Rapport du projet 5 d’Openclassrooms :

Utilisez les données publiques de l’OpenFoodFacts

Elève : Baptiste Fina

Date : 15/01/2018

Lien du projet :

Introduction :

Le but de ce projet est de créer un programme qui permet à un utilisateur de choisir un aliment qu’il consomme habituellement et de lui proposer une alternative plus saine ainsi que ses caractéristiques et éventuellement un point de vente.

Structure du projet :

├── main.py

├── requirements.txt

├── mdp.txt

├── app

├── interactions.py

├── databasemanager.py

├── constantes.py

├── printing.py

└── substitutes.py

└── substitutes.json

1ere partie : description du fonctionnement du programme

Au lancement, le programme affiche un menu où l’utilisateur peut soit chercher à remplacer un aliment pour un autre plus sain, soit consulter les aliments déjà substitués.

Dans le cas où l’utilisateur cherche à remplacer un aliment, le programme commence par lui demander la catégorie dont fait partie cet aliment. Ce choix permet de comparer des aliments similaires entre eux et de simplifier le choix de l’aliment à remplacer. Une fois la catégorie choisie, l’utilisateur se voit proposé vingt aliments provenant de cette catégorie.

Le score nutritionnel de l’aliment qu’il souhaite remplacer est ensuite comparé à ceux appartenant à la même catégorie. Le programme affiche ensuite simplement le substitut avec le meilleur score nutritionnel. En plus du nom du substitut, le programme affiche son id, ses ingrédients, un lien vers sa page sur le site OpenFoodFacts.org, son score nutritionnel et enfin un éventuel lieu d’achat.

L’utilisateur a ensuite le choix d’enregistrer le substitut proposé ou non dans la base de données. S’il choisit de ne pas le sauvegarder, on revient simplement au menu de base.

Enfin, dans le cas où l’utilisateur choisit d’afficher les aliments qu’il a déjà substitués, le programme affiche les substituts. L’utilisateur a ensuite le choix de soit retourner au menu, soit afficher le détail d’un aliment en entrant son numéro.

2eme partie : explication des choix et de l’avancement

Durant ce projet, j’ai dû prendre plusieurs décisions qui ont impacté son fonctionnement actuel. La première concerne la gestion de la taille de la base de données de l’OpenFoodFacts. De par sa taille, même en restreignant aux aliments français, il est illusoire de demander directement à l’utilisateur de choisir directement de remplacer un aliment. D’une part parce qu’il y a trop d’aliments, d’autre part parce que la comparaison des aliments entre eux n’aurait pas grand sens. J’ai donc commencé par vouloir demander à l’utilisateur plusieurs choix de catégories, de plus en plus fins, afin de pouvoir présenter à l’utilisateur tous les produits d’une même catégorie. Cependant, cette méthode, en plus d’être assez longue à l’utilisation présente deux inconvénients majeurs.

Le premier, c’est que l’API OpenFoodFacts ne propose pas d’arborescence des catégories très simple d’utilisation. Cela me contraindrait à devoir siphonner le contenu de la page d’une catégorie afin d’en extraire les sous-catégories. Cette méthode, en plus de ne pas respecter la consigne d’interagir avec l’API est aussi très lente puisqu’à chaque fois que l’utilisateur voudrait remplacer un aliment, il devrait à nouveau subir ces choix multiples.

En plus de ces deux raisons, si je voulais au départ effectuer ces choix multiples c’était dans le but de limiter autant que possible le nombre d’aliments proposés à l’utilisateur. Les premières catégories possédant plusieurs dizaines de milliers de produits, il était illusoire de demander à l’utilisateur de choisir de cette manière. Cependant, cette méthode restreint trop la comparaison des aliments entre eux et on ne trouve souvent pas de substitut, ou aux qualités nutritionnelles bien trop similaire.

Voilà pourquoi j’ai décidé de me restreindre aux vingt premières catégories (amendées des deux premières qui sont bien trop vagues dans leur description) et de n’afficher que la première page des aliments. Cela permet à l’utilisateur d’avoir un panel de vingt aliments très variés à la fois en genre et en qualité nutritionnelle.

En conclusion, ce projet m’a donné beaucoup plus de mal que les précédents. Tout d’abord car c’était la première fois que je devais interagir avec une API. J’ai eu au début du mal à trouver la documentation nécessaire afin de réaliser ce que je souhaitais, d’autant que j’ai assez régulièrement mal interprété ce qui était demandé. J’ai cependant beaucoup apprécié la philosophie du Doc Driven Development qui permet, quand on en prend le temps, de pouvoir développer le projet dans un cadre précis.

Enfin, j’ai pu aussi beaucoup m’améliorer en développement orienté objet sur un projet qui, à mes yeux, ne s’y présentait pas autant que le projet « Aidez MacGyver à s’échapper ».