

---

**UNIVERSITE KASDI MERBAH  
OUARGLA**

---

**Faculté des Nouvelles Technologies de  
l'Information et de la Communication**

**Département d'Informatique et des  
Technologies de l'Information**



**Mémoire MASTER ACADEMIQUE**

**Domaine :Informatique**

**Filière :Informatique**

**Spécialité : Informatique industrielle**

**Présenté par : Laouar Assia - Korichi Khadidja**

**Thème**

**Un système de recommandation pour l'assistance à la  
navigation sur Internet**

**Soutenu le : 04/06/2016**

**Devant le jury composé de :**

M. Mahdjoub	Med Bachir	Universié KASDI Merbah- Ouargla	Président
M. Zga	Adel	Universié KASDI Merbah- Ouargla	Examinateur
M. Herrouz	Abdelhakim	Université KASDI Merbah- Ouargla	Rapporteur

**Année universitaire :2015/2016**

---

# DÉDICACE

---

Dieu le tout puissant, mon créateur. À mon père, en signe d'amour, de reconnaissance et de gratitude pour tous les soutiens et les sacrifices dont il a fait preuve à mon égard. À ma mère, ma raison d'être, ma raison de vivre, la lanterne qui éclaire mon chemin et m'illumine de douceur et d'amour. À mes frères et mes sœurs. À toutes mes amies, qui m'ont beaucoup aidé à réaliser ce projet, et à tous mes proches.



---

# REMERCIEMENTS

---

Nous tenons tout d'abord à remercier Dieu le tout puissant et miséricordieux, qui nous a donné la force et la patience d'accomplir ce modeste travail. En second lieu, nous tenons à remercier particulièrement notre encadreur monsieur Abdelhakim Herrouz maitre-assistant A à Université Kasdi Merbah Ouargla pour son encadrement et pour l'intérêt qu'il a manifesté à notre travail. Nos vifs remerciements vont également aux membres du jury pour l'intérêt qu'ils ont porté à notre recherche en acceptant d'examiner notre travail et de l'enrichir par leurs propositions. Enfin, nous tenons également à remercier tous ceux qui ont contribué de près ou de loin à la réalisation de ce travail.



---

# TABLE DES MATIÈRES

---

<b>Table des matières</b>	<b>iii</b>
<b>Table des figures</b>	<b>v</b>
<b>Liste des tableaux</b>	<b>vi</b>
<b>Résumé</b>	<b>1</b>
<b>Introduction générale</b>	<b>2</b>
<b>1 Les systèmes de recommandation</b>	<b>4</b>
1.1 Introduction . . . . .	4
1.2 Historique . . . . .	4
1.3 Définitions des termes hypertextuelle et hypermédia . . . . .	6
1.4 Les modèles de navigation . . . . .	7
1.5 Problématique de la navigation sur Internet . . . . .	8
1.5.1 La surcharge cognitive . . . . .	9
1.5.2 La désorientation . . . . .	9
1.6 La recommandation sur Internet . . . . .	10
1.7 Définition des Systèmes de Recommandation (SR) . . . . .	10

1.8	Différence entre Système de Recommandation SR et Moteur de Recherche MR . . . . .	12
1.9	Les étapes principales de la recommandation . . . . .	13
1.9.1	La collecte d'information . . . . .	13
1.9.2	Modèle utilisateur . . . . .	14
1.9.3	Liste de recommandations . . . . .	15
1.10	Les techniques de recommandation . . . . .	15
1.10.1	L'analyse basée sur l'item . . . . .	16
1.10.2	L'analyse basée sur l'utilisateur . . . . .	16
1.11	Quelle analyse est la bonne? . . . . .	17
1.12	Conclusion . . . . .	17
<b>2</b>	<b>Classification des systèmes de recommandation</b>	<b>18</b>
2.1	Introduction . . . . .	18
2.2	Différences classifications des systèmes de recommandation . . . . .	18
2.3	Classification classique . . . . .	19
2.3.1	Le filtrage basé sur le contenu . . . . .	19
2.3.2	Le filtrage collaboratif . . . . .	20
2.3.3	Le filtrage hybride . . . . .	21
2.4	Autres classifications . . . . .	21
2.4.1	Recommandation Personnalisée . . . . .	22
2.4.2	Recommandation Objet . . . . .	22
2.4.3	Recommandation Sociale . . . . .	22
2.4.4	Recommandation Hybride . . . . .	22
2.4.5	Recommandation Éditoriale . . . . .	23
2.4.6	Recommandation Contextuelle . . . . .	23
2.5	Avantages et inconvénients des systèmes de recommandation . . . . .	24
2.6	Conclusion . . . . .	26

<b>3</b>	<b>Présentation des systèmes de recommandation existants</b>	<b>27</b>
3.1	Ezako-Un surplus d'intelligence . . . . .	27
3.2	Amazon . . . . .	28
3.3	Pigdata-La recommandation produit orienté métier et visuel . . . . .	29
3.4	Google . . . . .	29
3.5	Frosmo-Un boîte à outils au service de l'amélioration de la conversion . . . . .	30
3.6	Target2Sell-Pour les petits comme les gros marchands . . . . .	30
3.7	Nuukik-Plusieurs approches d'analyse prédictive . . . . .	31
3.8	Conclusion . . . . .	33
<b>4</b>	<b>Développement du système de recommandation</b>	<b>34</b>
4.1	Introduction . . . . .	34
4.2	Environnement de travail . . . . .	34
4.2.1	Matériel . . . . .	34
4.2.2	Python . . . . .	35
4.2.3	Delicious . . . . .	36
4.3	Partie développement . . . . .	36
4.3.1	La conception . . . . .	36
4.3.2	Possibilités graphiques sous Python . . . . .	37
4.3.3	Recueillir les préférences (Dictionnaire Local) . . . . .	38
4.3.4	Trouver les utilisateurs similaires . . . . .	39
4.3.5	Classement des critiques . . . . .	43
4.3.6	La recommandation d'articles . . . . .	44
4.3.7	Construire une recommandation pour (del.icio.us) . . . . .	47
4.4	Conclusion . . . . .	49
	<b>Conclusion générale</b>	<b>50</b>
	<b>Bibliographie</b>	<b>55</b>
	<b>Annexe</b>	<b>56</b>

---

# TABLE DES FIGURES

---

1.1	Les trois modèles de navigation. . . . .	8
1.2	Facteurs de recommandation. . . . .	11
1.3	Analyse basée sur l’item. . . . .	16
1.4	Analyse basée sur l’utilisateur. . . . .	16
2.1	Classification classique. . . . .	19
2.2	Le principe de filtrage basé sur le contenu. . . . .	20
2.3	Le principe du filtrage collaboratif. . . . .	20
2.4	Le filtrage hybride. . . . .	21
3.1	Amazon recommandation. . . . .	28
3.2	Pigdata-La recommandation produit orienté métier et visuel. . . . .	29
3.3	Frosmo moteur de recommandation. . . . .	30
3.4	Target2Sell. . . . .	30
3.5	Moteur _ nuukik. . . . .	31
4.1	Digrammes de conception. . . . .	36
4.2	Les taux d’utilisation des bibliothèques graphiques. . . . .	38
4.3	Dictionnaire local. . . . .	39

4.4	Test d'accès au dictionnaire. . . . .	39
4.5	Fonction de Sim Distance. . . . .	41
4.6	L'exécution de Sim Distance. . . . .	41
4.7	Fonction Sim Pearson. . . . .	43
4.8	L'exécution de Sim Pearson. . . . .	43
4.9	Fonction de topMatches. . . . .	44
4.10	L'exécution de topMatches. . . . .	44
4.11	Fonction de GetRecommendations. . . . .	45
4.12	L'exécution de GetRecommendations. . . . .	45
4.13	L'exécution Comparateur d'Objet. . . . .	46
4.14	Fonction de Comparateur d'Objet. . . . .	46
4.15	Fonction transformPrefs. . . . .	46
4.16	Dictionnaire transformé. . . . .	47
4.17	Site Delicious. . . . .	47
4.18	Get-popular(liste de dictionnaires URL ). . . . .	48
4.19	Fichier deliciousrec.py. . . . .	48
4.20	Fichier deliciousrec.py modifié. . . . .	48
4.21	Recommandation des voisins et des liens. . . . .	49



---

# LISTE DES TABLEAUX

---

1.1	Exemple de matrice d'usage . . . . .	15
2.1	Les avantages et les inconvénients des techniques de recommandation . . . . .	25
3.1	Comparatif des moteurs de recommandation d'e-commerce . . . . .	32
4.1	Espace de préférences. . . . .	40
4.2	Sim-Pearson. . . . .	42

---

# RÉSUMÉ

---

---

# INTRODUCTION GÉNÉRALE

---

Compte tenu de la très grande masse d'information aujourd'hui disponible sur Internet et le besoin en communication, en échange d'idées et en partage d'informations, les besoins de l'utilisateur sont devenus difficiles à traiter. D'une part, parce qu'ils ne sont pas formulés explicitement et, d'autre part, parce qu'ils sont évolutifs. Par ailleurs, de nombreux problèmes apparaissent au niveau de la navigation hypertextuelle (surcharge cognitive et la désorientation). Il est donc devenu primordial de concevoir des mécanismes qui permettent aux utilisateurs d'accéder à ce qui les intéresse le plus rapidement possible.

Ce travail présente le développement d'un système de recommandation sociale. Dans un premier temps, nous allons travailler sur un dictionnaire local puis nous généraliserons notre travail sur le site [del.icio.us](http://del.icio.us); car ce site de partage de signets ne possède pas de moteur de recommandation.

**Ce mémoire est organisé en quatre chapitres :**

**Le chapitre 1 :** est consacré à la présentation des systèmes de recommandation en général.

**Le chapitre 2 :** portera sur la présentation des différentes classifications des systèmes

de recommandation.

**Le chapitre 3 :** est consacré à une comparaison de quelques sites utilisant un système de recommandation.

**Le chapitre 4 :** proposera la présentation de l'application réalisée.

Enfin, en guise de conclusions, nous indiquerons quelques remarques sur ce travail en concluant par un bilan et nous exposerons les perspectives pour de futurs travaux.

---

# LES SYSTÈMES DE RECOMMANDATION

---

## 1.1 Introduction

Avec le problème de la croissance phénoménale de la quantité d'informations disponibles sur Internet et le besoin de changer d'un état de la rareté à l'état de l'abondance, il est nécessaire de disposer d'un système qui peut recommander des bons produits en tenant compte de certains paramètres. Il s'agit là du système de recommandation qui est devenu plus populaire durant ces 10 à 20 dernières années. Un système de recommandation fournit à des utilisateurs des suggestions qui répondent à leurs besoins et préférences informationnels. Les applications de recommandation peuvent être trouvées dans une grande variété d'industries, entreprises, service financier, musique / radio en ligne, TV et vidéos, les publications en ligne, et d'innombrables autres.

## 1.2 Historique

Les systèmes de recommandation sont reconnus assez tôt dans l'histoire de l'informatique : " Information Lens System " [11] peut être considérée comme le premier système

de recommandation. À l'époque, l'approche la plus commune pour le problème du partage d'informations dans l'environnement de messagerie électronique était la liste de distribution basée sur les groupes d'intérêt. La première définition pour le filtrage a été donnée aussi par Malone [11] : *" Même si le terme a une connotation littérale de laisser les choses dehors (filtrage négatif : enlèvement), nous l'utilisons ici dans un sens plus général qui consiste à sélectionner les choses à partir d'un ensemble plus large de possibilités (filtrage positif : sélection) "* .

Au cours des années 1990, les systèmes de recommandation sont devenus un important domaine de recherche avec la publication des premiers articles dans le domaine du filtrage collaboratif. La littérature académique a introduit le terme de filtrage collaboratif par le système Tapestry [8]. Il a été développé en 1992 par le centre de recherche de "Xerox" aux Etats Unis et qui a permis aux utilisateurs de créer des requêtes permanentes, basées sur les annotations des utilisateurs.

Quelques années plus tard, un certain nombre de systèmes académiques de recommandation ont été introduits dans des domaines différents tels que le système de recommandation d'articles d'actualités et de films développé par "GroupLens" [20] et le système de recommandation de musique "Ringo" proposé par Maes et Shardanand [10]. Ces deux systèmes sont également basés sur le filtrage collaboratif [16].

À la fin des 1990, Les déploiements de la technique de recommandations commerciales ont commencé à émerger. Peut-être la plus célèbre application sur une large gamme de systèmes techniques de recommandation est Amazon.com, Amazon.com à plus de 410 000 titres dans sa boutique Kindle seul seulement. Soutenir la découverte dans des espaces d'information de cette ampleur est un vrai challenge.

En suite, en 2006, Netflix a lancé Netflix Prize pour améliorer l'état de recommandation des films ; aussi Netflix a plus de 17.000 films dans sa sélection [13].

Aujourd'hui les systèmes de recommandation sont devenus très populaires et sont utilisés dans diverses applications Web.

### 1.3 Définitions des termes hypertextuelle et hypermédia

Quand on cherche des définitions précises sur les hypertextes, on s'aperçoit que les avis des auteurs divergent fortement. Celles qui attirent notre attention car utilisées dans notre approche, consiste à définir ces termes suivant deux points de vue : celui de la structure et celui de l'interaction entre l'utilisateur et le système.

Du point de *vue structurel*, Balasubramanian [27] définit un hypertexte comme un système composé de nœud et de liens. Les nœud peuvent être composés d'informations textuelles (cas d'hypertexte) ou d'informations multimédias (cas d'hypermédia ouverts comme Internet). Les nœuds sont reliés les uns aux autres par des liens, nous distinguons les nœuds qui sont à l'origine de liens (on parle de références) et les nœuds qui sont les destinations des liens (on parle de référés). Les liens peuvent être unidirectionnels, permettant d'aller d'une page a un autre ou bidirectionnels, afin de faciliter le retour au point de départ. Ils peuvent aussi être typés afin de spécifier la sémantique du lien et, être disposés n'importe où dans une page. L'ensemble constitué par ces liens définit l'architecteur du système et qui est nommé hyperespace.

Du point de *vue fonctionnel*, l'hypertexte peut être considéré comme étant un procédé informatique qui permet d'associer une entité souvent minimale (un mot, une portion d'image, un icône) a une autre entité souvent plus étendue (paragraphe, une image, ou une page). Ce mécanisme permet à l'utilisateur de se diriger librement dans l'hyperespace. En activant à l'aide d'un pointeur (ex : une souris) une zone du document (un mot ou un groupe de mots mis en évidence) qui est l'origine d'une association, l'utilisateur peut immédiatement atteindre une autre partie du document. L'utilisateur n'est pas obligé de suivre le cheminement prévu par l'auteur, il définit son propre parcours en fonction de ses envies et de ses centres d'intérêts. Ainsi, Rhéaume [19] écrit : "*l'hypertexte est par conséquent un document virtuel qui n'est jamais globalement perceptible dont l'actualisation d'une des potentialités est conditionnée par l'effectivité de la lecture. Cette propriété de l'hypertexte en fait un document " interactif " dans lequel le lecteur tient*

*une place prépondérante* ". Cette définition privilégie la problématique de la présentation des informations (en termes d'interface) et dénote de l'aspect éphémère et aléatoire des informations présentées dans les hypermédias. Les définitions que nous venons de donner ainsi que beaucoup d'autres tendent à dire qu'un hypertexte est un dispositif informatise permettant l'interconnexion de documents de divers types par des mécanismes associatifs sous contrôle de l'utilisateur. Il est représenté par un réseau de nœuds et de liens dans lequel les nœuds sont des documents et liens des références entre ces nœuds.

Il recouvre la notion d'intégration via les techniques d'interfaçage et celle d'organisation par l'accès associatif aux informations. Ainsi un hypertexte a deux aspects : un aspect organisation interne (nœuds et liens) et un aspect organisation externe (point d'ancrage et navigation) [3].

## 1.4 Les modèles de navigation

Généralement, on distingue trois modèles de navigation dans un espace d'information :

- La navigation spatiale
- La navigation sémantique
- La navigation sociale

Ces trois modèles ne s'excluent pas et entrent souvent en interactions (Cf. Figure 1.1 [ ]). Le modèle de navigation sociale privilégie les interactions humaines et la recommandation. C'est une technique relationnelle qui juge d'une information en tenant compte de sa notoriété ou de la notoriété de l'auteur qui donne l'information. Ceci permet à celui qui veut engager une recherche sur Internet, de bénéficier des recherches et des parcours effectués auparavant par d'autres internautes. Les principes de la navigation sociale peuvent être trouvés chez Michel de Certeau et Luce Giard dans "*l'ordinaire de la communication* ", lorsque ces auteurs montrent que pour trouver une information, la solution qui consiste à faire appel à des experts n'est pas la seule pratiquée. Elle s'appuie sur une distance sociale maximale, sur la légitimité du spécialiste, sur des autorités. Les pratiques, qui constituent



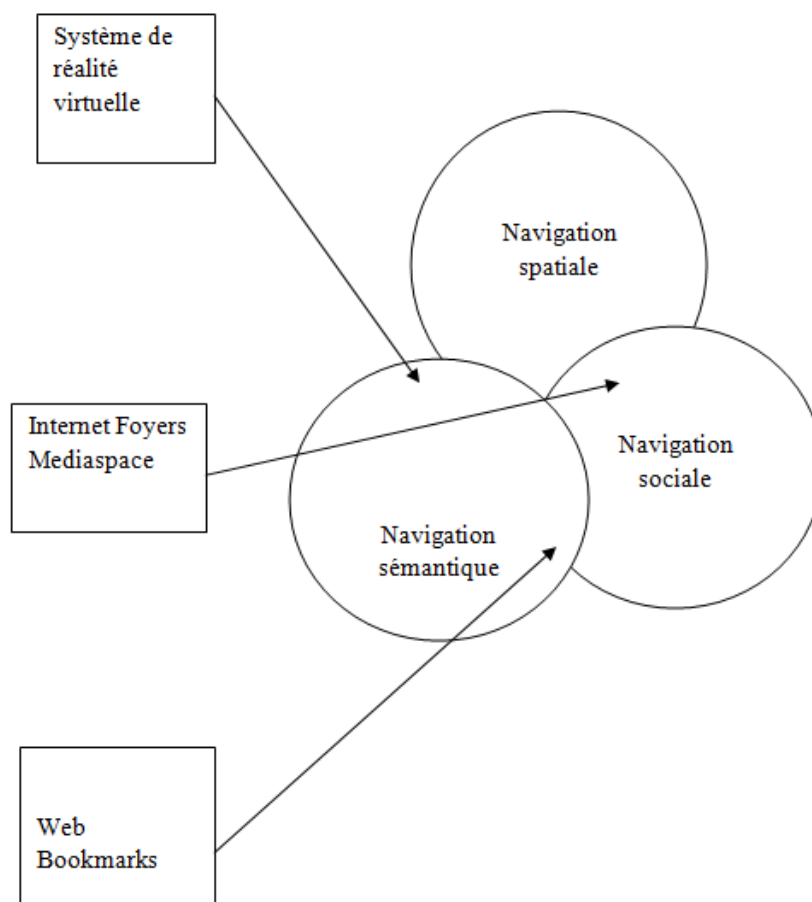


FIGURE 1.1 – Les trois modèles de navigation.

l’angle d’attaque privilégiée des auteurs, révèlent que l’on peut tout aussi bien chercher la même information auprès d’un ami, d’un voisin, d’un membre de sa famille, qui présentent tous l’avantage de la proximité sociale. Ce seul trait suffira à valider leurs informations, car elles seront exprimées dans les termes d’une culture et d’une expérience partagées [25].

## 1.5 Problématique de la navigation sur Internet

Dès la fin des années 1980, les recherches avaient mis en évidence des difficultés spécifiques de la navigation [5]. Ces problèmes ont pris une nouvelle dimension avec l’arrivée au début des années 1990 - du Web. On peut cependant les ramener à deux grands types de problèmes qui sont identifiés par Conklin [5] : la désorientation et la surcharge cognitive. Ces deux problèmes s’appellent mutuellement. Chaque type de problème a été l’objet de

nombreuses études spécifiques.

### 1.5.1 La surcharge cognitive

Un utilisateur est en surcharge cognitive quand il se trouve submergé par l'abondance d'informations et par l'excès de traitement à réaliser. Cette surcharge correspond à la diminution d'attention et de concentration sur l'objectif principal due à notre mémoire humaine (ou mémoire de travail) qui est trop faible et n'est pas capable de tout retenir. Car, comme l'avance Miller, un individu ne peut retenir plus de sept (07) éléments à la fois sans engendrer l'oubli ou la rétention superficielle [14].

### 1.5.2 La désorientation

Elle est définie comme un effet cognitif produit chez l'utilisateur qui perd le lien significatif entre ce qui est affiché sur son écran de travail et le but réel qu'il poursuivait au départ de sa navigation. Ce problème peut provenir de la mémoire à court terme qui est trop courte pour se rappeler les informations déjà reçues ou pour les relier aux objectifs poursuivis. Elle provient de la liberté de déplacement de noeud en noeud dans le système qui peut finir par troubler l'utilisateur au risque de se poser des questions du type "où suis-je?", "pourquoi suis-je là?" et "que dois je faire?". C'est cet effort de concentration nécessaire pour maintenir différents travaux ou chemins en même temps qui entraînent une surcharge cognitive de l'utilisateur. Gall et Hannafin (1994) [7] reprennent la définition de la désorientation proposée par Conklin [5] : "*Conklin (1987) described disorientation as either not knowing where the current location is in relation to the rest of the network, or being unable to find a location in the network that is known to exist*".

Edwards et Hardman rappellent, eux, celle proposée par Elm et Woods : "*The user not having a clear conception of the relationships within the system, or knowing his present location in the system relative to the display structure, and finding it difficult to decide where to look next within the system*" [6]. Trois niveaux de difficultés ont été mis en évidence :

1. la navigation dans l'espace logiciel : c'est le fonctionnement du système hypertexte

qui n'est pas connu. L'utilisateur ne sait pas ce qu'il faut faire ;

2. la navigation dans l'espace conceptuel : ce sont les catégories qui structurent l'information qui ne sont familières, l'utilisateur n'arrive pas à rentrer dans le système conceptuel de l'auteur ;
3. la navigation dans le texte : c'est le parcours des liens hypertextuels qui n'est pas familier. L'utilisateur perd le fil de sa navigation.

Enfin, la désorientation est un effet cognitif qui semble fréquent chez les usagers novices qui ont du mal à tirer profit des fonctionnalités offertes par les hypertextes.

## 1.6 La recommandation sur Internet

Internet est un nouvel outil d'information et de communication en pleine évolution offrant des perspectives de croissance exceptionnelles. C'est devenu un formidable moyen pour communiquer, d'échanger, de travailler, de rencontrer, d'apprendre et même de commercer. Internet est entraîné à modifier nos habitudes et de transformer complètement notre mode de vie.

Les systèmes de recommandation sont nés de la volonté de pallier le problème de surcharge d'information sur le Web. Combinant des techniques de filtrage d'information, personnalisation, intelligence artificielle, réseaux sociaux et interaction personne-machine, les systèmes de recommandation fournissent à des utilisateurs des suggestions qui répondent à leurs besoins et préférences informationnelles. En effet, les systèmes de recommandation sont particulièrement sollicités dans les applications de commerce électronique [23].

## 1.7 Définition des Systèmes de Recommandation (SR)

En raison de la popularité croissante des systèmes de recommandation dans de nombreux domaines, conduit l'apparition d'un conflit dans sa définition. Il existe de beaucoup des nombreuses définitions parmi eux la définition de Saimadhu [26], Saimadhu propose la définition suivante : " *un système de recommandation est une boîte noire qui analyse*

un certain ensemble d'utilisateurs et présentes les items où un seul utilisateur peut aimer ". Et la définition générale de Robin Burke [4] qui les défini comme suit : "des systèmes capable de fournir des recommandations personnalisées permettant de guider l'utilisateur vers des ressources intéressantes et utiles au sein d'un espace de données important ". En remarque dans les deux définitions que les systèmes de recommandation fonctionnent idéalement sur les manières suivantes : soit sur les propriétés des éléments que l'utilisateur aime ou non " les items ", soit sur la similarité entre les utilisateurs et de leur recommander des articles en conséquence. Il est également possible de combiner ces deux méthodes pour construire un moteur beaucoup plus robuste de recommandation. Beaucoup algorithmes ont été utilisées pour mesurer la similarité des utilisateurs ou des items dans les systèmes de recommandation par exemple le k-nearest neighbors (k-NN) et Pearson correlation ... Un système de recommandation prend en compte plusieurs facteurs en considération pour faire une recommandation à un utilisateur :

- **le profil de l'utilisateur** : âge, situation géographique, historique, ...
- **Informations sur les différents articles disponibles** : contenu associé à l'article
- **Les interactions des utilisateurs** : notent, contenu de navigation
- **Le contexte dans lequel les articles seront affichés** : sous-catégorie d'articles qui doivent être considéré

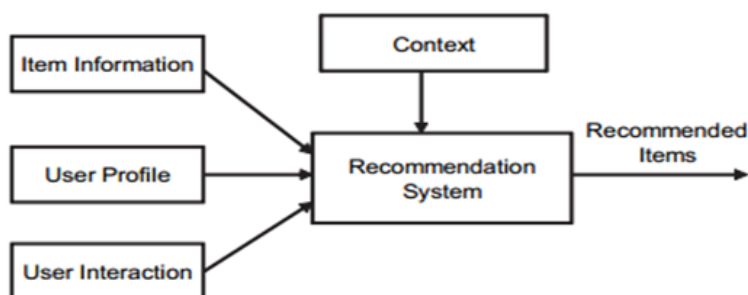


FIGURE 1.2 – Facteurs de recommandation.

## 1.8 Différence entre Système de Recommandation SR et Moteur de Recherche MR

**Moteur de Recherche :** Un moteur de recherche (Search engine en anglais) est un service en ligne permettant de trouver facilement des ressources grâce à un ou plusieurs mots-clés dans un formulaire de recherche. Les ressources peuvent être des pages web, des articles de forums Usenet, des images, des vidéos, des fichiers, etc..

Les exemples les plus connus sont Google et Yahoo !

**Fonctionnement d'un moteur de recherche :** Un moteur de recherche fonctionne grâce à un grand nombre de serveurs appelés robots chargés de parcourir la toile en suivant récursivement les liens des millions de pages Web et indexant le contenu dans de gigantesques bases de données afin de permettre leur interrogation [30].

**Moteur de Recommandation :** Proposer des objets susceptibles d'intéresser un individu, tel est l'objectif d'un moteur de recommandation. Le champ de ces objets est immense. Ce peut être des chansons, des programmes télévisés, des films, des livres, des appareils électroménagers, des destinations touristiques, des contacts professionnels et tant d'autres que l'individu pourrait apprécier, qu'il soit simple utilisateur ou bien consommateur. Un moteur de recommandation est donc un ensemble de règles applicables à des informations provenant de l'individu immergé dans un ensemble beaucoup plus vaste, celui des objets qu'on peut lui proposer et celui des autres individus. De façon simpliste, le raisonnement est le suivant : "Les utilisateurs regroupés selon des critères proches de ceux l'utilisateur ciblé ont apprécié une dizaine de livres. On proposera donc à notre utilisateur les livres à grand succès".

Comme pour les moteurs de recherche, la pertinence de ces suggestions sera un élément clef du choix de l'utilisateur lorsqu'il s'agira de recourir à un service. Les intérêts de la recommandation sont multiples. On retiendra principalement l'amélioration de la recherche première d'un utilisateur et la génération de ventes additionnelles [31].

Au dernier, Nous concluons que les moteurs de recherche ne sont pas les mêmes que les

systèmes de recommandation. Tous deux peuvent fournir du contenu personnalisé qui correspond à nos besoins.

## 1.9 Les étapes principales de la recommandation

Un système de recommandation requiert généralement trois(03) étapes :

1. La première consiste à recueillir de l'information sur l'utilisateur ;
2. La deuxième consiste à bâtir une matrice ou un modèle utilisateur contenant l'information recueillie ;
3. La troisième consiste à extraire à partir de cette matrice une liste de recommandations.

### 1.9.1 La collecte d'information

Pour être pertinent, un système de recommandation doit pouvoir faire des prédictions sur les intérêts des utilisateurs. Il faut donc pouvoir collecter un certain nombre de données sur ceux-ci afin d'être capable de construire un profil pour chaque utilisateur. Une distinction peut être faite entre 2 formes de collecte de données :

1. **Collecte de données explicites - Filtrage actif** : La collecte repose sur le fait que l'utilisateur indique explicitement au système ses intérêts.

**Exemple :** Demander à un utilisateur de commenter, taguer/étiqueter, noter, liker ou encore ajouter comme favoris des contenus (objets, articles...) qui l'intéressent. On utilise souvent une échelle de ratings allant de 1 étoile (je n'aime pas du tout) à 5 étoiles (j'aime beaucoup) qui sont ensuite transformées en valeurs numériques afin de pouvoir être utilisées par les algorithmes de recommandation.

**Avantage :** Capacité à reconstruire l'historique d'un individu et capacité à éviter d'agrèger une information qui ne correspond pas à cet unique utilisateur (plusieurs personnes sur un même poste).

**Inconvénient :** Les informations recueillies peuvent contenir un biais dit de déclaration [24].

2. **Collecte de données implicite - Filtrage passif :** Elle repose sur une observation et une analyse des comportements de l'utilisateur effectué de façon implicite dans l'application qui embarque le système de recommandation, le tout se fait en "arrière-plan" (en gros sans rien demander à l'utilisateur).

**Exemple :**

- Obtenir la liste des éléments que l'utilisateur a écoutés, regardés ou achetés en ligne.
- Analyser la fréquence de consultation d'un contenu par un utilisateur, le temps passé sur une page.
- Monitorer le comportement en ligne de l'utilisateur.
- Analyser son réseau social.

**Avantage :** Aucune information n'est demandée aux utilisateurs, toutes les informations sont collectées automatiquement. Les données récupérées sont a priori justes et ne contiennent pas de biais de déclaration.

**Inconvénient :** Les données récupérées sont plus difficilement attribuables à un utilisateur et peuvent donc contenir des biais d'attribution (utilisation commune d'un même compte par plusieurs utilisateurs). Un utilisateur peut ne pas aimer certains livres qu'il a achetés, ou il peut l'avoir acheté pour quelqu'un d'autre.

### 1.9.2 Modèle utilisateur

Le modèle utilisateur se présente généralement sous forme de matrice appelée "matrice d'usages". On peut se le représenter comme un tableau qui contient des données recueillies

sur l'utilisateur associées aux produits disponibles sur le site web.

	Item 1	Item 2	Item 3	Item 4	Item 5	...
User 1	😊			😊		...
User 2		😊	😊			...
User 3		😊				...
User 4			😊	😊	😊	...
...	...	...	...	...	...	...

TABLE 1.1 – Exemple de matrice d'usage

Le tableau 1 présente un exemple fictif de matrice binaire contenant des informations de type " l'utilisateur u a apprécié/n'a pas apprécié l'item i ". Ces informations peuvent également être " a acheté/n'a pas acheté ", "a consulté/n'a pas consulté ", etc. Elles peuvent également se mesurer sur un nombre plus élevé de classes : " a mis 1/2/3/4/5 étoiles " etc. Un autre point important est comment le temps influence le profil de l'utilisateur. Les intérêts des utilisateurs, généralement, évoluent au cours du temps. Les données du modèle utilisateurs devraient donc constamment être réajustées pour rester conformes aux nouveaux centres d'intérêts de l'utilisateur.

### 1.9.3 Liste de recommandations

Pour extraire une liste de suggestions à partir d'un modèle utilisateur, les algorithmes utilisent la notion de mesure de similarité entre objets ou personnes décrits par le modèle utilisateur. La similarité a pour but de donner une valeur ou un nombre (au sens mathématique du terme) à la ressemblance entre 2 choses. Plus la ressemblance est forte, plus la valeur de la similarité sera grande. A l'inverse, plus la ressemblance est faible, et plus la valeur de la similarité sera petite. On verra quelques exemples plus tard dans ce texte.

## 1.10 Les techniques de recommandation

Les techniques sont divisées en deux analyses : SR à base d'items, et SR à base d'utilisateurs, nous allons expliquer chacune de ces deux analyses.



### 1.10.1 L'analyse basée sur l'item

Les approches dans cette classe appliquent les calculs statistiques sur les items. L'idée fondamentale est de proposer à l'utilisateur des items similaires à ceux qu'il a aimés. Deux approches sont principalement connues dans cette classe : la recommandation à base de contenu et le filtrage collaboratif à base d'items [2] [20].

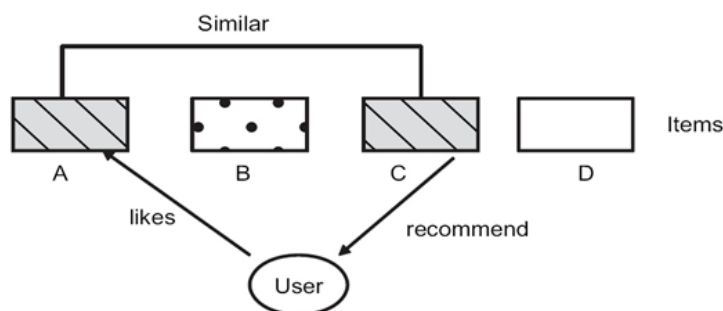


FIGURE 1.3 – Analyse basée sur l'item.

### 1.10.2 L'analyse basée sur l'utilisateur

Les approches basées sur les utilisateurs considèrent qu'un utilisateur a tendance à privilégier les items appréciés par les utilisateurs qui lui sont similaires. Chaque approche

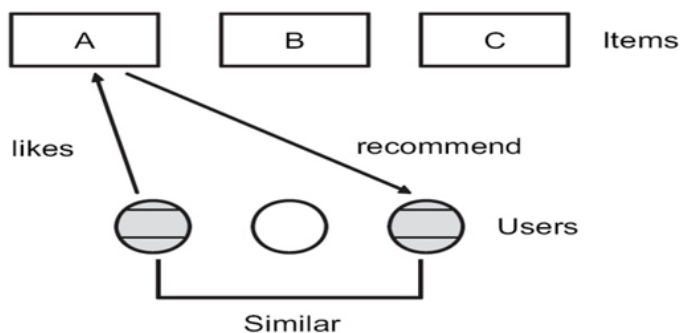


FIGURE 1.4 – Analyse basée sur l'utilisateur.

interprète la similarité différemment, le FC considère que les utilisateurs sont voisins s'ils ont un comportement similaire à l'égard des items (ils attribuent des notes proches aux items) [18]. En revanche, le SRC considère qu'un utilisateur a une tendance à imiter le comportement et suivre les conseils des utilisateurs auxquels il fait confiance [12].

## 1.11 Quelle analyse est la bonne ?

- Si notre liste d'articles ne change pas beaucoup, il est utile de créer une corrélation point-à-point en utilisant l'analyse basée sur l'article. Le tableau obtenu peut ensuite être utilisé dans le moteur de recommandation.
- Si notre liste d'articles change fréquemment, par exemple des articles liés à l'actualité, il peut être utile de trouver des utilisateurs connexes pour les recommandations.
- Si l'article est recommandé à un utilisateur, il n'y a pas d'autre choix que de trouver des utilisateurs liées.
- La dimension de l'espace de l'article et l'utilisateur peut être utile dans le choix des approches et ceci peut être plus facile à mettre en œuvre. Par exemple, si nous avons des millions d'utilisateurs et un ordre de grandeur de moins d'éléments, il peut être plus facile de faire une analyse basée sur l'article. Chaque fois que les utilisateurs se voient, vous traitez avec des matrices creuses.

## 1.12 Conclusion

Trouver des informations sur un grand site peut être un processus long et difficile. Un système de recommandation peut aider l'utilisateur à trouver des informations en leur fournissant des suggestions personnalisées. Dans ce chapitre, nous avons présenté tout d'abord la définition de la recommandation en général et ses objectifs principaux. Ensuite, nous avons souligné les étapes de construction d'un système de recommandation. Comme nous avons convenu qu'un système de recommandation a pour objectif de fournir à un utilisateur des ressources pertinentes en fonction de ses préférences. Ce dernier voit ainsi réduit son temps de recherche mais reçoit également des suggestions de la part du système auxquelles il n'aurait pas spontanément prêté attention. On a essayé d'expliquer les deux principaux techniques visant à produire des systèmes de recommandation qui sont : l'analyse basée sur l'item et l'analyse basée sur l'utilisateur.

---

# CLASSIFICATION DES SYSTÈMES DE RECOMMANDATION

---

## 2.1 Introduction

Durant ces dix dernières années, plusieurs classifications sont apparus, suivant les données à recommander, suivant les informations disponibles et bien évidemment suivant l'objectif visé. Ce qui conduit l'émergence d'un débat sur quelle classification est la meilleure. Ce chapitre présente et analyse les différentes classifications existants et met en évidence les différentes approches de chaque classification. Enfin, nous allons préciser quelle est la classification générale ou la classification de base.

## 2.2 Différences classifications des systèmes de recommandation

Il est possible de classer les systèmes de recommandation en différentes manières mais la classification la plus fréquente est la classification classique énoncée ci-dessous.

## 2.3 Classification classique

La plupart des systèmes de recommandation prennent l'une des deux approches de base : filtrage collaboratif ou filtrage par le contenu. D'autres approches (telles que les approches hybrides) existent également. A.Tuzhiline et G.Adomavicius [1] ils ont dit : "*the current generation of recommendation methods that are usually classified into the following three main categories : content-based, collaborative, and hybrid recommendation approaches.*".

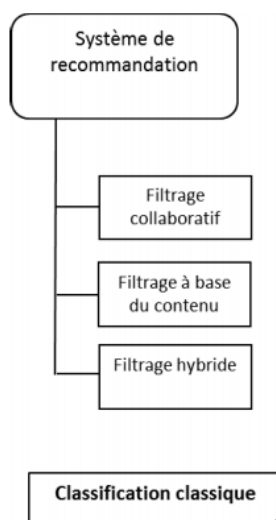


FIGURE 2.1 – Classification classique.

### 2.3.1 Le filtrage basé sur le contenu

Dans cette approche aussi appelée filtrage cognitif, le choix des documents à recommander est basé sur une comparaison des thèmes abordés dans les documents par rapport aux thèmes intéressant l'utilisateur. Méthodes de filtrage basées sur le contenu sont basées sur une description de l'article et le profil de la préférence de l'utilisateur. Dans un système de recommandation basée sur le contenu, les mots clés qui sont utilisés pour décrire les éléments : à côté, le profil d'utilisateur est conçu pour indiquer le type d'article que cet utilisateur aime. En d'autres termes, ces algorithmes essaient de recommander des éléments qui sont similaires à ceux que l'utilisateur a aimés dans le passé (ou examine dans le présent). En particulier, divers articles de candidats sont comparés aux éléments

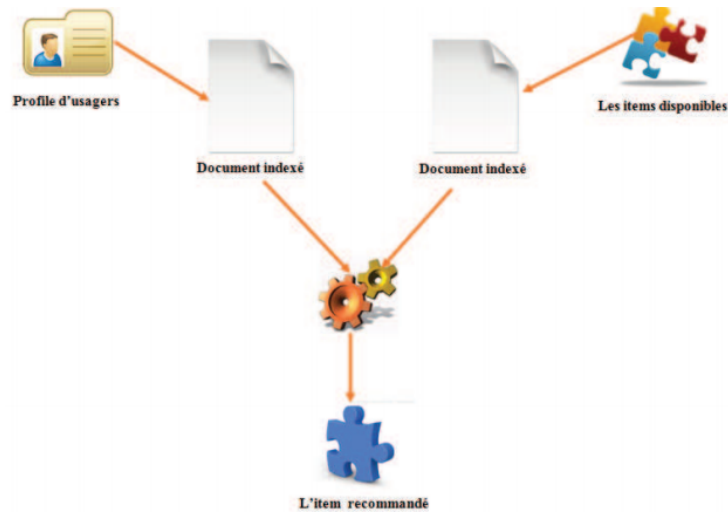


FIGURE 2.2 – Le principe de filtrage basé sur le contenu.

précédemment notés par l'utilisateur et les éléments les plus correspondants sont recommandés. Cette approche a ses racines dans la récupération de l'information et filtrage de recherche de l'information.

### 2.3.2 Le filtrage collaboratif

Les méthodes de filtrage collaboratif sont basées sur la collecte et l'analyse d'une grande quantité d'informations sur les comportements, les activités et les préférences des utilisateurs et de prédire ce que les utilisateurs apprécieront en fonction de leur similarité avec d'autres utilisateurs. Le principal avantage de l'approche de filtrage collaboratif est qu'il ne repose pas sur la machine contenue analysable et il est donc capable de recommander de façon précise les éléments complexes tels que les films, sans nécessiter une compréhension de l'article lui-même.

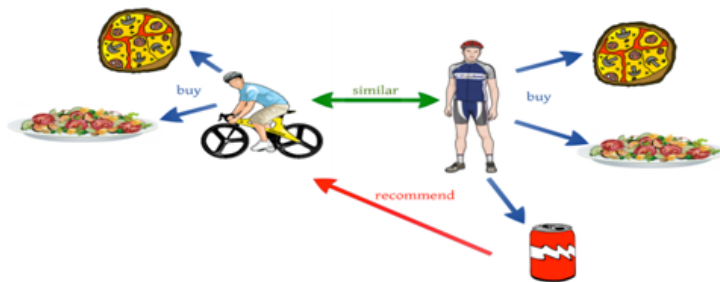


FIGURE 2.3 – Le principe du filtrage collaboratif.

### 2.3.3 Le filtrage hybride

Dans [4], Burke définit le filtrage hybride comme étant une combinaison des méthodes traditionnelles précédemment présentées afin d'en pallier les limites. Ces dernières sont actuellement les plus représentées dans la littérature, notamment à cause du fait qu'elles soient jugées comme étant les plus efficaces [21] [16].

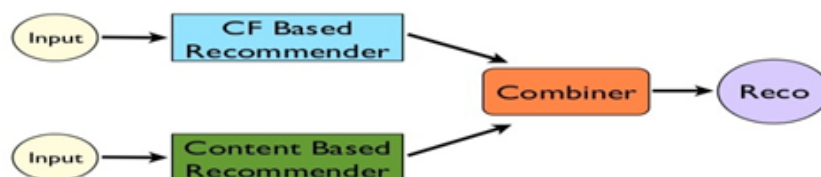


FIGURE 2.4 – Le filtrage hybride.

## 2.4 Autres classifications

En 2002, Burke [4] a ajouté trois autres approches : La recommandation basée sur les données démographiques, la recommandation basée sur l'utilité et la recommandation basée sur la connaissance. Mais ces trois approches peuvent être considérées comme des formes particulières de recommandation basée sur le contenu, aussi Nakagawa et Mobasher [15] ont annexé à ce qui précède, un autre approche de la recommandation basée sur la détection de motifs et qui n'est incluse ni dans la recommandation basée sur le contenu, ni dans le filtrage collaboratif. Notons que Mathieu, Mathieu [24] énonce qu'il y a seulement quatre(4) approches possibles pour un système de recommandation et qui sont :

- Recommandation Personnalisée ;
- Recommandation Objet (Content-Based filtering CB) ;
- Recommandation Sociale (Collaborative Filtering CF - Context Aware) ;
- Recommandation Hybride.

### 2.4.1 Recommandation Personnalisée

Il s'agit de recommander des objets sur la base du comportement passé de l'utilisateur.

### 2.4.2 Recommandation Objet

Il s'agit de recommander des objets (ou contenus) en se basant sur les qualités et propriétés intrinsèques de l'objet lui-même et en les corrélant avec les préférences et intérêts de l'utilisateur. Ce type de système va donc extraire un certain nombre de caractéristiques et attributs propres à un contenu, afin de pouvoir recommander à l'utilisateur des contenus additionnels possédant des propriétés similaires. Cette méthode crée un profil pour chaque objet ou contenu, c'est-à-dire un ensemble d'attributs/propriétés qui caractérisent l'objet.

### 2.4.3 Recommandation Sociale

Il s'agit de recommander des choses sur la base du comportement passé des utilisateurs similaires, en effectuant une corrélation entre des utilisateurs ayant des préférences et intérêts similaires. On utilise des méthodes qui collectent et analysent des données sur le comportement, les activités, les préférences des utilisateurs. Ensuite, les algorithmes tentent de prédire ce que l'utilisateur aimera en cherchant des utilisateurs qui ont les mêmes comportements que l'utilisateur à qui l'on souhaite faire des recommandations. L'idée sous-jacente est de dire que si une personne A a la même opinion (ou les mêmes goûts) qu'une personne B sur un objet  $x$ , alors la personne A a plus de chance d'avoir la même opinion que B sur un autre objet  $y$ , plutôt que d'avoir la même opinion que quelqu'un choisi au hasard pour l'objet  $y$ . L'idée de base est donc de dire que si des utilisateurs ont partagés des mêmes intérêts dans le passé, il y a de fortes chances qu'ils partagent aussi les mêmes goûts dans le futur.

### 2.4.4 Recommandation Hybride

Une combinaison des trois approches ci-dessus permet de résoudre des problèmes comme le cold Start et la sparsity (rareté) qu'on retrouve dans une approche de recommandation uniquement sociale. En appliquant préalablement un algorithme de recommandation ob-

jet sur les contenus pour en exploiter leurs descriptions et caractéristiques, accompagné ensuite d'un algorithme de recommandation sociale pour effectuer les recommandations peut aider à résoudre ces limitations. Autrement dit, pour les cas de rareté, lorsque peu d'items ont été évalués par les utilisateurs et qu'un filtrage collaboratif n'est pas possible, ce qu'on fait, c'est qu'on assigne en premier lieu un pseudo-rating ou vote artificiel par défaut à l'utilisateur sur les contenus disponibles en utilisant préalablement un algorithme objet. Puis, on applique sur la matrice (contenant peu de vrais ratings et beaucoup de pseudo-ratings) un filtrage collaboratif [21]. Damien Poirier [17] dit " *Il existe différentes formes de recommandation, suivant les données à recommander, suivant les informations disponibles et bien évidemment suivant l'objectif visé. Nous présentons ici les formes de recommandation les plus répandues* " de même Damien a proposé une classification en (04) approcher Il a ajouté en plus de la recommandation personnalisée et la recommandation sociale la suit :

#### **2.4.5 Recommandation Éditoriale**

La recommandation éditoriale est généralement utilisée lorsqu'aucun autre système de recommandation n'est présent ou encore lorsque le système n'a aucune connaissance sur le visiteur du site. Cette forme de recommandation a pour principal objectif d'attirer rapidement l'œil de l'utilisateur novice afin de lui procurer l'envie de parcourir une partie du catalogue, comme le ferait la une d'un journal. Pour cela, on peut mettre en avant les produits les plus populaires, les nouveautés, les articles les mieux notés, les promotions ...

#### **2.4.6 Recommandation Contextuelle**

Le principe de la recommandation contextuelle est de proposer des items proches de l'item consulté. Les techniques de rapprochement des items peuvent être simples. On peut par exemple sélectionner des items du même univers, du même auteur, du même réalisateur, du même compositeur, de même couleur, etc. Elles peuvent également être plus complexes, comme avec les méthodes basées sur les usages. Youtube ou Amazon sont des exemples parfaits de cette technique. Lorsqu'un item est consulté par un internaute, par exemple un



disque de Georges Brassens, le système recommande une liste d'items qui ont été appréciés par les utilisateurs ayant également apprécié ce disque de Brassens.

## 2.5 Avantages et inconvénients des systèmes de recommandation

- **Adaptabilité** : Au fur et à mesure que la base de données des évaluations augmente, la recommandation devient plus précise.
- **Nouvel utilisateur** : un nouvel utilisateur qui n'a pas encore accumulé suffisamment d'évaluations ne peut pas avoir de recommandations pertinentes.
- **Nouvel item** : un item doit avoir suffisamment d'évaluations pour qu'il soit pris en considération dans le processus de recommandation.
- **Démarrage à froid** : le démarrage à froid est un problème pour les nouveaux utilisateurs qui commencent à jouer avec le système, parce que le système ne dispose pas d'assez d'informations à leur sujet. Si le profil d'utilisateur est vide, il doit consacrer une somme d'efforts à l'aide du système avant d'obtenir une récompense (les recommandations utiles). D'autre part, quand un nouvel item est ajouté à la collection, le système doit avoir suffisamment d'informations pour être en mesure de recommander cet item aux utilisateurs.
- **Le cas du système débutant** : provient lors du lancement d'un nouveau service de recommandation. Le système ne possède alors aucune information sur les utilisateurs et sur les items. Les méthodes de filtrage collaboratif ne peuvent pas fonctionner sur une matrice d'usages vide. La solution consiste en général à trouver des informations descriptives des items afin d'organiser le catalogue et inciter les utilisateurs à le parcourir jusqu'à ce que la matrice d'usages soit assez remplie et permette de passer en mode collaboratif. Le tableau 2.1 résume les forces et faiblesses des méthodes traditionnelles utilisées par les systèmes de recommandation, en l'occurrence le Filtrage Collaboratif (FC), le Filtrage à Base de Contenu (FBC) [9].

Techniques	Avantages	Inconvénients
Filtrage à base du contenu	<ul style="list-style-type: none"> <li>- Pas besoin d'une large communauté d'utilisateurs pour pouvoir effectuer des recommandations.</li> <li>- Une liste de recommandations peut être générée même s'il n'y a qu'un seul utilisateur.</li> <li>- La qualité croit avec le temps.</li> <li>- Pas besoin d'information sur les autres utilisateurs.</li> <li>- Prendre en considération les goûts uniques des utilisateurs.</li> </ul>	<ul style="list-style-type: none"> <li>- L'analyse du contenu est nécessaire pour faire une recommandation.</li> <li>- Problème de recommandation des images et de vidéos en absence de Métadonnées.</li> <li>- Nécessité du profil d'utilisateur.</li> </ul>
Filtrage collaboratif	<ul style="list-style-type: none"> <li>- Ne demande aucune connaissance sur le contenu de l'item ni sa sémantique.</li> <li>- La qualité de la recommandation peut être évaluée.</li> <li>- Plus les nombre d'utilisateurs est grand plus la recommandation est meilleure</li> </ul>	<ul style="list-style-type: none"> <li>- Démarrage à froid.</li> <li>- Nouvel Item.</li> <li>- Nouvel utilisateur.</li> <li>- Problème de confidentialité.</li> <li>- La complexité : dans les systèmes avec un grand,nombre d'items et d'utilisateurs le calcul croit linéairement.</li> </ul>

TABLE 2.1 – Les avantages et les inconvénients des techniques de recommandation

## 2.6 Conclusion

Dans ce chapitre, nous avons vu la classification générale qui contient les deux approches principales filtrage basé sur le contenu et filtrage collaboratif donnent des résultats très intéressants. Le filtrage collaboratif se propose de recommander aux utilisateurs certains articles qu'ils n'ont pas encore lus en se basant sur les opinions du groupe d'utilisateurs similaires. En revanche, le filtrage basé sur le contenu évalue si un texte est pertinent ou non pour un utilisateur, en fonction de son profil thématique. Néanmoins, ces deux aspects du profil semblent importants. Ainsi, une approche hybride combinant les goûts thématiques personnels et les opinions des utilisateurs du groupe paraissent la meilleure méthode pour fournir des recommandations. Nous avons décrit les principaux avantages et inconvénients de chacune de ces méthodes. Il est à noter que ces techniques de filtrage ne prennent pas en compte un bon nombre d'informations utiles pour expliquer le jugement qu'effectue l'utilisateur. Ceci requiert la prise en compte de différentes sources d'informations sur l'utilisateur (ses préférences précédemment déclarées, son réseau social,...) et leurs corrélations avec le contenu des documents qui lui sont proposés. Ainsi, nous avons assisté, ces dix dernières années, à la naissance de plusieurs nouvelles familles de systèmes de filtrage d'information, comme les systèmes de filtrage d'information basés sur l'utilité, les systèmes de filtrage d'information à base d'ontologies, les systèmes d'annotations collaboratives, les systèmes de filtrage d'informations à base de réseaux sociaux, etc.

---

## PRÉSENTATION DES SYSTÈMES DE RECOMMANDATION EXISTANTS

---

*Un nombre assez important de systèmes de recommandation sont proposés dans la littérature. Nous citons entre autres :*

### 3.1 Ezako-Un surplus d'intelligence

L'outil Ezako collecte des informations et permet de former des groupes de clients aux goûts similaires mais aussi de découvrir les goûts individuels des clients. Le tableau de bord est conçu pour piloter le plus simplement possible les actions.

Les recommandations basées sur l'historique de ventes et les clients similaires peuvent être orientées par des règles métiers (fin de stock..) et une stratégie de vente (offres spéciales, top ventes,...).

La version custom plus puissante propose une personnalisation du moteur de recherche, une recherche instantanée et une autocorrection des requêtes [29].

## 3.2 Amazon

Amazon, star des entreprises d'e-commerce est un modèle de réussite pour beaucoup. Son succès grandissant depuis sa création en 1994 est impressionnant, notamment avec son taux de transformation atteignant plus de 6% , soit le double du score moyen des sites e-commerce français et le triple de Pixmania (Source : Internet retailer). Un de ces points forts les plus remarqué Incontestablement son système de recommandation. 30% du CA total d'Amazon est généré grâce à lui [32]. Il est frappant dès les premières visites. A peine

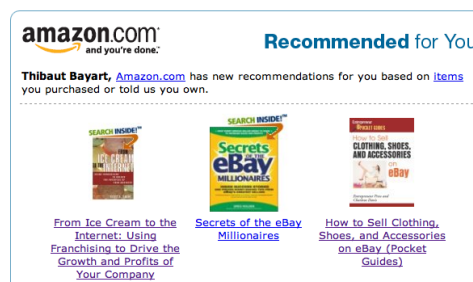


FIGURE 3.1 – Amazon recommandation.

le premier produit consulté sur le site que des suggestions très précises nous sont proposés. Souvent bluffés, les utilisateurs naviguent sur le site au gré des recommandations basées sur les achats réalisés précédemment par des acheteurs intéressés par des produits similaires. Nos meilleurs conseils ne viennent donc plus de notre libraire mais d'un algorithme de recommandation qui n'a plus à faire ses preuves.

**Mais comment Amazon a-t-il développé un outil aussi sophistiqué et performant ?**

Les consommateurs aiment être conseillés et avisés par des personnes partageant leurs goûts. C'est sur cette théorie qu'Amazon a développé son système de recommandation automatique. Misant ainsi sur du contenu (2 produits peuvent être plus en moins liés en fonction du nombre de mots similaires dans leur description par exemple.) et sur le filtrage collaboratif (la construction d'une matrice de relation entre les différents achats des consommateurs), Amazon a développé son mystérieux algorithme item-to-item collaborative filtering. La priorité est clairement la personnalisation des produits recommandés en fonction des intérêts de chaque consommateur. La tendance serait de recommander les

produits similaires entre des clients ayant des similitudes (age, ville, sexe, CSP,...) mais Amazon a préféré faire le choix de recommander des produits en fonction d'une liste établie avec les consultations de produits, achats et classements réalisés par les consommateurs eux-mêmes.

### 3.3 Pigdata-La recommandation produit orienté métier et visuel

L'outil de recommandation produit Pigdata fonctionne en mode Saas avec un algorithme de collaborative filtering avec plus de 100 paramètres analysés. Pigdata est custo-

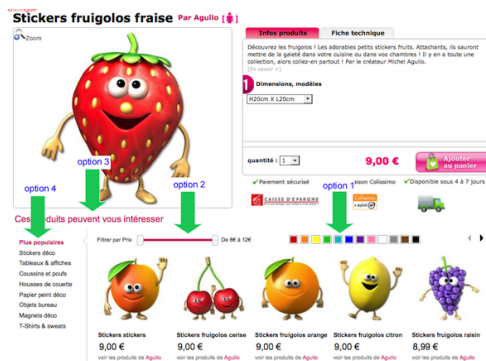


FIGURE 3.2 – Pigdata-La recommandation produit orienté métier et visuel.

mizable avec 120 paramètres analysés comme le temps passé sur la page, les clics effectués ou, bien entendu, les achats réalisés [29].

### 3.4 Google

Google personnalise nos résultats de recherche, quand cela est possible, en se basant sur notre localisation et/ou nos dernières recherches. Lorsqu'on est connecté à notre compte Google, il propose un contenu encore plus pertinent en fonction de notre historique de recherche. L'algorithme du PageRank est de manière intrinsèque un outil basé sur de la recommandation sociale dans la mesure où il utilise les liens entre les pages web. L'utilisation de contenus provenant de nos cercles de Google+ est aussi une forme de recommandation sociale [17].

### 3.5 Frosmo-Un boîte à outils au service de l'amélioration de la conversion

Cette solution en mode SaaS venue de Finlande est très riche et permet une approche globale pour l'amélioration de la conversion. Frosmo propose ainsi en plus du moteur de recommandation produit, un outil d'A/B Testing et du Retargeting. L'outil est notamment



FIGURE 3.3 – Frosmo moteur de recommandation.

intégré à Google Adwords. Sa prise en main est donc plus impliquante et une prestation de consulting technique et/ou sur la data est parfois nécessaire en plus de l'abonnement mensuel conçu par paliers. L'outil coûte 500 €/mois pour moins de 100 000 visiteurs/mois et 1500 €/mois pour moins d'1 million de visiteurs/mois. Pour les petits sites, un autre éditeur Finlandais propose Nosto qui équipe 1600 sites dans 64 pays et qui fonctionne avec un algorithme de collaborative filtering (analyse comportementale) [29].

### 3.6 Target2Sell-Pour les petits comme les gros marchands

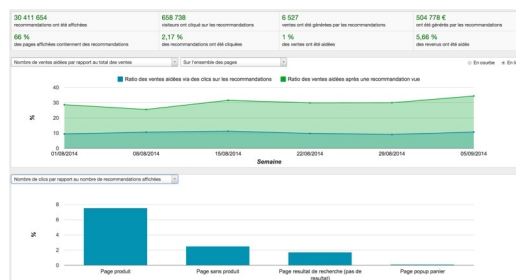


FIGURE 3.4 – Target2Sell.

Solution SaaS spécialisée dans la recommandation, Target2Sell vise aussi bien les gros marchands, tels que Auchan, Monoprix, Raja, que les petits commerçants, notamment via un récent accord signé avec Oxatis. Les concepteurs de la solution ont mis l'accent sur la simplicité tant dans l'interface du backoffice de l'outil que dans son mode de fonctionnement. Ainsi, le service privilégie les algorithmes auto-apprenants. Il faudra attendre une version future pour développer ses propres algorithmes prédictifs sur la plateforme. Ceux-ci intègrent un fonctionnement multicanal, avec une recommandation optimisée en fonction du canal, qu'il soit le site web, le site mobile, ou le canal e-mail - avec une intégration menée avec Mailchimp et Smart Focus [28].

### 3.7 Nuukik-Plusieurs approches d'analyse prédictive

Le lancement de Nuukik fait suite à 3 ans de R & D menée avec l'INRIA pour développer les algorithmes de recommandation de ce service. Celui-ci panache le Collaborative filtering, c'est-à-dire une recommandation qui s'appuie sur les ventes déjà réalisées ainsi que la navigation des internautes, avec une recommandation basée sur les contenus, typiquement les attributs de la fiche produit. Ce mélange de deux approches doit permettre, selon les ingénieurs de la start-up de rester pertinent même lorsqu'il y a peu d'historique sur un produit, au moment des changements de collection dans le textile. De base, Nuukik propose sa technologie sous forme de service Cloud, avec une facturation à la requête. Néanmoins, l'éditeur n'exclut pas la possibilité de déployer son logiciel en mode on-premise pour les sites qui en exprimeront le besoin [28].



FIGURE 3.5 – Moteur \_ nuukik.



CHAPITRE 3. PRÉSENTATION DES SYSTÈMES DE RECOMMANDATION  
EXISTANTS

---

Editeur	Amazon	Google	Target2Sell	Frosmo solution Ltd	Nuukik	Ezako	Pigdata
Plateformes	Linux, Windows, Unix, MacOS	Mac, Linux, Windows	Windows, linux	Sans objet	Sans objet	Sans objet	Sans objet
BD	NoSQL, Dynamo.db	Bigtable (A Distributed Storage System for Structured Data)	Via API	Via API	Oracl, postgreSQL, etc	Via API	Sans objet
Algorithmes implémentés	algorithme item-to-item collaborative filtering	content-based filtering	Algorithme auto apprenant guide par règles	algorithme de collaborative filtering (analyse comportementale)	Mix de collaborative filtering et content based	Mix de collaborative filtering et content based	collaborative filtering
facteurs de recommandation	l'historique des achats et des consultations, le temps passé sur une page, comment l'utilisateur est arrivé sur Amazon, le parcours de la découverte à l'achat, les habitudes et les goûts en fonction de son pays, son climat,...	la localisation de l'utilisateur et/ou les dernières recherches d'même (historique de recherche)	en fonction du visiteur (presque comme Amazon)	en fonction des préférences et des comportements des utilisateurs.	(notes, commentaires), les comportements de l'utilisateur en "arrière-plan" ainsi que le contenu qui consiste à définir des caractéristiques (objectives) aux objets	l'historique de ventes et les clients similaires peuvent être orientés par des règles métiers (fin de stock..) et une stratégie de vente (offres spéciales, top ventes,...).	le temps passé sur la page, les clics effectués ou, bien entendu, les achats réalisés.
Type de algorithme	Machine learning	Algorithme preectif et machine learning	algorithme prédictif	algorithme prédictif	Machine learning	Machine learning	algorithme prédictif

TABLE 3.1 – Comparatif des moteurs de recommandation d'e-commerce

## 3.8 Conclusion

Dans ce chapitre, nous avons comparé certains systèmes de recommandation du domaine du e-commerce. Notre moteur de recommandation que nous allons construire est assez proche de celui d'Amazon. En effet, ce moteur utilise l'algorithme de filtrage collaborative item-to-item de type machine learning pour trouver les utilisateurs similaires à partir de quelques facteurs tels que (les liens disponibles, l'historique, le profil, ...) et ensuite il recommande les items (dans notre cas les liens) à travers cette similarité.

---

# DÉVELOPPEMENT DU SYSTÈME DE RECOMMANDATION

---

## 4.1 Introduction

Nous allons essayer d'implémenter un système de recommandation sociale. Tout d'abord, nous allons travailler sur un dictionnaire local et dans un deuxième temps, nous généraliserons notre travail sur la base de données DELICIOUS.

## 4.2 Environnement de travail

### 4.2.1 Matériel

**Marque :** Nom HP 630.

**Mémoire (RAM) :** 2 Go.

**Processeur :** intel Core(TM) 2Duo (2.27GHz).

**Système d'exploitation :** Windows 7 Professional SP1 (32bits).

## 4.2.2 Python

Python est un langage de script de haut niveau, portable, dynamique, extensible, gratuit, structuré et open source conçu pour être orienté objet. Il est multi-paradigme et multi-usage. Développé à l'origine par Guido Van Rossum en 1993, il est, comme la plupart des applications et outils open source, maintenu par une équipe de développeurs un peu partout dans le monde.

### Caractéristiques du langage :

- Python est portable, non seulement sur les différentes variantes d'UNIX, mais aussi sur les OS propriétaires : MacOS, BeOS, NeXTStep, MS-DOS et les différentes variantes de Window.
- Python est gratuit, mais on peut l'utiliser sans restriction dans des projets commerciaux.
- La syntaxe de Python est très simple et combinée des types des données évolués (listes, dictionnaires,...), conduit à des programmes à la fois très compacts et très lisibles.
- Python gère ses ressources (mémoire, descripteurs de fichiers...) sans intervention du programmeur, par un mécanisme de comptage de références (proche, mais différent, d'un garbage collector).
- Python est orienté-objet. Supporte l'héritage multiple et la surcharge des opérateurs.
- Python est dynamique (l'interpréteur peut évaluer des chaînes de caractères représentant des expressions ou des instructions Python), orthogonal (un petit nombre de concepts suffit à engendrer des constructions très riches), réflexif (il supporte la méta programmation, par exemple la capacité pour un objet de se rajouter ou de s'enlever des attributs ou des méthodes, ou même de changer de classe en cours d'exécution) et introspectif (un grand nombre d'outils de développement, comme le debugger ou le profiler, sont implantés en Python lui-même).

- La bibliothèque standard de Python, donnent accès à une grande variété de services : chaînes de caractères et expressions régulières, protocoles Internet (Web, FTP, HTML...), persistance et bases de données, interfaces graphiques.
- Python est un langage qui continue à évoluer et un langage de choix pour traiter le XML [22].

### 4.2.3 Delicious

C'est un service web de bookmarking social pour le stockage, le partage et la découverte de signets Web. Le site a été fondé par Joshua Schachter en 2003 et acquis par Yahoo en 2005. A la fin de 2008, le service a fait plus de 5,3 millions d'utilisateurs et 180 millions d'URL uniques signet. Yahoo a vendu Delicious aux systèmes AVOS en Avril 2011. En mai 2014, AVOS a vendu le site à science INC [33].

## 4.3 Partie développement

### 4.3.1 La conception

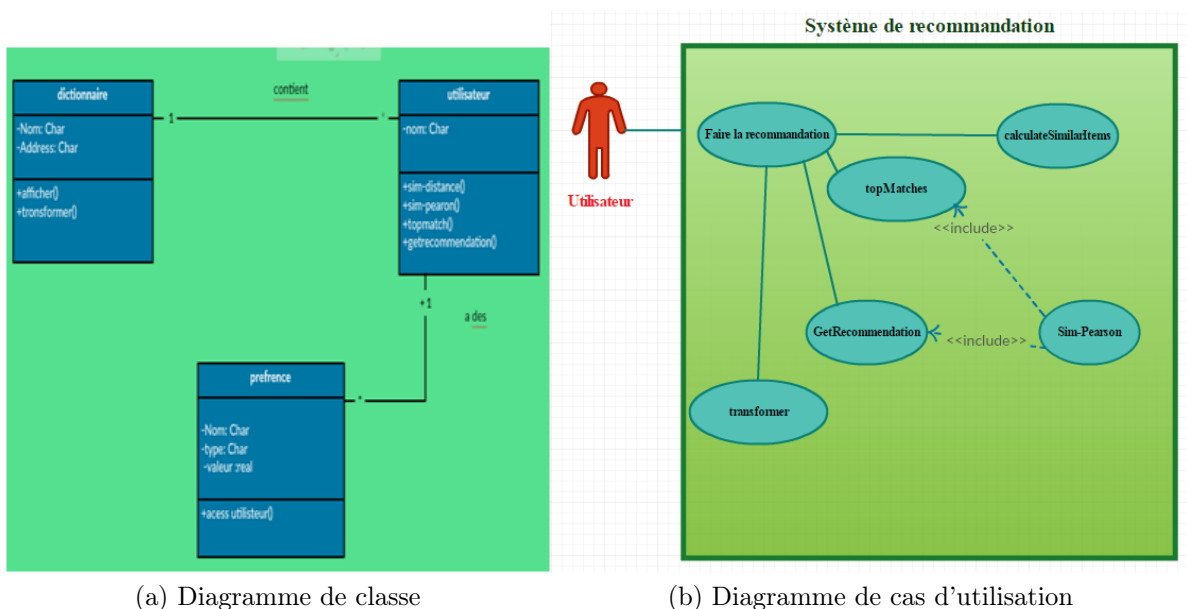


FIGURE 4.1 – Diagrammes de conception.

Pour la partie de conception on a utilisé le langage UML. Dans notre application on essayer de construire un diagramme de cas d'utilisation (b) qui est un diagramme dynamique, chaque cas d'utilisation est associée à une fonction d'algorithmes. Les sections suivantes expliquent chacune de ces fonctions et l'utilisateur qui peut utilise ces recommandations et un diagramme de classe (a) qui est un diagramme statique.

### 4.3.2 Possibilités graphiques sous Python

Trois bibliothèques graphiques sous Python sont largement utilisé : wxPython, tkinter, pyGTK.

**Boa Constructor** : est un éditeur pour Python et un constructeur de GUI sur le toolkit wxPython. Il permet la création et la manipulation visuelle de fenêtres graphiques. wxPython est une collection de modules Python réalisée sur la base des wxWidgets de wxWindows, un framework multi-plateformes écrit en C++ [34].

**wxPython** : met à disposition des développeurs un nombre impressionnant de classes permettant de réaliser des interfaces homme-machine (IHM) complètement indépendante de l'OS sur lequel ils sont mis en oeuvre. Mais au-delà, d'une simple IHM, wxPython fournit également des classes de haut niveau aussi diverses que les sockets réseau, le traitement de l'image ou encore l'interprétation HTML ou XML [35].

**Le module Tkinter** : module de Python permet de créer des interfaces graphiques (GUI : graphical user interface). De nombreux composants graphiques (ou widgets) sont disponibles parmi : fenêtre (classe Tk), bouton (classe Button), case à cocher (classe Checkbutton), étiquette (classe Label), zone de texte simple (classe Entry). On peut gérer de nombreux événements : clic sur la souris, déplacement de la souris, appui sur une touche du clavier, top d'horloge... [37]

**PyGTK** : module Python permettant la création d'interfaces graphiques utilisant la bibliothèque GTK2. À partir de GTK3, PyGObject remplace PyGTK dans ce rôle.

### Pourquoi wxpython est la meilleure bibliothèque graphique ?

Comme nous pouvons le voir dans l'image ci-dessous que le taux d'utilisation de wxPython est la plus élevée après PyQt. Puisque wxPython ils plus facile à travailler que Tkinter et PyGTK et aussi gratuit. Vous pouvez obtenir des choses vers le haut et en cours d'exécution à été plus facilement et avec moins de code. WxPython est multiplateforme, et le même logiciel peut être exécuté sur Windows, UNIX et apparenté, et Mac OS X sans modification. wxPython est un outil levant avec une communauté active de



FIGURE 4.2 – Les taux d'utilisation des bibliothèques graphiques.

développement, tandis que Tcl / tk et Tkinter sont vieux et usés mais wxpython est l'outil de l'avenir. Enfin et pour toutes ces raisons, nous avons adopté l'environnement Boa qui utilise la bibliothèque wxPython pour le développement de l'interface graphique de notre application.

### 4.3.3 Recueillir les préférences (Dictionnaire Local)

La première étape que nous avons besoin est une façon de représenter des différentes personnes et leurs préférences. Pour cela, nous allons tout d'abord, créer un fichier appelé `recommendations.py`. Ce dictionnaire utilise un classement de 1 à 5 comme un moyen d'exprimer combien chacun des ces critiques de navigation ont aimé un lien donné. Peu importe la façon dont les préférences sont exprimées, on doit trouver un moyen de les mapper sur des valeurs numériques. Si c'était le cas d'un site commercial, on pourrait utiliser une valeur de 1 pour indiquer que quelqu'un avait acheté un article dans le passé et une valeur de 0 pour indiquer qu'ils n'avaient pas. Pour un site où les gens votent sur



FIGURE 4.3 – Dictionnaire local.

des reportages, des valeurs de -1, 0, 1 pourraient être utilisés pour représenter " détesté ", " ne pas voter " et " aimé ". Utiliser un dictionnaire est pratique pour expérimenter avec les algorithmes et il est facile de chercher et modifier le dictionnaire mais pour un très grand ensemble de données, on aura probablement à stocker les préférences dans une base de données.

Test d'accès au dictionnaire :

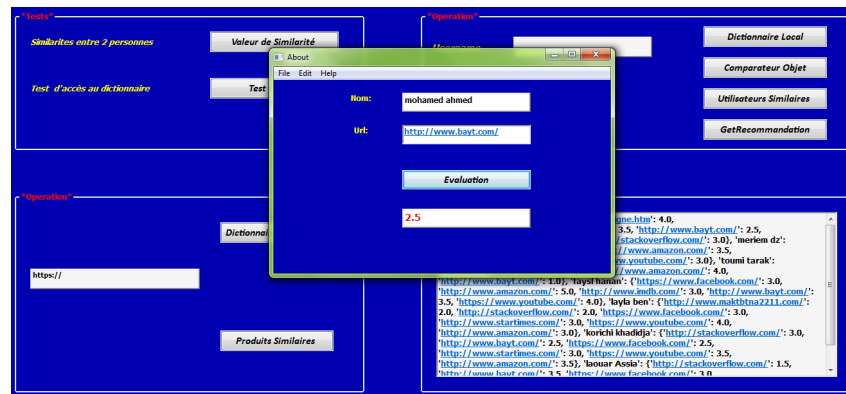


FIGURE 4.4 – Test d'accès au dictionnaire.

### 4.3.4 Trouver les utilisateurs similaires

Après la collection des données sur les articles que les gens aiment, nous devons trouver un moyen de déterminer comment les gens sont similaires dans leurs goûts. Pour ce faire,



en comparant chaque personne à chaque autre et de calculer un score de similarité. Il y a plusieurs façons de procéder, et dans notre réalisation, nous pouvons travailler avec deux systèmes pour le calcul des scores de similarité : la distance euclidienne et la corrélation de Pearson.

**Distance euclidienne :** Une façon très simple de calculer un score de similarité est d'utiliser un score de distance euclidienne.

$$E(x, y) = \sqrt{\sum_{i=0}^n (x_i - y_i)^2}$$

Il prend les liens pour lesquels les gens ont un classement en commun et les utilise comme axes d'un graphique. Nous pouvons ensuite tracer les gens sur le tableau et voir à quel point ils se ressemblent. Cette figure montre les gens cartographiés dans l'espace de préférence

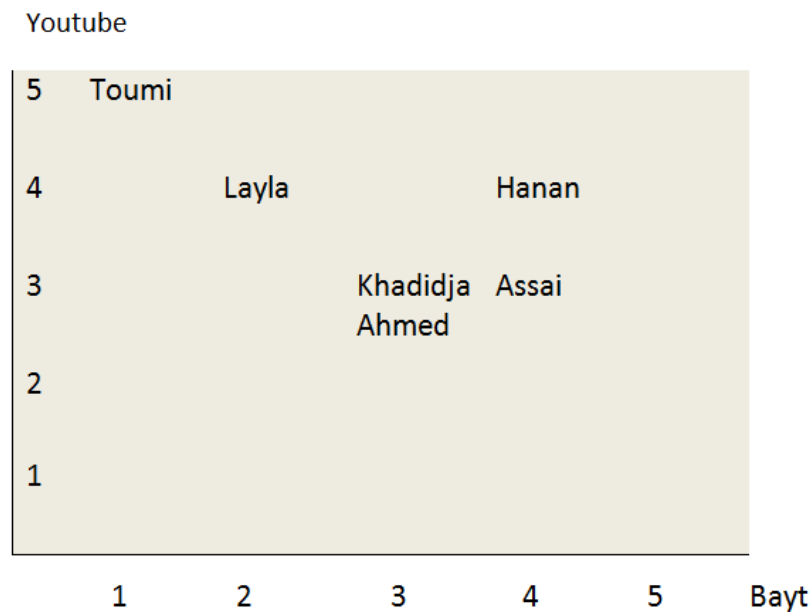


TABLE 4.1 – Espace de préférences.

par exemple Toumi a été tracé à 4.5 sur l'axe Youtube et à 1,0 sur l'axe bayt. Si deux personnes sont plus étroites dans l'espace de préférence, on peut conclure qu'ils sont plus similaires dans leurs préférences. Parce que la carte est en deux dimensions, nous ne pouvons pas regarder deux classements à la fois, mais le principe est le même pour les grandes séries de classements. Pour calculer la distance entre toumi et layla dans le tableau,

il suffit de prendre la différence dans chaque axe, les ajouter ensemble, puis prendre la racine carrée de la somme. Python, on peut utiliser le  $\text{pow}(n, 2)$  la fonction à ajuster un nombre et prendre la racine carrée de la fonction  $\text{sqrt}$  :  $\text{sqrt}(\text{pow}(5-4,2)+\text{pow}(4-1,2))$  Cette formule calcule la distance, qui sera plus faible pour les personnes qui sont plus similaires. Cependant, nous aurons besoin d'une fonction qui donne des valeurs plus élevées pour les personnes qui sont similaires. Cela peut être fait en ajoutant 1 à la fonction (si on ne reçoit pas une erreur division par zéro) et l'inverser :  $1/(1+\text{sqrt}(\text{pow}(5-4,2)+\text{pow}(4-1,2)))$

Cette nouvelle fonction renvoie toujours une valeur comprise entre 0 et 1, où une valeur de 1 signifie que deux personnes ont des préférences identiques. On peut tout mettre ensemble pour créer une fonction de calcul de similarité.

Nous allons ajouter ce code à recommendations.py.

```

from math import sqrt

def sim_distance(prefs, person1, person2):
    si={}
    for item in prefs[person1]:
        if item in prefs[person2]:si[item]=1
    if len(si)==0: return 0
    sum_of_squares=sum([pow(prefs[person1][item]-prefs[person2][item], 2)
        for item in prefs[person1] if item in prefs[person2]])
    return 1/(1+sum_of_squares)
    
```

FIGURE 4.5 – Fonction de Sim Distance.

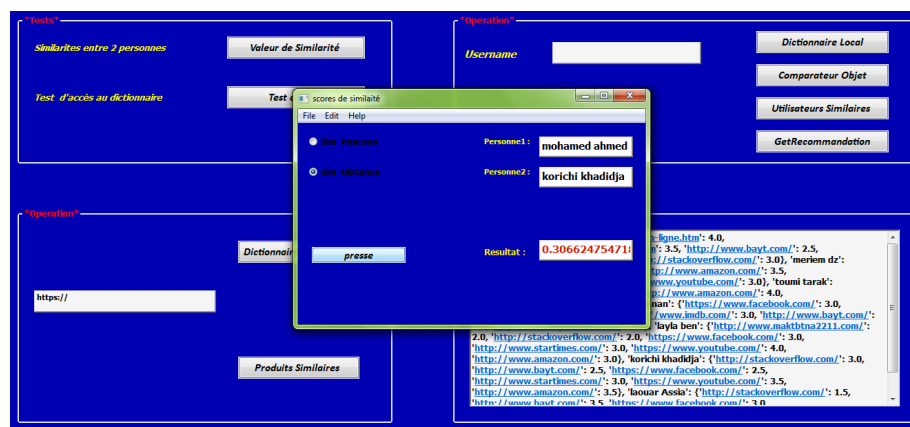


FIGURE 4.6 – L'exécution de Sim Distance.

**Corrélation de Pearson Score :** Une manière un peu plus sophistiqué pour déterminer la similitude entre les intérêts des personnes est d'utiliser un coefficient de corrélation de Pearson. Le coefficient de corrélation est une mesure de la façon dont deux ensembles de données tiennent sur une ligne droite. La formule est plus compliquée que le score de la distance euclidienne, mais elle a tendance à donner de meilleurs résultats dans le cas où les données ne sont pas bien normalisées.

$$r = \frac{\sum xy - \frac{\sum x \sum y}{N}}{\sqrt{(\sum x^2 - \frac{(\sum x)^2}{N})(\sum y^2 - \frac{(\sum y)^2}{N})}}$$

Pour mettre en évidence cette méthode, nous pouvons tracer les notes de deux des critiques sur un tableau. Nous pouvons également voir une ligne droite sur le graphique. C'est ce

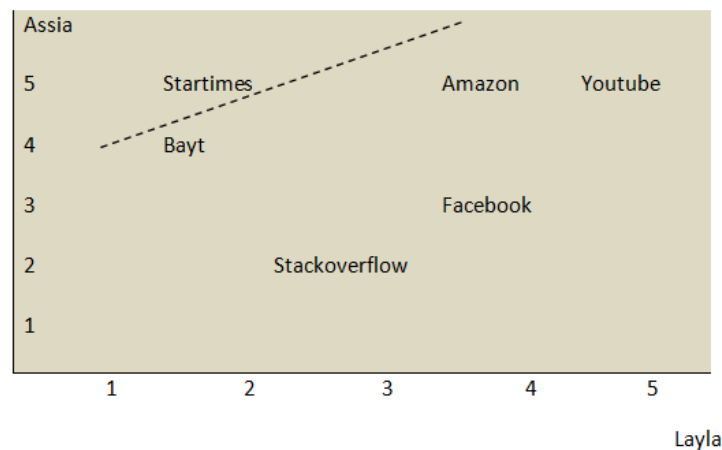


TABLE 4.2 – Sim-Pearson.

qu'on appelle la ligne de meilleur ajustement, car il vient au plus près possible de tous les éléments sur le tableau. Si les deux critiques avaient des notes identiques pour chaque lien, cette ligne serait diagonale et toucherait tous les articles dans le tableau, donnant un score de corrélation parfaite de 1. Dans le cas illustré, les critiques sont en désaccord sur quelques liens, de sorte que le score de corrélation est d'environ 0,4. Le code pour le score de corrélation de Pearson trouve d'abord les articles les mieux notés par les critiques. Il calcule ensuite les sommes et la somme des carrés des notes pour les deux critiques, et calcule la somme des produits de leurs évaluations. Enfin, on utilise ces résultats pour calculer le coefficient de corrélation de Pearson. Contrairement à la mesure de la distance,

```

def sim_pearson(prefs,p1,p2):
    si={}
    for item in prefs[p1]:
        if item in prefs[p2]:si[item]=1
    if len(si)==0: return 0
    n=len(si)
    sum1=sum([prefs[p1][it] for it in si])
    sum2=sum([prefs[p2][it] for it in si])
    sum1sq=sum([pow(prefs[p1][it],2) for it in si])
    sum2sq=sum([pow(prefs[p2][it],2) for it in si])
    psum=sum([prefs[p1][it]*prefs[p2][it] for it in si])
    num=psum-(sum1*sum2/n)
    den=sqrt((sum1sq-pow(sum1,2)/n)*(sum2sq-pow(sum2,2)/n))
    if den==0: return 0
    r=num/den
    return r
    
```

FIGURE 4.7 – Fonction Sim Pearson.

cette formule n'est pas très intuitive.

Cette fonction retourne une valeur comprise entre -1 et 1. Une valeur de 1 signifie que les gens ont exactement les mêmes notes pour chaque article. Contrairement à la mesure de la distance, nous n'avons pas besoin de modifier cette valeur pour obtenir la bonne échelle.

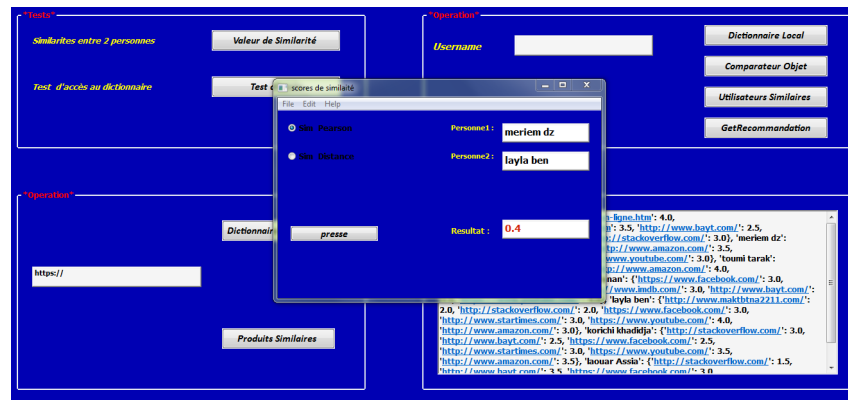


FIGURE 4.8 – L'exécution de Sim Pearson.

### 4.3.5 Classement des critiques

Maintenant que nous avons les fonctions pour comparer deux personnes, nous pouvons créer une fonction pour comparer les scores de tout le monde contre une personne donnée et trouver les plus proches. On va ajouter cette fonction à recommendations.py pour

obtenir une liste ordonnée des personnes ayant les mêmes goûts pour la personne en question :

```

74 recommendations.py - C:\Users\User\Desktop\recommendations.py
File Edit Format Run Options Windows Help
def topMatches(prefs, person, n=5, similarity=sim_pearson):
    scores=[(similarity(prefs, person, other), other)
             for other in prefs if other!=person]
    scores.sort()
    scores.reverse()
    return scores[0:n]
    
```

FIGURE 4.9 – Fonction de topMatches.

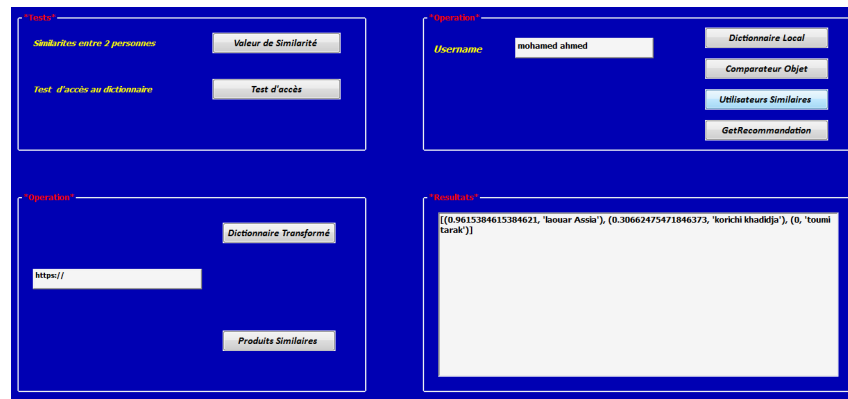


FIGURE 4.10 – L'exécution de topMatches.

### 4.3.6 La recommandation d'articles

Nous avons besoin de marquer les articles en produisant un score pondéré qui classe les critiques. Prenons les voix de tous les autres critiques et multiplions leur ressemblance par le score qu'ils ont donné à chaque liens. Nous devons diviser par la somme de toutes les similitudes pour les critiques qui ont examiné ce lien. La dernière ligne indique les résultats de cette division. Le code est ci-dessous. Maintenant, nous savons comment trouver des gens semblables et recommander des produits pour une personne donnée. Mais, comment pouvons-nous voir quels sont les produits qui sont semblables les uns aux autres. Cette situation est fréquente sur les sites Web commerciaux, en particulier lorsque le site n'a pas recueilli beaucoup d'informations sur les utilisateurs. nous pouvons déterminer la

```

74 recommendations.py - C:\Users\User\Desktop\recommendations.py
File Edit Format Run Options Windows Help
def getRecommendations(prefs, person, similarity=sim_pearson):
    totals={}
    simSums={}
    for other in prefs:
        # don't compare me to myself
        if other==person: continue
        sim=similarity(prefs, person, other)

        # ignore scores of zero or lower
        if sim<=0: continue
        for item in prefs[other]:

            # only score movies I haven't seen yet
            if item not in prefs[person] or prefs[person][item]==0:
                # Similarity * Score
                totals.setdefault(item,0)
                totals[item]+=prefs[other][item]*sim
            # Sum of similarities
            simSums.setdefault(item,0)
            simSums[item]+=sim

    # Create the normalized list
    rankings=[(total/simSums[item], item) for item, total in totals.items()]

    # Return the sorted list
    rankings.sort()
    rankings.reverse()
    return rankings
    
```

FIGURE 4.11 – Fonction de GetRecommendations.

similitude en regardant qui ont aimé un article en particulier et de voir les autres choses qu'ils aimaient.

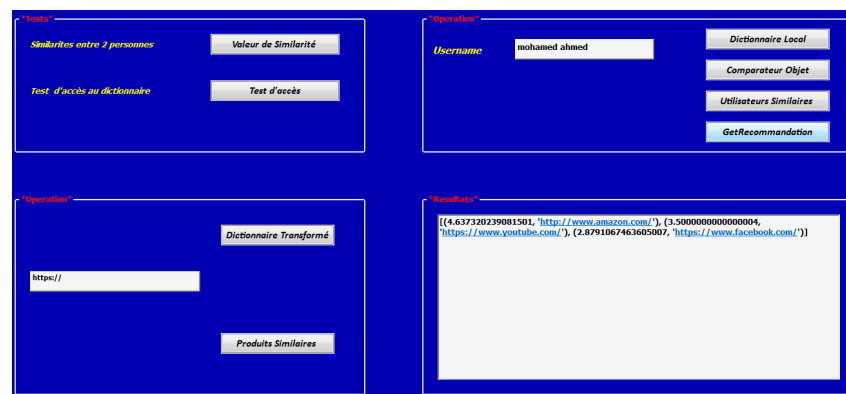


FIGURE 4.12 – L'exécution de GetRecommendations.

**Le filtrage basé sur l'article :** La technique générale consiste à pré-calculer les éléments les plus similaires pour chaque article.

**(dataset) de Comparateur d'Objet :**

Pour comparer des objets, la première chose qu'on doit faire est d'écrire une fonction pour construire l'ensemble des données des articles similaires. Cela ne doit pas être fait à chaque

fois qu'une recommandation est nécessaire. Nous construisons l'ensemble de données une fois et le réutiliser chaque fois que nous en avons besoin.

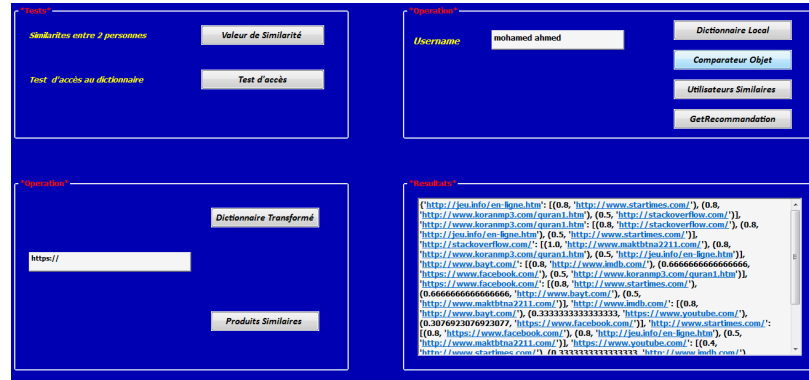


FIGURE 4.13 – L'exécution Comparateur d'Objet.

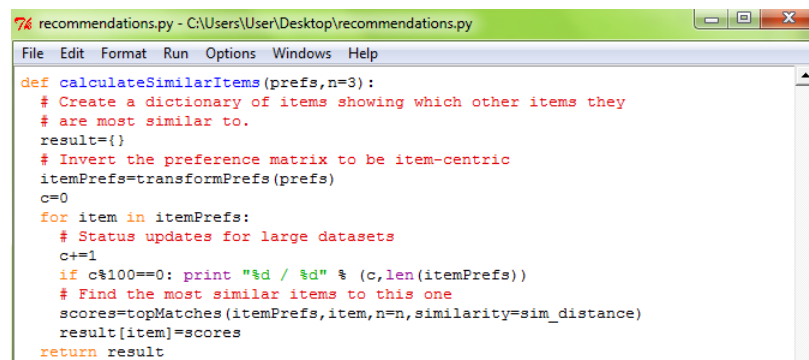


FIGURE 4.14 – Fonction de Comparateur d'Objet.

### dictionnaire transforme :

Dans ce cas, nous pouvons déterminer la similitude en regardant qui ont aimé un article en particulier et de voir les autres choses qu'ils aimait. Nous allons ajouter le code suivant pour réaliser la transformation :

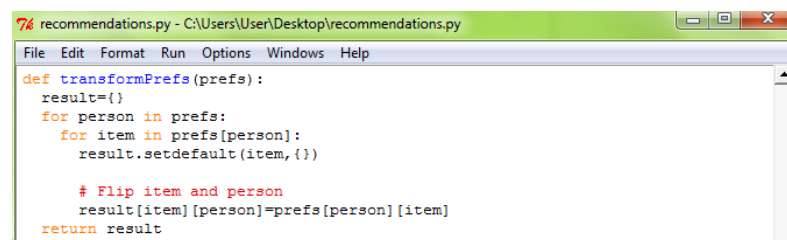


FIGURE 4.15 – Fonction transformPrefs.



FIGURE 4.16 – Dictionnaire transformé.

### 4.3.7 Construire une recommandation pour (del.icio.us)

Cette section montre comment récupérer des données à partir de l'un des plus populaires sites en ligne de bookmarking, et comment utiliser ces données pour trouver des utilisateurs similaires et recommander des liens qu'ils n'ont pas vus auparavant. Ce site, auquel nous pouvons accéder à l'URL <http://del.icio.us>, permet aux gens de mettre en place un compte et poster des liens qui les intéressent. Nous pouvons visiter le site et regarder les liens que d'autres personnes ont postés, et aussi revoir des liens "populaires" qui ont été publiés par de nombreuses personnes différentes. Un aperçu de la page del.icio.us est illustré à la figure suivante :

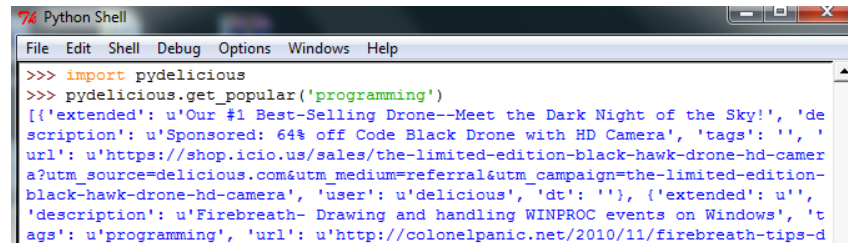


FIGURE 4.17 – Site Delicious.

Contrairement à certains sites de partage, del.icio.us ne comprend pas La façon de trouver des gens semblables ou recommander des liens qu'on aime. Heureusement, on peut utiliser les techniques présentées dans ce mémoire pour ajouter cette fonctionnalité nous-même.



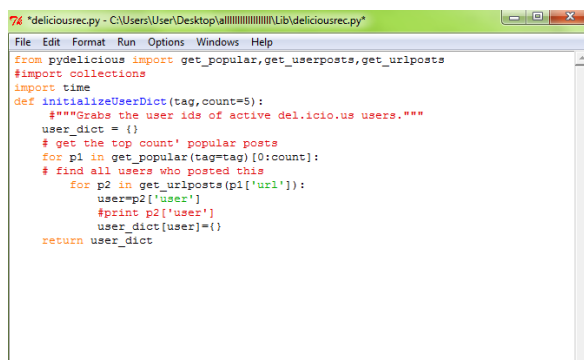
**L'API del.icio.us :** Les données de del.icio.us sont mises à disposition par le biais d'une API (pydelicious 0.6) qui renvoie des données au format XML. Une API Python est téléchargeable à l'adresse <http://code.google.com/p/pydelicious/source>. On peut



```
Python Shell
File Edit Shell Debug Options Windows Help
>>> import pydelicious
>>> pydelicious.get_popular('programming')
[{'extended': u'Our #1 Best-Selling Drone--Meet the Dark Night of the Sky!', 'description': u'Sponsored: 64% off Code Black Drone with HD Camera', 'tags': '', 'url': u'https://shop.icio.us/sales/the-limited-edition-black-hawk-drone-hd-camera?utm_source=delicious.com&utm_medium=referral&utm_campaign=the-limited-edition-black-hawk-drone-hd-camera', 'user': u'delicious', 'dt': ''}, {'extended': u'', 'description': u'Firebreath- Drawing and handling WINPROC events on Windows', 'tags': u'programming', 'url': u'http://colonepanic.net/2010/11/firebreath-tips-d
```

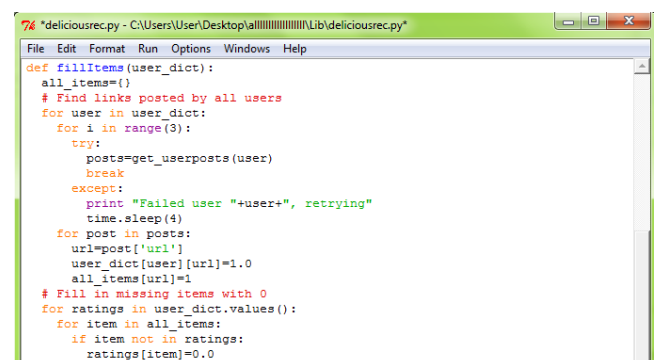
FIGURE 4.18 – Get-popular(liste de dictionnaires URL ).

voir qu'il retourne une liste de dictionnaires, chacun contenant une URL, description, et l'utilisateur qui l'a publié. Puisque nous travaillons sur des données en temps réel, les résultats seront probablement différents de ceux des exemples. Il existe deux autres appels que nous allons utiliser, `get_urlposts`, qui retourne tous les messages pour une URL donnée, et `get_userposts`, qui retourne tous les messages pour un utilisateur donné. Les données de ces appels sont renvoyées de la même manière. Cela nous donnera un dictionnaire avec certains utilisateurs, référençant tous un dictionnaire vide attendant d'être remplis avec des liens. L'API ne renvoie que les 30 dernières personnes ayant posté de lien. Contrairement à l'ensemble de données de critique de différent domaine, il n'y a que deux notes possibles dans ce cas : 0 si l'utilisateur n'a pas posté de lien, et 1 s'il le faisait.



```
*deliciousrec.py - C:\Users\User\Desktop\al\deliciousrec.py
File Edit Format Run Options Windows Help
from pydelicious import get_popular, get_userposts, get_urlposts
import collections
import time
def initializeUserDict(tag, count=5):
    """Grabs the user ids of active del.icio.us users."""
    user_dict = {}
    # get the top count' popular posts
    for p1 in get_popular(tag+tag)[0:count]:
        # find all users who posted this
        for p2 in get_urlposts(p1['url']):
            user=p2['user']
            #print p2['user']
            user_dict[user]={}
    return user_dict
```

FIGURE 4.19 – Fichier deliciousrec.py.

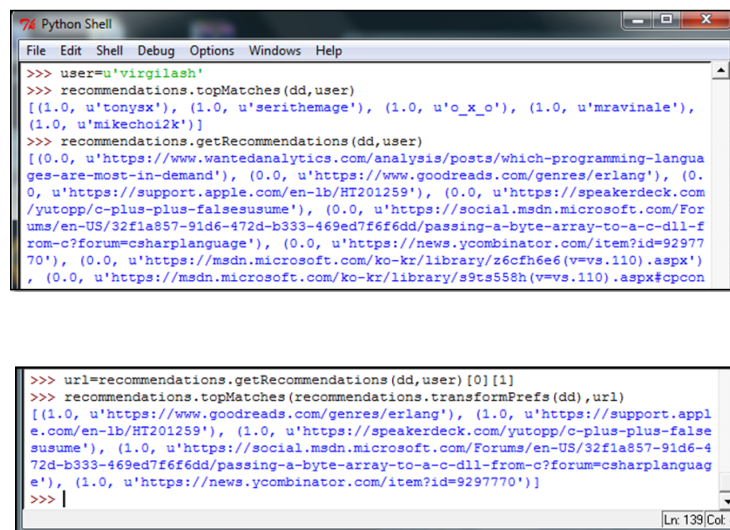


```
*deliciousrec.py - C:\Users\User\Desktop\al\deliciousrec.py
File Edit Format Run Options Windows Help
def fillItems(user_dict):
    all_items={}
    # Find links posted by all users
    for user in user_dict:
        for i in range(3):
            try:
                posts=get_userposts(user)
                break
            except:
                print "Failed user "+user+", retrying"
                time.sleep(4)
        for post in posts:
            url=post['url']
            user_dict[user][url]=1.0
            all_items[url]=1
    # Fill in missing items with 0
    for ratings in user_dict.values():
        for item in all_items:
            if item not in ratings:
                ratings[item]=0.0
```

FIGURE 4.20 – Fichier deliciousrec.py modifié.

### Recommandation des voisins et des liens :

Maintenant qu'on a construit un ensemble de données, on peut appliquer les mêmes fonctions qu'on a utilisées avant sur le jeu de données de critique des liens de navigation. Pour sélectionner un utilisateur au hasard et trouver d'autres utilisateurs qui ont les mêmes goûts que lui, nous utiliserons le code suivant dans la session Python :



```

Python Shell
File Edit Shell Debug Options Windows Help
>>> user='virgilash'
>>> recommendations.topMatches(dd,user)
[(1.0, u'tonyx'), (1.0, u'serithemage'), (1.0, u'o_x_o'), (1.0, u'mravinale'),
(1.0, u'mikechoi2k')]
>>> recommendations.getRecommendations(dd,user)
[(0.0, u'https://www.wantedanalytics.com/analysis/posts/which-programming-langua
ges-are-most-in-demand'), (0.0, u'https://www.goodreads.com/genres/erlang'), (0.
0, u'https://support.apple.com/en-lb/HT201259'), (0.0, u'https://speakerdeck.com
/yutopp/c-plus-plus-falsesusume'), (0.0, u'https://social.msdn.microsoft.com/Foru
ms/en-US/32f1a857-91d6-472d-b333-469ed7f6f6dd/passing-a-byte-array-to-a-c-dll-f
rom-c?forum=csharpplanguage'), (0.0, u'https://news.ycombinator.com/item?id=92977
70'), (0.0, u'https://msdn.microsoft.com/ko-kr/library/26cfh6e6(v=vs.110).aspx')
, (0.0, u'https://msdn.microsoft.com/ko-kr/library/s9ts558h(v=vs.110).aspx#cpcon

```

```

>>> url=recommendations.getRecommendations(dd,user)[0][1]
>>> recommendations.topMatches(recommendations.transformPrefs(dd),url)
[(1.0, u'https://www.goodreads.com/genres/erlang'), (1.0, u'https://support.appl
e.com/en-lb/HT201259'), (1.0, u'https://speakerdeck.com/yutopp/c-plus-plus-false
susume'), (1.0, u'https://social.msdn.microsoft.com/Forums/en-US/32f1a857-91d6-4
72d-b333-469ed7f6f6dd/passing-a-byte-array-to-a-c-dll-from-c?forum=csharpplanguag
e'), (1.0, u'https://news.ycombinator.com/item?id=9297770')]
>>> |
Ln: 139|Col: 4

```

FIGURE 4.21 – Recommandation des voisins et des liens.

## 4.4 Conclusion

Dans ce dernier chapitre nous avons essayé de construire un moteur de recommandation pour le site Delicious. Dans un premier temps nous avons travaillé sur un dictionnaire local, puis nous avons généralisé le travail sur le site lui-même. Toutefois, des problèmes relatifs à l'API de Delicious sont à signaler tout au cours des premiers mois de l'année 2016.

Source : <http://www.wandlesoftware.com/2016/03/delicious-api-broken.html>.

---

## CONCLUSION GÉNÉRALE

---

Il va sans dire que la surcharge cognitive d'un utilisateur due à une trop grande quantité d'information est un problème majeur des systèmes hypermédia. L'utilisateur navigant sur le Web se trouve ainsi submergé par l'abondance d'informations. Les systèmes de recommandation ont été introduits pour résoudre ce problème et sont maintenant largement utilisés pour améliorer l'expérience utilisateur sur le Web.

Notre travail nous a permis :

- de montrer l'importance des algorithmes de recommandation qui permettent de produire des recommandations de haute qualité, d'effectuer de nombreuses recommandations par seconde pour des millions d'utilisateurs.
- de mettre en œuvre ces algorithmes en Python qui offre des avantages dans le traitement d'une grande quantité d'information.
- de montrer la puissance de ces algorithmes quand il s'agit de recommander des liens.
- de souligner que les systèmes de recommandations sont apparus pour essayer de résoudre et limiter les problèmes liés à la surcharge informationnelle (surcharge cognitive et la désorientation) par proposition de recommandations d'items.

Cependant, la tendance actuelle des systèmes de recommandation est plutôt axée sur des méthodes nouvelles, multicritères, multidimensionnelles ou encore se fondant sur des notions psychologiques comme les émotions, les opinions.

Le système réalisé porte sur le site de partage de signets *del.icio.us* ; car il ne possédait pas de moteur de recommandation. Il peut se prévaloir d'apporter des solutions à bon nombre de problèmes de navigation en conformité avec les objectifs annoncés.

Par ailleurs, il semble maintenant nécessaire d'aborder des questions qui relèvent de l'usage et du processus de contrôle que l'utilisateur doit exercer sur sa propre activité. On compte tester notre système dans le cadre de l'apprentissage coopératif – à distance ou à présence-. Cette évaluation nous permettra de vérifier si les utilisateurs perçoivent eux-mêmes les avantages de cette simplification de procédures de navigation et de s'assurer des atouts de notre système.

Enfin, nous entrevoyons pour nos travaux de généraliser notre expérience avec *DELICIOUS* à d'autres sites Web ainsi que la mise en œuvre de techniques adéquates pour pouvoir travailler sur des bases de données volumineuses.

---

# BIBLIOGRAPHIE

---

## Références bibliographiques

- [1] Adomavicius, G. and Tuzhilin, A. (2005). *Toward the next generation of recommender systems : A survey of the state-of-the-art and possible extensions*. IEEE Transactions on Knowledge and Data Engineering, 17(6) :pages 734-749.
- [2] Basu, C., Hirsh, H., Cohen, W. and Manning, N. C. (2001). *Technical paper recommendation : A study in combining multiple information sources*.
- [3] BERKANE, T. (mars 2012). *Système d'information & Intelligence Artificielle*, mémoire de Magister hypermedia adaptatif éducatif : interface adaptative et gestion des profils apprenants, université mouloud Mammeri.
- [4] Burke, R. (2002). *Hybrid recommender systems : Survey and experiments*. *User Modeling and User-Adapted Interaction*.
- [5] Conklin, J. (1987). *Hypertext : an introduction and survey* , Microelectronic and Computer Technology Corp., pages 18-40.
- [6] EDWARDS Deborah, M.,HARDMAN, L. (1989). *Chapter 7 : "Lost in Hyperspac"*. *Cognitive mapping and navigation in a hypertext environment*, in McAleese Ray (dir.),

- 
- Hypertext : Theory into practice, Oxford, Blackwell Scientific Publications Ltd., coll. Intellect books, pp. 105-125.
- [7] GALL James, E. HANNAFIN Michael, J. (1994). *A framework for the study of hypertext*, in Instructional Science, Vol. 22, n°3, pp. 207-232.
- [8] Goldberg, D., Nichols, D., Oki, M., and Terry, D. (1992). *Using collaborative filtering to weave an information tapestry*. In Commun. ACM, volume 35, pages 61-70.
- [9] KARAOUZENE, M. (2015). *système de recommandation des services web sémantiques*. Université Abou Bakr Belkaid, Tlemcen.
- [10] Maes, P. and Shardanand, U. (1995). *Social information filtering : algorithms for automating "word of mouth"*. In the SIGCHI conference on Human factors in computing systems, Denver, Colorado, United States. ACM Press/Addison-Wesley Publishing Co.
- [11] Malone, T., Brobst, S., Cohen, S., Grant, K., and Turk, F. (1987). *Intelligent information des systèmes de partage*. In Communications of the ACM, volume 30, pages 390-402.
- [12] Massa P. and Avesani P. (2004). *Trust-Aware Collaborative Filtering for Recommender Systems*, in Proceedings of the Int. Conf. on Cooperative Information Systems.
- [13] Michael D. Ekstrand, John T. Riedl, Joseph A. Konstan. (2010). *Collaborative Filtering Recommender Systems, Foundations and Trends R in Human Computer Interaction*, Boston-Delft.
- [14] Miller, G.A. (1956). *the magical number seven, plus or minus two : some limits on our capacity processing information* , Psychological Review, 63, 81-97.
- [15] NAKAGAWA, M., MOBASHER, B. (2003). *A Hybrid Web Personalization Model Based on Site Connectivity*. In Proceedings of WebKDD, pages 59-70.
- [16] Nicolas, B. *Etat de l'art sur les Systemes de Recommandation*, Projet AXIS de l'INRIA, dans le cadre du projet Addictrip

- 
- [17] Poirier, D. (2011). *Des textes communautaires à la recommandation*. PhD thesis, Université d'Orléans et Université Pierre et Marie Curie - Paris 6.
- [18] Resnick, P. and Varian, H. (1997). *Recommender systems*. In *Communications of the ACM*, volume 40, pages 56-58.
- [19] Rhéaume, J. (décembre 1993). *les hypertextes et les hypermédias*, Revue EducaTechnologie, volume 1, numéro 2.
- [20] Sarwar, B., Karypis, G., Konstan, J., and Riedl, J. (2001). *Item-based collaborative filtering recommendation algorithms*. In WWW10, pages 285-295, Hong Kong. ACM.
- [21] Schein, A. I., Popescul, R., Ungar, L. H., Pennock, D. M. (2002). *Methods and Metrics for Cold-Start Recommendations*. In In Proceedings of the 25th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval pp. 253-260, ACM Press.
- [22] Swinnen, G. (2000-2008). *apprendre à programmer avec Python*.
- [23] Zaïer, Z. (2010). *Modèle multi-agents pour le filtrage collaboratif de l'information* Thèse. Doctorat en informatique cognitive, Montréal (Québec, Canada).

## Références hypertextes

- [24] Mathieu, 25.04.2012, in Dossiers, Mathieu Les algorithmes de recommandation, <http://www.podcastscience.fm/dossiers/2012/04/25/les-algorithmes-de-recommandation/> consulté le 14/03/2016.
- [25] Navigation-sociale, <https://fr.wikipedia.org/wiki/Navigation-sociale>, consulté 11/03/2016;
- [26] Samaidhu, (January 24, 2015), *recommendation engine*, <https://dataaspirant.com/2015/01/24/recommendation-engine-part-1/>.

- [27] Balasabramian,V., (1994), *State of the Art Review on Hypermedia Issues and Applications* , Graduate School of Management, Rutgers University, Newark, NJ, <http://www.isg.sfu.ca/~duchier/misc/hypertext-review/index.html>
- [28] Benchmark Group, 2016, *comparatif de 7-moteurs de recommandation e-commerce*,<http://www.journaldunet.com/solutions/saas-logiciel/comparatif-de-7-moteurs-de-recommandation-e-commerce/>
- [29] Par Senee, 2015, *5 moteurs de recommandation de merchandising à la loupe*, Skeelbox - Tous droits réservés Mentions légales | Skeelbox, 13 rue de Charenton, 75012 Paris, <http://www.skeelbox.com/etude-5-moteurs-de-recommandation/>
- [30] Jean-François., P, *Moteur de recherche*, <http://www.commentcamarche.net/contents/1321-moteur-de-recherche>, consulté 28/04/2016
- [31] *Le traitement des donnees nouvelle source de pouvoir le cas des moteurs de recommandation*, <http://www.journaldunet.com/ebusiness/expert/62393/le-cas-des-moteurs-de-recommandation.shtml>, consulté 11/03/2016
- [32] *L'EFFICACITÉ DE LA RECOMMANDATION PRODUIT CHEZ AMAZON*, POSTÉ LE 10 JUIN 2013,<https://www.antvoice.com/e-commerce2/lefficacite-de-la-recommandation-produit-chez-amazon/>
- [33] <http://dist.emencia.net/RAPPEL-PYTHON.html>, consulté 12/02/2016
- [34] <http://python.developpez.com/telecharger/detail/id/1923/Boa-Constructor-wxPython>, consulté 10/05/2016
- [35] <http://alain72.developpez.com/tutos/wxPython/>, consulté 10/05/2016
- [36] <https://fr.wikipedia.org/wiki/PyGTK>, consulté 15/05/2016
- [37] <http://webcache.googleusercontent.com/search?q=cache:http://fsincere.free.fr/isn/python/cours-python-tkinter.php>, consulté 15/05/2016



---

# ANNEXE

---

## Liste des abréviations :

**API** : *Application Programming Interface.*

**CA** : *Chiffre d’Affaires.*

**FC** : *Filtrage Collaboratif.*

**FBC** : *Le Filtrage à Base de Contenu.*

**k-NN** : *K-plus proche voisin(k-nearest neighbor).*

**MR** : *Moteur de Recherche.*

**MIT** : *Massachusetts Institute of Technology.*

**SR** : *Système de Recommandation.*

**SRC** : *Systèmes de Recommandation à base de Confiance.*

**SaaS** : *Software as a service.*

**UML** : *Unified Modeling Language.*

**GUI** : *Graphical User Interface.*

---

## Résumé

---

Les systèmes de recommandations sont des systèmes automatiques qui permettent, par des techniques de Data Mining, de fournir à des utilisateurs des suggestions qui répondent à leurs exigences. Un grand nombre de systèmes de recommandation existent dans divers domaines. Leur objectif est de filtrer et d'adapter les informations pour chaque utilisateur. Les méthodes généralement utilisées pour le calcul de la recommandation sont soit basées sur le contenu soit sur la similarité de l'utilisateur avec les autres utilisateurs (approches collaboratives).

Ce travail a pour objectif de développer un système de recommandation social sous Python. Il sera utilisé dans le domaine de la navigation sur le Web pour suggérer à un utilisateur des ressources pertinentes susceptibles de l'intéresser en se basant sur les avis d'autres utilisateurs qui partagent avec lui les mêmes centres d'intérêt.

**Mots clés :** *Data Mining, Système de recommandation social, Approches collaboratives, filtrage basé sur le contenu, Python.*

---

## Abstract

---

Recommendations systems are automatic systems. that, by Data « Mining techniques » provide to users suggestions that meet their requirements. Many recommendation systems exist in various areas. Their aim is to filter and adapt the information for each user. The methods generally used for the calculation of the recommendation is based on the content or on the similarity of the user with other users (collaboratives approaches).

This project aims to develop a social recommendation system in Python to be used in the area of Web browsing who suggest to user resources may interest him based on the opinions of other users who share with him the same interests.

**Keywords:** *Data Mining, Social recommendation system, Collaboratives approaches, Filtering based on content, Python.*

---

## ملخص

---

إن أنظمة التوصية هي الانظمة الآلية التي تسمح بواسطة تقنيات " تعدين البيانات " تزويد المستخدمين بالاقترحات التي تلي متطلباتهم. عدد كبير من أنظمة التوصية موجودة في مختلف المجالات هدفها هو تصفية وتكيف المعلومات لكل مستخدم. الأساليب المستخدمة عادة لحساب التوصية إما تكون على أساس المحتوى أو على تشابه المستخدم مع مستخدمين آخرين (مناهج التعاونية). هذا المشروع يهدف الى تطوير نظام توصية اجتماعي باستعمال لغة البرمجة بيثون الذي سيتم استخدامه في مجال تصفح الإنترنت والذي بدوره يقترح على مستخدم موارد على الارجح قد تهمة استنادا إلى آراء مستخدمين آخرين الذين يشتركون معه في نفس الاهتمامات.

**كلمات الدالة:** *تعدين البيانات ، نظام توصية الاجتماعي ، مناهج التعاونية ، تصفية على اساس المحتوى ، بيثون.*