Manger Pas Cher

Bastien Baude
Etienne Taillefer de Laportalière
Alexis Olivari
Loïc Themyr

2018

Table des matières

1 Introduction

Notre site web est une plate-forme qui permet de relier les consommateurs et les commerçants en donnant la possibilité aux consommateurs d'acheter les produits presque périmés à prix réduit.

Cela permet aux consommateurs d'acheter de la nourriture à un prix défiant toute concurrence, et aux commerçant d'éviter le gaspillage et de réduire les pertes associées.

2 Organisation du goupe

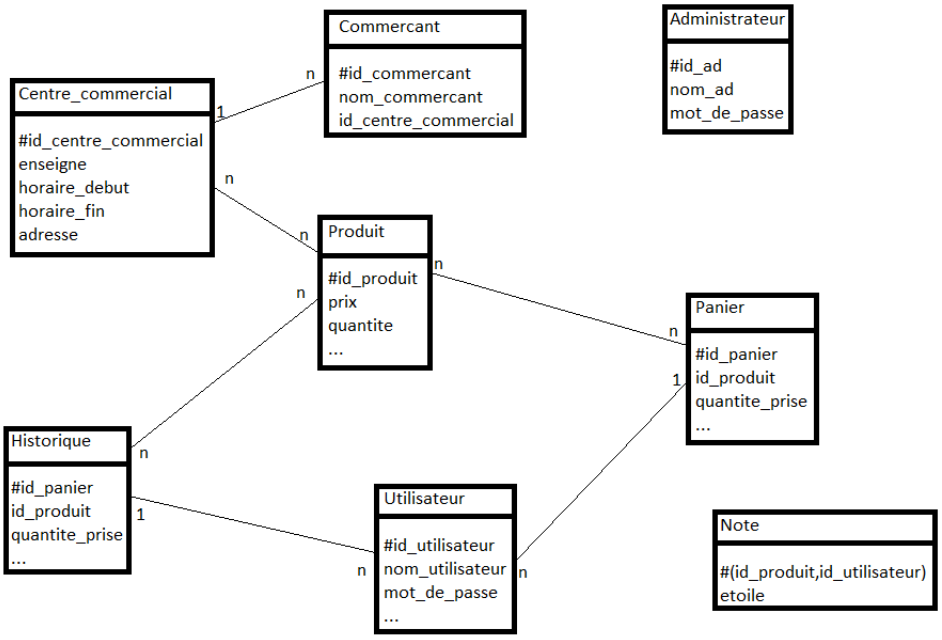
Dès le début du projet nous nous sommes répartis les rôles pour que chacun puisse travailler de son côté. Etienne Taillefer de Laportalière s'est occupé de toute la partie sql (bases et trigger) ainsi que les différents droits pour le site. Bastien Baude a fait les parties PHP et JavaScript, Alexis Olivari s'est occupé du CSS et Loïc Themyr s'est consacré à la partie Machine learning pour implémenter le système de recommandation.

3 Les bases de données

3.1 Les différentes tables

Pour notre projet web nous avons dû créer de nombreuses tables dans la base de données. En effet, dans notre site il existe trois types de personnes : l'utilisateur, les commerçants et les administrateurs avec leur table respective. En plus de ces tables, il y a une table pour les produits, une pour le panier (détaillée plus tard), une pour les historique, et enfin une pour les centres commerciaux.

De plus, il s'agit d'un site de vente en ligne donc il est nécessaire d'ajouter une possibilité de recommander des produits aux utilisateurs, pour cela, nous avons créé un système de machine learning (recommender system). Ce système nécessite une table supplémentaire qui sauvegarde les notes des produits, c'est la table Note.



3.2 Les trois rôles

Comme dit précédemment il y a trois trois types de personnes, cela implique que des droits différents pour chacun d'entre eux. Ainsi seuls les administrateurs peuvent créer un centre commercial, le principe est le suivant, une fois un accord est signé entre notre site et un centre commerciale l'administrateur crée le centre commerciale dans la base puis envoie les différents identifiants aux commerçants du centre commerciale. De plus seul un administrateur peut remplir les bases de données à l'aide d'un tableau csv.

Grâce à cela les commerçants peuvent alors se créer un compte lié à leur centre commercial et ainsi ils sont alors les seuls à pouvoir ajouter des produits dans la base de données. Les utilisateurs ne peuvent qu'afficher les produits et en acheter.

3.3 Les mises à jour des bases de données

L'enjeu des bases de données pour notre site est l'automatisation de la mise à jour. Ainsi si un utilisateur achète un produit il faut l'enlever, si un produit est périmé il faut alors l'enlever, si toutes les quantités d'un produit ont été achetées il faut alors supprimer le produit. Pour régler ces problèmes j'ai utilisé des triggers.

Mon premier trigger enlève automatiquement de la table produit, toutes les quantités (d'un produit) prises par un utilisateur qui sont mis dans son panier. Ainsi ce trigger permet de maintenir la table produit à jour des différentes opérations. De plus, un autre trigger a été crée pour permettre de supprimer tous les produits périmés après l'ajout d'un produit par un commercant dans la table. A chaque mise à jour de la table produit (par exemple dû au premier trigger), il y a un trigger qui supprime tous les produits dont la quantité est égale à 0. Enfin, il y a un autre trigger, qui lorsque que l'utilisateur valide son panier, supprime et transfère vers la table historique tout le contenu du panier . Cela permet de garder une trace des différents achats de l'utilisateur et de mettre à jour la table à chaque fois. Au contraire un des triggers remet dans la table produit tous les éléments du panier dans le cas où le client décide d'annuler son panier. Une des choses que je n'ai pas réussi à faire est de programmer une requête quotidienne pour vérifier les dates de péremption. Ainsi là je supprime les produits périmés qu'après un insert dans la table produit ce qui n'est pas optimal.

4 La partie PHP/JavaScript

4.1 CRUD

Un utilisateur doit avoir la possibilité de créer son compte, mais aussi de se connecter, se déconnecter, d'afficher son compte et modifier son mot de passe. Dans un premier temps, un formulaire de création de compte a été créé pour répondre à cette demande, cela permet d'ajouter le nouvel utilisateur à notre Table avec les variables superglobales POST. Ensuite, pour la connexion et la déconnection, on utilise les variables superglobales de SESSION. Cela permet à l'Utilisateur de s'authentifier et de notre côté, on conserve sa clé qui sera utilisée pour ajouter un Produit au Panier par exemple.

De même pour les Commerçants qui ont la possibilitée de créer leur compte et de le modifier. Néanmoins, ils peuvent aussi ajouter des Produits à la table dédiée grâce à un formulaire.

4.2 Affichage de l'ensemble des Produits

Il s'agit d'un site de vente en ligne donc l'utilisateur doit avoir accès à l'ensemble des produits disponibles sur le site. On affiche donc cet ensemble à l'aide d'une requête SQL SELECT * FROM Panier. Ensuite, on le stocke dans un tableau avant de l'afficher. Cependant, on ne peut pas se limiter à un simple affichage, il faut ajouter des possibilités telles que noter le produit ou l'ajouter au Panier avec une certaine quantité prédéfinie.

Pour permettre aux Utilisateurs de noter un produit, nous avions pensé à utiliser un formulaire simple avec des boutons radio. Cependant, avec ce type de formulaire on ne pouvait pas récupérer la clé du Produit associé et l'Utilisateur devait noter le Produit puis valider sa note. Dans un second temps, on a ajouté dans le tableau d'affichage 5 balises ancres alignées en forme d'étoiles par produits. Si l'Utilisateur appuie sur l'une d'entre elles, alors on vérifie que l'Utilisateur est bien connecté et ensuite grâce à un formulaire caché on ajoute la note à la table par le biais de JavaScript. De cette manière, l'Utilisateur n'a pas besoin de valider sa note.

De même, pour permettre aux Utilisateurs de choisir une quantité d'un Produit avant de l'ajouter au Panier, on a utilisé 6 balises ancres alignées à l'image de celles des Notes et la variable Quantité est ensuite stockée dans une variable de SESSION. Enfin, après avoir sélectionné une quantité, l'Utilisateur est en droit de l'ajouter dans son Panier. Cependant, on ne peut pas utiliser un formulaire classique car on a déjà les données mais il faut pouvoir y accéder, elles sont dans un tableau. Pour cela, je récupère les caractéristiques du Produit associés à la ligne du bouton Ajouter au Panier en question, en JavaScript et ensuite on utilise un formulaire caché pour l'ajouter à la table Panier avec les caractéristiques du Produit et les variables de SESSION associée.

4.3 Affichage des Produits sous condition

Sur la page d'accueil, l'Utilisateur peut afficher des Produits en choisissant la catégorie ou la marque. Pour répondre à cette attente, j'utilise une requête SQL avec des conditions LIKE sur la catégorie et la marque des Produits.

4.4 Affichage du panier

Lorsque l'Utilisateur a finalisé son Panier, il peut l'afficher et supprimer des éléments de son Panier. Enfin, il peut cliquer sur Payer et l'ensemble de son Panier est ajouté à la Table Historique.

5 Machine Learning

On a voulu intégrer à notre site web un système de recommandation de produits. C'est-à-dire, une fonctionnalité qui propose à chaque client des produits qui correspondent à leur goût. Pour créer ce système de recommandation, je me suis tourné vers un algorithme de machine learning qui s'appuie sur les notes des utilisateurs.

La mise en place du programme commence donc avec l'implémentation d'un système de notation. On souhaite que chaque client note certains produits et ces notes seront stockés dans une table sql. Quelque chose qui marche bien en général sur internet est le système de notation en étoile. Grâce à cela, on récupère une note allant de 1 à 5 qui indique l'affiliation d'un client avec un produit.

Ensuite, il faut récupérer la table de notation afin de l'utiliser dans notre algorithme de machine learning. Comme cet algorithme est implémenté en octave, il est plus facile de créer un convertisseur pour notre table en fichier csv. Le fichier ml data.php permet donc de télécharger le csv correspondant à la table où l'on a l'identifiant du client, l'identifiant du produit qui sont notés par le client et la note correspondante. Il y a notamment dans ce fichier une jointure entre la table note et la table utilisateur. Une fois ce fichier téléchargé, on peut le charger avec notre programme en octave.

Produits	Jean	Marie	Pierre	Célestin
Poulet	2	?	?	5
Boeuf	0	1	4	5
Nutella	3	1	5	?
Lait	?	5	0	0
Fromage	4	5	1	?

Exemple table de notation

Dans le fichier recomand.csv il y a l'implémentation de notre algorithme. Tout d'abord, on crée une matrice qui indique si oui ou non un utilisateur a noté un produit. Notre matrice peut se visualiser de la façon suivante :

Produits	Jean	Marie	Pierre	Céléstin
Poulet	1	0	0	1
Boeuf	1	1	1	1
Nutella	1	1	1	0
Lait	0	1	1	1
Fromage	1	1	1	0

Puis, on crée une nouvelle matrice qui représentera les caractéristiques des produits. Chaque produit ayant un nombre de paramètres arbitraires et ces paramètres sont initialisés aléatoirement.

Voici un exemple de cette matrice :

Produits	x1	x2	x3
Poulet	?	?	?
Boeuf	?	?	?
Nutella	?	?	?
Lait	?	?	?
Fromage	?	?	?

Ensuite, on crée une matrice similaire à la précédente mais qui représente l'affiliation d'un client à certaine caractéristique de produit. En voici un exemple :

Produits	Jean	Marie	Pierre	Céléstin
x1	?	?	?	?
x2	?	?	?	?
x3	?	?	?	?

L'objectif de l'algorithme et de prédire les notes manquantes afin de savoir quels produits plairont à quels clients. Ainsi, on note Θ^i le vecteur associé aux caractéristiques par le client i et X^j le vecteur associé aux paramètres du produit j. On aura alors la valeur $(\Theta^i)^T X^j$ qui correspondra à la note estimée du produit j par le client i.

Le but maintenant est donc d'optimiser les paramètres des X^j et Θ^i (initialisés aléatoirement) afin d'obtenir un modèle qui donne les meilleurs estimations. Or, pour ce faire, on utilise un algorithme de descente de gradient qui va optimiser simultanément les paramètres de ces deux vecteurs en minimisant l'erreur quadratique $J(\Theta, X)$ entre $(\Theta^i)^T X^j$ et la note réelle du client $Y^{(i,j)}$.

$$J(\Theta, X) = \frac{1}{2} \sum_{(i,j):r(i,j)=1} ((\Theta^i)^T X^j - Y^{(i,j)})^2$$

avec $r(i,j) = 1$ si le client i a noté le produit j.

Après cette étape d'optimisation, on effectue une prédiction pour tous les produits non notés et on sauvegarde le résultat dans un fichier csv. Enfin, le fichier recommandation.php, qui est appelé à chaque connexion, permet d'initialiser les 5 variables de sessions correspondants aux 5 produits recommandés (les mieux notés par l'algorithme) pour le client connecté.

Pour finir, j'ai créé, grâce à javascript, un carrousel qui permet de faire défiler dans le temps la photo ainsi que la réduction de ces produits recommandés sur la page d'accueil.

Enfin, il existe plusieurs façon pour améliorer notre système de recommandation. Il faudrait ajouter des features à nos paramètres soit en optimisant le nombre de vecteurs initialisés aléatoirement dans nos matrices Θ et X, soit en y ajoutant des vecteurs à valeurs fixées correspondants aux informations que l'on possède déjà sur les produits (marque, type, prix, commerçant associé), ou même effectuer ces deux opérations.
Il faudrait également adapter notre programme à du online learning en utilisant dans un premier temps un algorithme de descente de gradient stochastique afin

de pouvoir optimiser nos paramètres à chaque nouvelle notation de la part d'un client. Et ensuite relier notre programme dynamiquement à notre site web et notre base de donnée afin de se passer de l'intervention d'un administrateur et d'obtenir des recommandations qui s'actualisent à chaque notation.

NB : pour que le système de machine learning soit opérationnel après un make install, il faut suivre ces quelques étapes :

- Se connecter en tant qu'administrateur : id : taillefer ; mdp : titi.

- Cliquer sur le lien "create sql from csv". (ajout de produits à la table)

- Ajouter quelques clients à la base en faisant des inscriptions.

- Effectuer des notations avec ces clients.
(il n'est pas nécessaire de noter tous les produits. Mais il est interressant de faire une notation respectant un schema, par exemple créer des clients qui aiment la viande et rien d'autre, des clients qui aiment les fruits mais pas les pates, ... Le but étant de créer des groupes de clients qui semblent aimer les même produits mais qui n'ont pas tout notés les mêmes produits. Une notation aléatoire ne poserait aucuns problèmes mais entraînerait une recommandation aléatoire.

- Repartir dans l'espace admin et cliquer sur le lien "create_csv_from_dt". Un fichier data.csv sera téléchargé.

- Copier ce fichhier dans le dossier src du site web en remplaçant l'ancien.

- Executer le programme recommand-sys avec Octave.

- Repartir dans l'espace admin et cliquer sur le lien "recommandation systeme".