cette liste de données, cette visualisation permets de voir les catégories les plus récurrentes dans la liste.

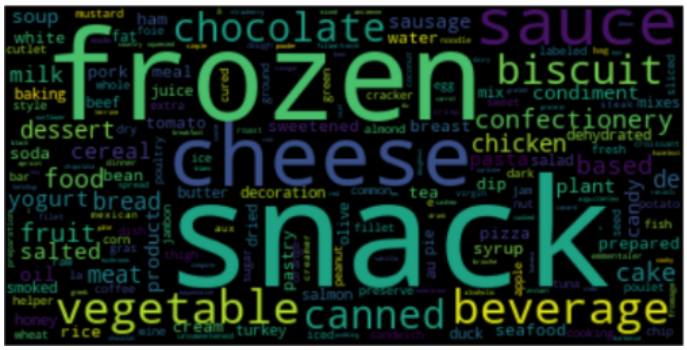

FIGURE 2 – Nuage de mots représentant les mots les plus fréquents parmi les catégories principales en anglais des produits

3 Data cleaning & Processing

Cette partie contient le data cleaning général du dataset qui a été réalisé. Vous pourrez également retrouver le data cleaning de la colonne "ingrédients" dans la question 1 qui est une sous section de cette partie.

Chaque ligne représente un produit alimentaire faisant partie de la base de données d'Open Food Facts. Nous allons d'abord supprimer des colonnes qui ne sont pas nécessaires à la suite du projet de manière à ne garder que celles qui nous serviront à lier les produits entre eux et créer nos clusters. Le dataset est composé de colonnes contenant des informations plus ou moins importantes concernant les produits. Certaines de ces colonnes peuvent être considérées comme des doublons étant donné qu'elles contiennent des données doublons d'autres colonnes, plus précisément les colonnes contenant les suffixes '_tag' ou '_en' ou encore les url des images.

Concernant les colonnes que nous avons gardé, nous avons fait nos choix selon le taux de valeurs nulles par colonne. En effet dans les colonnes des valeurs nutritionnelles (celles qui ont le suffixe '_100g' dans le nom de colonne), nous avons gardé que celles qui ont au minimum 30% de valeurs non nulles. Nous avons gardé les 15 colonnes suivantes :

```
Pourcentage de valeurs Null par colonnes :

   - energy-kcal_100g : 18.62%
   - energy_100g : 16.89%
   - fat_100g : 17.17%
   - saturated-fat_100g : 21.73%
   - trans-fat_100g : 69.61%
   - cholesterol_100g : 69.15%
   - carbohydrates_100g : 17.23%
   - sugars_100g : 19.03%
   - fiber_100g : 61.53%
   - proteins_100g : 17.14%
   - salt_100g : 19.71%
   - sodium_100g : 19.71%
   - calcium_100g : 69.62%
   - iron_100g : 69.79%
   - nutrition-score-fr_100g : 53.99%

Nombre colonnes :  15
```

FIGURE 3 – Sélection des colonnes ayant -70% de valeurs nulles dans les colonnes à suffixe '_100g'

On retrouve dans l'exploration des moyennes des valeurs nutritionnelles. Ici nous pouvons constater que le carbohydrates (glucides) est le composant qui a la plus grande moyenne pour 100g parmi les produits du dataset.

	Mean value
carbohydrates_100g	27.798542
fat_100g	13.227092
sugars_100g	13.073532
proteins_100g	9.392422
nutrition-score-fr_100g	9.128108
saturated-fat_100g	5.279597
fiber_100g	2.858071
salt_100g	2.642948
sodium_100g	1.057228
calcium_100g	0.192252
trans-fat_100g	0.109410
cholesterol_100g	0.044346
iron_100g	0.005055

FIGURE 4 – Moyenne des valeurs nutritionnelles non nulles des produits du dataset

3.1 Question 1

Définir et nettoyer le vocabulaire des ingrédients, trouvez vous des erreurs ? Comment y remédiez vous ? Proposez des solutions pour gérer et identifier les erreurs constatées.

Le but de cette question est de livrer une liste des ingrédients récurrents parmi les produits du dataset étudié. Dans cette partie nous allons expliquer les problèmes rencontrés sur la colonne des ingrédients, et donner une solution utilisée pour gérer ces problèmes. Notre objectif sera de constituer un ”bag-of-words” pour avoir les ingrédients les plus couramment utilisés.

Pour le premier traitement de la colonne ingrédients, nous sommes partis d'une hypothèse assez simple : la colonne des ingrédients contient une liste d'ingrédients et chaque élément est séparé par une virgule. Nous créons alors une seule liste qui contient tous les ingrédients de la colonne.

Ce premier ”split” de la colonne ingrédients nous a permis de relever différents points :
- La langue principale de la colonne est l'anglais
- Il faut gérer les majuscules et les minuscules dans le but d'éviter d'avoir deux éléments qui représentent le même ingrédient (Exemple : sugar/Sugar)
- Il faut gérer la présence de caractère spéciaux et de chiffres dans les ingrédients
- Parmi les ingrédients certains sont constitués d'un mot, certains sont des bigrames et certains sont des trigrames.
- Pour chaque langue, on observe que les ingrédients les plus courants sont les mêmes. En effet on remarque que ”sucre” et ”sugar” figurent parmi les éléments les plus utilisés. De même pour ”sel” et ”salt” ou encore ”eau” et ”Water”.

	A	B	C	D	E
1	**index**	**count**		**index**	**count**
2	salt	217958		corn starch	19104
3	sugar	120997		Sugar	18562
4	water	110715		whey	18499
5	citric acid	57139		Water	18386
6	sel	44490		vinegar	18025
7	niacin	43517		garlic	17529
8	dextrose	39439		yeast	17214
9	riboflavin	39371		garlic powder	16782
10	corn syrup	37793		enzymes)	16601
11	natural flavor	35471		modified corn starch	16125
12	soy lecithin	33610		caramel color	16005
13	reduced iron	32560		onion powder	15933
14	folic acid)	31642		palm oil	15551
15	spices	30638		enzymes	14822
16	sea salt	27063		cheese culture	14685
17	eau	26857		cheese cultures	14162
18	xanthan gum	25758		canola oil	14122
19	soybean oil	25727		thiamin mononitrate	13359
20	sucre	25265		carrageenan	13109
21	thiamine mononitrate	25081		modified food starch	12972
22	natural flavors	23906		malted barley flour	12599
23	high fructose corn syrup	21004		lactic acid	12378
24	salt)	20465		cream	12108

FIGURE 5 – Termes les plus récurrents suite au premier split

Pour avoir une meilleure division des termes de la colonne, nous passons par la tokenisation. Avec cette méthode, nous souhaitons avoir une liste de chaque élément distinct présent dans la colonne. Cela permettra par la suite d'éliminer les éléments parasites tels que les caractères spéciaux et les chiffres.

Dans un premier temps nous nous intéressons aux ingrédients composés d'un seul mot (1-gram). Pour cela, nous supprimons tous les stopwords. Pour identifier les ingrédients, nous procédons au calcul de fréquence de chaque mot.

Pour réaliser ce traitement, nous nous basons sur la librairie nltk et ses attributs pour supprimer les stopwords. Mais avant de réaliser cette tâche, nous passons par une étape de normalisation : nous mettons tous les éléments en minuscule. Par la suite, pour supprimer tous les caractères spéciaux nous filtrons notre liste par **l'expression régulière (RegEx)** suivante : [a-zA-Z]+. Pour finir, nous utilisons la lemmetization pour avoir la racine de chaque mot.

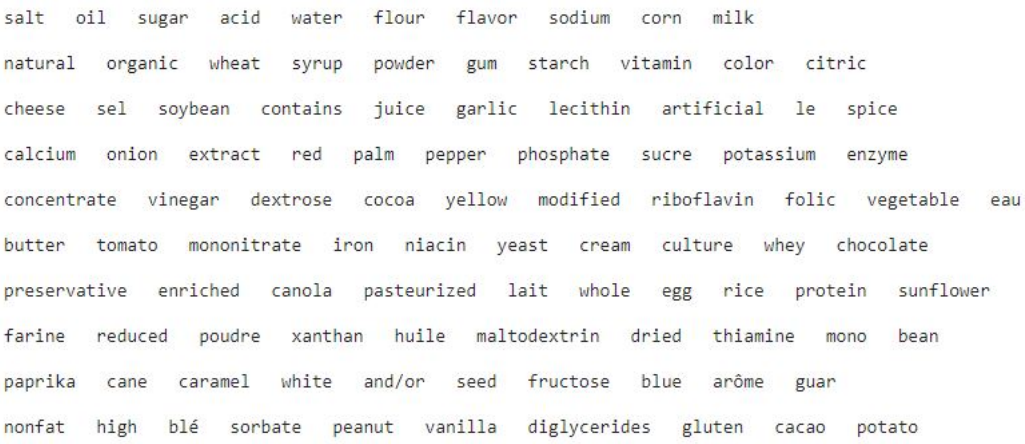

FIGURE 6 – 100 éléments les plus fréquents après la tokenisation

En explorant les termes les plus fréquents, nous pouvons voir que certains sont des adjectifs. Pour palier à ce problème nous utilisons les tags. Cette fonctionnalité de la librairie nltk nous permet de ne garder que les tokens qui sont des noms propres.

Dans les listes d'ingrédients, certains ingrédients sont eux-mêmes composés d'ingrédients, nous avons donc décidé de prendre le sous ensemble des ingrédients au lieu de prendre l'ingrédient lui même pour avoir un degré de détails plus important. Par exemple, si le chocolat noir est un ingrédient dans le produit étudié, nous allons garder les ingrédients du sous ensemble "chocolat noir" et ne pas mettre "chocolat noir" dans la liste d'ingrédients, ainsi le degré de détails sera plus commun à tous et générique.

Il était alors intéressant d'explorer les bigrams. De la même façons que les unigrams, nous procédons à un calcul de fréquence des bigrams possibles dans notre colonne d'ingrédients. Cela nous permet d'élargir notre bag-of words avec l'ensemble des ingrédients. Comme on peut le voir sur la figure ci-dessous, un grand nombre de ces bigrams font sens et ont bien leur place dans la liste d'ingrédients.

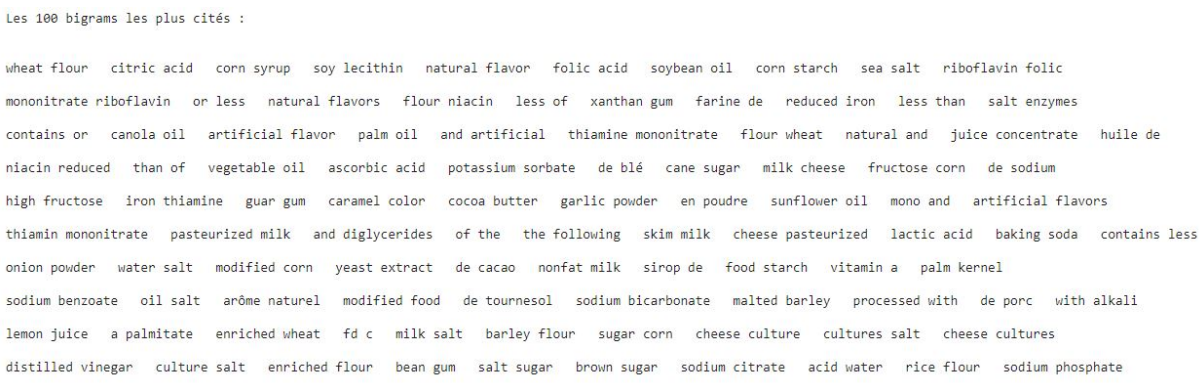

FIGURE 7 – 100 bigrams les plus fréquents après la tokenisation

Nous retrouvons dans cette colonne des fautes d'orthographes sur certains produits étant donné que les particuliers peuvent alimenter la base de données open source, ce qui serait potentiellement à l'origine de ces anomalies. Ces fautes d'orthographes peuvent être problématiques car elles peuvent nous empêcher de mener à bien tout notre étape de data processing (notamment pour la lemmatization pendant laquelle la racine du mot ne risque pas d'être reconnue) et préparation de la donnée pour l'entraînement.

Il y a également des cas dans lesquels l'encodage des caractères n'est pas bien traduit (UTF-8), nous nous sommes retrouvés avec quelques caractères inconnus dépourvus de sens. De plus il y a également des espaces manquants ce qui crée des mots inexistants formant une fusion de mots.

Le bag of word permet de contourner un grand nombre de ces anomalies suites au différents traitementd appliqués à la colonne des ingrédients. Le dataset étant assez large, nous ne pouvons malheureusement pas contourner l'ensemble des anomalies. L'idée de bag of words généralisé nous a donc paru intéssante pour avoir un ensemble d'ingrédients le plus général possible pour décrire les aliments.

4 Modélisation

Une fois la préparation des données faite, nous pouvons ensuite commencer la modélisation. Le machine learning se divise en deux types d'apprentissages : l'apprentissage supervisé et l'apprentissage non supervisé.

Dans notre cas, nous travaillerons sur un sujet d'apprentissage non supervisé, c'est-à-dire que nous n'allons pas chercher à prédire une catégorie ni une valeur mais à rapprocher les instances similaires dans la donnée qui nous appartient.

4.1 Question 2

Sur la base des faits nutritionnels et/ou des catégories d'aliments, proposer des approches de clustering et une visualisation de certaines catégories de produits. Trouver des valeurs aberrantes (un produit très différent des autres du même groupe). Il existe des produits très similaires en termes de valeur nutritive mais très différents en termes de catégories ou ingrédients ?

Le but de cette question selon nous est de rassembler les produits selon leurs catégories d'aliments, les composants nutritionnels et les ingrédients. Dans cette question nous avons réalisé **3 KMeans** suivant différentes approches.

Pour commencer, la première approche que l'on a eu était de réaliser un K means basé sur les données nutritionnelles numériques du dataset, c'est-à-dire les colonnes dont le noms contiennent '_100g'. Pour **ce premier KMeans**, nous avons voulu faire un modèle en nous concentrant sur les deux colonnes que nous trouvions les plus intéressantes, en l'occurrence 'energy_kcal_100g' et 'fat_100g'.

Le KMeans est un algorithme non supervisé de machine learning qui va trouver des patterns entre les données et les clusteriser en un nombre k de clusters.

En explorant la donnée sur les colonnes 'energy_kcal_100g' nous avons remarqué la présence d'outliers (valeurs aberrantes). En effet, entre la valeur du 3ème quartile (367 kcal pour 100g) et la valeur maximale (8.86*10e10 kcal pour 100g), nous constatons la présence de valeurs qui fausse la répartition des clusters notamment en engendrant la création d'un cluster pour certains des outliers.

	energy-kcal_100g	fat_100g
count	8.500000e+05	850000.000000
mean	1.045739e+05	10.955891
std	9.615167e+07	38.275926
min	0.000000e+00	0.000000
25%	2.600000e+01	0.000000
50%	1.790000e+02	3.400000
75%	3.670000e+02	17.307692
max	8.864745e+10	29000.000000

FIGURE 8 – Présence d'outliers dans la colonne 'energy_kcal_100g'

Nous avons décidé de ne garder que les données entre le premier et le troisième quartile pour ne pas avoir trop de valeurs aberrantes et ne pas fausser la formation des k clusters. En comparant les données nutritionnelles, nous avons constaté que par exemple des frites surgelées à frire se situent dans la moyenne de répartition des produits avec 136 kcal pour 100g pour une valeur médiane de 179 kcal pour 100g. C'est ainsi que nous avons vérifié la qualité de la donnée concernant cette colonne.

Ingrédients / Composition

Ingrédients

Ingrédients : pommes de terre (96,5%), huile de tournesol (3,5%).

Valeurs nutritionnelles

	Valeurs nutritionnelles pour 100 g	Taux d'apports journaliers pour 100 g*
valeur énergétique (kJ)	570 kJ / 100 g	8 %
valeur énergétique (kcal)	136 kcal / 100 g	-

FIGURE 9 – Comparaison des frites avec les données statistiques de l'énergie en kcal

Ensuite, nous avons appliqué l'algorithme des K means en choisissant d'avoir 4 clusters. Nous obtenons grâce aux valeurs nutritionnelles des clusters qui sont bien répartis et équitables. Nous voyons ensuite que la répartition de produits selon les 4 clusters est équitable car nous avons presque le même nombre de produits dans les 3 premières catégories tandis que dans la dernière catégorie il y a moins de produits.

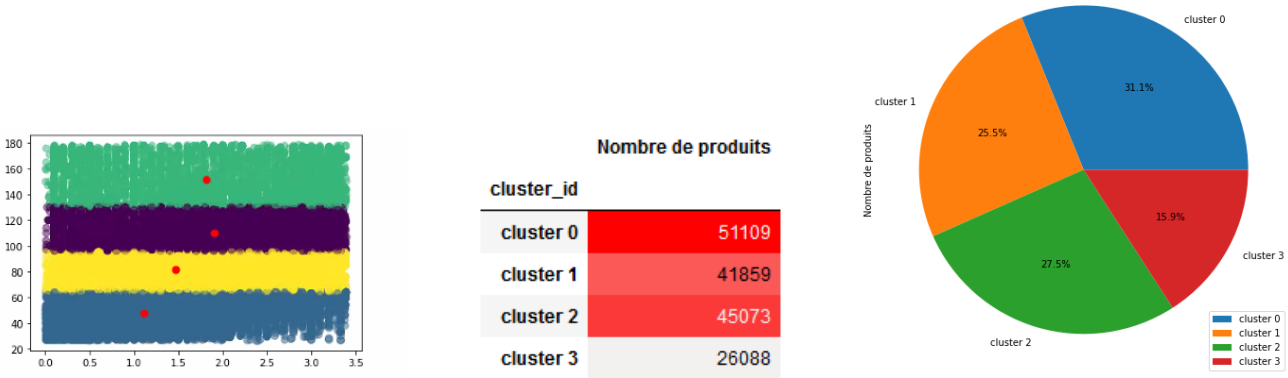

FIGURE 10 – Représentation des clusters selon la première approche

Nous avons ensuite voulu visualiser les produits qui appartenaient à chacun des clusters pour voir s'ils étaient bien regroupés. Selon nous, les clusters réalisés pour ce premier KMeans sont encourageants mais pas suffisants pour notre étude, cela s'explique par le fait que nous avons beaucoup de valeurs nulles parmi les valeurs nutritionnelles ce qui va fausser la creation de nos clusters. C'est pourquoi nous allons réaliser ensuite un second KMeans dans lequel nous allons prendre uniquement des produits ayant des valeurs non nulles.

FIGURE 11 – Produits des 4 clusters du premier KMeans

Pour le deuxième KMeans, nous allons étudier les données des 15 colonnes contenant le suffixe '_100g' que nous avons préalablement sélectionné. Dans un premier temps, nous avons réalisé un KMeans avec 8 clusters pour les produits qui contiennent uniquement des données non nulles sur l'ensemble des 15 colonnes. Nous décidons d'en étudier les centroids des 8 clusters : nous constatons que certaines des standards deviations (écart-types) des 15 colonnes de ce KMeans sont supérieurs à 100 ce que nous considérons dispersé. Les colonnes concernées sont :

- 'energy-kcal_100g'
- 'energy_100g'
- 'fat_100g'
- 'carbohydrates_100g'
- 'sugars_100g'

	energy-kcal_100g	energy_100g	fat_100g	saturated-fat_100g	trans-fat_100g	cholesterol_100g	carbohydrates_100g	sugars_100g	fiber_100g	proteins_100g	salt_100g	sodium_100g	calcium_100g	iron_100g	score-fr_100g
count	8.000000	8.000000	8.000000	8.000000	8.000000	8.000000	8.000000	8.000000	8.000000	8.000000	8.000000	8.000000	8.000000	8.000000	8.000000
mean	5324.486993	22232.305392	77.191451	12.615867	0.042312	0.029338	340.384686	111.923678	27.830848	44.823894	2.979961	1.194484	0.075602	0.015141	13.148852
std	14012.769337	58502.310505	152.557730	16.275231	0.038697	0.020567	872.825981	278.146892	69.582929	103.145061	5.351244	2.147556	0.048851	0.035527	10.333943

FIGURE 12 – Statistiques des centroids des 15 colonnes

Nous décidons donc de supprimer 10% des outliers de ces 5 colonnes, ce qui fait qu'il nous restera 112 000 produits restantes. On refait ensuite un Kmeans avec 8 clusters. Les résultats de ce second KMeans sont également intéressants car nous retrouvons une tendance au sein des 8 clusters avec plusieurs produits qui se ressemblent cependant ce n'est pas assez pour affirmer que toutes nos données sont parfaitement clusterisées.

Enfin pour le troisième KMeans, nous avons décidé de nous concentrer sur des données textuelles de la colonne "catégories_en" rédigée en anglais. Tout d'abord, nous avons appliqué un nettoyage textuel qui est similaire à celui qui a été appliqué dans la question 1.

Puis pour modéliser, nous avons utilisé le **Word2vec** qui est un algorithme utilisé pour le word embedding. Le **word embedding** consiste à vectoriser des mots, cela va permettre de représenter un mot par un vecteur de valeurs numériques.

Nous avons ensuite un vecteur de 100 valeurs qui va représenter un mot. Tous nos mots sont représentés par des vecteurs de taille 100. Notre objectif est d'avoir **un seul** vecteur de 100 valeurs pour un produit, cependant 1 produit peut avoir plusieurs catégories. C'est pourquoi nous allons faire la moyenne de tous les vecteurs qui représentent une catégorie pour un produit. Ensuite nous appliquons notre KMeans à nos produits.

Les résultats de ce KMeans sont très satisfaisants car nos produits sont bien regroupés selon le nom de leurs catégories. Les résultats de cette étude peuvent être retrouvés dans le notebook auquel vous pourrez accéder via le lien dans la dernière section de ce rapport.

Pour avoir une visualisation des données, nous avons opté pour une simple ACP (Analyse en composantes principales) pour avoir une représentation en deux dimensions. Cela permet d'avoir une approche visuelle des clusters.

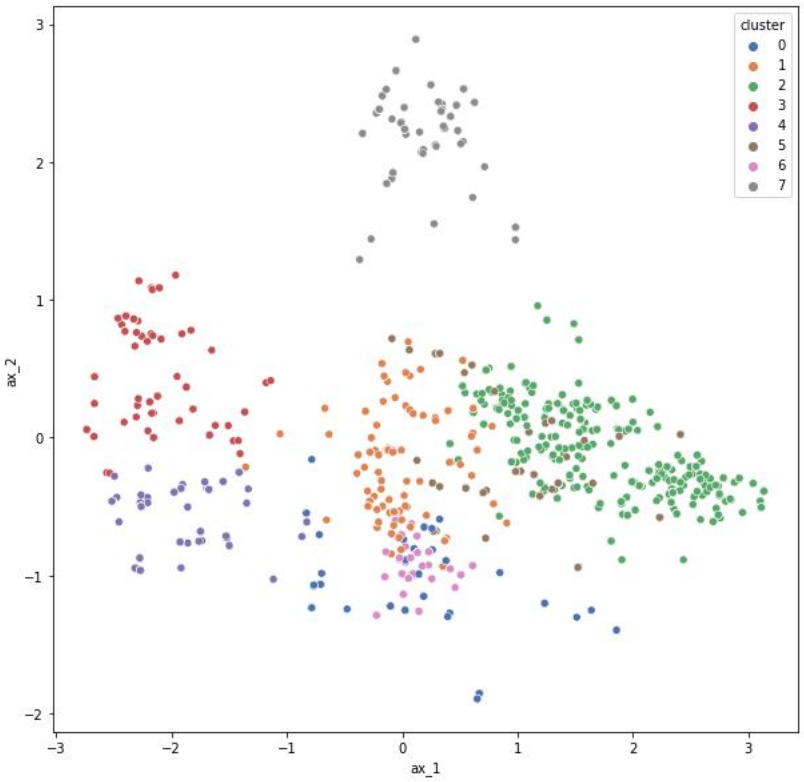

FIGURE 13 – Projections des données suite à l'ACP

Nous avons divisé nos données en 8 clusters (nombre de cluster par défaut pour la librairie Python Scikit-Learn). Certains clusters sont bien démarqués des autres comme on peut le voir, notamment les clusters 7, 3 et 2. Cependant on remarque qu'un grand nombre des clusters se chevauchent sur la visualisation. Selon nous cela est dû au nombre de catégories de produits présents dans le dataset. Le nombre de clusters n'est pas suffisant pour avoir une bonne démarcation des différents groupes.

Pour avoir le nombre optimal de clusters, nous avons utilisé la méthode Elbow. Cette méthode consiste à réaliser plusieurs K-means en faisant varier la valeur K des clusters. Pour chaque modèle réalisé, nous calculons la distorsion des clusters : c'est la somme des carrés des distances des points avec le centre de leur cluster. Cela nous permet d'anticiper la valeur de K qui diminuerait la distorsion du modèle.

Nous avons calculés différentes distorsions pour des valeurs de K allant de 15 jusqu'à 250 avec un pas d'avancement de 30. Cela nous permet d'avoir des calculs plus rapides à exécuter et une visualisation plus précise. On remarque que la tendance de la courbe est quasiment la même à partir de 100 clusters. Cette analyse nous permet de confirmer l'hypothèse portée sur le nombre important de catégories de produits présents dans le dataset.

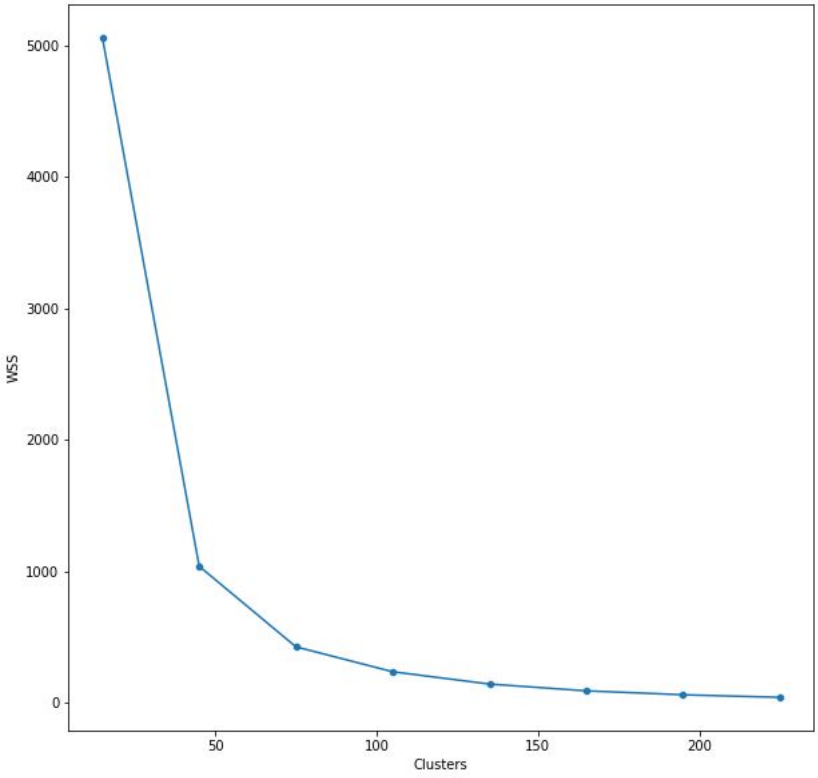

FIGURE 14 – Distorsion du modèle en fonction du nombre K de clusters

Enfin nous avions pensé à une autre piste pour clusteriser nos données sur cette étude, cependant étant donné le travail conséquent déjà réalisé sur le clustering, nous n'avons pas pu approfondir la partie suivante : nous avons créé une colonne que l'on a intitulé "composition nutritionnelle", dans laquelle nous fusionnons le contenu des colonnes terminant par le suffixe '_100g' et le nom de la colonne. Cette colonne est la composition nutritionnelle de chaque produit. Nous avons concaténé la valeur nutritionnelle et le nom du nutriment comme dans l'exemple suivant.

L'objectif de cette idée était de clusteriser les données selon les compositions nutritionnelles mais en utilisant les données d'un point de vue textuel étant donné que Word2vec prend en compte le contexte et les mots entourant le mot étudié, nous avons déduit que la structure de la colonne sur

```
Product name : Compote banane framboise lait de coco acérola
Liste nutritionnelle : energy-kcal_100g 77.0 energy_100g 322.0 fat_100g 2.1 saturated-fat_100g 1.8 carbohydrates_100g 13.0 suga
rs_100g 11.0 proteins_100g 0.7 salt_100g 0.01 sodium_100g 0.004
```

FIGURE 15 – Exemple de la colonne créée sur la liste nutritionnelle pour le produit "Compote banane framboise lait de coco acérola"

les valeurs nutritionnelles qui a été créée pouvait être intéressante. En effet, la quantité nutritionnelle suit le nutriment dans la chaîne de caractère, nous déduisons donc selon nos recherches que le modèle prendra en compte le contexte de la phrase. Ensuite, après avoir vectorisé nos données textuelles, l'idée est de clusteriser en utilisant le modèle des K-Means.

5 Perspectives : Question 3

Sur la base de votre expertise sur cet ensemble de données, proposez et décrivez un modèle (aucun code requis) qui serait intéressant pour enrichir le projet OpenFoodFacts.

Cette question vise à tester notre compréhension du sujet et à constater les problématiques réelles d'un projet. Parmi les axes de réflexion abordés concernant l'optimisation et l'enrichissement de ce projet nous avions pensé à **la problématique de la perte de la donnée**. Nous entendons par la perte de données, le fait de ne pas être capable de tirer avantage de l'intégralité des données simultanément.

C'est pourquoi nous avons pensé à grouper les données qui sont dans des langues différentes. En effet, nous avons constater dans ce projet, qu'il y avait plusieurs langues, par conséquent les produits similaires décrits dans des langues différentes ne peuvent être clusterisés. Pour résoudre ce problème de différence de langue, nous pourrions utiliser dans un premier temps un outil (librairie) qui permettrait de détecter une langue. Nous choisirions une langue de référence pour notre dataset en fonction de la langue qui sera la plus présente dans le dataset. Ensuite nous ferions appel à un traducteur qui permettrait ainsi de traduire toute la donnée dans la langue que nous avons ciblée, tout en évitant de traduire les données qui sont déjà dans la bonne langue d'où la présence du détecteur de langue, cela permettrait d'économiser une important complexité temporelle. Cette action nous permettrait de regrouper la quantité de données à notre disposition et de renforcer notre modèle en lui donnant des données qui sont dans la même norme, qui est dans notre cas : la langue. Cela éviterait de devoir faire du scrapping (récupération de données libre), ni de créer nous mêmes des données.

Nous pensions également à une autre innovation concernant ce dataset. En faisant notre exploration, nous nous sommes rendus compte que 42.02% des valeurs dans la colonne 'categories' sont nulles.

FIGURE 16 – Pourcentage de valeurs nulles dans la colonne catégories

Selon nous, la catégorie d'un aliment est une donnée essentielle dans sa fiche de présentation, ne serait-ce pour identifier ou pour rechercher le produit. C'est pourquoi, nous avons pensé qu'il serait intéressant d'implémenter un modèle qui prédirait la catégorie de l'article en fonction de sa liste d'ingrédients et de sa composition nutritionnelle.

Pour la partie apprentissage de cette étude, nous utiliserions tous les produits qui ont une catégorie et leur liste d'ingrédients et nutritionnelle correspondant. Pour exploiter les données textuelles, nous aurions recours au word embedding (la vectorisation des données textuelles) afin de pouvoir trouver des patterns entre les descriptions des produits.

Nous utiliserions ensuite notre modèle sur les données non catégorisées pour réaliser nos prédictions afin de leur attribuer un label. Cette évolution aurait permis de pouvoir réaliser un clustering selon les catégories avec un dataset d'entraînement plus fournit.

6 Lien du notebook sur Google Collaboratory

Voici le lien menant à notre notebook :

Notebook : `https://drive.google.com/file/d/101SHMLwmeJNqD01Kt7Muv1SWzaoEhvZ1/view?usp=sharing`
Datasets : `https://drive.google.com/drive/folders/1mV3DiYvMLAzEow9pa_KtnF2a4k4l6rFf?usp=sharing`