Fouille de données

Rapport Préliminaire sur le projet Eat A Week

De Zhao, Vincent Segonne

Introduction

Dans ce document nous présentons l'avancée du projet nommé "EatAWeek" sur lequel nous travaillons dans le cadre du cours de fouille de données. Nous commençons par rappeler l'idée générale du projet, ses objectifs ainsi que l'intérêt qu'à celui-ci dans le cadre du cours. Ensuite, nous détaillerons l'avancement de nos travaux, nous expliquerons ce que nous avons fait et implémenté ainsi que les difficultés que nous avons rencontrées jusque là. Enfin nous terminerons par donner les prochaines étapes du projet.

I - Rappel du projet

Le projet EatAWeek est un projet qui vise à automatiser la tâche pénible de faire les commissions. Le but final étant de ne plus à se préoccuper de ce que nous allons manger dans la semaine au moment de faire les courses. Pour cela, l'idée principale du projet est de pouvoir générer automatiquement un certain nombre de repas pour la semaine avec la liste des ingrédients à acheter. Nous précisons que ce type de système existe déjà sur le web. L'intérêt qu'à ce projet dans le cadre de ce cours est qu'en plus de générer automatiquement des recettes, le système doit pouvoir apprendre à connaître les goûts de l'utilisateur afin de pouvoir au mieux choisir des plats qui lui conviennent. Le machine learning est donc au coeur de ce projet. Il s'agit en fait simplement de monter un moteur de recommandation de recettes. Pour cela nous nous sommes largement inspiré de ce que nous avons vu en cours et avons implémenté les méthodes utiles pour les moteurs de recommandations (représentation de document via TFIDF, clustering etc..)

II - Jusqu'à Aujourd'hui

La première étape du projet a été de récupérer les données. En effet pour pouvoir générer des recettes et proposer un contenu à l'utilisateur, il nous a fallu constituer une base de données de recettes. Pour faire cela, nous avons utiliser un module de webcrawling en php pour parcourir le site Marmiton.org afin de récupérer le plus de recettes possibles. C'est une tâche qui n'a pas été très facile et qui nous pose encore quelques difficultés pour plusieurs raisons. La première étant que nous n'avions jamais fait de webcrawling et que le module que nous avons trouvé est en php, langage que nous ne connaissons pas.

Ensuite il y a le problème du repérage des robots par le site qui nous bloque au bout d'un moment. Enfin comme la majeur partie des données extraites du web, celles-ci ne sont pas très propres et il est très fréquent d'avoir des bouts de recettes manquant ou bien des fautes d'orthographes qui font planter le système.

Nous avons donc tant bien que mal récupérer des recettes, nous souhaiterions en récupérer au moins 1000 pour les tests. Nous stockons ces données dans une base de données que nous gérons via le module sqlite3 en python et qui nous sert de banque de recettes pour les recommandations.

La deuxième étape a consisté à représenter les recettes de manière numérique afin de pouvoir faire des calculs dessus. Pour cela, nous avons commencé par une représentation assez basique mais quand même efficace, la représentation de document par les termes, c'est à dire le TFIDF. Nous avons donc transformé les recettes issues de la base de données en vecteur obtenu par TFIDF que nous avons implémenté via Sklearn.

Les recettes sont donc pour l'instant représentée par les termes trouvés dans le titre, les ingrédients ainsi que les instructions. Encore une fois il faut rappeler que ceci peut être biaisé par des fautes d'orthographes ou par une erreur de tokenisation par exemple. C'était particulièrement le cas lorsque nous n'avions pas à notre disposition beaucoup de données.

Ensuite, une fois que nous avons pu vectoriser les données, nous nous sommes attaqués à la modélisation du profil utilisateur. Nous avons choisi d'implémenter un modèle de recommandation par contenu car nous n'avons pas accès à un nombre de profils d'utilisateurs. En raison du problème que l'on appelle fréquemment le problème de "cold start", c'est à dire que le système ne dispose pas à l'initialisation de données sur l'utilisateur et dans le but de modéliser son profil il nous faut demander à l'utilisateur de "liker" un certain nombre de recettes. C'est ce que nous faisons en premier lorsque le système détecte un nouvel utilisateur, il lui fait défiler des recettes choisies aléatoirement et lui demande de liker celles qui lui plaisent. Une fois que cela est fait, nous pouvons passer à la phase de modélisation avec les données obtenues grâce aux likes de l'utilisateur. Faire le vecteur moyen de tout ce qu'il a aimé n'aurait pas beaucoup de sens car les recettes peuvent être très différentes les unes des autres, nous avons plutôt opté pour la deuxième méthode qui consiste à faire un clustering des recettes qui se ressemblent.

Ainsi, nous pouvons désormais commencer la recommandations de recettes en comparant les autres recettes aux différents clusters obtenus grâce à la similarité cosinus.

On garde pour chaque recette l'argmax (ici, le centroide) obtenu via la similarité cosinus avec tous les clusters et puis nous trions les recettes dans l'ordre croissant.

Nous obtenons alors les recettes supposément les plus proches des goûts de l'utilisateur. Nous lui proposons ces recettes et le soumettons à une nouvelle validation de ces recettes, si celles-ci sont likées alors elles sont ajoutées à son profil. Ceci va enrichir le moteur de recommandation et devrait lui permettre d'être encore meilleur par la suite.

Un petit test a été implémenté dans le main.py
=> python3 main.py
=> Suivre les instructions.

III - Ce qui nous attend

Maintenant que nous avons implémenté une version alpha du programme, nous avons deux principaux objectifs à atteindre d'ici le rendu de projet.

Tout d'abord nous allons travailler sur l'optimisation du modèle de recommandations. Nous avons plusieurs pistes, par exemple nous allons implémenter un grid search pour pouvoir trouver les meilleurs paramètres pour le clustering. Nous pouvons également penser à

d’autres modèles que le K-means. Cela rejoint également une phase importante que nous devons traiter, l’évaluation du système.

Nous allons également essayer de modifier la représentation des données, on pense notamment à rajouter des features au TFIDF existant, peut être même pondérer certains traits comme le temps de préparation ou bien le prix du repas par exemple. De plus, il sera possible de “disliker” des recettes ce qui sera ajouté au profil et ce qui permettra d’écarter les recettes qui n’ont pas plu à l’utilisateur.

Enfin il sera temps de passer à la version graphique du programme pour que ce soit plus facilement utilisable, nous y implémenterons peut être également des paramètres additionnels comme des filtres (repas végétariens, etc..).