Fouille de données : Rapport de projet

Binôme

Dakhlia Cyrille et Jaouani Mohamed-Amine

1. But du Projet

Le but de notre projet de fouille de données est d'implémenter un système de recommandation basé sur le sport, proposant aux utilisateurs inscrits des sports qu'ils seraient susceptibles d'aimer, mais également des personnes avec qui ils pourraient de devenir ami sportif, voire plus !

2. Environnement du programme

Le programme se sert des informations sur les différents sports qui sont référencés dans la base de données, ainsi que des informations des utilisateurs, fournies par leurs soins lors de leur inscription (sur l'application / le site en devenir) :

- Les informations d'un sport sont définies selon les critères suivants :
  - **Pack :** *bien-être, combat, danse, raquette, sports collectifs, …*
  - **Nombre de joueurs** : *individuel, deux, équipe*
  - **Equipement** : *projectiles (balle, ballon, boule (sans rebond), …), moyen de projection (raquette, ~~marteau,~~ …), moyen de déplacement (jambes, véhicule sans moteur~~,~~ cheval, perche, …), support de déplacement (sol, tapis, barre parallèles, tremplin, …)*
  - **Lieu, environnement :** *intérieur, extérieur ; terre, air, eau, neige ; solide, liquide, élastique*
  - **Partie du corps entraînée**
  - **Capacités requises** : *équilibre, mouvement, coordination avec l'équipe, vitesse, endurance, anticipation, trajectoire, positionnement*
- Les informations d'un utilisateur sont définies selon les critères suivants :
  - **Âge**
  - **Localisation** : *ville, (pays)*
  - **Niveau sportif :** *pas d'activité, peu d'activité, faible activité (1 à 3 jours par semaine), activité modérée (3 à 5 jours par semaine), activité intense (6 à 7 jours par semaine), activité très intense (2x par jours, activités intenses en plus)*
  - **Genre :** *homme, femme, ~~etc.~~*
  - **Domaine d'étude :** *informatique, langues, lettres, cinéma, mathématiques, physique, chimie, etc.*

- **Niveau d'étude :** *Licence 1, Licence 2, Licence 3, Master 1, Master 2*
- **Sports aimés**
- **Packs aimés**

A partir de ces données, le programme se base sur les principes « Item-based » et « User-based » pour effectuer ses recommandations de sports et ses recommandations d'amis.
Considérons dans la suite trois utilisateurs A, B et C

Recommandation de sport selon les deux principes mis en œuvre :

1. ***Item-based*** :
   a. Si on aime le Taekwondo…
   b. … et que le Taekwondo ressemble au Krav Maga…
   c. … alors on est susceptible d'aimer le Kra Maga
2. ***User-based*** :
   a. Si A aime le Taekwondo, le Basket et l'Aïkido…
   b. … et qu'on aime aussi le Taekwondo et le Basket…
   c. … alors on aimera sûrement l'Aïkido

Recommandation d'utilisateur selon les deux principes mis en œuvre :

1. ***Item-based*** :
   a. Si on est ami avec A…
   b. … et que C ressemble à A (ils aiment les mêmes sports, habitent dans la même ville, ont le même âge par exemple) …
   c. … alors on est susceptible de devenir ami avec C
2. ***User-based*** :
   a. Si A (et C) est (sont) ami(s) avec B…
   b. … et qu'on est ami avec A (et C) …
   c. … alors on s'entendra sûrement bien avec B

3. Problèmes à éviter et solutions

A. Problème du cold-start

Le problème du cold-start est courant dans les systèmes de recommandation, comme par exemple Netflix, où on demande directement à l'utilisateur de sélectionner des films qu'il aime parmi une liste afin de prévoir ses futurs 'like'.
Nous utilisons nous aussi un système similaire demandant à l'utilisateur de choisir des sports qu'il aime parmi une liste (partielle/complète) générée (aléatoirement ou non), nous permettant de jauger les goûts de cet utilisateur et de pouvoir lui proposer des sports adéquats dans un premier temps avant d'avoir plus d'informations sur ses goûts et sur ses activités vis-à-vis du programme. Il reste toutefois possible de mettre à jour les préférences d'un utilisateur.
Nous avons envisager d'implémenter ce mécanisme de choix lorsque nous développerons ce système de recommandation au sein d'une application mobile Android/iOS.

B. Problème des goûts opposés

Si un utilisateur indique qu'il aime plusieurs sports très différents, par exemple Taekwondo et danse classique, on ne dispose pas de plus d'informations concernant ses préférences car cela revient au problème du cold-start. Nous pouvons alors lui proposer des sports aléatoirement dans les packs qu'il aime. Par exemple, si l'utilisateur aime ces deux sports, nous pouvons lui proposer de faire du Krav Maga, du Hip-hop, ou encore mieux : de la Capoeira qui est un mix entre danse et sport de combat.
Nous avons donc pensé à implémenter un système de tag qui attribue à chaque sport un ou plusieurs tags correspondant à des mots-clefs décrivant le sport en question, permettant ainsi de calculer les TF-IDF très facilement et ainsi calculer les distances entre deux sports afin de les recommander.

Par exemple :

Football : #balle #collectif #terrain
Basket-ball : #balle #collectif #terrain
Tennis : #balle #raquette #individuel #terrain
Badminton : #raquette #individuel #terrain
Taekwondo : #combat #tatami #kimono
Hip-Hop : #danse #salle #street
Kapoera : #combat #danse

On peut voir ici que le Football et le Basket-ball contiennent les mêmes mots-clefs, on peut ainsi supposer qu'un utilisateur qui aime le Football va aussi aimer le Basket-ball donc il semble très intéressant de proposer ce sport.
Cela servira aussi lors d'un ajout d'un nouveau sport dans la base de donnée. En effet, en ajoutant un sport on ajoute aussi ses 'tag' (mots-clefs), et on peut calculer le plus proche voisin et ainsi déterminer dans quelle classe (cluster) ce nouveau sport se situe pour pouvoir le proposer à un utilisateur qui aime des sports dans ce même cluster.

4. Implémentation

Nous utilisons le langage Python pour développer ce projet. Nous utilisons aussi une API Java pour parser un fichier Excel. Ce fichier Excel nous a été délivré par l'Université Paris 7 Diderot et contient la liste des utilisateurs anonymisés et les sports qu'ils ont choisis ainsi que les horaires des créneaux de ces choix. Cette API se nomme jexcelapi et se trouve dans notre GIT à l'arborescence suivante : spotmate-finder/src/ext_data/jexcelapi/

Une fois le fichier parsé, nous stockons les données dans une base de donnée SQLite en utilisant Python.
L'avantage d'utiliser SQLite et que lorsque nous exporterons le projet sur une application mobile Android nous pourrons simplement réutiliser le même code de création de la base car les bases de données sont implémentées en SQLite sur Android.

Nous disposons de plusieurs tables contenant toutes les données pertinentes des utilisateurs. Ces données sont séparées en plusieurs tables pour éviter à recréer complètement la base si on ajoute de nouvelles colonnes (champs).

5. Algorithme

Nous utilisons la similarité de Dice-Sorensen pour calculer le nombre de tag en comment entre deux utilisateur. Plus ils ont de tag en commun et plus la similarité sera importante. La similarité de Dice-Sorensen se calcul comme suit :

S(A,B) = 2* (|A inter B| / (|A| + |B|))

On peut par la suite faire la moyenne des TF-IDF associés à ces tags mais si l'utilisateur à choisi des sports très différents la moyenne n'aura aucun sens.

Nous utilisons un algorithme de recommandation basé sur le TF-IDF des tags attribués aux sports.
Nous mesurons la similarité entre le TF-IDF d'un sport avec celui d'un autre sport, si la valeur de leur TF-IDF se ressemble fortement, alors nous en déduisons que ces deux sports ont un profil similaire et nous pourrons proposer ce nouveau sport à l'utilisateur.

Source :

- http://projetconnaissance.free.fr/classement/classements/classement_sports.html