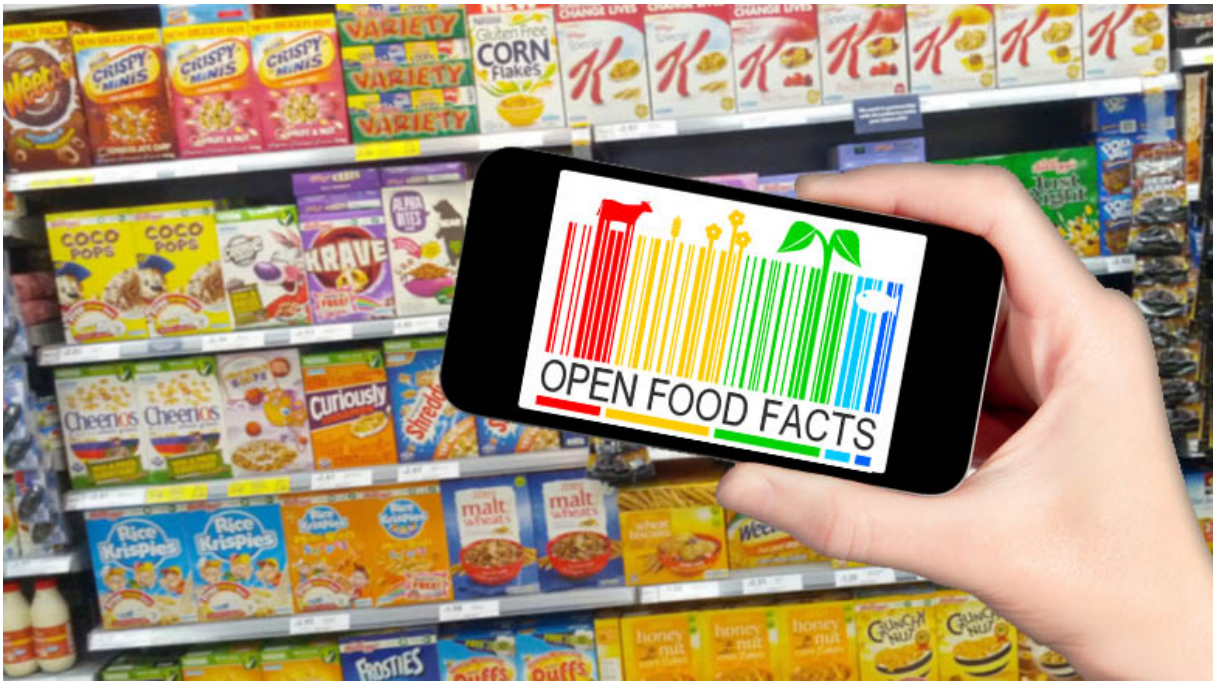


Advanced Machine Learning for NLP and Text Processing

Project 1 : OpenFoodFacts

Yanis FADILI & Lies HAOUAS
Professeur : Kezhan SHI

Sommaire

Introduction

L'objectif du projet est ici de travailler sur un jeu de données très lourd (2 millions de données) provenant du site *openfoodfacts.org*, afin de trouver certains liens entre différents produits référencés selon leurs informations nutritionnelles dans une démarche d'optimisation du site en faisant intervenir l'intelligence artificielle.
Le projet est développé en Python 3 sur Google Colab, dont le lien se trouve ci-après :
https://colab.research.google.com/drive/1RjWBO0jUrwsEyTfr-HDycDXaqAOoDvMG?usp=sharing

Importation des librairies et du dataset

Tout d'abord, nous nous sommes rendus sur le site Open Food Facts, catégorie *data* afin d'importer les données du site, au format .csv.
Le dataset étant très lourd : plus de 2 millions de données repartis sur 4,62 Go, nous avons donc été contraints de travailler sur un sample, de 50000 lignes et 187 colonnes pour des raisons de rapidité des traitements.

1) Define and clean the vocabulary of ingredients, do you find some mistakes ? How do you manage them?

On s'intéresse à la colonne « ingredients_text », qui concerne la liste des ingrédients d'un produit.

Tout d'abord, comme les données ne sont pas toutes en français, on utilise la librairie deep_translator en important le module Google_translator, qui permet de traduire un texte inférieur à 5000 caractères, lorsque des termes non-français sont présents dans le texte.

Propose solutions to manage/identify errors.

Pour identifier les erreurs, on créé des tableaux contenant les données texte à analyser, et on les parcourt afin de trouver ou non certains caractères problématiques présentes dans le texte.

Pour gérer les erreurs, on fait un « data cleaning » afin de rendre le texte exploitable : on enlève la ponctuation, les caractères non-alphabétiques, les accents, les espaces en trop, ainsi que les « stop-words ».

Entrée [26]:
```
from deep_translator import GoogleTranslator
translated = GoogleTranslator(source='auto', target='fr').translate(ingredient.iloc[5780,1])
translated
```

Out[26]:
```
'FARINE ENRICHIE (FARINE DE BLÉ, NIACINE, FER RÉDUIT, MONONITRATE DE THIAMINE, RBOFLAVINE, ACIDE FOLIQUE), SUCRE, SHO
RTENING VEGETAL AQUEUX (HUILE DE SOYA INTERESTÉRIFIÉE, SOJA OL HYDROGÉNÉE HUILE DE COTON HYDROGÉNÉE, ALT DE COTON CON
TIENT : BLÉ, LAIT DISTRIBUÉ PAR KROGER CO, CINCINNATI, OHIO 45202'
```

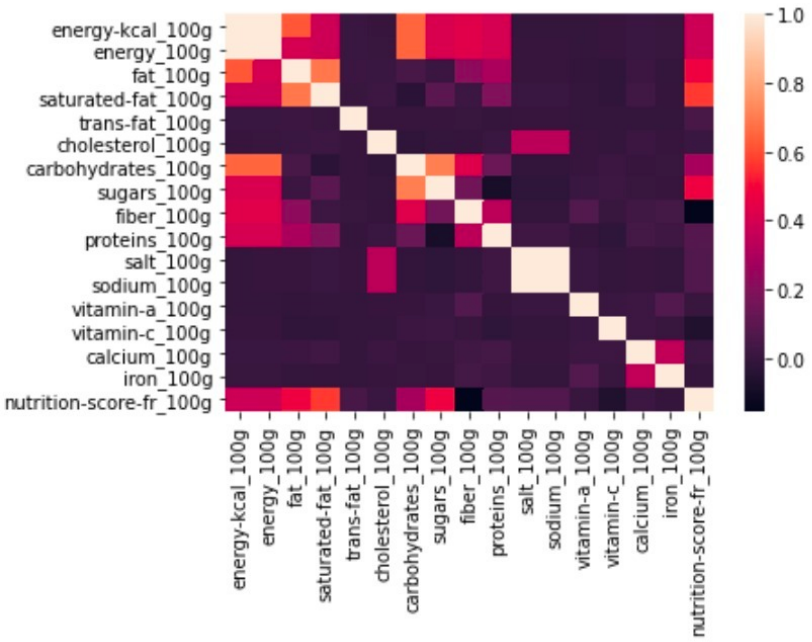
Entrée [27]:
```
#Je travaille que sur 5000 lignes car la traduction prend enormément de temps

traduction=[]

for i in range(0,5000):
    if ingredient['ingredients_text'].isnull()[i]==True:
        #print('vide')
        traduction.append("")
    if ingredient['ingredients_text'].isnull()[i]==False:
        if ingredient.iloc[i ,2]=="en:france":
            traduction.append(ingredient.iloc[i ,1])
        elif ingredient.iloc[i ,1][1] in ['0','1','2','3','4','5','6','7','8','9']:
            traduction.append("")
        else :
            translated = GoogleTranslator(source='auto', target='fr').translate(ingredient.iloc[i,1][:5000])
            #print(translated)
            traduction.append(translated)
```

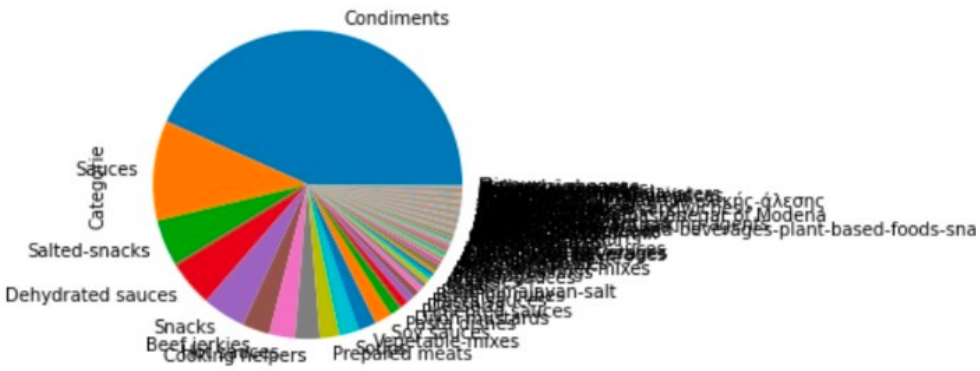
2) Based on nutrition facts and/or food categories, propose clustering approaches and a visualisation of some categories of products.

En explorant les différentes colonnes d'informations nutritionnelles, on peut de prime abord trouver certains liens entre ces dernières. Par exemple, on remarque une corrélation entre la quantité de sel et de sodium. Pour vérifier cela, on décide de mettre en place une matrice de corrélation afin de mieux visualiser quelles informations sont liées.



Ensuite, on tente une approche de clustering afin de remarquer quels liens existe-t'il entre les différentes catégories de produits et les différentes informations nutritionnelles ? Pour cela, on décide de faire un pie chart afin de visualiser dans quelle catégorie trouve t'on le plus de sodium ou de gras. Logiquement, on trouve de sodium dans les snacks, sauces, et les condiments, et de gras dans les huiles végétales.

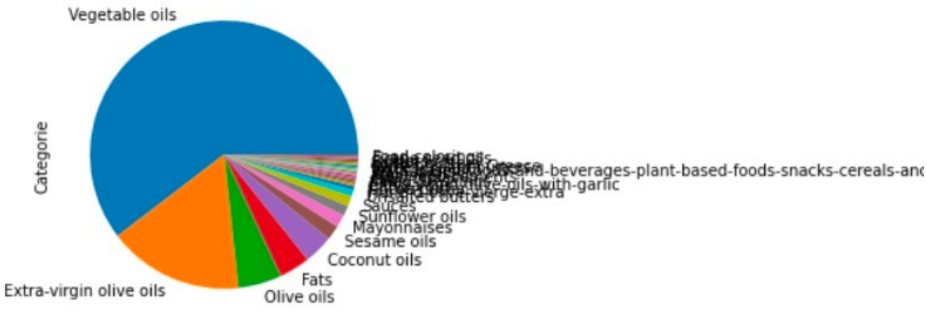
```
Entrée [129]: Categorie_nutrition[Categorie_nutrition.sodium_100g>2].C
Out[129]: <AxesSubplot:ylabel='Categorie'>
```



```
Entrée [142]: #On va effectuer le même travail avec la variable fat_100g

              Categorie_nutrition[Categorie_nutrition.fat_100g>80].Categorie.

Out[142]: <AxesSubplot:ylabel='Categorie'>
```



Find outliers (a product very different from others of the same group).

Pour ce qui est des outliers, c'est-à-dire les données qui contrastent grandement avec les valeurs « normalement » mesurées, on remarque que par exemple certains produits de ne contiennent pas de sucres alors qu'ils sont dans les catégories d'aliments sucrés. Cela est dû au fait que le sucre dans ces aliments est remplacé par de la stévia, plante naturelle qui donne ce goût sucré.

```
Entrée [32]: #On remarque que l'index 1113 ne contient pas de sucre alors qu'il est classé dans la catégorie des sucres
             #On va regarder dans le data frame de départ de quel produit il s'agit

             df.loc[1113].sugars_100g #Spray sweetener
             #Ce produit ne contient pas de sucre mais du stevia ce qui donne le coup sucré

Out[32]: 0.0
```

It exists products very similars in terms of nutrition facts but very different in terms of categories or ingredients ?

Oui, on remarque que certains produits présentent les mêmes capacités nutritionnelles alors qu'ils sont totalement différents

en termes de catégorie de produits. Pour cela, on créé un nouveau dataframe contenant d'une part, par exemple, la quantité de sodium pour 100g d'un produit, ainsi que la catégorie du produit en question. On remarque que par exemple, les 5 premiers produits sont riches en sodium alors qu'ils ne sont pas du tout dans la même catégorie.

```
Entrée [79]: sortedDf.head(5)
```

Out[79]:

	sodium_100g	main_category	pnns_groups_1	pnns_groups_2
53457	1300.000	en:hot-sauces	Fat and sauces	Dressings and sauces
39845	120.000	en:salad-dressings	Fat and sauces	Dressings and sauces
3027	80.000	en:whole-milk-yogurts	Milk and dairy products	Milk and yogurt
21071	52.832	en:biscuits	Sugary snacks	Biscuits and cakes
4411	41.700	en:bonbons	Sugary snacks	Sweets

L'hypothèse est plus ou moins vérifiée avec les produits riches en fer, et d'autres encore.

	iron_100g	main_category	pnns_groups_1	pnns_groups_2
34115	19.20000	en:breakfast-cereals	Cereals and potatoes	Breakfast cereals
55004	14.28570	en:dehydrated-beverages	Beverages	Artificially sweetened beverages
82962	13.39290	en:meats	Fish Meat Eggs	Meat
11561	13.27430	en:meats	Fish Meat Eggs	Meat
48376	3.57143	en:poultries	Fish Meat Eggs	Meat

3) Based on your expertize on this dataset, propose and describe a model (no code required) that would be interesting to enhance the OpenFoodFacts project.

Pour optimiser l'utilisation du projet OpenFoodFacts, on peut par exemple ajouter une clause qui demande que tous les champs, notamment les catégories (*main_category, pnns_groups_1 & 2*) soient remplies. De ce fait, il est plus facile par la suite de catégoriser les données.

Ensuite, pour ce qui est de l'approche Machine Learning, on peut par exemple établir un modèle de clustering comme KNN, afin de prédire dans quelle catégorie se trouve le produit en se basant sur ses informations nutritionnelles. En effet, on remarque que c'est souvent l'information qu'il manque le plus dans le jeu de données du projet OpenFoodFacts.

Enfin, il conviendrait d'imposer une langue par défaut lors de l'ajout de données, afin d'éviter aux data scientists d'avoir recours à des méthodes de traduction, qui peuvent parfois mal traduire certains termes, et donc fausser l'étude.