

LA GESTION DE PROCESSUS D'AMÉLIORATION DU SYSTÈME GRC (GESTION DE LA RELATION CLIENT) D'ENTREPRISE ET L'OPTIMISATION DE L'ALGORITHME DE RECHERCHE DES DONNEES

Mykyta KUDINOV

*Université nationale d'économie de Kharkiv Simon Kuznets, Ukraine, Université Lumière Lyon 2, France,
e-mail: nikita.kudinov@gmail.com*

Cet article présente les résultats de deux projets qui ont été effectués au sein de l'entreprise Lamudi GmbH. La première partie décrit l'utilisation du logiciel MailSolution et l'automatisation obtenue en utilisant l'API XML, dans le but de diminuer la quantité de travail manuel. La deuxième partie décrit le statu quo de Stack de la technologie existante utilisé pour la recherche à travers la base de données et montre la solution pour les tests de logique du moteur de recommandation qui pourrait être utilisé après le test mis en œuvre.

Mots clés: *commerce électronique (e-commerce), gestion de projets, Apache Solr, système de recommandation, CRM, MailSolution*

1. Introduction

Presque tous les sites web e-commerce modernes, offrant un grand nombre de produits variés, pratiquent la gestion de la relation clientèle (CRM). Cette pratique permet de gérer les relations avec les clients par l'envoi de courriels ou SMS, mais aussi de communiquer avec les abonnés du bulletin hebdomadaire ou encore d'utiliser les e-mails pour la vérification des personnes au moment de leur enregistrement sur le site. C'est aussi un bon moyen de promouvoir les nouvelles offres, et d'informer les utilisateurs de tout changement concernant l'utilisation de la plateforme.

Toutes les entreprises (sociétés filles) créent par l'incubateur Rocket Internet GmbH bénéficient d'un soutien conséquent afin de développer le nouveau produit. Bien que toutes les entreprises de Rocket Internet soient différentes, nombres d'entre elles sont des e-commerces et ont une stratégie électronique similaire. Abonné à une stratégie de service CRM unique, Rocket Internet donne accès au même service pour de nombreuses entreprises. C'est toujours techniquement et financièrement raisonnable. Mais même si une entreprise a le soutien de l'accélérateur, si l'interface utilisateur du logiciel respectif n'est pas efficace dans le sens de la gestion de plusieurs comptes de pays ayant des préférences différentes, il est encore possible de l'améliorer. En utilisant l'API officiel, il est possible pour chaque entreprise de créer sa propre interface pour gérer des campagnes d'emailing. Cette amélioration est décrite plus loin. La deuxième partie contient l'explication de la manière dont Lamudi utilise un logiciel OpenSource pour afficher des résultats de recherche pertinents. Le système actuel a aussi été développé au sein de Rocket Internet et ensuite utilisé dans plusieurs entreprises, mais avec des personnalisations. Pour Carmudi et Lamudi – deux entreprises qui représentaient initialement une seule entreprise, une logique personnalisée pour la pertinence des recherches a été mise au point. La raison pour laquelle les deux entreprises utilisent Apache Solr comme moteur de recherche Open Source, est que la logique initiale a été écrite dans Solr bloc de fonction de recherche. Cela signifie qu'au lieu d'utiliser la version standard de calcul de la pertinence des entités dans la base de données, il est possible de modifier la logique. Mais il est difficile d'utiliser Solr comme moteur de recommandation parce que Solr calcule la pertinence des recherches basée uniquement sur les caractéristiques des objets et n'inclut pas les données de préférences de l'utilisateur comme l'histoire de demandes de recherche, et ne comprend pas non plus les éventuels changements dans les préférences. Aussi pour appliquer les modifications à l'expérience plus personnalisée, la structure des tableaux de base de données doit changer. Par exemple si le calcul de pertinence de la recherche devrait inclure des vues par certaine période de temps ou conduit par certaine période de temps ou pire encore, des valeurs moyennes, elles peuvent être calculées le même temps ou pré-calculée et inscrite dans la base de données. Comment surmonter ces obstacles est décrit ci-dessous.

2. Description de l'UMS MailSolution d'Experian

Avec UMS MailSolution d'Experian il est possible d'effectuer des opérations principales de campagnes e-mail et les notifications par SMS. Ici vous pouvez voir la structure principale de la logique MailSolution (Figure 1):

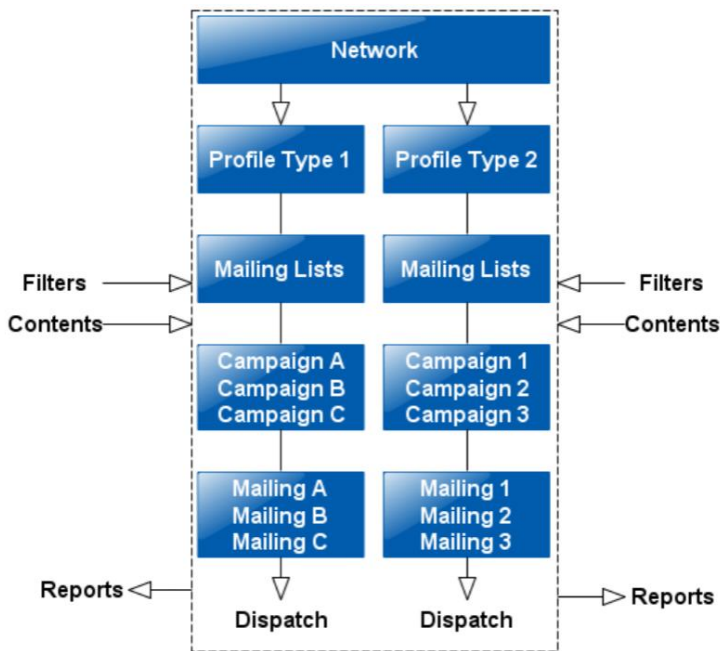


Figure 1. Schéma principal de MailSolution

Comme vous pouvez le voir, toute la gestion des e-mails est affectée au profil spécifique. Par exemple, Lamudi est présent dans plus de 30 pays à travers le monde et cela signifie qu'ils utilisent plus de 30 types de profil. D'un certain point de vue, il est nécessaire de séparer toutes les opérations pour les différents pays, car ils utilisent différentes langues, différents types d'e-mails et différentes campagnes promotionnelles sont expédiées dans un contexte différent. Donc, l'utilisation de différents profils est plus facile. Mais de l'autre côté, il est plus difficile d'effectuer des changements dans le contenu ou la structure. Par exemple, lorsque l'administration de l'entreprise avec le Gestionnaire de produit (Product Manager) a décidé de modifier la conception et la structure de la plateforme, ce serait beaucoup plus difficile à mettre en œuvre tous les changements sont acceptés dans l'interface standard MailSolution. Imaginez combien de temps il faudrait déposer pour changer emails pour tous les pays. Chaque pays où la société représentée doit avoir au moins minimal quantité d'e-mails. Parmi eux se trouvent des emails transactionnels (confirmation d'inscription, la réussite de l'enregistrement, l'enregistrement des souscriptions et succès, des alertes par courriel des nouveaux biens similaires par le dernier filtre utilisé dans la recherche, téléchargement réussi d'une «fiche», la confirmation de message délivré au propriétaire du bien et beaucoup d'autres) compter environ 22 types au total et des e-mails bulletin, plus de deux types: l'un pour le vendeur et autre pour l'acheteur d'un. En multipliant par 24 le nombre de pays 30 nous obtenons un nombre minimum de changements manuels dans le système, égale à 720, pour effectuer des changements de style ancien e-mail à nouveau mise en page. Si un e-mail (après la conception est prêt) prend environ 15 minutes de mise en place, écrit le code du pays ISO, contrôle de Google Analytics paramètres de suivi etc. (sans compter les erreurs), il faudrait $720 * 15 = 10800$ minutes ou 180 heures ou 22,5 jours de travail (en supposant que 8 heures par jour). Cela signifie que le système doit certainement avoir procédé qui permettra de faire des changements dans un délai aussi court que possible automatisé. MailSolution dispose d'une documentation sur l'API XML qui permet de construire ces outils d'aide par entreprise elle-même [1].

3. Proposition de schéma d'automatisation pour CRM

L'interface XML pour MailSolution soutient l'entreprise en automatisant le travail. Vous pouvez facilement gérer de grandes quantités de données et réaliser des importations de lots. Par le biais de ce qu'on appelle les demandes, vous soumettez vos données à MailSolution. Vous pouvez soumettre une demande HTTPS à <https://xml.fagms.net/cgi-bin/xmlinterface> ou télécharger votre fichier XML à un compte SFTP à Experian. MailSolution traite les données soumises et renvoie une réponse. La réponse est soit affichée immédiatement dans le navigateur, retournée par e-mail ou HTTPS ou enregistré dans votre compte SFTP.

Les données présentées contiennent, entre autres, un ou plusieurs éléments de commande. Chaque élément de commande est associé à un certain processus exécuté par MailSolution. Par exemple, vous pouvez importer un grand nombre de profils de membres ou mis en place Mailings pour votre campagne e-mail.

Le concept sous-jacent de l'interface est XML (eXtensible Markup Language). Ses composants essentiels sont

appelés éléments (aussi connu comme “tags”), qui sont définis par un nom et une valeur. Un élément peut contenir d'autres éléments. Cet élément est ensuite transmis à l'élément en tant que parent, tandis que les éléments qui y sont contenus sont connus comme des éléments enfants. La structure d'un fichier XML est définie par le type de Document Type Definition (DTD). Une DTD définit le contenu, à savoir les éléments autorisés et les valeurs d'un fichier XML, ainsi que l'ordre dans lequel les éléments doivent apparaître dans le fichier XML. Si vous ne respectez pas strictement à l'ordre spécifié par la DTD, votre demande ne peut être traitée et vous recevrez un message d'erreur. La DTD pour l'interface XML MailSolution est disponible sous <https://xml.fagms.net/XMLInterface.dtd>. Une méthode pour importer des données dans MailSolution soumet une demande HTTPS <https://xml.fagms.net/cgi-bin/xmlinterface> aide d'un script par Experian ou votre propre script. Lorsque vous soumettez votre demande via HTTPS, il y a une particularité que vous devez garder à l'esprit. Si votre demande ne contient que le fichier d'importation XML (par exemple import.xml nommé), vous devez entrer le type de contenu suivant dans le champ d'en-tête HTTP: Content-Type: text / xml. Si votre requête contient également d'autres fichiers (par exemple, les pièces jointes et images), le type de contenu doit être multipart/form-data. Les fichiers individuels de votre demande sont ensuite séparés et ont chacun leur propre type de contenu. Le nom du type de contenu qui contient le fichier import.xml doit être importation. En suivant des règles définies d'un script a été créé pour gérer des campagnes d'email. Image suivante montre les flux de données et l'architecture du système.

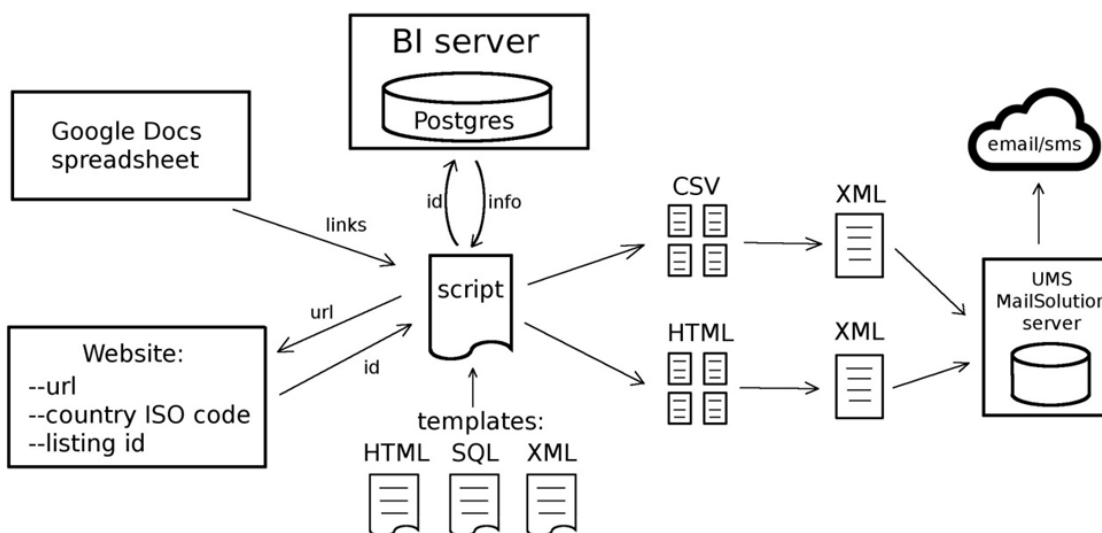


Figure 2. Schéma principal d'automatization UMS MailSolution

Comme vous pouvez le voir sur l'image (Figure 2), le processus de création des instructions pour MailSolution à l'aide de fichiers XML est le suivant :

1. S'il y a besoin d'avoir une entrée de données des équipes locales, ils peuvent utiliser Google Documents Spreadsheet pour remplir les cellules avec des liens ou du texte. Il est généralement utilisé pour les bulletins. Il est beaucoup plus facile d'utiliser des URL pour les questionnaires que de remplir toutes les entrées de données manuellement.
2. Si urls de Google Docs sont utilisés) En obtenant réponse de sites Web en utilisant HTTP GET demander au script analyse des données de réponse pour obtenir les informations nécessaires.
3. Utilisation de la liste des identifiants données et les identifiants de l'agent et des articles, script se connecte à la base de données DataWarehouse (DWH) et obtient toutes les informations sur les agents, des listes et des articles pour l'utiliser comme contenu des bulletins d'information ou d'autres emails.
4. Lorsque l'étape précédente est terminée, le script utilise HTML et TXT templates pour créer des bulletins et autres e-mails avec un contenu spécifique pour chaque pays. Cette séparation permet de diminuer la quantité de travail en apportant des changements qui ne sont pas en 720 e-mails, mais seulement 24. Également en divisant le contenu des e-mails en différentes parties, “header”, “footer” de page et le “body”, il est possible de réduire la quantité de travail encore plus si l'utilisation même en-tête et pied de page dans de nombreux courriels.
5. Utilisation d'une connexion à DWH et le contenu des modèles le script crée des listes d'e-mails à qui la campagne d'email se réfère (abonnés à la newsletter pour ex.) Et les fichiers HTML avec le contenu que l'on verra dans

le courrier électronique entrant boîte. Le contenu de tous les fichiers créés (csv, html, txt) sera utilisé comme valeurs pour les fichiers XML outre s'appuyer sur des modèles XML. Chaque modèle XML correspond à une instruction spécifique à UMS MailSolution. Par exemple on télécharger la liste à jour des abonnés. Un autre ensemble nouveau contenu html pour e-mails et troisième effectue la configuration pour l'envoi de date donnée et l'heure locale et pour des pays donnés. Un autre envoi des instructions à envoyer email de test. Tous les autres MailSolution des opérations exercées par lui-même.

Dans la suite de mon travail le script a été créé. Il est extensible, prend en charge différents cas, comme la vérification de pays donnés uniquement, la séparation des processus pour vérifier les données ou tout faire seule étape. Il peut également être intégré dans l'outil local avec l'aide de l'interface web conviviale. Mais il y a des entreprises qui créent des logiciels plus intelligents et complexe comme un service à offrir l'intégration de la gestion des données et d'appuyer la gestion de CRM avec les dernières technologies.

1. Description du moteur de recherche et l'état actuel du système

Pour les plateformes Web qui utilisent la grande quantité de données et de créer des catalogues en ligne, il est raisonnable que les utilisateurs ne seront pas en mesure d'embrasser toutes les données représentées. C'est pourquoi les moteurs de recherche et les filtres sont utilisés. Chez Lamudi et Carmudi le moteur de recherche Open Source Apache Solr appelé est utilisé. Il présente de nombreux avantages à utiliser pour la recherche, mais a également des restrictions en matière de personnalisation ou de construction des recommandations.

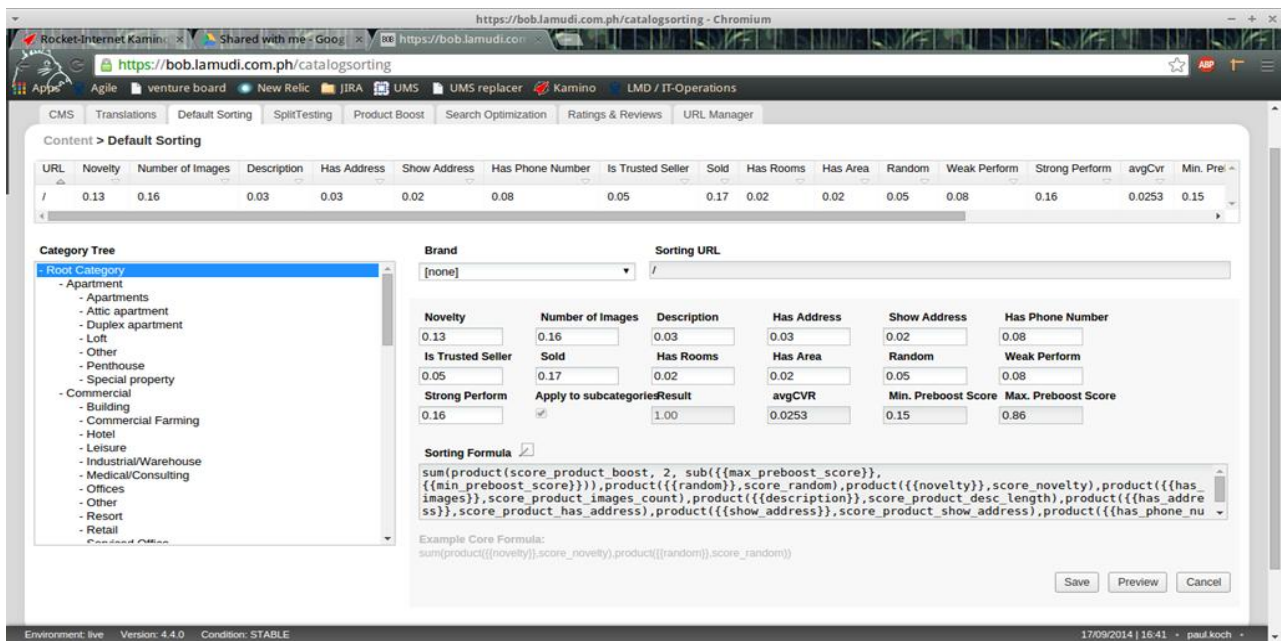
Solr est ultra-rapide OpenSource plateforme de recherche populaire du projet Apache Lucene. Ses caractéristiques principales sont puissant moteur de recherche en texte intégral, frappé en évidence, la recherche à facettes, près de l'indexation en temps réel, le regroupement dynamique, l'intégration de bases de données, riche de documents (par exemple, Word, PDF) la manipulation, et la recherche géospatiale. Solr est très fiable, évolutive et tolérante aux pannes et fournir distribué indexation, la réplication et l'interrogation à charge équilibrée, le basculement automatique et de récupération, la configuration centralisée et plus. Solr permet les fonctions de recherche et de navigation d'un grand nombre des plus grands sites Internet du monde.

Solr est écrit en Java et fonctionne de façon autonome en texte intégral serveur de recherche au sein d'un conteneur de servlet comme Jetty. Solr utilise la bibliothèque de recherche Lucene Java à sa base pour l'indexation de texte intégral et la recherche, et RESTful comme HTTP / XML et JSON API qui font qu'il est facile à utiliser à partir de pratiquement n'importe quel langage de programmation. Configuration externe puissant Solr permet de l'adapter à presque n'importe quel type d'application sans codage Java, et il possède une architecture vaste plugin lors de la personnalisation plus avancée est nécessaire [2].

Voici une liste non exhaustive des fonctionnalités utilisées par Solr:

- Un schéma réel des données, avec des types numériques, des champs dynamiques, clés uniques
- Extensions puissants à la Lucene Query Language
- Recherche à facettes et filtrage
- Geospatial Recherche avec le soutien de plusieurs points par document et polygones géo
- Avancée, l'analyse de texte configurable
- Hautement configurable et extensible utilisateur Caching
- Optimisation des performances
- Configuration externe via XML
- Une interface d'administration basée sur AJAX
- Logging contrôlable
- Rapide en temps quasi réel indexation incrémentielle et la réplication de l'indice
- Hautement évolutive recherche distribuée avec index fragmenté sur plusieurs hôtes
- JSON, XML, CSV / délimité-texte et aux formats binaires de mise à jour
- Une façon simple de tirer des données des bases de données et des fichiers XML à partir des sources de disques et HTTP locales
- Rich document analyse syntaxique et de l'indexation (PDF, Word, HTML, etc.) en utilisant Apache Tika
- Intégration Apache UIMA pour l'extraction de métadonnées configurable
- Plusieurs indices de recherche

1. Développement de moteur de recommandation



Spécialement pour les sites d'annonces (Carmudi et Lamudi ventures de Rocket Internet) la logique pour le calcul de la pertinence a été créée. La logique de recherche Solr peut être modifiée avec la formule personnalisée écrite en format spécial qui Solr comprends (Figure 3). Exécuter un serveur Solr possède une interface Web aussi facile à utiliser. Vous pouvez modifier les poids de la formule ou de modifier ou exécuter des requêtes d'une manière simple. Pour la raison que l'objet de catalogue («listing») est une propriété de l'immobilier, la durée de vie d'un utilisateur peut être décrit par la suite. D'abord, il découvre la plateforme, puis tente d'utiliser la recherche pour trouver la propriété de son intérêt. Dans le cas où il trouve la plateforme utile, il peut enregistrer des résultats de recherche, puis revenir plus tard pour continuer la recherche. Il passe un peu de temps sur la recherche pour obtenir des données sur les propriétés disponibles et dans quelques semaines ou même des mois accepte enfin de finaliser l'accord avec un agent qui inscrit la propriété sur la plateforme. Si nous voulons construire un moteur de recommandation intelligent qui s'adapte parfaitement les deux côtés du marché et si nous utilisons ce modèle de comportement de l'utilisateur, alors nous devrions envisager d'autres hypothèses. Le premier est le problème de la séparation de l'intérêt des acheteurs et l'intention du vendeur. L'acheteur aimerait trouver un appartement qui correspond exactement à ses besoins. Il s'agit d'une tâche classique de moteur de recommandation. Mais le vendeur, un agent, a un point de vue différent. D'habitude, il a beaucoup plus d'appartements (200-300 propriétés en moyenne) et il voit la plateforme comme un canal de distribution. Pour avoir une comparaison avec d'autres canaux de marketing de distribution, il utilise ces indicateurs comme le nombre de «leads» par période de temps. Par «leads», nous entendons un utilisateur qui a non seulement jeté un coup d'œil sur certains des biens immobiliers, mais qui a aussi contacté le propriétaire par le biais du bouton "show telephone number", en cliquant ou en lui envoyant un message interne.

Ensuite, en utilisant le rapport de l'argent dépensé sur le canal de distribution de nombre de leads, il peut calculer le prix d'acquisition d'un lead unique. De toute évidence, il trouve beaucoup plus intéressant de dépenser moins et obtenir plus de prospects par période de temps donnée [3, 4].

Figure 3. Capture d'écran de l'interface Solr

Le problème caché ici est que toutes les propriétés diffèrent par la qualité. Et parfois, la quantité d'information utilisée pour décrire une propriété ajoute à la différence de qualité ou le rend moins intéressant. C'est pourquoi de fonctionner avec un taux de conversion supposée à une propriété, nous devons utiliser les vues et conduit par période de temps fixe. Il nous permet de voir quelles sont les propriétés ont plus de potentiel pour obtenir plus de prospects. Ces types de propriétés sont dits «très performants» et doit être «boosted» au début de la liste de résultats de recherche. Et cette action augmente la probabilité de conversion parce que les premières pages sont visionnées plus souvent que suivant. Un tel comportement peut être illustrée par l'histogramme classique donné par résultats de recherche Google analyse. Si vous calculez CTR (taux de clics), le trafic en fonction de la position d'une entité donnée dans la liste, vous obtiendrez environ des taux: la première page obtient 70-80% du trafic, alors chaque page suivante obtient le trafic de moins en moins (Figure 4). La même chose peut être dit à propos de premières positions de page - devient supérieure de plus qu'en bas de la page. Ces résultats s'intègrent parfaitement la théorie sur le comportement de l'utilisateur quand il

cherche quelque chose dans un énorme catalogue et lorsque la tâche de présentateur est de montrer les entités les plus pertinentes. Mais en cas de Lamudi telle tâche devrait être distraire à l'évaluation de la qualité et dans l'intention de pousser propriétés des agents sélectionnés avec une plus grande priorité que d'autres [5].

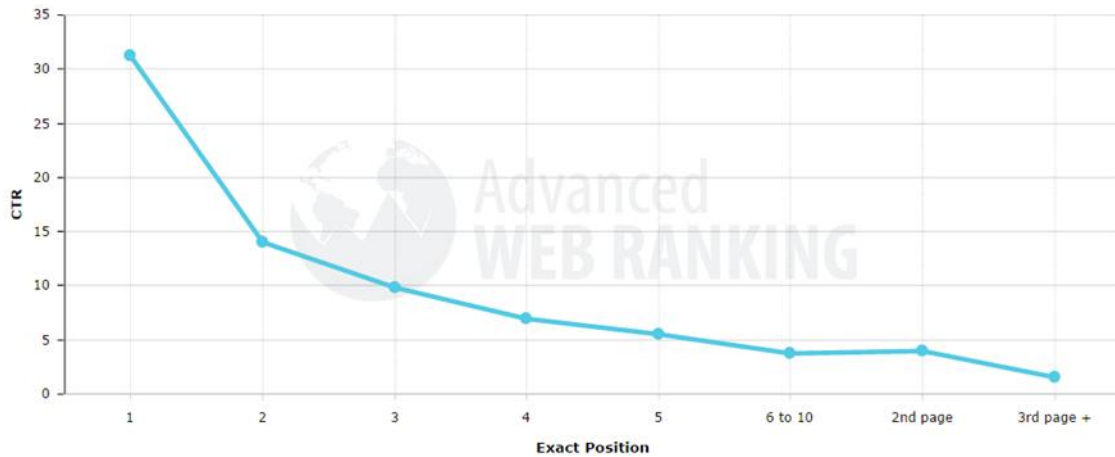


Figure 4. Distribution disproportionnée de CTR par page

L'administration de Lamudi a décidé de construire un prototype de moteur de recommandation premier et tester la façon dont il fonctionne avant la mise en œuvre de la production. Sur le schéma, vous pouvez voir l'architecture du système qui a été construit à des fins de test (Figure 5).

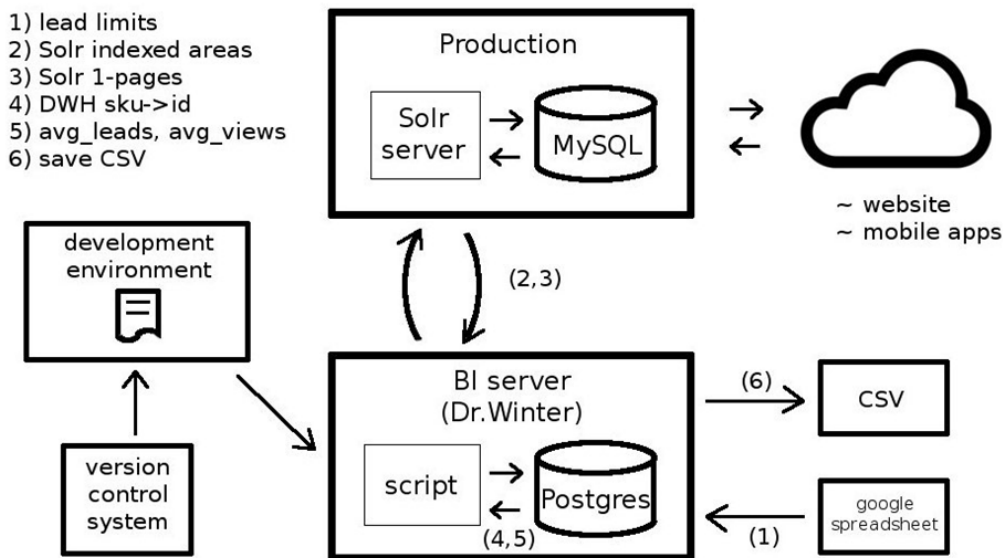


Figure 5. Schéma principal de prototype de moteur de recommandation

Il existe deux types les plus populaires de recommandation algorithmes : filtrage de collaboration et de filtrage basé sur l'article. La raison pour laquelle il existe une architecture complexe derrière le filtrage de collaboration qui comprend l'analyse des utilisateurs, de leur histoire de recherche et des notes qu'ils ont laissé et faible corrélation possible entre les préférences des utilisateurs, il est plus fiable pour construire un moteur de filtrage basé sur l'article qui utilise les caractéristiques des biens immobiliers.

Exigences formelles pour le système sont:

1. donner un sentiment général d'une qualité d'un bien ;
2. pousser automatiquement la plupart des propriétés convertibles ;
3. donner un moyen de manuellement les résultats tune par des équipes locales

Points d'attention supplémentaire:

4. éviter de renforcer automatique de mauvais résultats ;

5. nous ne pouvons pas prévoir le taux de conversion avec la qualité fiable, c'est pourquoi nous utilisons les exigences de base pour le téléchargement pour maintenir le niveau de qualité moyen.

Trier note calcul comprend trois points principaux:

6. La qualité de l'inscription de base: listes de fantaisie ne convertissent pas mieux que les simples, mais une norme minimale de qualité est important pour la prime la convivialité du site ;

7. Performances d'affichage: 1 seuil d'un rendement élevé (des annonces attrayantes = haute CVR), 1 seuil de mauvais résultats (faible CVR). Rien entre les deux, afin d'éviter le cercle vicieux (malchance dans les premiers jours => downboost => malchance pour toujours) ;

8. Les considérations commerciales: une grande flexibilité d'apporter manuellement les annonces au sommet. Pour la logique donnée une formule a été élaborée (Figure 6).

$$s = \vec{weights} * \left(\text{Boost}, \text{Random}, 2^{-(\text{ageDays}/30)^2}, 1 - 3^{-\text{numImages}}, 1 - 3^{-(\text{charsInDescription}/100)}, 1(\text{yes/no}), \dots, -1(\text{views} > 150 \wedge \text{leads} = 0), 1(\text{CVR} > 2 * \text{avg}(\text{CVR})) \right)^T$$

Figure 6. Les parties de « boosting » logique.

Le paramètre de boost comme un paramètre de propriété peut être mise à jour et devrait être utilisé pour faire des corrections manuelles dans la logique de tri. Précisément cette partie qui est utilisé par le script.

1. Conclusions

Le développement d'un outil utilisable nécessite pas seulement la qualité et devrait envisager extensibilité pour de plus amples maintien ou le développement, mais aussi des contraintes de temps dicté par la stratégie de l'entreprise. Parfois, la "meilleure" solution ne correspond pas aux exigences de courte durée de développement. Mais même si cela signifie seulement que chaque projet devrait contenir aussi des risques estimation et une meilleure gestion des temps d'intégrer possibilité de produits de construction de haute qualité tout en battant des restrictions de temps.

Références

1. Experian UMS MailSolution [Ressource Electronique] – Mode d'accès: <http://www.experian.de/>
2. Apache Solr documentation [Ressource Electronique] – Mode d'accès: <http://lucene.apache.org/solr/>
3. Recommender System introduction [Ressource Electronique] – Mode d'accès: http://en.wikipedia.org/wiki/Recommender_system
4. Recommender Systems [Ressource Electronique] – Mode d'accès: <http://www.forbes.com/sites/lutzfinger/2014/09/02/recommendation-engines-the-reason-why-we-love-big-data/>
5. Click-through rate (CTR) on google pages [Ressource Electronique] – Mode d'accès: <http://moz.com/blog/google-organic-click-through-rates-in-2014>

Sous la supervision de (Під керівництвом):

Jean-Hugues Chauchat (PhD, HdR, professeur, ERIC lab)

Орлов П.А. (д.е.н. професор, зав. кафедри економіки та маркетингу)

Керівник з іноземної мови ст. викладач кафедри іноземних мов та перекладу Безугла І.В.