Boosting de profil

# Le principe

Le boosting consiste à effectuer des jointures plusieurs tables représentatives d’un aspect du profil d’un utilisateur afin de déterminer un boosting global pour celui-ci.

Ce n’est pas la méthode la plus efficace mais elle peut s’effectuer quasi exclusivement au sein de la base de données contrairement aux clustering en PHP qui prend trop de temps.

# La base de donnée

## Les tables

### Intérêt

Répertorie les différents centres d’intérêts possibles

### IP

L’IP en tant que telle n’est pas enregistrée mais à la place on trouve les colonnes

* Id
* Pays
* Département (Région serait plus juste)
* Ville

Chaque utilisateur dispose d’une IP, on peut ensuite calculer un « score\_utilisateurs\_ip » entre deux utilisateurs : plus il est élevé et plus les utilisateurs sont géographiquement proches.

### Mot

La liste des mots apparaissant dans les produits une fois qu’ils ont été nettoyés : suppression des mots vides et de caractères spéciaux

### Produit

La liste des produits référencés avec un titre, un auteur et une description

### Produit\_mot\_poids

Plus un mot a de l’importance dans un produit et plus son poids est élevé

### Requête

Pour chaque requête effectuée (dans une barre de recherche par exemple), on a l’id de l’utilisateur effectuant la recherche ainsi que celle du mot recherché.

### Utilisateur

La table de l’utilisateur avec un âge, une ville et une référence à l’IP lui correspondant

### Utilisateur intérêt

Référence pour chaque utilisateur de ses centres d’intérêt avec une valeur comprise entre 0 et 10 : plus cette valeur est élevée et plus l’utilisateur aime un centre d’intérêt.

### Utilisateur produit

Recense les données fournies par l’utilisateur sur un produit : on a le nombre de visite, s’il a acheté le produit et la note qu’il lui a attribuée

## Les fonctions

Elles servent à calculer certains éléments pour les vues.

## Les procédures

### Recommander produits

Prend trois paramètres : « id », « nombre » et « début »

Retourne les « nombre » produits à partir de la position « début » correspondant le plus à l’utilisateur représenté par « id ».

Pour ce faire la procédure effectue une jointure entre les vues « profil\_score\_utilisateur » et « profil\_score\_utilisateur\_produit » et multiplie les scores en étant issus.

Plus un score global est élevé et plus les utilisateurs ayant des profils similaires l’ont apprécié

### Score utilisateurs requête produit

Remplacé par des vues.

## Les vues

Elles permettent essentiellement de calculer pour chaque utilisateur deux profils :

* Le profil « score utilisateur » qui calcule un score entre deux utilisateurs : plus le score est élevé et plus les utilisateurs sont proches au niveau de leurs profils
* Le profil « score utilisateur produit » qui calcule un score entre un utilisateur et un produit : plus le score est élevé et plus l’utilisateur apprécie le produit

### Profil score utilisateur

Pour chaque couple d’utilisateurs le score est obtenu en additionnant les scores intermédiaires issus des vues suivantes :

* Score utilisateurs âge : le score est compris entre 0 et 4
  + Deux utilisateurs ayant un âge similaire ont un score élevé
* Score utilisateurs intérêt : le score est compris entre 0 et n
  + Deux utilisateurs ayant des intérêts similaires ont un score élevé
* Score utilisateurs IP : le score est compris entre 0 et 3
  + Deux utilisateurs venant d’un lieu similaire ont un score élevé

On peut calculer d’autres scores intermédiaires selon les données qu’on arrive à récupérer de l’utilisateur.

### Profil score utilisateur produit

Pour chaque utilisateur on calcule un score symbolisant son « appréciation d’un produit » en additionnant les scores intermédiaires suivant :

* Score basé sur les produits visités et effectué à partir de la table utilisateur produit :
  + Score = note, si la note est renseigné dans la table utilisateur produit
  + Sinon score = 6, si l’utilisateur a acheté le produit
  + Sinon le score correspond à la valeur minimale entre le nombre de visites du produit par l’utilisateur et le chiffre 4
* Score basé sur les recherches de produits effectués à partir de la table requête et des vues « score utilisateur requête… » :
  + Pour chaque recherche on réalise un « scoring », selon la méthode OkapiBM25, avec chaque document au sein duquel au moins un mot de la recherche apparait. Ce score intermédiaire est représenté par la vue « score utilisateur requête produit »
  + Puis on calcule le score entre un utilisateur U et un produit P en :
    - Addition des scores intermédiaires des requêtes entre un utilisateur U et un produit P
    - Division par le nombre de requêtes totales effectuées par l’utilisateur U
  + De ce fait, plus un utilisateur fait apparaître certains termes dans ses recherches et plus les documents liés à ces recherches auront de l’importance dans le résultat