

# Projet tutoré

---

## Tableau de bord d'un système de recommandation

---

**Auteur :**

Quentin PARMENTIER

**Encadrant :**

Azim ROUSSANALY

**Année : 2018**

# Tableau de bord d'un système de recommandation

---

**Etudiant** : Quentin PARMENTIER

**Encadrant** : Azim ROUSSANALY

**Année universitaire** : 2017 – 2018



## Remerciements

Tout d'abord, je souhaite remercier mon tuteur, Azim ROUSSANALY, pour m'avoir fait confiance, m'avoir guidé et accompagné tout au long de ce projet.

Je tiens aussi à remercier l'équipe KIWI et toutes les personnes ayant travaillé sur PERICLES pour le travail qu'ils avaient déjà fourni.

# Table des matières

Remerciements .....	3
Table des matières .....	4
1. Introduction.....	5
2. Les systèmes de recommandations .....	6
2.1 Recueil d'information .....	6
2.2 Matrice ou Modèle utilisateur et prédiction.....	7
2.3 Les autres types de système de recommandations .....	8
2.3.1 Recommandation personnalisée.....	8
2.3.2 Recommandation objet.....	8
2.3.3 Recommandation hybride.....	8
2.4 Les ressources .....	9
3. Evaluation d'un système de recommandation.....	10
3.1 Méthodes d'évaluation .....	10
3.2 Les indicateurs de performance .....	11
4. Le projet et sa réalisation .....	14
4.1 L'architecture du projet.....	14
4.2 Le SR au sein des UNT .....	15
4.3 Les choix et difficultés .....	17
4.4 La réalisation .....	18
4.5 La base de données .....	19
5. Etat du projet et améliorations possibles .....	21
6. Conclusion .....	23
A. Annexes.....	24
A.1 Structures de la base de données .....	24
A.2 Mode opératoire .....	24
A.3 Bibliographie .....	28

## 1. Introduction

Durant la première année du Master SCA (Sciences Cognitives et ses Applications), j'ai été amené à faire un projet tutoré sur la période du second semestre pour lequel je devais mettre en place un logiciel pouvant évaluer un système de recommandation (SR) mis en place grâce à différents indicateurs choisis par mes soins en fonction de mes recherches.

Ce projet est mené dans le cadre d'un Appel à Manifestation d'Intérêt (AMI), par le Laboratoire Lorrain de Recherche en Informatiques et ses Applications (LORIA), plus particulièrement l'équipe KIWI (Knowledge Information and Web Intelligence) dont mon encadrant, Azim ROUSSANALY fait partie. Leurs recherches se portent sur des systèmes complexes, de l'intelligence artificielle et de la robotique.

Il y a deux systèmes de recommandation dans ce projet, l'un sur le site de cours en ligne ARCHE et un autre sur chaque Universités Numériques Thématiques (UNT), ils ont pour but de proposer des ressources universitaires, pour faire découvrir de nouvelles ressources inexploitées ou d'aiguiller un étudiant dans sa recherche de ressources.

Dans la première partie de ce rapport, nous verrons qu'est-ce qu'un système de recommandation et comment il est mis en place dans notre cas. Dans une deuxième partie, on abordera comment l'évaluer et pour finir la réalisation de l'application.

## 2. Les systèmes de recommandations

Dans un premier temps, pour traiter notre sujet, il me paraît évident de devoir comprendre « qu'est-ce qu'un système de recommandations ? ».

« Un système de recommandation est une forme spécifique de filtrage de l'information visant à présenter des éléments d'information (films, musique, livres, news, pages Web, etc.) qui sont susceptibles d'intéresser l'utilisateur ».

[\[Wikipédia2016\]](#)

Pour cela le système procède en plusieurs étapes :

- Recueillir de l'information sur l'utilisateur
- Construire une matrice ou un modèle utilisateur en y mettant les informations recueillies
- Constituer une liste de recommandation à partir du modèle

### 2.1 Recueil d'information

On distingue deux façons de recueillir des informations.

La première étant à la vue de l'utilisateur, on lui demande par quoi est-il intéressé, ses informations personnelles ou bien ce qu'il aimerait voir, comme sur Pinterest par exemple, on appelle cela le filtrage actif.

La deuxième est celle que l'on utilise dans notre système, le filtrage passif, qui consiste à récupérer des informations sans rien demander à l'utilisateur sur ses dernières connexions au site, grâce à un système de cookie, ce qu'il a visité, par quel domaine est-il attiré de par ses recherches etc. , comme sur Amazon où quand on regarde un article, on nous en propose des similaires sans avoir renseigné ce que l'on recherche.

## 2.2 Matrice ou Modèle utilisateur et prédiction

Notre modèle ou notre matrice et la restitution d'une liste de recommandation vont être construits en fonction de l'approche de création et de l'algorithme choisi. Différentes approches sont alors possibles, dans notre cas on a deux SR différents. Le premier en local utilise la recommandation sociale. Cette recommandation se base sur plusieurs utilisateurs qui ont un comportement ou des profils similaires. Par exemple, vous et un autre utilisateur êtes passionnés des films fantastiques et que vous avez visité vingt pages de films en commun et le lendemain cet autre utilisateur visite un vingt et unième film que vous n'avez pas vu, alors le système vous le propose car vous avez des comportements similaires.

Le deuxième utilise le Vector Space Model (VSM) qui utilise les mots-clés des documents pour créer des vecteurs et calculer des similitudes à l'aide de distances entre vecteurs. Plus les vecteurs ont une distance faible, plus les documents auxquels ils appartiennent sont similaires.

La prédiction est le fait de choisir les recommandations envoyées à l'utilisateur. Dans le projet PERICLES il y a six algorithmes différents.

Le premier étant basique, chaque mot a le même poids.

Le deuxième modifie le poids des mots, plus un mot est présent dans le document, plus son poids est élevé.

Le troisième, au lieu de regarder les mots en tant que tel, l'algorithme va se baser sur la sémantique du mot. Par exemple, on a le mot livre. Si on retrouve ce mot dans un autre document il aura un poids de 1 mais contrairement aux autres algorithmes si on trouve un mot similaire, il aura aussi un poids, comme bouquin qui est proche de livre et aura un poids de 0,85. Encyclopédie aussi, mais sémantiquement plus éloigné que bouquin, il aura un poids de 0,4.

Ces six algorithmes vont alors retourner des propositions et les trois ayant le plus haut score seront affichées à l'écran de l'utilisateur.

Quel que soit le SR ou l'algorithme utilisé, les données de ceux-ci sont standardisées pour aller être inscrites dans la même base de données que l'on analysera dans la partie réalisation.

## **2.3 Les autres types de système de recommandations**

On a vu ensemble celui utilisé, il en existe d'autres connu que l'on va voir succinctement.

### **2.3.1 Recommandation personnalisée**

Basée sur les visites de l'utilisateur, ce système recommande des articles, pages en fonction de ceux déjà visités. Un exemple très simple, sur Amazon ou les publicités Google.

### **2.3.2 Recommandation objet**

Ce système prend en compte « des qualités et des propriétés de l'objet, ici des ressources universitaires, pour les corrélés avec les préférences et les intérêts de l'utilisateur ». [\[Mathieu2014\]](#)

### **2.3.3 Recommandation hybride**

Le système hybride est une combinaison des trois autres pour contrer les différents inconvénients de chacun des autres systèmes.

## 2.4 Les ressources

Pour pouvoir retourner des ressources, il faut en amont en avoir. Pour cela le système va piocher dans des Universités Numériques Thématiques (UNT) qui regroupent environ trente-cinq mille ressources différentes ayant des thèmes propres à leurs UNT respectives. On compte parmi les sept UNT, l'Université Numérique Thématique en Economie et Gestion (AUNEGE), l'Université Numérique Francophone des Sciences de la Santé et du Sport (NUF3S), l'Université Numérique Ingénierie et Technologie (Fondation UNIT), l'Université des Sciences en Ligne (UNISCIEL), l'Université Numérique Juridique Francophone (UNJF), l'Université Ouverte des Humanités (UOH) et l'Université Numérique Francophone des Sciences de la Santé et du Sport (UVED).

Les ressources sont principalement des ressources scientifiques et multimédias. Elles sont écrites par des membres des UNT c'est-à-dire par des professeurs d'universités et d'écoles du supérieur et sont ensuite validées par un conseil scientifique pour juger si oui ou non elles ont une place dans une UNT. C'est en finançant ces ressources qu'elles peuvent être libres et disponibles pour tout le monde.

Dans la base de données (BDD) du projet SR les recommandations sont sous formes de métadonnées. « Une métadonnée est littéralement une donnée sur une donnée. Plus précisément, c'est un ensemble structuré d'informations décrivant une ressource quelconque ». [\[Patrick Peccatte 2007\]](#)

Dans d'autres termes, la ressource va être décrite en fonction de plusieurs descripteurs, ce sont ses métadonnées.

Tous les jours, il y a un moissonnage des métadonnées, c'est-à-dire qu'un robot va toutes les lire et voir s'il y en a de nouvelles. Dans le cas positif, les nouvelles ressources sont stockées dans notre base comme nouvelle ressource.

Ce projet de système de recommandation a alors plusieurs buts. Dans un premier temps aider les étudiants à trouver des articles, des cours, des informations qu'ils recherchent, communément appelées ressources pédagogiques ou divertir leurs connaissances grâce à la banque de données que fournit les UNT mais par la même occasion faire connaître l'existence des UNT à un bien plus grand nombre de personnes.

Maintenant que la notion et le fonctionnement d'un SR est claire, on va aborder la qualité. Dans le prochain chapitre nous verrons comment nous pouvons juger de l'efficacité d'un SR et quels indicateurs de performance nous avons choisi.

### **3. Evaluation d'un système de recommandation**

Dans cette partie nous allons voir les différentes méthodes d'évaluation et les indicateurs pour définir les performances d'un SR. Le but est de vérifier que les algorithmes nous retournent des ressources pertinentes.

#### **3.1 Méthodes d'évaluation**

Il existe trois grandes méthodes pour évaluer des performances.

La première à placer l'utilisateur au cœur de l'évaluation en prenant un échantillon de ceux-ci, souvent en les invitant directement dans des locaux, pour tester une application ou un nouveau système. Le but sera alors de relever des avis à chaud de l'utilisation, ou bien la qualité des ressources dans notre cas, en en retirant des données qualitatives, par exemple « bien », « ergonomique » et des données quantitatives qui noteront l'acceptabilité ou les algorithmes. On appelle cela l'analyse centrée utilisateur.

Malheureusement, cette méthode est onéreuse, non durable dans le temps et peut contenir des biais de déclaration où l'utilisateur ment ou se laisse influencer sur ce qu'il pense.

La deuxième consiste en l'analyse de « la qualité d'un modèle prédictif à partir de données historiques, sans interagir avec les clients ou utilisateurs » [\[Arnaud De Myttenaere 2016\]](#). C'est l'évaluation hors ligne. Généralement on s'en sert pour faire de l'A/B testing. Il y a deux populations, une avec un ancien SR (A) et une autre avec un nouveau SR (B). On compare ensuite les résultats des deux populations, par exemple, le nombre de ventes sur Amazon.

L'inconvénient étant qu'il faut une base de données considérable pour pouvoir appliquer cette méthode, que l'on n'a pas forcément.

La dernière et celle que l'on va utiliser s'appelle l'analyse en ligne. Elle permet d'analyser les données des utilisateurs en laissant une notion de temps. En mettant alors en place des paramètres d'analyse que l'on souhaite regarder on pourra indépendamment du temps ou des personnes connectées, analyser nos données. C'est bien pour ces raisons que nous nous dirigeons vers cette méthode. La contrepartie est que nous n'avons pas de données qualitatives.

### 3.2 Les indicateurs de performance

Maintenant que nous avons notre méthode d'évaluation, il nous faut connaître les différents indicateurs nous permettant de juger si notre système est efficace et lequel ou lesquels de nos algorithmes sont plus performants.

Nous parlerons ici uniquement des indicateurs basés sur des données quantitatives. Nous énoncerons une première équation, qui sera générale, puis une deuxième qui sera transposée sur notre projet en fonction de notre base. ([Figure 4,5 et 6](#))

Dans le cas d'un système de recommandation on a quatre possibilités pour une ressource. Soit elle est recommandée ou non, soit elle est consultée ou non. [[G. Shani & A. Gunawardana 2010](#)]

	Recommandée	Non recommandée
Utilisée	True-Positive (tp)	False-Negative (fn)
Non Utilisée	False-Positive (fp)	True-Negative (tn)

Tableau 1 : Tableau des possibilités

Différentes approches sont possibles. On peut se demander si notre SR retourne des recommandations de qualité, c'est-à-dire est-ce qu'elles intéressent nos utilisateurs. On utilise la précision pour cela. [[G. Shani & A. Gunawardana 2010](#)]

$$Précision = \frac{\#tp}{\#tp + \#fp} = \frac{\sum(nb\_vue)}{\sum(nb\_montre)}$$

Dans le cas d'une précision trop basse, on a une requête trop large sur nos recommandations.

On peut utiliser le rappel pour savoir quelle est la probabilité qu'une recommandation intéressante soit proposée. [\[G. Shani & A. Gunawardana 2010\]](#)

$$Rappel = \frac{\#tp}{\#tp + \#fn}$$

Dans notre cas nous n'avons pas la possibilité d'avoir #fn, donc le rappel ne sera pas un de nos indicateurs.

Par exemple, s'il y a 10 suggestions pertinentes mais que seules 4 sont proposées, on a un rappel de 0,4. Le fait qu'il y a des ressources pertinentes non proposées s'appelle des silences. Dans le cas d'un grand silence, c'est que les requêtes sont trop précises ou que les ressources sont mal indexées. [\[N. Friburger 2009\]](#)

Mais si uniquement ces quatre suggestions sont proposées, on a une précision de 1.

On peut donc combiner les deux pour avoir une mesure harmonieuse, cela se nomme la F-mesure.

$$F - mesure = 2 * \frac{(Précision * Rappel)}{(Précision + Rappel)}$$

Pour évaluer nos suggestions, nous pouvons aussi évaluer les écarts entre les scores donnés par nos algorithmes et le score réel des suggestions, on utilise pour cela la Mean Absolute Error (MAE). Dans notre cas, nous n'avons pas la possibilité de savoir quel est le score réel de nos suggestions, ça ne serait donc pas très significatif en le demandant à un seul utilisateur.

Autre point important, la diversité de nos suggestions. Une étude menée par Lathia en 2010 démontre que les utilisateurs attendent une certaine diversité dans les suggestions d'un SR sans quoi ils se sentiraient frustrés par un manque de choix. [\[Lathia 2010\]](#) [Équipe KIWI 2013]

On veut aussi que notre SR suggère toutes nos ressources disponibles, on appelle cela la couverture de notre SR, c'est le pourcentage des ressources existantes proposées comme suggestion.

$$\begin{aligned} \text{Couverture} &= \frac{\#tp + \#fp}{\text{Nombre totale de ressources}} \\ &= \frac{\text{Count}(\text{Distinct suplom\_id})}{\text{Count}(id\_items)} \end{aligned}$$

Un top des ressources est aussi un indicateur pertinent pour un tableau de bord.

$$\begin{aligned} \text{Top 20 des ressources proposées} &= \text{Top}(\#tp + \#fp) \\ &= \text{Top 20 } \sum(nb\_montre) \text{ group by } id\_items \end{aligned}$$

$$\text{Top 20 des ressources suivies} = \text{Top}(\#tp) = \text{Top 20 } \sum(nb\_vue)$$

$$\begin{aligned} \text{Top 20 des ressources précision} &= \frac{\text{Nombre de fois proposée}}{\text{Nombre de fois cliquées}} \\ &= \text{Top 20 } \frac{\sum(nb\_montre)}{\sum(nb\_vue)} \text{ group by } suplom\_id \end{aligned}$$

Avoir une vision absolue pour connaître l'état d'utilisation du système était un besoin énoncé par les futurs utilisateurs du logiciel d'évaluation de notre système, je les ai donc mis en place.

$$\text{Nombre absolu de recommandation} = \#tp + \#fp = \sum(nb\_montre)$$

$$\text{Nombre absolu de recommandation suivies} = \#tp = \sum(nb\_vue)$$

$$\text{Nombre absolu d'utilisateur} = \#utilisateur = \text{Count}(\text{Distinct } id\_user)$$

$$\text{Nombre de ressources disponible} = \#ressource = \text{Count}(id\_items)$$

De plus pour chacun des indicateurs nous pouvons faire des agrégats pour comparer plusieurs éléments entre eux. Comme les UNT ou les catégories des ressources, les algorithmes. Un agrégat par unité de temps aurait été

intéressant pour voir l'apprentissage et la différence d'utilisation, malheureusement notre système ne mémorise que la dernière proposition et la dernière suivie d'une ressource par un utilisateur. Pour pallier cela, nous avons choisi d'extraire notre base de données à différents moments et dire à quel numéro d'extraction notre recommandation appartient (1<sup>ère</sup>, 2<sup>e</sup>, 3<sup>e</sup>, ...). Par exemple, je fais une extraction tous les 15 du mois à partir de janvier, ma première extraction sera le 15 janvier 2018, la deuxième le 15 février 2018, la troisième le 15 mars 2018, etc.

## 4. Le projet et sa réalisation

Le but final du projet, rappelons-le, était de créer un logiciel permettant d'évaluer un système de recommandation en utilisant des indicateurs spécifiques, vu au-dessus. Dans cette partie nous verrons l'architecture du projet SR, les difficultés que cela impose, les choix et la réalisation en tant que telle.

### 4.1 L'architecture du projet

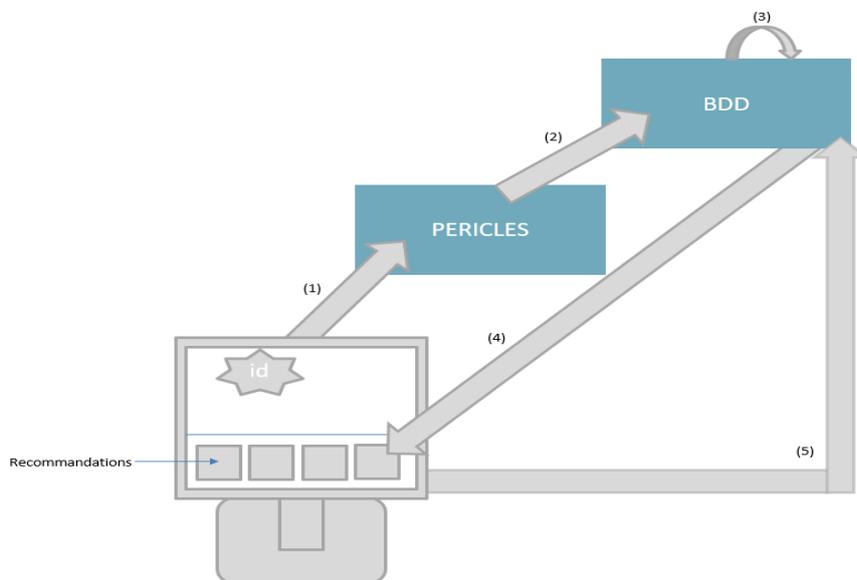


Schéma fonctionnel 1 : Le système de recommandations

Dans la BDD, de manière simplifiée, une recommandation est sous la forme (Ressource proposée, ID Utilisateur, Nombre de fois proposée, Nombre de fois suivie). ([Figure 6](#))

Le SR est actuellement déployé sur deux sites d'UNT, AUNEGE et UOH. L'utilisateur a un identifiant (ID) qui lui est attribué et enregistré dans un cookie pour qu'à une visite future nous sachons qui il est. Quand il regarde une ressource, un couple utilisateur-ressource est envoyé à Périclès (1), outils de recommandation personnalisé.

(2) Ensuite, Périclès envoie à la base de données (BDD) une requête pour récupérer les ressources à envoyer à l'utilisateur (4). En même temps la BDD enregistre les recommandations retournées couplées à l'ID de l'utilisateur (3).

En bas de page se trouve les recommandations personnalisées pour notre utilisateur, s'il décide de suivre une recommandation, l'information est stockée dans la BDD.

## 4.2 Le SR au sein des UNT

Le SR est actuellement déployé sur plusieurs UNT. Il permet d'intégrer une partie recommandation sur les pages de ressources à l'endroit souhaité par les UNT.

The screenshot shows a resource page with the following elements:

- Header:** 'Voyage interactif dans l'univers de Jules Verne' and 'PÉRICLÈS - PÉRICLÈS Éditeur - 2012'.
- Main Content:** A map titled 'Voyage dans l'univers de Jules Verne' showing various locations and routes.
- Metadata Sidebar:**
  - TYPE PÉDAGOGIQUE:** Guide pédagogique, Au travail, Banque
  - NIVEAU:** Enseignement supérieur, Autre LL, Licence
  - ACTIVITÉ INDUITE:** Apprendre à apprendre
  - PROPOSITION D'UTILISATION:** Ce module thématique s'adresse à celles et ceux qui souhaitent découvrir l'univers de Jules Verne.
  - LANGUE DE L'APPRENTISSAGE:** Français
- Access and Description:** 'Accéder à la ressource' and 'DESCRIPTION' section.
- Recommendations:** A section titled 'Recommandations' with several items:
  - 'L'anthologie française de la cartographie - France'
  - 'Jules Verne - Science-Fiction - 184-1848'
  - 'L'extension des Sciences humaines'
  - 'Les îles françaises à travers l'histoire contemporaine'
  - 'Grenoble - Illustrations (Cours - Le paléolithique supérieur européen)'
  - 'L'évolution (Ce que nous apprend l'anthropologie, 12/23)'
- Share Section:** 'Partager' with social media icons for Facebook, Twitter, LinkedIn, YouTube, Dribbble, and SoundCloud.

Le site UOH a choisi de mettre les recommandations en bas de page en dessous de l'article.

La rubrique est non visible si on ne fait pas défiler la page, il y a des articles sans images mais elle occupe une bonne partie de la page.

Je pense qu'il y a une incohérence ergonomique sur ce choix.

Exemple 1: Exemple ressource de l'UNT d'UOH

**AUNEGE** PERICLES ET L'EUROPE

**Canal-u**

"Pericles et l'Europe"

Conférenciers : Florence DUCREAU, Jacques DANG

Journée Aunège, Printemps UNT, 3 juin 2014, Amphithéâtre BENE de la Faculté de pharmacie - Université de Lorraine

**CONTACTER LE CONTRIBUTEUR**

★ ★ ★ J'AIME

➤ AJOUTER À MON CANAL-U

➤ SÉLECTIONNER UNE SÉQUENCE POUR MON CANAL-U

IMPRIMER PARTAGER

**LES CHAPITRES**

Présentation du projet	00:00
Pericles dans l'environnement européen et international	17:14
Questions de la salle	27:23

**RÉSUMÉ** | DIAPORAMA | AUTEURS | DOCUMENTS

**PERICLES ET L'EUROPE**

Printemps UNT 2014 - Première conférence commune des UNT sur le numérique dans les établissements de l'enseignement supérieur.

Journée AUNEGE - 3 juin 2014

Amphithéâtre BENE de la Faculté de pharmacie - Université de Lorraine

"Pericles et l'Europe"

par Jacques Dang (HEC) et Florence Ducreau (Université de Lorraine, AUNEGE)

PERICLES (Projet pour l'Évaluation et la Recherche Informatisée autour des Compléments dans l'Enseignement Supérieur) est un projet dans le cadre du Programme d'Investissements d'Avenir. Son but est la création d'un outil universel d'évaluation de la qualité des enseignements. Dans cette intervention le projet PERICLES est présenté par le coordinateur du projet, HEC, et par l'un des partenaires, AUNEGE.

Label UNT : AUNEGE

Date de réalisation : 3 Juillet 2014  
Durée du programme : 30 mn  
Classification Dewey : Méthodes d'enseignement

Catégorie : Conférences  
Niveau : Tous publics / hors niveau, niveau Licence (LMD), niveau Master (M2), Formation continue  
Disciplines : Méthodes d'enseignement  
Collections : Printemps AUNEGE 2014  
fiche1.com - Voir la fiche LOM

Auteur(s) : DANG Jacques, DUCREAU Florence  
producteur : AUNEGE, Université de Lorraine

Langue : Français  
Mots-clés : qualité, enseignement supérieur, évaluation, PERICLES, évaluation  
Conditions d'utilisation / Copyright : Creative Commons (BY-NC)

**COMMENTAIRES**

AJOUTER UN COMMENTAIRE | LIRE LES COMMENTAIRES

Aucun commentaire sur cette vidéo pour le moment (les commentaires font l'objet d'une modération)

**DANS LA MÊME COLLECTION**

"Synthèse de la Journée"

Media & Learning, une association européenne d'enseignants, un projet

"AUNEGE et les MOOC"

Comment édifier une ressource numérique

Exemple 2: Exemple de ressource de l'UNT d'AUNEGE

Pour l'UNT d'AUNEGE, le même choix en bas de page, des images pour chacune des miniatures générées en fonction du titre de la ressource. Une taille correcte, mais non accessible sans scroll non plus.

**universcience.tv** FACEBOOK CHANGE-T-IL LE MONDE ?

**Canal-u**

FACEBOOK CHANGE-T-IL LE MONDE ?

Avec Stéphane Hugon, sociologue, responsable du Groupe de recherche sur la technologie et le quotidien, et Dominique Carion, chercheur au laboratoire de sociologie de France Telecom R&D, centre d'études des mouvements sociaux EHESS/CNRS.

Date de réalisation : 1 Décembre 2012  
Durée du programme : 27 mn  
Classification Dewey : Interaction sociale, communication, médias et communication

Catégorie : Entretiens  
Niveau : Tous publics / hors niveau  
Disciplines : Sociologie sociale, changements sociaux, Théorie de la communication  
Mots-clés : Facebook, médias sociaux, communication, interaction sociale

producteur : AUGIER Florence  
Réalisateur(s) : ALCHINGEAU Sylvie

Langue : Français  
Mots-clés : communication, Internet, Facebook, médias sociaux, Sociologie, communication

**COMMENTAIRES**

AJOUTER UN COMMENTAIRE | LIRE LES COMMENTAIRES

Aucun commentaire sur cette vidéo pour le moment (les commentaires font l'objet d'une modération)

**DANS LA MÊME COLLECTION**

Sport, le fin des temps du monde ?

Le stockage des sports est-il malade ?

OGU, vers une crise de confiance ?

On en finit avec le pas de chatte ?

La science va-t-elle trop vite ?

Science et religion, plus que de la méfiance ?

L'univers est-il des limites ?

L'impossible partage de l'eau ?

L'homme, un animal impati ?

L'homosexualité revue par la science

Vers de nouvelles espèces animales ?

L'univers est-il accessible à nos sens ?

Clonal, OGM, biopuce... à quel sera l'espèce humaine ?

Am et science... un beau mariage ?

Faut-il encore croire au Big Bang ?

Qu'est-ce qui fait durer le temps ?

Exemple 3: Exemple de ressource de l'UNT canal-u

Canal-u est le seul à laisser apparaitre les recommandations sans avoir à scroll. Il y a un grand nombre de recommandation possible, toutes avec des miniatures.

Une analyse plus complète sur le positionnement et les différents choix ergonomiques sur le placement de recommandations pourrait être un bon sujet d'analyse.

### 4.3 Les choix et difficultés

Le problème majeur est de ne pas avoir la main mise sur les données récoltées par notre SR, il nous faut alors un moyen d'accéder à celles-ci. Pour pallier à ce souci, une extraction totale de la table de recommandations et de ressources (items) sera nécessaire pour que l'on puisse faire appel à ces données de manière externe à la BDD d'origine. Cela sera possible grâce à un script Structured Query Language (SQL) qui nous permettra d'extraire les données à une date *d*. Pour stocker ces données on utilise une BDD et SQL pour communiquer avec celle-ci. Ce choix nous permettra des requêtes simples, modulables et un ajout de différentes tables si on le souhaite. Notre base sera aussi constituée d'une table d'items représentant les différentes ressources, qui nous sont disponibles dans le but de pouvoir avoir des informations sur les types de ressources consultées, aussi extraites durant notre script.

Un autre problème est le manque de log, on n'a pas de moyen de suivre un utilisateur sur une connexion et voir son chemin en fonction des recommandations, ni même avoir une quelconque information sur celui-ci. Avoir ce type d'information permettrait d'avoir plus de finesse dans nos réponses.

Par exemple, le SR fonctionne et des gens l'utilisent. Mais qui sont ces gens, plus des étudiants ou plus des professeurs ? L'utilisent-ils à chaque connexion ? Suivent-ils plusieurs fois de suite les recommandations ?

Pour ce qui est de la réalisation, une application simple, durable et nécessitant aucune installation étant voulue, c'est pourquoi je me suis dirigé vers le langage JAVA qui nous assurera cela.

Les autres avantages de ce langage sont, la présence d'une très grande communauté, des mises à jour fréquentes et il dispose de nombreuses interfaces de programmation applicative (API). Les API permettent d'utiliser des fonctions déjà programmées pour les affilier à notre projet très facilement.

Les analyses de performances retournées à l'utilisateur du logiciel seront sous forme de graphiques, graphiques dépendant de l'indicateur. Pour ce fait nous utiliserons JavaFX, une API Java permettant d'utiliser différents outils pour la mise en page, mais aussi, des graphiques.

#### 4.4 La réalisation

Pour la réalisation, on a opté pour une interface unique où on pourra sélectionner la date de début  $d1$  et la date de fin  $d2$  d'extraction des données, un affinage des données sur une UNT unique et choisir les éléments à étudier, c'est-à-dire algorithmes et indicateurs de performance.

Il existe aussi une fonction nous permettant de faire la différence entre les résultats d'une extraction et ceux de l'extraction à une date  $d-1$ , cela permet d'analyser l'évolution de la croissance selon les différents indicateurs.

De plus chaque champ est marqué d'une aide, indiqué par le rond bleu avec un « ? » dedans, pour que l'utilisateur puisse comprendre à quoi correspond chacun des champs et leurs subtilités avec un exemple à chaque fois.

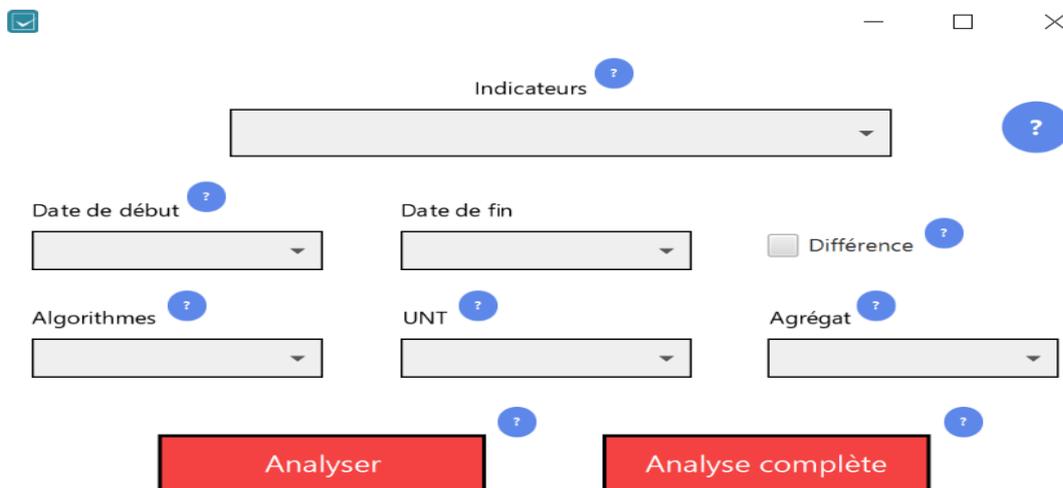


Figure 1 : Maquette de l'application

Et le schéma fonctionnel de notre application sera le suivant :

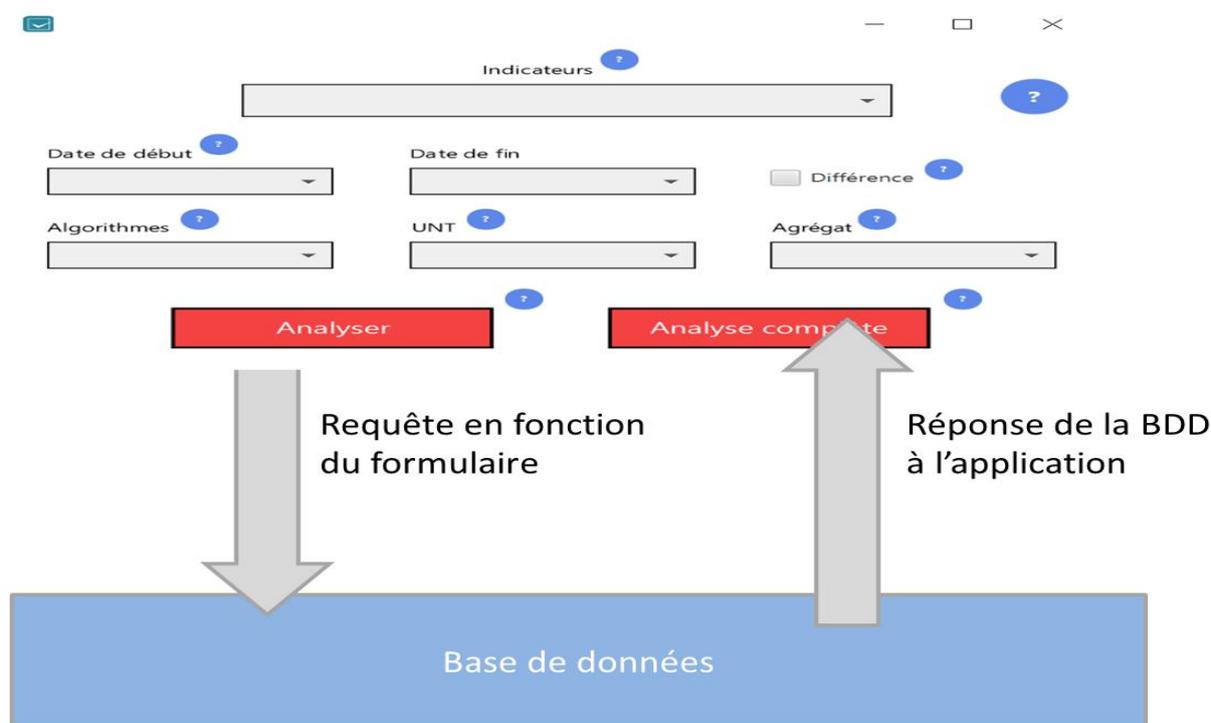


Figure 2 : L'application tableau de bord

L'utilisateur choisit son indicateur, les différents paramètres qu'il souhaite et appuie sur « Analyser » pour créer une requête adaptative en fonction de ses choix, la requête interroge la BDD, la BDD répond et le programme génère une interface graphique. Pour plus de détails sur l'utilisation du logiciel, un mode opératoire fût écrit. [\[Mode opératoire\]](#)

#### 4.5 La base de données

Comme je le disais précédemment, le programme est à destination d'utilisateurs travaillant dans les UNT et ayant aucune connaissance informatique poussée. C'est pour cela que le programme se veut clé en main et que la base de données doit être externe. La BDD a pour système de gestion PostgreSQL qui est gratuit et ayant bien plus de fonctions que MySQL.

Enoncé précédemment, la façon de fonctionner du SR nous force à faire des injections de la base PERICLES à la notre à un intervalle de temps que l'on choisit. Cette action ne pouvait pas non plus leurs être confiée, un script d'injection avec un planificateur de tâches qui l'exécute a été mis en place.

Dans la première version du programme on injectait toutes les données telles quelles, mais après une petite analyse, il y avait 1 100 000 lignes minimums qui allaient être injectées à chaque fois. A hauteur d'une injection par semaine à terme, c'est plus de 57 millions de lignes par an. Un ralentissement du traitement des données, un poids de base grossissant aurait été un souci à long terme.

Pour remédier à ce problème, une nouvelle version de l'injection a été déployée. Désormais seules les recommandations ayant eu une modification, sur nb\_monstre ou/et nb\_vue, sont désormais injectées. Les requêtes ont aussi dû être mises à jour pour être adaptées à cette nouvelle version.

## 5. Etat du projet et améliorations possibles

Le projet est actuellement fonctionnel et prêt à l'emploi. Une mise en production de l'outil couplé à un retour utilisateur pourrait permettre d'ajouter d'autres fonctionnalités ou d'autres indicateurs voulus comme pour l'analyse complète sous forme de fichier qui nous a été proposé par Carole Stefan-Schorle, directrice d'UOH.

D'autres améliorations sont possibles comme la fiabilisation des données dans la base de données PERICLES, des noms de catégories de ressources ne sont pas sous forme UTF-8 alors que parfois les mêmes le sont, cela double les catégories et sépare les données l'une de l'autre.

Analyser les algorithmes un à un est actuellement impossible car dans la base il n'y a que « RANDOM », correspondant au choix aléatoire d'un algorithme pour la recommandation, ou « VSM » pour décrire un algorithme.

De même pour les ressources UNT, la spécification de la provenance de la ressource se fait uniquement sur UOH, pour les autres il est impossible de les dissocier car elles sont marquées d'un blanc.

Une autre possibilité d'amélioration serait de créer une base de logs qui traquerait le chemin parcouru par nos utilisateurs. Cela permettrait de voir le chemin de décision et comment il se laisse influencer par les recommandations ou non, combien il est prêt à en suivre d'affilée, reste-t-il toujours sur le même thème ou il en change. Beaucoup de questions pourraient être posées et analysées grâce à ce changement. Cette possibilité avait déjà été envisagée durant la création du SR, mais il y avait un nombre bien trop grand de logs, c'est donc pour un souci d'optimisation que le modèle actuel est tel.

Le troisième point d'amélioration n'est pas technique, il serait intéressant de démocratiser ces UNT, ce système de recommandation, avoir une vraie communication sur leurs existences et permettre un flux de visiteurs plus conséquent pour avoir un jeu de données plus complet et des analyses plus justes et plus complètes.

De notre côté, notre base de données est alimentée grâce à des injections. La fréquence actuelle est quotidienne pour des soucis de temps et pour peupler la base dans le but de pouvoir commencer des analyses sur les données. La fréquence est à revoir mais est facilement changeable dans le planificateur de tâches.

Bien que les premiers résultats étaient encourageant à partir du 15 mai 2018, des aberrations apparaissent, comme des recommandations faites 1450 fois et

seulement un seul utilisateur actif. Le problème vient de la BDD PERICLES qui n'est pas toujours alimentée et nous n'avons malheureusement pas la main sur ces incidents.

Une fois la BDD PERICLES stable et fiabilisée avec les améliorations citées ci-dessus, le programme pourra être utilisé à son maximum, mais nous sommes conscients que c'est un point important à corriger.

## 6. Conclusion

La première étape était de comprendre le fonctionnement des systèmes de recommandations et plus particulièrement le nôtre.

Une fois cela fait, nous avons fait des recherches bibliographiques pour déterminer ce qu'était un bon indicateur pour analyser un SR.

Différents indicateurs ont été sélectionnés pour notre SR. Malheureusement certains n'étaient pas implantables car nous n'avions pas les données nécessaires.

Une fois le travail de recherche fini, nous avons développé une première version de l'application, qui n'était pas très ergonomique, donc une autre version a pris jour mettant plus en avant les indicateurs comme élément central.

Tous les indicateurs retenus ont été mis à disposition dans l'outil d'analyse développé.

La question d'optimisation s'est alors posée. Effectivement, en travaillant avec des injections récurrentes pour récupérer les données de la base PERICLES, nous injectons une grande quantité de données fixes d'une injection à une autre. Une redondance des données était alors dans notre base qui aller croître de manière linéaire et contenir de plus en plus de redondance.

Pour pallier à ce problème une nouvelle version fût mise en place permettant l'injection uniquement des recommandations ayant changée de statut.

On retiendra que certaines améliorations sont possibles, comme la fiabilisation des données dans la BDD PERICLES, un journal d'actions des utilisateurs, l'ergonomie des UNT vis-à-vis des recommandations et la communication sur l'existence de celles-ci en plus d'une incitation à aller les parcourir.

Pour finir, une mise en place du programme au sein des UNT couplée à un retour des besoins et de remarques permettraient de perfectionner l'outil qui est tout à fait utilisable et fonctionnel à l'heure actuelle.

# A. Annexes

## A.1 Structures de la base de données



Figure 4: Architecture de la base d'items

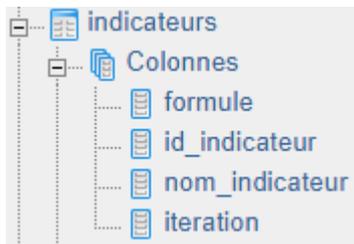


Figure 5 : Architecture de la table des indicateurs

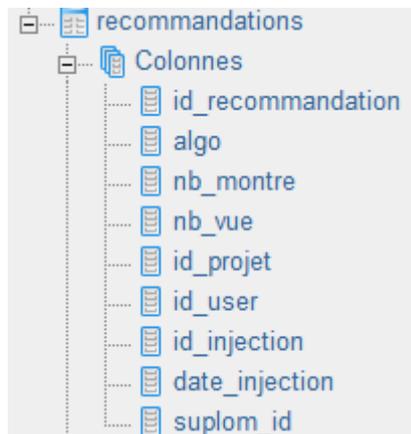


Figure 6 Architecture de la table de recommandations

## A.2 Mode opératoire

Deux grands choix s'offre à nous :

- 1) L'analyse d'un indicateur
- 2) Une analyse complète

### 1) L'analyse d'un indicateur

Ce choix se sépare en trois étapes :

- Sélection d'un indicateur
- Choix des paramètres d'affinage
- Analyser l'indicateur et le retour

## Première étape : Sélection de l'indicateur

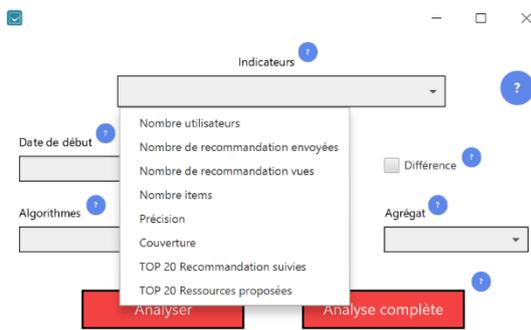


Figure 7: Interface du logiciel après avoir appuyé sur la barre des indicateurs

Après avoir appuyé sur la barre des indicateurs, les différents choix possibles, représentés par les différents indicateurs mis en place, se déroulent.

Ce fonctionnement est le même pour toutes les autres barres défilantes.

Chaque indicateur est décrit dans ce document ou bien dans le bouton « ? » situé au-dessus de la barre.

Dans le cas où vous ne sélectionnez pas un message d'erreur apparaît vous signalant de sélectionner un indicateur.

## - Deuxième étape : Choix des paramètres d'affinage

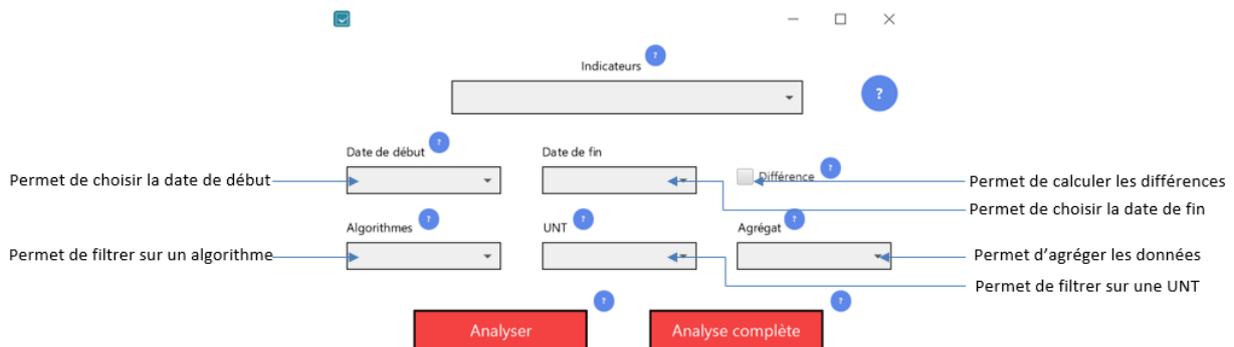


Figure 8: Explication des paramètres d'affinage

Pour le filtre des dates, cela permet de cadrer les résultats au niveau des dates. S'il n'y a pas de choix de date de début, cela prendra la première date possible, pour la date de fin, cela sera la dernière.

Trier sur un algorithme en particulier ou sur une UNT permet de sélectionner seulement notre choix dans la requête.

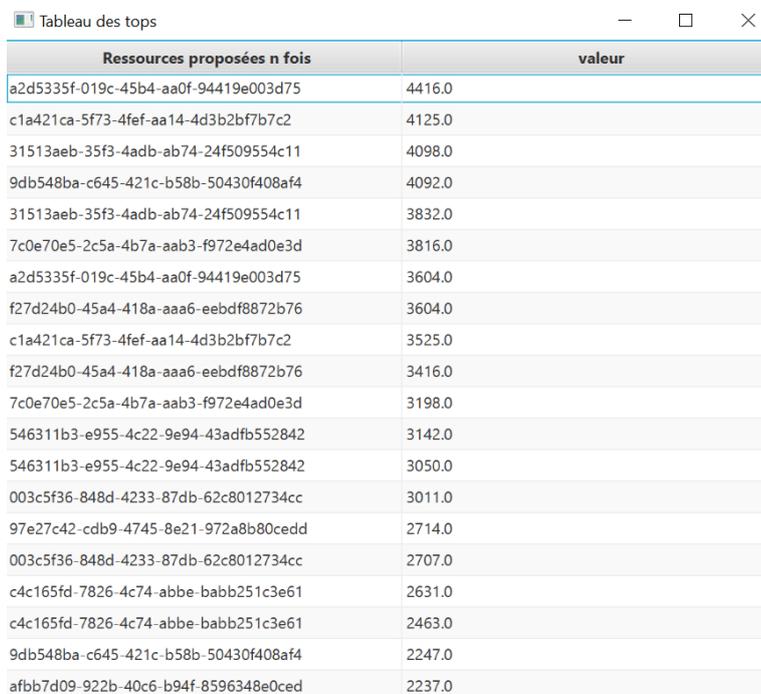
L'agrégat permet d'agréger nos résultats. Par exemple en sélectionnant comme indicateur nombre d'items avec un agrégat langue, j'aurais comme résultat le nombre de ressources par langue.

Différence permet d'avoir une courbe d'évolution, par exemple, toujours avec le même indicateur, si j'ai 100 items à la première injection, 120 à la deuxième et 125 à la troisième j'aurai une courbe de progression de 0, +20, +5.

### - Troisième étape : Analyser l'indicateur et le retour

Pour cela il suffit d'appuyer sur le bouton « Analyser ».

Deux retours possibles, un tableau pour les indicateurs commençant par « TOP ».



Ressources proposées n fois	valeur
a2d5335f-019c-45b4-aa0f-94419e003d75	4416.0
c1a421ca-5f73-4fef-aa14-4d3b2bf7b7c2	4125.0
31513aeb-35f3-4adb-ab74-24f509554c11	4098.0
9db548ba-c645-421c-b58b-50430f408af4	4092.0
31513aeb-35f3-4adb-ab74-24f509554c11	3832.0
7c0e70e5-2c5a-4b7a-aab3-f972e4ad0e3d	3816.0
a2d5335f-019c-45b4-aa0f-94419e003d75	3604.0
f27d24b0-45a4-418a-aaa6-eebdf8872b76	3604.0
c1a421ca-5f73-4fef-aa14-4d3b2bf7b7c2	3525.0
f27d24b0-45a4-418a-aaa6-eebdf8872b76	3416.0
7c0e70e5-2c5a-4b7a-aab3-f972e4ad0e3d	3198.0
546311b3-e955-4c22-9e94-43adfb552842	3142.0
546311b3-e955-4c22-9e94-43adfb552842	3050.0
003c5f36-848d-4233-87db-62c8012734cc	3011.0
97e27c42-cdb9-4745-8e21-972a8b80cedd	2714.0
003c5f36-848d-4233-87db-62c8012734cc	2707.0
c4c165fd-7826-4c74-abbe-babb251c3e61	2631.0
c4c165fd-7826-4c74-abbe-babb251c3e61	2463.0
9db548ba-c645-421c-b58b-50430f408af4	2247.0
afbb7d09-922b-40c6-b94f-8596348e0ced	2237.0

Figure 9: Résultat top 20 des ressources proposées

Pour les autres, le retour se fait sous forme d'une courbe.

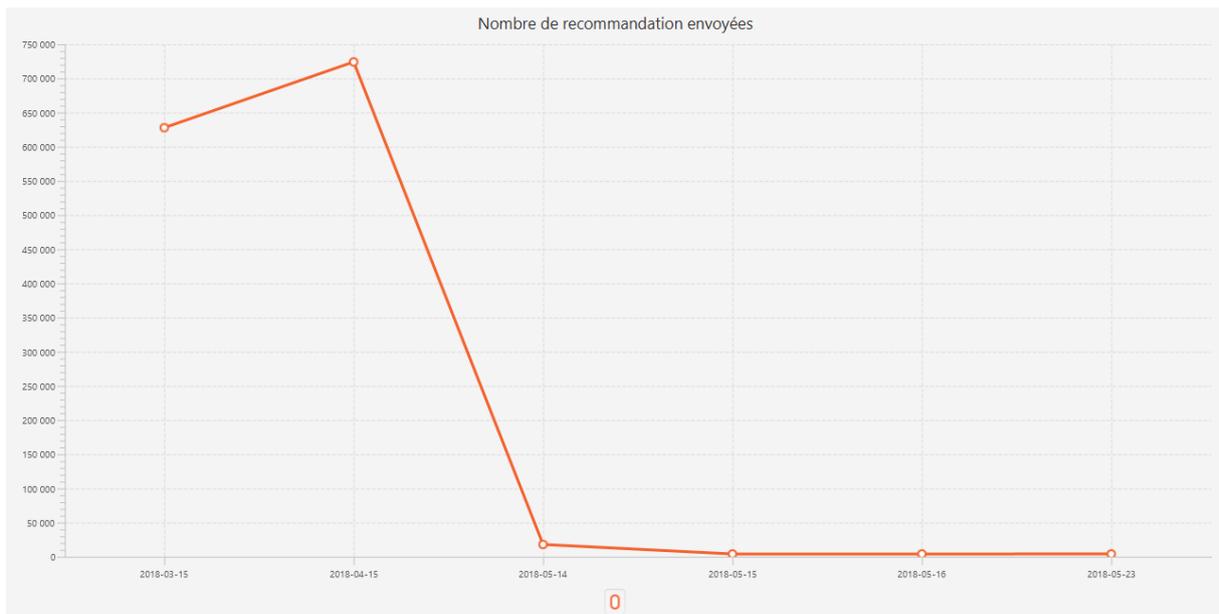


Figure 10: Graphique du nombre de recommandations envoyées date par date

Le même résultat en utilisant le bouton différence.

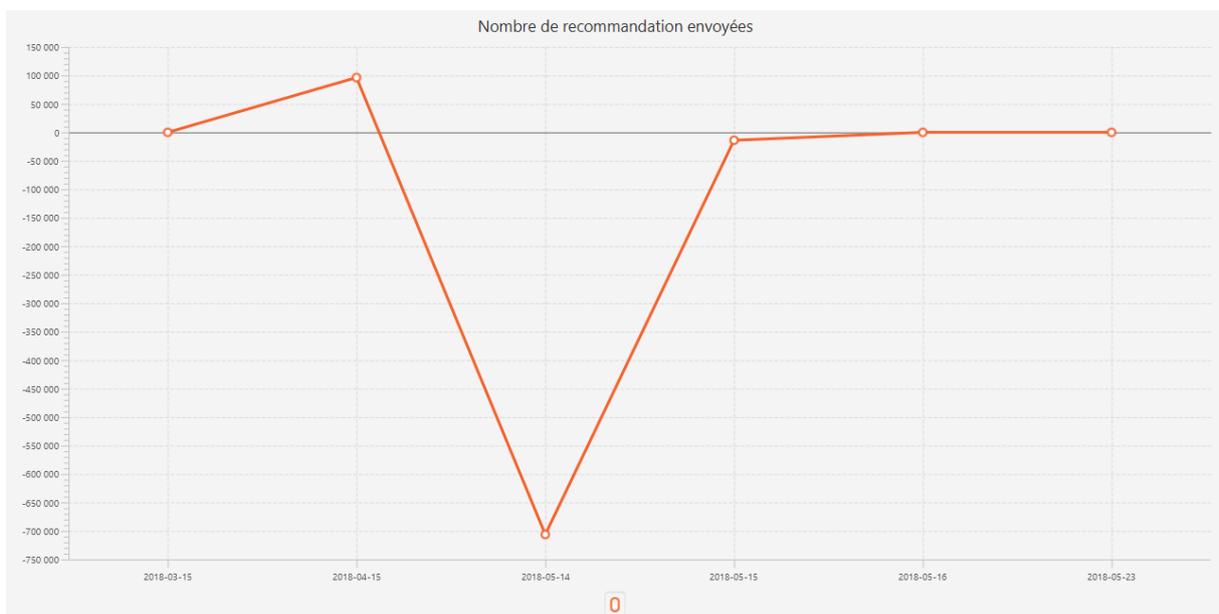


Figure 11: Graphique du nombre de recommandations envoyées date par date avec option soustraction

Dans chacun des deux cas, avec le bouton droit de la souris il est possible d'enregistrer sous forme .png le tableau ou la courbe.

## 2) Une analyse complète

Pour ce choix, il suffit de suivre la deuxième étape décrite ci-dessus et d'appuyer sur « Analyse complète » et un fichier au format .docx sera généré screens de tous les indicateurs en fonction des paramètres choisis.

### A.3 Bibliographie

[Wikipédia2016] :

[https://fr.wikipedia.org/wiki/Syst%C3%A8me\\_de\\_recommandation](https://fr.wikipedia.org/wiki/Syst%C3%A8me_de_recommandation), 2016

[Mathieu2014] :

Mathieu, <https://www.podcastscience.fm/dossiers/2012/04/25/les-algorithmes-de-recommandation/>, 2014

[Arnaud De Myttenaere 2016] : Arnaud De Myttenaere, 2016, Évaluation hors-ligne d'un modèle prédictif : application aux algorithmes de recommandation et à la minimisation de l'erreur relative moyenne, page 1, <https://tel.archives-ouvertes.fr/tel-01395290/document>

[Patrick Peccatte 2007] : Patrick Peccatte, 2007, Soft Expérience, <http://peccatte.karefil.com/Software/Metadata.htm>

[N. Friburger 2009] : N. Friburger, 2009, Métriques et TAL, p11

[Lathia 2010] : Lathia N. K., 2010, Temporal diversity in recommender systems.

[Équipe KIWI 2013] : Sylvain Castagnos, Armelle Brun, Anne Boyer, Utilité et perception de la diversité dans les systèmes de recommandation. CORIA 2013 - 10ème Conférence en Recherche d'Information et Applications, avril 2013, <https://hal.inria.fr/hal-00783329/document>

[G. Shani & A. Gunawardana 2010] : Guy Shani et Asela Gunawardana, Evaluation Recommendation System, 2010