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

├── app

├── interactions.py

├── databasemanager.py

├── constantes.py

├── printing.py

└── substitutes.py

└── substitutes.json

Première partie : choix de la catégorie

En allant sur le site d’OpenFoodFacts, on constate qu’il y a plus de 10000 catégories existantes. Il est donc irréaliste de demander à un utilisateur de faire son choix de cette manière. L’une des solutions aurait pu être de télécharger toute la base de données de l’OpenFoodFacts pour la France. Cependant, cette solution me paraît peu élégante et ne règle pas le problème du choix de la catégorie, en plus de prendre beaucoup de place. Je décide plutôt d’obtenir toutes les catégories et d’en proposer les 20 premières par ordre de taille à l’utilisateur. Ces catégories sont enregistrées dans le fichier constantes.py.

On demande tout d’abord à l’utilisateur de choisir une catégorie parmi les 20, puis, grâce à la librairie « request », on récupère le contenu de la page de cette catégorie. On obtient ainsi toutes les sous-catégories. On répète l’opération choix-demande 3 fois afin d’affiner suffisamment le résultat. Au-delà, soit on obtient un pool d’aliments trop restreint, et donc la recherche d’un meilleur substitut n’aboutit pas. En-deçà on obtient au contraire trop d’aliments et le résultat des comparaisons peut sembler absurde.

Une fois que l’utilisateur a choisi la dernière catégorie d’aliments on peut lui demander de choisir l’aliment qu’il souhaite remplacer.

Deuxième partie : choix de l’aliment à remplacer

Une fois que l’utilisateur a choisi la dernière catégorie, on télécharge tous les aliments provenant de cette catégorie dans un fichier JSON. On extrait ensuite de ce fichier les informations qui nous intéressent pour les placer dans une table MySQL. Cette étape est gérée par la classe DatabaseManager.

Lorsque l’utilisateur a choisi l’aliment qu’il souhaite remplacer, on compare son score nutritionnel (et non sa note) à ceux de sa catégorie et on affiche les aliments plus sains. Il reste ensuite à l’utilisateur à choisir le substitut puis on affiche ses caractéristiques (ingrédients, score nutritionnel, lieu d’achat éventuel) et on l’enregistre dans le fichier substitute.json.

Conclusion :

Ce projet m’a donné beaucoup plus de fil à retordre que les précédents. La première difficulté a été pour moi de trouver un moyen pour l’utilisateur de choisir l’aliment à remplacer de façon claire. La méthode que j’ai choisie et que j’ai détaillée dans la première partie. J’ai choisi de ne pas sauvegarder la base de données au fur et à mesure que l’utilisateur fait des choix à cause d’une limitation de pymysql qui affiche un warning au lancement du programme. L’utilisateur a donc nécessairement besoin d’une connexion internet pour faire fonctionner le programme.

La seconde difficulté que j’ai rencontrée concerne l’interaction avec la base de données MySQL. J’avais commencé par écrire simplement une fonction par commande MySQL requise, ce qui ne correspond à la philosophie de développement de l’orienté objet. J’ai donc réécrit cette partie pour obtenir la classe DatabaseManager qui possède une fonction qui prend en paramètre une commande MySQL, l’exécute et renvoie le résultat.

J’ai aussi, suivant les conseils du mentor qui m’a fait passer la soutenance, écrit une classe printing.py qui gère l’affichage des éléments complexes tels que les listes de catégories et d’aliments ou les détails d’un produit.

Grâce à ce projet j’ai pu me familiariser avec le « doc driven development » que je trouve très utile pour se fixer des guidelines mais parfois difficile à respecter quand on avance dans le développement lorsqu’on se rend compte de certaines difficultés. J’ai aussi eu l’occasion de découvrir MySQL