Analyse comparative de modèles statistique et d'apprentissage profond pour l'étude du parcours utilisateur sur un site internet

Mathilde Gorieu

Ingénieur Agronome
Parcours Data Science pour la biologie

Rapport de stage présenté par :

Mathilde Gorieu

Sujet :

Analyse comparative de modèles statistique et d'apprentissage profond pour l'étude du parcours utilisateur sur un site internet

Maîtres de stage :
Emeric Dynomant et Arnaud Desfeux

Année : 2017-2018
Promotion 166

OmicX
72 rue de la république
76140 Le-Petit-Quevilly, France
☎ 02 79 02 75 22
www.omictools.com

Maîtres de stage :

Emeric Dynomant

Arnaud Desfeux

Remerciements

Je souhaite remercier les personnes travaillant à OmicX, pour leur bienveillance, leur joie de vivre et leur simplicité. Vous m'avez aidée à prendre conscience de l'importance d'une ambiance de travail décontractée et que l'échange et la communication sont essentiels au bon fonctionnement d'une équipe. Merci en particulier à Tiphaine et Léa pour la relecture de ce travail.

Merci à l'équipe de Data Science pour les échanges riches et la confiance qu'ils m'ont accordée durant ces six mois.

En particulier, merci Emeric. Tu mérites ton propre paragraphe. Pour le temps que tu m'as accordée, pour les conseils précieux en Python, pour l'écoute toujours attentive et bienveillante, pour la relecture de ce rapport et pour simplement tous les moments passés.

Je souhaite aussi remercier, sans pouvoir toutes les citées, les personnes m'ayant permis de construire ma vie professionnelle. Personne, pas même un algorithme je crois, n'aurait pu prédire mon métier actuel il y a six ans après mon baccalauréat. Merci à mes encadrants précédants, Pierre et Pascal, merci à mes professeurs de classe préparatoire et d'école d'ingénieur. Vos conseils sont précieux.

Enfin, la construction d'une vie professionnelle n'est pas possible sans un entourage présent et là encore bienveillant. Merci à mes parents et à mes frères pour leur soutien, à mes colocataires pour leur accueil à Rouen et le bonheur qu'ils m'ont apporté. À mon équipe pour m'avoir accueillie pour une fin de saison mémorable.

Même si je ne sais pas ce que me réserve l'avenir, je suis persuadée d'avoir trouver ma voie. À moi maintenant d'aller explorer les sentiers.

Résumé / Abstract

La visibilité d'un site internet est mesurée par des indicateurs précis comme le nombre de visites et le nombre de pages vues par visite. Or les systèmes de mesure actuelle des visites sur un site internet font appel à des cookies, qui peuvent être bloqués par l'utilisateur. On s'est donc demandé si cette moyenne était représentative du trafic réel du site internet. Une étude des données enregistrées automatiquement sur un serveur (appelé données de logs apache) est effectuée. Au fur et à mesure de l'exploitation, d'autres enjeux sont apparus et ont été abordés. Ce travail consistera donc en trois points clés. Premièrement, un filtre est appliqué afin de récupérer des données de qualité et de permettre un stockage dans une base de données. Ensuite, une interface de visualisation a été construite pour obtenir des informations utiles à l'équipe marketing et à la direction en lien avec les visiteurs du site. La comparaison avec d'autres interfaces est alors effectuée. Pour finir, une comparaison de modèle est réalisée entre un modèle classique statistique de graphe et un modèle de Réseau Récurrent à Mémoire à Court Terme Résiduelle afin de prédire une page intéressante à proposer aux visiteurs du site. Ce modèle permettrait ainsi d'augmenter le nombre de pages vues par visite. Les deux modèles donnent au final des prédictions similaires, avec une précision autour de 30 %. Cette étude pourra être reproduite et utilisée pour tout site internet ayant accès aux données de leur serveur.

Mots clés : base de données, SQL, Prédiction, Graphe, apprentissage profond

A website's popularity can be evaluated using precise indicators such as the number of visits per day and the number of viewed webpages per visit. However, sytems that are used nowadays are mostly based on cookies, which can be blocked by users. We asked ourselves whether or not these analyses gave an accurate overview of a website's actual traffic. A study of data which is automatically saved onto a server (log apache data) was carried out. During the study, other challenges arose and were dealt with. The work presented here consists of three keys steps. Firstly, a filter was applied to the data so as to store clean data in a database. Then, a dashboard was put in place to give marketing teams and management access to information regarding user habits. A comparison between this dashboard and a classic cookie based dashboard was done. Finally, another comparison was done, this time, between a classic statistics model and a deep-learning LSTM model, to see if they could similarly predict how a page can interest users based on their navigation data. It turned out that both models had a very similar accuracy, roughly 30 %. The study can be reproduced by any website team with server data.

Keywords : Database, SQL, LSTM, deep-learning, prediction

Abréviations

DPD : Directeur de la Protection des Données

IP : Internet Protocol

noSQL : Not Only Stuctured Query Language

SQL : Stuctured Query Language

REGEX : EXpression REGulière

RGPD : Réglementation Générale de la Protection des Données

Table des matières

Table des matières

Introduction **1**

1 Des données de grande dimension à structurer pour les exploiter **3**
- 1.1 Des données à caractère personnel . . . 3
- 1.2 Des données particulières à nettoyer . . . 4
- 1.3 Construction d'une base SQL . . . 7

2 Exploration des données dans une interface graphique **10**
- 2.1 Analyse de la qualité des données . . . 10
- 2.2 Mise en place d'une interface interactive . . . 12
- 2.3 Comparaison avec d'autres outils de calcul d'indicateurs . . . 15

3 Comparaison de modèle de prédiction **17**
- 3.1 Un jeu de données basé sur les sessions des visiteurs . . . 17
- 3.2 Mise en place des modèles de prédiction . . . 18
  - 3.2.1 Modèle statistique . . . 18
  - 3.2.2 Modèle d'apprentissage profond . . . 19
- 3.3 Analyse de la précision des modèles . . . 24
  - 3.3.1 Précision automatique des modèles . . . 24
  - 3.3.2 Précision par validation manuelle . . . 25
  - 3.3.3 Précision en prenant en compte la structure du site internet . . . 26

Conclusion **28**

Bibliographie **30**

Annexes **32**

Table des figures

Table des figures

1 Capture d'écran de l'interface Google Analytics 2
2 Du pré-traitement des données à l'insertion en base 4
3 Comparaison globale du nombre de ligne avant et après l'application du filtre pour chaque fichier journalier 6
4 Boxplot de la durée d'exécution du script Filtre puis Insertion 8
5 Boxplot du nombre de pages vues par session 10
6 Nombre de requête sur le site par session 11
7 Visualisation des indicateurs de trafic du site 12
8 Visualisation des données utilisateurs 13
9 Visualisation des indicateurs dits globaux des données visiteurs 13
10 Diagramme de Sankey des données utilisateurs 14
11 Visualisation des indicateurs concernant les pages "catégories", "outils" et le moteur de recherche 14
12 Nombre moyen de visiteurs uniques par jour sur une durée d'un an 15
13 Explication de la construction du modèle à partir de 3 sessions fictives . . . 18
14 Représentation schématique du modèle LSTM 20
15 Représentation complète du modèle LSTM sour *TensorFlow* 21
16 Comparaison de la précision du modèle avec différents hyper-paramètres . 22
17 Comparaison de la précision du modèle avec différentes combinaisons de paramètres 23
18 Interface de notation manuel des modèles 25
19 Pop-up s'affichant sur le site 28

Table des tables

Liste des tableaux

1 Récapitulatif des données contenues dans la base SQL 9
2 Précision obtenue par validation automatique 24
3 Précision finale des différents modèles par validation manuelle 26
4 Précision par validation automatique prenant en compte la structure du site internet 27

Introduction

Ces dernières années, le nombre d'objets connectés a augmenté exponentiellement et a dépassé les 8 milliards en 2017 (Gartner, 2017). Chaque connexion crée des données de sources variées. Par exemple, une connexion sur un site internet crée des données de cookies. Ces cookies sont régulièrement utilisés de manière à conserver les mots de passe et les connexions des utilisateurs (Google, 2018). Ces données sont stockées et peuvent être valorisées pour améliorer les fonctionnalités des sites internet. Ils peuvent aussi être utilisés afin de cibler les publicités selon les utilisateurs (Chung and Paynter, 2002; Google, 2018). Les logs de connexion sont également stockés par les serveurs de ces sites. En effet, l'hébergeur a obligation, d'après l'article 1 du décret n° 2006-358, de conserver les données de trafic de son site durant un an minimum (De villepin, 2006). Ces données peuvent être utiles en cas de litige (copie des données du site par un robot par exemple).

Néanmoins, elles sont moins valorisées que les données de cookies (Dongre and Raikwal, 2015) et utilisent un espace de stockage qui peut s'avérer coûteux pour les sites générant beaucoup de trafic.

L'entreprise OmicX est à l'origine du site internet omictools.com (Perrin et al., 2017). La plate-forme, mise en ligne en 2013, est un répertoire d'outils bioinformatiques associé à un moteur de recherche. Sur le site, 28 852 outils sont disponibles, répertoriés dans 149 catégories principales et 2 683 sous-catégories (Données du 1er août 2018). Comme tout site internet, il est important de comprendre la navigation des utilisateurs afin d'augmenter le nombre et la durée des visites (Dongre and Raikwal, 2015). Des outils tels que GoogleAnalytics (figure 1) permettent d'avoir accès à des statistiques intéressantes sur les utilisateurs (Chande, 2015).

Néanmoins, ces graphiques sont proposés à partir de données de cookies qui peuvent être bloqués par les utilisateurs (Chande, 2015). En effet, leur collecte nécessite l'approbation de l'internaute (CNIL, 2017). Une approche à partir des données de logs de connexion pourrait alors être envisagée pour avoir accès aux statistiques de manière plus fiable et adaptée aux besoins réels de l'entreprise. Ces données sont automatiquement enregistrées par les serveurs et l'internaute n'a pas moyen de les bloquer.

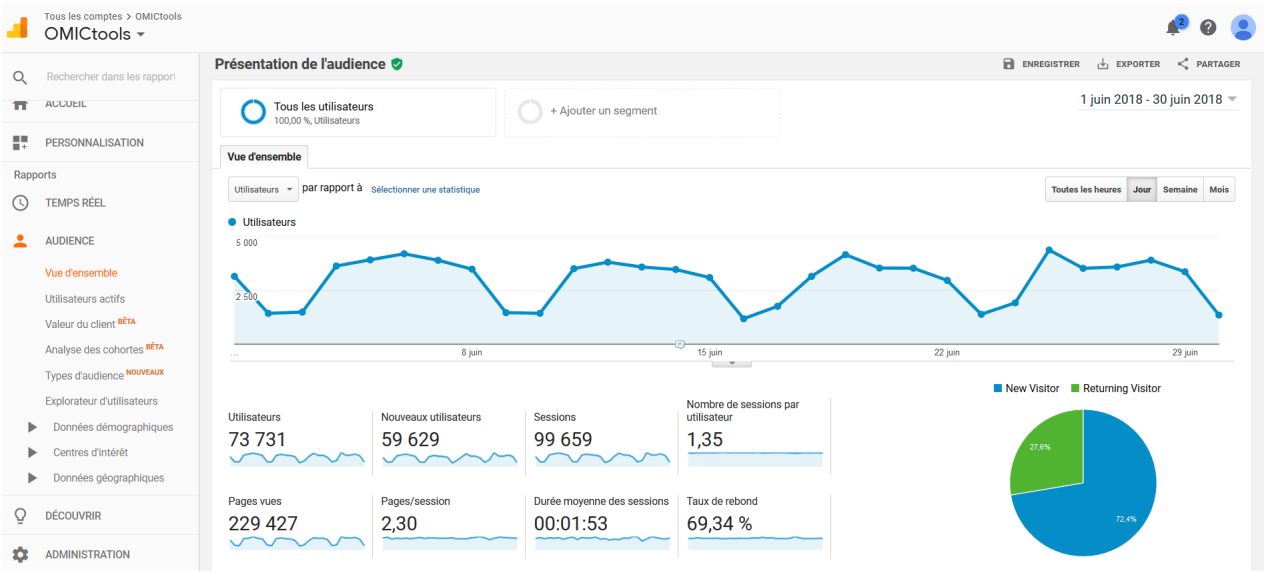

FIGURE 1 – Capture d'écran de l'interface Google Analytics

Visualisation du trafic du mois de juin 2018 à partir des données du cookie '–ga' de GoogleAnalytics

En effet, les visiteurs du site omictools sont généralement des informaticiens ou bioinformaticiens, sensibilisés aux problématiques de collecte des données. Il est donc probable qu'ils utilisent un moyen de blocage des cookies comme Ghostery (Knowlton, 2018). Il est donc intéressant pour la société d'avoir un outil permettant de connaître le nombre de visiteurs uniques, le nombre de sessions et de bien d'autres indicateurs. Ainsi, la problématique de stockage et d'exploitation de ces données s'est posée, allant jusqu'à la prédiction du parcours utilisateur sur le site internet.

Pour cela, une base de donnée relationnelle a été créée afin de structurer la donnée puis cette base a été connectée à une interface graphique de visualisation. Enfin, ces données ont servi de jeu d'entraînement pour des modèles de prédiction statistiques et d'apprentissage profond (deep-learning) afin de proposer une aide à la navigation pour les visiteurs. Ces modèles seront didactiques afin de leur faire découvrir de nouveaux outils que d'autres bioinformaticiens avec les mêmes problématiques utilisent.

1 Des données de grande dimension à structurer pour les exploiter

1.1 Des données à caractère personnel

Avant même de commencer le stockage et l'exploitation des logs de connexion, il paraissait essentiel de rappeler le cadre légal d'un tel projet. En effet, les données d'utilisateurs sont des données à caractère personnel. Une donnée est dite à caractère personnel quand elle permet d'identifier la personne de manière directe ou indirecte (Meunier, 2018). Les données de logs contiennent l'adresse IP et l'agent utilisateur des personnes se connectant au site, qui permettent une identification des personnes. Il s'agit donc bien de données personnelles. Le cadre légal de la Réglementation Générale de la Protection des Données (RGPD) définit donc le stockage et l'exploitation de ces données. Les données à caractère personnel doivent ainsi être collectées pour des finalités déterminées, explicites et légitimes, adéquates, pertinentes et limitées à ce qui est nécessaire au regard des objectifs pour lesquels elles sont traitées. Cette loi est appliquée depuis le 25 mai 2018 pour toutes les entreprises de l'Union Européenne (UE) ou internationales dont les personnes ayant accès au service résident dans l'Union Européenne.

Premièrement, le stockage et l'exploitation sont soumis au consentement explicite de la personne. Il faut informer précisément les individus sur cette exploitation et sur le temps de stockage des données. Dans notre cas, le stockage des données de logs est obligatoire en France. Il nous faut par contre obtenir l'autorisation des personnes pour l'exploitation des données. Avant le 25 mai 2018, la page *terms-of-use* du site omictools.com informait les utilisateurs que, par l'utilisation du site, ils acceptaient l'exploitation de ces données à des fins statistiques et prédictives. Depuis le 25 mai 2018, il a donc fallu obtenir un consentement explicite : une fenêtre s'affiche à l'arrivée de l'utilisateur sur le site, lui expliquant l'exploitation faite à partir de ses données et l'invitant à nous indiquer s'il en refuse l'exploitation.

De plus, chaque utilisateur peut demander à ce que les données le concernant soient effacées sans justification. Un script permettant cela a donc été mis en place. Il supprime l'adresse IP et l'agent utilisateur de cette personne. Ce sont, en effet, les deux informations à caractère personnel stockées en base. L'utilisateur doit, pour cela, envoyer une requête à l'adresse email de contact avec ses informations afin que ses données puissent être supprimées, comme expliquer dans la page *terms-of-use* du site omictools.com.

Cette loi permet à chacun de mieux contrôler ses données. Un Directeur de la Protection des Données (DPD) travaille actuellement avec la structure OmicX afin de s'assurer de la conformité des projets avec cette nouvelle réglementation RGPD.

Pour chaque requête sur le site internet, par un humain ou par un robot ou *bot* en anglais, des lignes sont écrites dans le fichier de logs correspondants aux requêtes envoyées par le navigateur au serveur. Un *bot* est un programme informatique capable de communiquer et d'échanger des informations avec des serveurs de manière automatique. Ils permettent, par exemple, d'indexer des pages, de vérifier que toutes les pages d'un site fonctionnent ou de copier l'ensemble d'un site internet en quelques minutes. Ces données liées au *bot* sont donc non informatives dans le cas du traitement des données des utilisateurs. Un exemple de lignes de données écrites sur le serveur est présenté sur la figure 2A, il s'agit du premier encart de données brutes.

Ces fichiers sont stockés sur un serveur, de manière hebdomadaire entre février 2017 et novembre 2017 et journalière depuis décembre 2017, changement dû à l'évolution des serveurs de la société. La première étape a donc été d'obtenir un fichier par jour contenant les données de logs. De plus, en avril 2018, deux nouveaux serveurs ont été mis en place, un aux États-Unis et un à Singapour afin de diminuer le temps de chargement des pages en Amérique et en Asie. Les données sont donc actuellement réparties en trois fichiers. Il a ainsi fallu adapter le script d'insertion en base à ces nouvelles contraintes. Ce programme tourne actuellement chaque jour pour récupérer le fichier du jour précédent.

Une ligne du fichier est constituée d'informations de l'utilisateur (adresse IP, agent utilisateur) et sur la requête effectuée (nom de la page, date et heure de la requête...). Il a fallu construire une première expression régulière (REGEX) pour structurer correctement ces informations. En effet, les données ne sont pas séparées classiquement par des tabulations ou des virgules (*.tsv/*.csv), mais par des caractères variables entre chaque information. C'est la première étape de pré-traitement permettant de structurer les données (figure 2A, 2B).

La deuxième étape est un filtre (figure 2B, 2C), permettant de récupérer uniquement les données utiles à notre étude. Il y a en effet un premier nettoyage à effectuer pour enlever les bots informatiques. Les agents utilisateurs ont ici été utilisés pour les identifier. En effet, on peut voir sur la figure 2B, en rouge, le mot clé **bingbot** nous permettant de dire que cette ligne est écrite par l'interaction entre le serveur et un robot. Cette liste de mots clés a été constituée en interne par observation des données. Il s'agit d'un travail conséquent afin d'obtenir des données cohérentes et utilisables pour faire des statistiques sur les visiteurs. Ce filtre aurait été bien différent si le but de l'étude avait été d'effectuer des statistiques de robots passant sur le site.

De plus, un second nettoyage a été effectué sur les données pour éliminer les requêtes d'affichage d'images. Lorsque l'on affiche une page avec une image, deux lignes sont écrites, l'une d'elles correspond à la requête permettant d'obtenir le contenu HTML et l'autre, à l'affichage de l'image stockée par exemple en .png (figure 2B). Là encore, les mots clés (extensions de fichiers) ont été identifiés et utilisés pour ne pas stocker ces données dans notre base (Annexe 1).

Enfin, des indicateurs internes ont été trouvé afin de ne pas inclure les visites du personnel dans les statistiques. En effet, certains sont amenés à passer beaucoup de temps sur le site, et ces données ne sont pas pertinentes. Les adresses IP de l'entreprise, des prestataires ainsi que des mots clés liés à l'administration du site ont été ajoutés au filtre (Annexe 1).

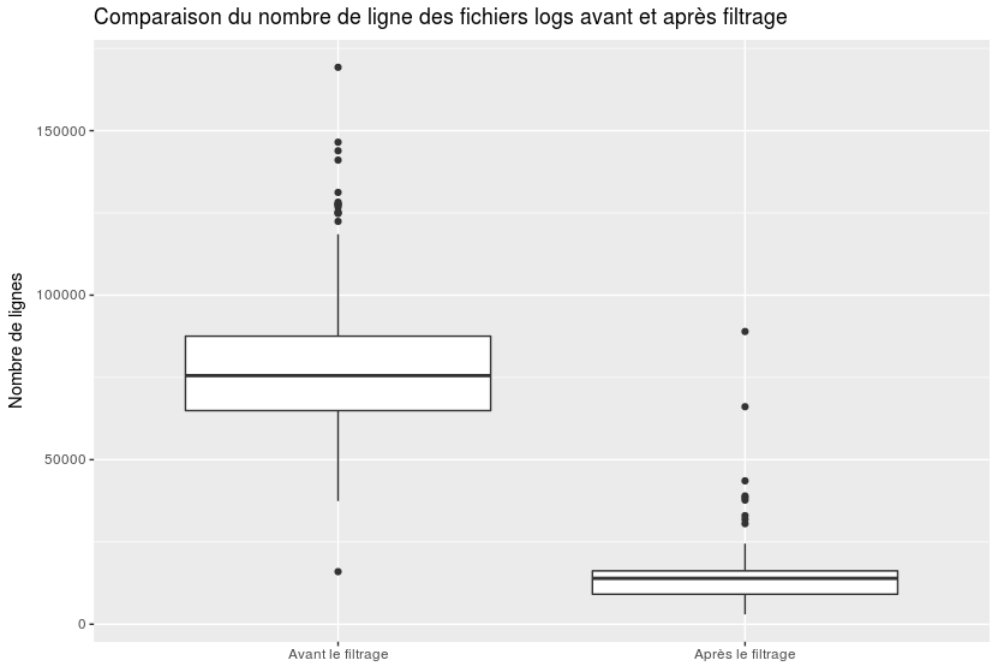

FIGURE 3 – Comparaison globale du nombre de ligne avant et après l'application du filtre pour chaque fichier journalier
Entre le 13 février 2017 et le 12 juin 2018 : Fichier en moyenne 77062 lignes avant filtrage et 13615 lignes après

Ce pré-traitement est classique et nécessaire pour effectuer des statistiques correctes par la suite (Dongre and Raikwal, 2015). En effet, le bruit a été fortement réduit, le nombre de données étant divisé par 6 (figure 3). Ce filtrage effectué en amont de l'insertion en base permet ainsi de ne pas stoker d'informations non exploitées dans le cadre de ce projet. Les éléments du filtre sont fournis en annexe 1.

Les éventuels oublis de robot seront ainsi facilement repérables en utilisant des statistiques simples. Le nombre de page vues par jour pour un utilisateur sera, par exemple, un bon indicateur. S'il s'agit d'un humain, on peut penser qu'il n'ira pas voir 400 pages en une seule journée comme le ferait un robot. On pourra ainsi regarder si ce pré-traitement a été suffisant ou s'il doit être amélioré.

1.3 Construction d'une base SQL

La création d'une base de données relationnelle a semblé être la meilleure solution afin de pouvoir effectuer des statistiques rapides et de conserver nos données dans un schéma structuré pour créer un modèle de prédiction.

Une ligne de logs est donc constituée de l'adresse IP de l'utilisateur, de la date et l'heure de la requête, du type de réponse de la page, du nom de la page requêtée, du code réponse de la page, de la page d'origine de l'utilisateur, de son agent utilisateur qui correspond aux informations transmises par son navigateur internet et de son pays d'origine (figure 2A).

La base MariaDB de type Structured Query Language (SQL) a été choisie. En effet, on connaît les clés avant l'insertion et les fichiers de logs n'évoluent pas ou très peu au cours du temps. Les données de la base sont donc stables et liées entre elles. Une base document de type NoSQL aurait également été possible aux vues du nombre de requêtes mais n'a pas été choisie ici du fait des relations entre les données. De plus, l'entreprise m'a également encouragée à choisir une base relationnelle afin d'assurer une homogénéité du parc des bases de données déjà en place.

Le schéma simplifié de la base est disponible sur la figure 2D. Le principe d'une base SQL est d'attribuer des indices (clés) aux différents champs. Par exemple, la page d'accueil du site va être présente dans les requêtes de plusieurs utilisateurs. On va donc lui attribuer un identifiant unique et stocker cet identifiant et sa valeur dans une table *request_value*. Cette opération est aussi effectuée pour les valeurs des origines, les réponses des pages, les adresses IP, les pays et les agents utilisateurs. Ces valeurs uniques sont chacunes stockées dans des tables indépendantes afin d'éviter la redondance des informations stockées.

La table *Country* est associée à la table *IP*. En effet, chaque adresse IP a une localisation unique. Les tables *User_agent* et *IP* sont liées à la table *User_name*. Il est considéré ici qu'un utilisateur est défini de manière unique par l'association de son agent utilisateur et de son adresse IP. Cela peut donc introduire un biais. En effet, plusieurs personnes travaillant dans une structure de taille moyenne (possédant une unique adresse IP), certains ordinateurs de même configuration peuvent renvoyer un même agent utilisateur et donc être associés à un seul utilisateur. Ce biais est jugé acceptable car les structures de taille importante vont posséder plusieurs adresses IP, ce qui permettra de distinguer les utilisateurs.

Toutes ces tables sont ensuite liées par des clés étrangères à une table centrale nommée *Request*. Cette table contient une ligne par requête utilisateur sur le site internet.

Afin d'optimiser la rapidité de la base, des index ont été créés sur certains champs comme les dates. Un index est une structure de données entretenue automatiquement par la base elle-même, qui permet de localiser facilement des enregistrements dans un fichier (Coronel et al., 2012).

Une dernière table *Session* a été créée. Elle contient l'heure de début, l'heure de fin, la durée, la date et l'identifiant de l'utilisateur d'une session. Elle a été définie comme dans Google Analytics (Chande, 2015) afin de pouvoir comparer ce travail de statistiques descriptives aux statistiques de Google, habituellement utilisées par l'équipe marketing. Une session démarre à la première requête de l'utilisateur et se termine après 30 minutes d'inactivité (c'est à dire, 30 minutes sans requête par cet utilisateur) ou à minuit, heure française. Ce calcul induit donc un biais, car si un utilisateur navigue en journée dans une autre zone de fuseau horaire, sa session pourra être divisée en deux sessions. Ce biais existe déjà dans les statistiques de Google, et les personnes de l'équipe marketing ont souhaité conserver cette même façon de calculer une session.

Les fichiers de logs se créant en temps réel, ils sont compressés chaque matin vers 6 heures et stockés sur leurs serveurs respectifs. Un planifieur a été mis en place pour permettre la récupération automatique de ces données chaque nuit, leur filtrage, leur insertion en base et le calcul des sessions. Cette méthode permet l'exécution automatique du script et donc d'assurer une base à jour. Ce script prend en moyenne 2 h 16 chaque matin pour s'exécuter. Cette moyenne est établie à partir des durées d'exécution du script entre le 1er juillet et le 19 juillet 2018, ce qui correpond à 14 jours ouvrés et 5 jours de week-end.

Cela conforte l'idée que l'application d'un filtre était nécessaire, un plus gros volume de données aurait nécessité un temps d'insertion beaucoup plus long. On peut également noter que le trafic étant moins important le week-end, l'insertion des fichiers du samedi et dimanche sont plus courts (figure 4).

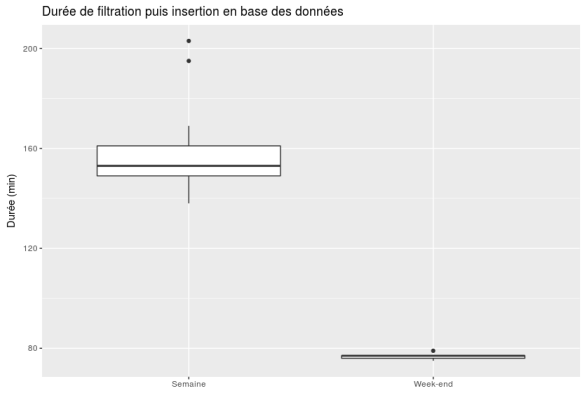

FIGURE 4 – Boxplot de la durée d'exécution du script Filtre puis Insertion
Distinction de l'insertion des fichiers contenant des données des jours ouvrés et contenant des données du week end. Entre le 1er juillet 2018 et le 19 juillet 2018.

Le passage de fichiers de données brutes structurées à une seule base de données a été un travail conséquent. Il était nécessaire afin de pouvoir correctement les exploiter. Il est aussi intéressant de noter que cette première partie répond à une problématique d'exploitation particulière : on cherche ici à effectuer des statistiques sur les utilisateurs du site.

Un premier bilan global est fait à ce stade de notre étude. Il permet une vue globale des données à disposition. Ce bilan est effectué le 19 juillet 2018 (Table 1).

Du 13 février 2017 au 18 juillet 2018	
Nombre de requêtes	5.557.118
Nombre de visiteurs	1.572.878
Nombre de sessions	2.367.066

TABLE 1 – Récapitulatif des données contenues dans la base SQL

Il y a donc dans cette base plus de 5 millions de lignes (Table 1). Il s'agit de données massives, une visualisation plus fine des données sera nécessaire. En effet, même sur une journée, l'affichage en brut dans un tableau des requêtes est illisible, il s'agit d'un tableau de plus de 10 000 lignes. Il va donc être nécessaire de visualiser ces données afin d'obtenir des informations pertinentes à partir des données de cette base.

2 Exploration des données dans une interface graphique

2.1 Analyse de la qualité des données

Avant d'exploiter cette base de données, il faut en vérifier la qualité. En effet, pour le moment, rien n'indique que les étapes de pré-traitement précédemment définies sont suffisantes pour affirmer que toutes les données insérées en base sont des données d'utilisateur et non de robots par exemple. S'il reste trop de données liées au robots, cela aurait pour conséquence de sur-estimer nos indicateurs statistiques utilisés par le marketing.

La cohérence des données est facilement vérifiable en les mettant en rapport avec leur nature même : l'activité d'un humain et d'un programme automatisant la recherche de lien *hypertexts* par exemple seront en effet bien différentes. Le nombre de clics par session et la durée de celles-ci ont donc été utilisés pour vérifier cette qualité.

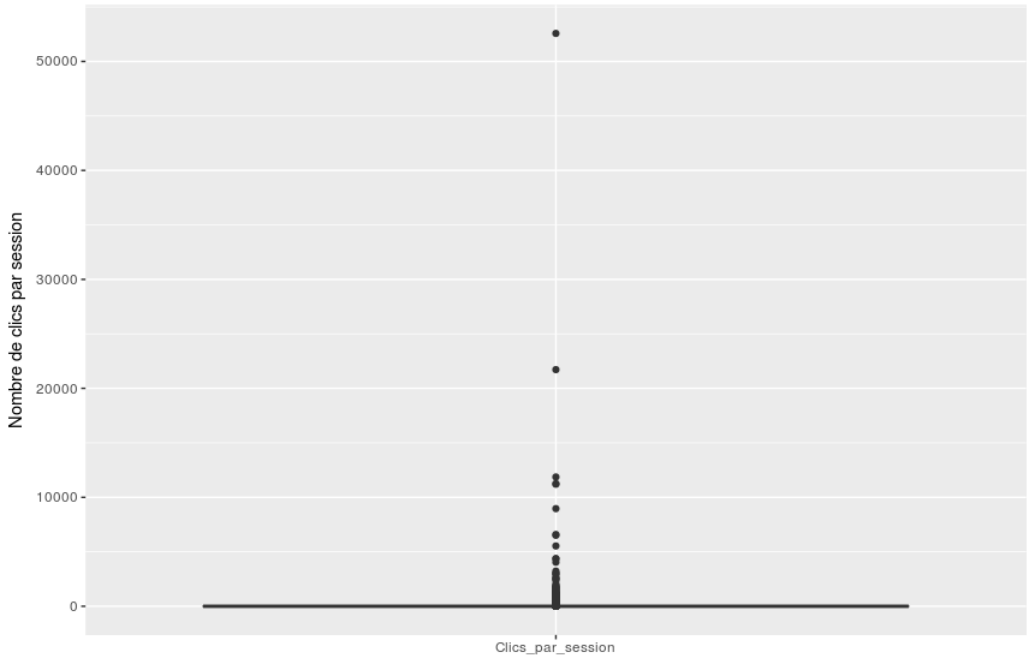

FIGURE 5 – Boxplot du nombre de pages vues par session
Données entre le 13 février 2017 et le 17 juin 2018

On voit sur la figure 5 qu'il y a quelques valeurs aberrantes dans notre base de données. La question centrale est donc de comprendre d'où viennent ces données aberrantes afin d'améliorer le filtre et de les supprimer de la base. L'étude des données brutes n'a pas permis de trouver un indicateur à ajouter au filtre. Il s'agit donc de robot qui ne sont pas détectables à l'aide de leur agent utilisateur. Pour, néanmoins avoir une base de

données sans robot, il faut estimer à partir de combien de requêtes dans la session, il est probable que les données ne soient pas correctes et qu'elles ne devraient pas être prises en compte. On pourra alors fixer un seuil.

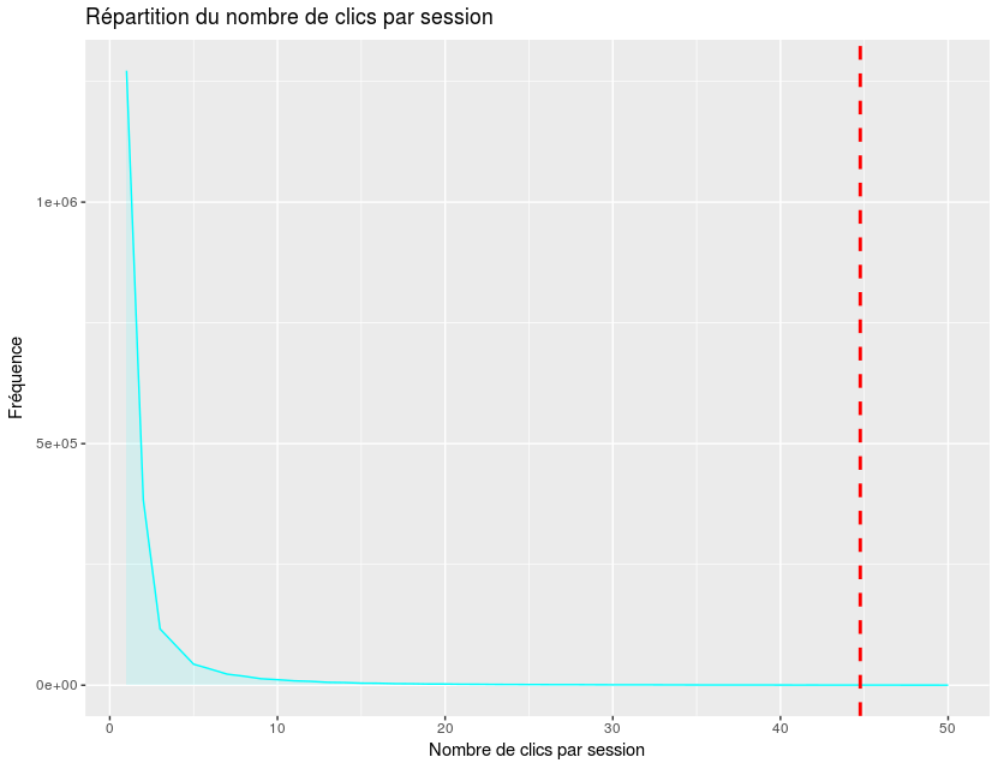

FIGURE 6 – Nombre de requête sur le site par session
Données entre le 13 février 2017 et le 17 juin 2018, avec en rouge la valeur de l'écart-type du nombre de clics

Comme cela est visible sur la figure 6, peu de session ont plus de 10 clics. On a établi que seulement 3.65 % des sessions ont plus de 10 clics. Néanmoins, on a jugé qu'un humain pouvait voir plus de 10 pages durant sa navigation. La valeur seuil a été fixée à 45 (Figure 6), cela permet de réduire le bruit lié aux derniers robots restants tout en minimisant l'opportunité d'éliminer une vraie visite. En nombre, cela représente 0.3 % des données, ce qui montre que la marge d'erreur résiduelle générée par notre filtrage précédant est faible. Ces données aberrantes sont supprimées de la base afin de ne pas stocker de l'information inutile. Après chaque insertion, un script de vérification automatique se lance pour effectuer cette suppression.

2.2 Mise en place d'une interface interactive

Le premier travail sur la base nouvellement créee a été de sortir les statistiques utilisateurs du site internet. En discussion avec l'équipe marketing, et par analyse des graphiques fournis par Google Analytics, une liste de besoins a été définie. Cette liste s'est enrichie au fur et à mesure des demandes et des possibilités offertes par les données.

Les nécessités de l'équipe marketing ont pu être divisées en trois parties :

— Les indicateurs de visites, à comparer d'une semaine à l'autre pour connaître l'évolution du trafic ;

— Les données globales de parcours utilisateur dans chacun des sous-types de pages du site omictools.com (pages "outils", pages "catégories" ...) ;

— Les statistiques détaillées pour chacun des sous-types précédemment cités.

Un besoin d'interactivité évident est ressorti de ces réunions : ces chiffres et graphiques devront être disponibles entre deux dates fournies par l'utilisateur. Un autre paramètre est également au choix, il s'agit du paramètre *limite* (figure 7). Ce paramètre intervient dans les graphiques et établit la limite d'identifiants uniques à afficher dans une figure (le nombre n défini dans "les n pages les plus vues sur le site" par exemple).

Une visualisation de ces données (figure 7 et figure 8) à l'aide d'une interface web, codée en python via le framework Dash (Dash, 2015) a été mise en place et est disponible en interne. Les données sont obtenues à partir de requêtes SQL directement sur la base, entre les deux dates données par l'utilisateur.

Les indicateurs (figure 7) sont assez classiques comme le nombre moyen de visiteurs uniques par jour, le nombre moyen de page vues par session, le nombre moyen de sessions. D'autres sont en lien avec le moteur de recherche comme le nombre de mots uniques recherchés sur la période donnée.

FIGURE 7 – Visualisation des indicateurs de trafic du site

Entre le 16/07/2018 et le 22/07/2018. L'utilisateur peut entrer les dates qu'il souhaite, et l'interface lui permet de visualiser les données sur cette période.

La première partie des graphiques concerne le trafic en général sur le site (figure 8). Il permet de visualiser au quotidien des informations tels que le nombre de visiteurs uniques, le nombre de session (figure 8A), le nombre de mots clés recherchés (figure 8B). Il permet aussi de voir le pays où sont les utilisateurs, si l'information a été enregistrée (figure 8C).

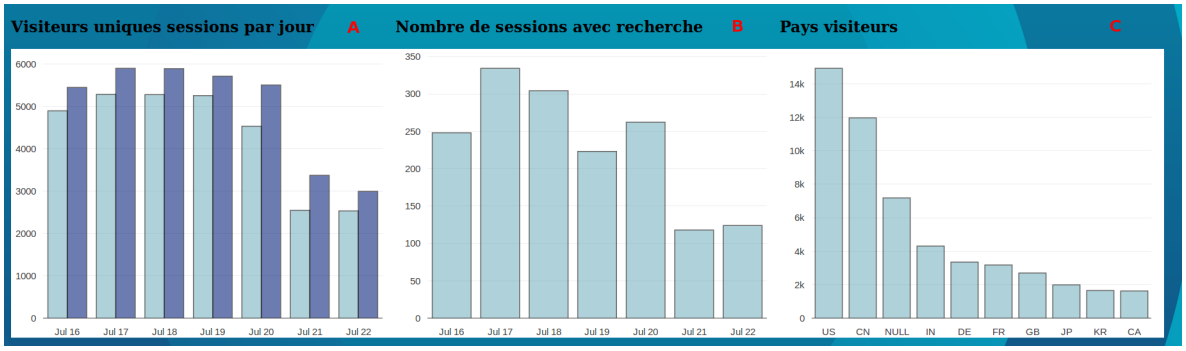

FIGURE 8 – Visualisation des données utilisateurs
Entre le 16/07/2018 et le 22/07/2018

La deuxième partie des graphiques est une vue macro puis micro des pages du site. Le premier graphique (figure 9A) permet d'afficher la proportion de pages vues, regroupées par sous-type de page. En effet, les 31 693 pages qui composent le site ne sont pas toutes de même nature (données du 1er août 2018). Par exemple, toutes les pages "outils" du site sont groupées (les pages "outils" étant les pages détaillant les spécifications techniques d'un programme bioinformatique unique référencé sur omictools.com). Cela permet de comprendre le type de pages le plus vu en global sur le site. La figure 9B permet de visualiser la première page vue par l'utilisateur, là encore groupées par type de page. Le dernier est un diagramme de Sankey (figure 10), visualisant les trajectoires utilisateur entre les pages les plus retrouvées dans le trafic. Ces trois graphiques dépendent du paramètre d'affichage *limite*, ils afficheront par défaut les 10 premiers.

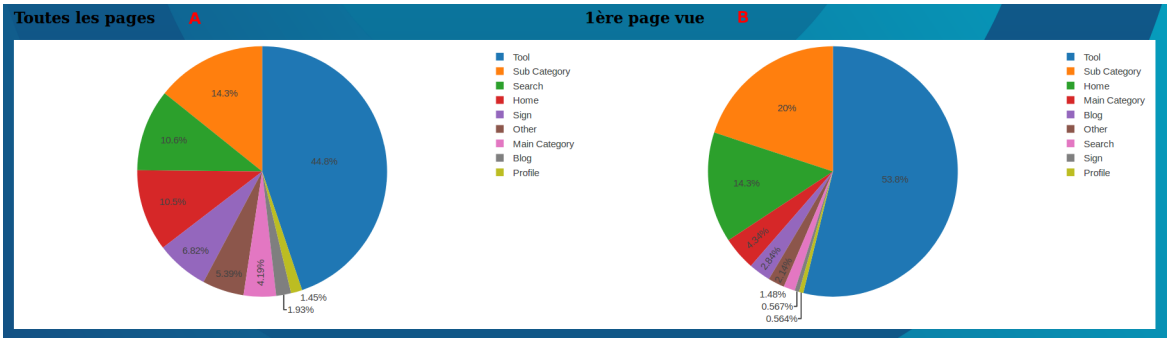

FIGURE 9 – Visualisation des indicateurs dits globaux des données visiteurs
Entre le 16/07/2018 et le 22/07/2018

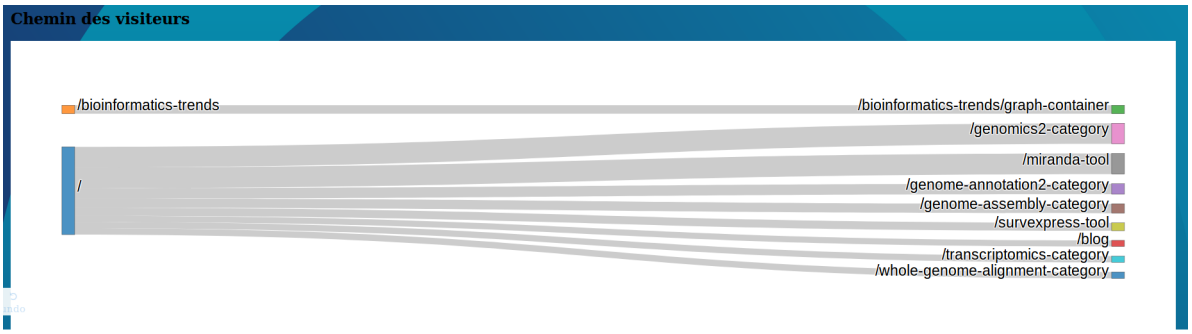

FIGURE 10 – Diagramme de Sankey des données utilisateurs

Entre le 16/07/2018 et le 22/07/2018

La dernière partie permet une échelle plus locale de travail. Les graphiques permettent de visualiser les catégories les plus vues (figure 11A), les outils bioinformatiques les plus vus (figure 11B) et les mots les plus entrés sur le moteur de recherche (figure 11C). Ces graphiques permettent d'étudier les domaines de la bioinformatique les plus actifs en temps réel ainsi que l'intérêt de utilisateurs pour certains outils.

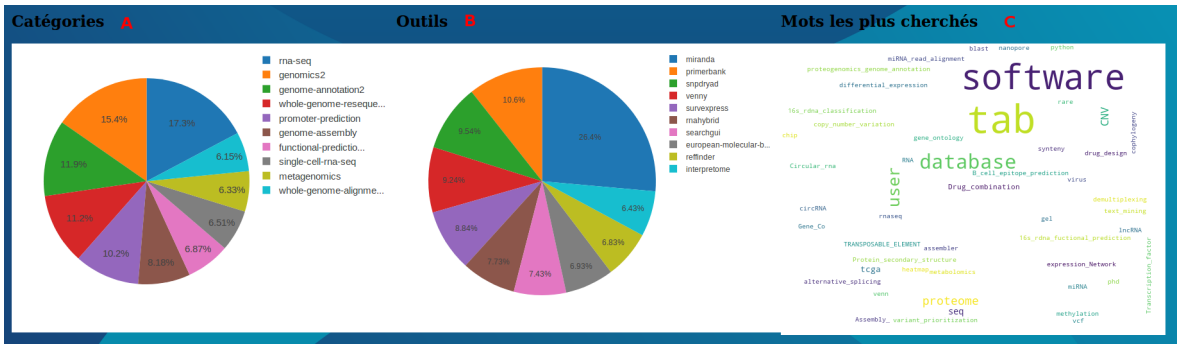

FIGURE 11 – Visualisation des indicateurs concernant les pages "catégories", "outils" et le moteur de recherche

Entre le 16/07/2018 et le 22/07/2018

La création de cette interface a permis un dialogue intéressant avec l'équipe marketing. Il a été nécessaire de bien se comprendre et de définir les termes utilisés afin de représenter ce dont ils avaient réellement besoin.

Cette interface est implémentée sur un serveur local de l'entreprise. Elle peut donc être utilisée quotidiennement par le gestionnaire de trafic du site, et par les responsables de l'entreprise.

2.3 Comparaison avec d'autres outils de calcul d'indicateurs

Pour finir, la comparaison avec d'autres interfaces telles que GoogleAnalytics est intéressante pour montrer que ce travail était essentiel pour l'équipe marketing. En effet, l'outil fourni par Google est utilisé pour gérer les indicateurs de visites d'un site internet (Bender, 2018) et est l'outil majoritairement utilisé dans les entreprises, Google étant un acteur central dans la technologie actuelle (Cox, 2017).

L'indicateur central d'un site internet fournissant un service gratuit comme omictools.com est le nombre de visiteurs uniques par jour. On va tester si la moyenne des visiteurs uniques données par Google est la même que celle obtenue par analyse des logs de connexion. Pour cela, un test de Wilcoxon est utilisé car la distribution du jeu de données ne suit pas une loi normale. Cela est logique car le nombre de visites entre les jours ouvrés et entre les week-ends est très différent. Un test de Shapiro-Wilk ($pvalue < 2e^{-15}$) le confirme.

Pour le test de Wilcoxon, on pose l'hypothèse H0 que les moyennes des données de visites de GoogleAnalytics et du projet logs sont identiques. Or, $pvalue < 2.2e^{-16}$, on rejette donc H0, les moyennes sont significativement différentes. Le nombre de visiteurs uniques par jour calculé par GoogleAnalytics est d'en moyenne 2 700 et celui calculé à partir des données logs est d'en moyenne 4 224 (figure 12). Cette différence montre qu'une partie de nos utilisateurs possède des bloqueurs de cookies.

Environ 36 % des visiteurs ne sont pas détectés par Google. Ce chiffre signifie qu'avant la mise en place de cette base, les indicateurs de performance du site internet étaient sous-estimés de 36 %. Étant dans une start-up, les levées de fonds sont liées à ces indicateurs et les sous estimer représente une importante perte économique.

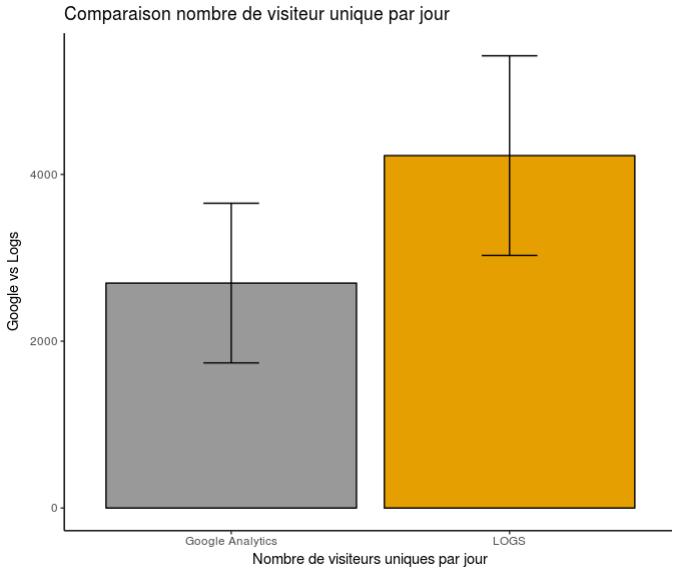

FIGURE 12 – Nombre moyen de visiteurs uniques par jour sur une durée d'un an

Ce chiffre se retrouve sur le nombre de pages vues par jour (42 % de pages vues en plus) et le nombre de session (36 %).

Le premier objectif de ce stage était de valoriser ces données et de construire un outil pratique et utilisable par les équipes en interne pour utiliser les informations contenues dans les logs de connexion.

Un deuxième projet a été imaginé à la suite de cette valorisation. En effet, les informations sur les sessions peuvent nous permettre, en utilisant un modèle prédictif, de construire une aide à la navigation et donc d'augmenter le nombre de page vues par visite.

3 Comparaison de modèle de prédiction

3.1 Un jeu de données basé sur les sessions des visiteurs

L'objectif est ici de proposer aux utilisateurs des pages du site omictools.com qui pourraient les intéresser, en se basant sur les données de navigations globales des visiteurs.

L'idée est donc de récupérer des informations en temps réel et, à l'aide d'un modèle prédictif préalablement entraîné, de leur proposer une page à visiter. Il s'agit, en effet, bien d'un algorithme de prédiction, on cherche à prédire la page vue après un certain nombre d'autre pages. Cependant, aucune donnée personnelle (adresse IP, agent utilisateur) n'a été utilisée afin de garantir le respect de la RGPD.

Pour construire ces modèles de prédiction, un jeu de données a été extrait à partir des sessions de la base de données. Toutes les sessions de cette base de plus de 6 millions de requêtes constituent un volume de données très important.

Ces navigations sont, de plus, influencées par les liens implémentés par l'équipe sur le site internet : en effet, chaque page est reliée à une autre par un maillage interne. Ces liens sont créés par un algorithme permettant de relier par exemple des pages outils entre elles et, étant suggérés et utilisés par l'utilisateur, ils apportent donc une information moins riche qu'une session intégrant une recherche par exemple. Effectivement, elle va créer un lien entre une page vue précédemment et une page proposée par le moteur de recherche. On peut estimer que l'utilisateur travaille dans un domaine en particulier de la bioinformatique et utilise le site d'une façon active.

Un choix a donc été effectué à cette étape, les données qui semblent plus informatives ont été sélectionnées. Les sessions utilisées pour le jeu de données contiendront au moins une recherche.

Ce choix est aussi justifié par la puissance des serveurs disponibles en interne. Un jeu de données trop volumineux serait beaucoup plus long à traiter et probablement induirait des problèmes de mémoire. Pour avoir un ordre d'idée, le jeu de données chargé en RAM utilise plus de 400 Go sur les 500 Go disponibles. Prendre toutes les sessions auraient quadruplé la taille du jeu de données et il aurait fallu louer d'autres serveurs.

Ce jeu de données contient les données entre le 1er février 2017 et le 9juillet 2018 et représente 64 806 sessions.

3.2 Mise en place des modèles de prédiction

A l'aide de ces sessions, deux types de modèles de prédiction ont été mis en place. Ils permettent à partir du chemin de l'utilisateur de prédire la page qu'il irait potentiellement voir après cette navigation. Une approche comparative a été effectuée afin de définir le type de modèle qui entrera en production sur le site internet.

3.2.1 Modèle statistique

Le premier modèle permet de prédire la page suivante à partir de la page actuelle de l'utilisateur. Il s'agit d'un modèle statistique de type graphe où un nœud représente une page et une arrête représente la probabilité p $(0 < p < 1)$ que deux pages se suivent dans un cheminement de navigation.

Une matrice est tout d'abord définie avec en colonne la page d'origine, en ligne la page d'arrivée et la valeur à la jonction correspond au nombre de fois où la page 1 est suivie de la page 2 dans le jeu de données. Ces valeurs numéraires sont ensuite transformées en probabilités. Un exemple est donné sur la figure 13 en prenant un sous-jeu de donnée de 3 sessions. Cette matrice servira de base pour la création du graphe orienté.

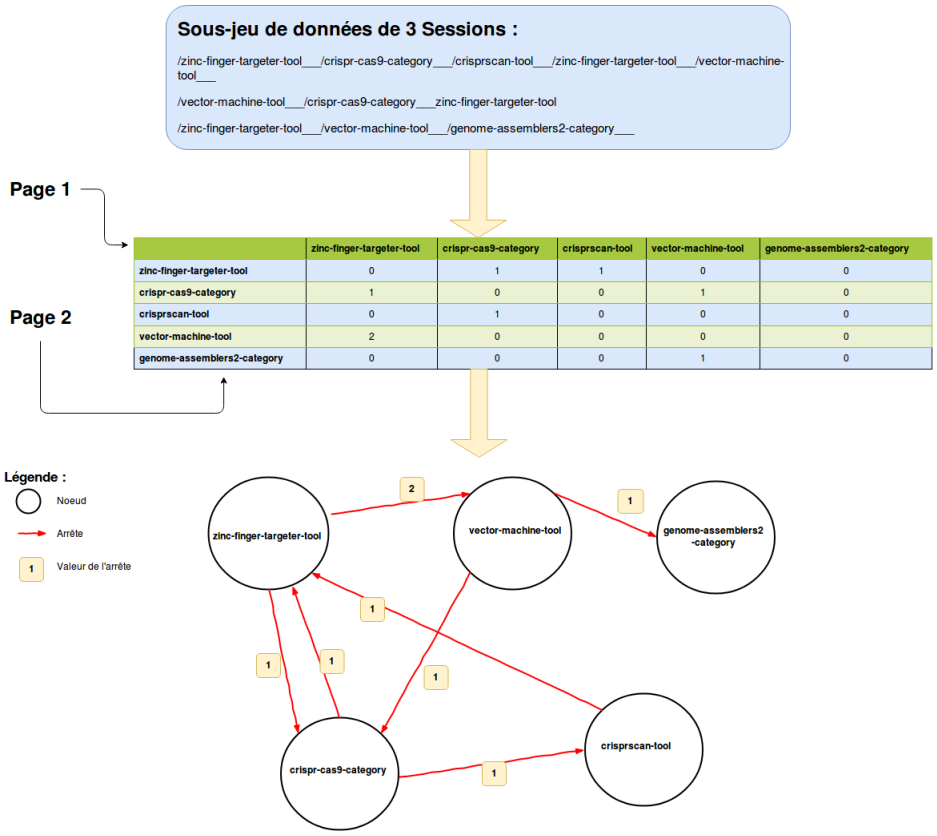

FIGURE 13 – Explication de la construction du modèle à partir de 3 sessions fictives

Ce graphe est orienté car un parcours dans le site à un sens défini. Dans l'exemple, le chemin *vector-machine-tool* vers *genome-assemblers2-category* est possible mais l'inverse n'est pas possible selon la construction du graphe. Ce graphe est créé à l'aide du package Python *networkx* (Hagberg et al., 2018) et à partir du jeu de données décrit dans la partie 3.1. Il possède 22 737 nœuds et 12 1781 arrêtes.

À partir de ce modèle synthétisant toute l'information de navigation sur le site (quand la session comporte une recherche), il est ensuite possible à partir d'un noeud de prédire la page suivante. Pour cette prédiction, deux types de statistiques sont possibles.

La première est la page la plus probable, c'est-à-dire le noeud relié à notre noeud de départ par l'arrête présentant une probabilité maximale. La deuxième est d'effectuer un tirage parmi les noeuds reliés à notre page initiale en pondérant le choix par les poids des arrêtes. Cela permet de laisser un peu de plasticité au modèle et ainsi de diversifier les prédictions. En reprenant notre exemple, une prédiction à partir du noeud *zinc-finger-targeter-tool* donnera avec le premier modèle toujours le noeud *vector-machine-tool* alors que le deuxième proposera 2 fois sur 3 ce noeud mais aussi 1 fois sur 3 le noeud *crispr-cas9-category*.

Une limite est vite apparue dans ce modèle : la richesse de nos données réside dans l'information temporelle des sessions. Prédire à partir seulement d'une page entraîne une perte d'information, certaines sessions faisant plus de quatre pages. Pour prendre en compte cette information, un modèle d'apprentissage profond ou deep-learning en anglais, a été mis en place.

3.2.2 Modèle d'apprentissage profond

Présentation

Un modèle de Réseau Récurrent à Mémoire à Court Terme Résiduelle (Long Short-Term Memory, LSTM en anglais) a été utilisé (Hochreiter and Schmidhuber, 1997). Ce type de modèle est apte à apprendre et à générer des séquences ordonnées (ie. temporelles), des séquences de mots par exemple (Chiu and Nichols, 2017). Cette mémoire vient du fait que la prédiction va dépendre de plusieurs items en entrée et non uniquement du dernier comme dans le modèle précédant (chaque mot composant le début d'une phrase sera pris en compte lors de la génération du mot suivant, pour rester dans le même exemple).

Le but global de l'entraînement d'un tel modèle est de lui fournir une donnée d'entrée X et une donnée de sortie $Y_{attendue}$. Le modèle va lui-même essayer de prédire une valeur $Y_{prédie}$ en utilisant X, et celle-ci sera comparée avec la valeur $Y_{attendue}$. Si celles-ci sont différentes, le modèle doit se corriger afin de se rapprocher de la bonne prédiction.

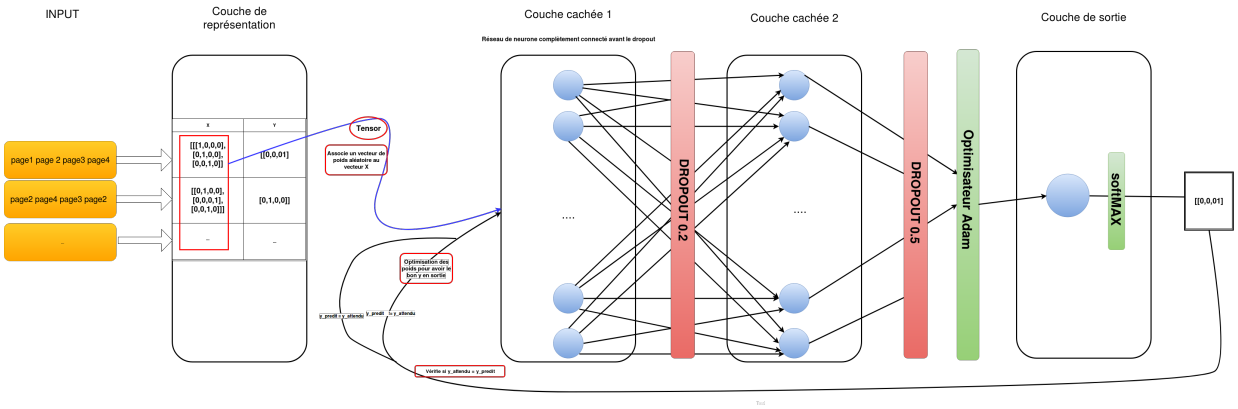

FIGURE 14 – Représentation schématique du modèle LSTM
Dropout_1 0.2, Dropout_2 0.5

Un schéma présentant la structure du modèle est présenté en figure 14. Il comporte une couche d'entrée, deux couches cachées et une de sortie.

La couche d'entrée va permettre de structurer les données comme le modèle le nécessite. Les réseaux LSTM sont assez robustes lors de l'encodage des données en entrée en *one-hot*. Cela assigne, dans notre exemple, à chaque page du site internet une dimension 0 dans un vecteur. Lors de l'analyse d'un parcours, on modifie en 1 les dimensions correspondant aux indices des pages retrouvées dans la navigation. Nous avons décidé d'utiliser des séquences de navigations de 3 pages afin de prédire la quatrième. En effet, le nombre moyen de pages consultées lors d'une session de navigation sur omictools.com est de 3 pages. Comme vu précédemment, notre jeu d'entraînement représente 64 806 sessions qui, une fois décomposées en chemins de 3 pages, représentent 26 4667 parcours. 22 737 pages internet étant représentées dans ces données, la matrice d'entraînement aura donc une structure définie par $X = [[X_1, X_2, ...X_i...., X_{22737}]_1, ..., [X_1, X_2, ...X_i...., X_{22737}]_n]$ avec $n \in [1, 264667]$. La matrice de valeurs de sorties correspondra aux 264667 pages attendues.

Chaque couche cachée est constituée de 256 neurones indépendants. Le but de chaque neurone individuel est d'approximer une fonction qui modifiera le vecteur de poids internes associés aux vecteurs X. Ce vecteur comporte donc un nombre de dimension égal à celui des vecteurs d'entrée X, soit 22 737. A la sortie de la première couche, chaque neurone est relié avec les 256 neurones de la seconde couche cachée. Il s'agit d'un réseau complètement connecté. C'est lors de l'échange d'informations entre ces couches que les fonctions modifiant le vecteur de poids vont être optimisées par les neurones. Le *dropout* représente la probabilité qu'un neurone retrouve son état initial (Pham et al., 2014). Cette subtilité permet d'éviter un phénomène de sur-apprentissage. Courant dans le domaine de l'apprentissage supervisé, cela revient à construire un modèle avec une très bonne précision sur un jeu de donnée en particulier, mais qui devient vraiment moins précis lorsque l'on change ces données. Le modèle s'est adapté de manière trop précise aux données. Chaque couche cachée est ainsi associée avec un *dropout*.

Ensuite, un optimisateur est utilisé pour fusionner les fonctions unitaires, chacune optimisée par un neurone de la seconde couche cachée, en une seule et même fonction globale. L'optimisateur Adam est utilisé, il s'agit d'un des optimisateurs les plus stables et les plus utilisés dans le monde de l'apprentissage profond (Kingma and Lei, 2015; Reimers, 2017). Pour établir la prédiction finale, un *softmax* est utilisé afin de rendre plus plastique le modèle. Ainsi, la sortie du modèle est un score pour chacune des 22 737 pages possibles et non une page unique, rendant le modèle plus adaptatif par la suite.

Avant l'entraînement final du modèle, certains paramètres doivent être optimisés, car leur valeur influence grandement la précision des prédictions (on les nomme alors *hyper-paramètres*). Ce genre de modèle n'est pas capable de travailler sur toutes les données en une seule fois. Elles sont donc séparées en k groupes, chaque groupe allant être utilisé indépendamment pour optimiser le modèle. Ce k correspond au nombre de *batchs*, il s'agit d'un des hyper-paramètres que l'on cherchera à optimiser. Un autre hyper-paramètre e représente le nombre de fois où le modèle parcourra chaque *batch* lors de l'entraînement. Ce nombre représente le nombre d'*epochs*. Le dernier des hyper-paramètres qui sera étudié est la vitesse d'apprentissage (*learning rate*, LR), correspondant globalement à la "vitesse" où seront modifiées les fonctions internes en cas d'erreur de prédiction. Trop haute, la fonction ne pourra jamais se stabiliser à une valeur proche de son optimum, trop faible, le temps d'entraînement nécessaire serait beaucoup trop grand.

Pour ce faire, on entraîne un modèle avec un sous-jeu de données, et on mesure sa précision. L'optimisation se poursuivra tant que la précision n'a pas atteint un plateau.

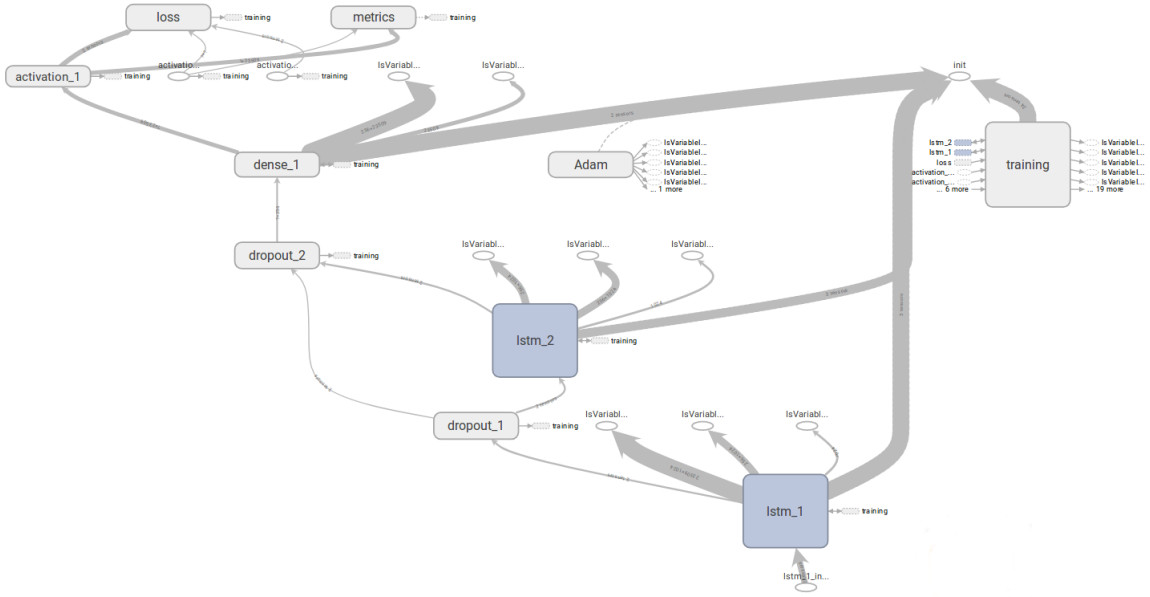

FIGURE 15 – Représentation complète du modèle LSTM sour *TensorFlow*
Dropout_1 0.5, Dropout_2 0.5, loss et accuracy pour la validation des prédictions

Création et optimisation

Ce modèle a donc été défini en utilisant l'architecture *TensorFlow* (Abadi et al., 2016), et cette implémentation est présentée en figure 15. Puis, les sessions ont donc été pré-traitées afin d'obtenir ces quadruplets de pages. On obtient alors un jeu de données X constitué des listes de trois premières pages et un jeu de données Y constitué de la quatrième page vue. À intervalles réguliers, le modèle va donc utiliser les fonctions approximées afin de prédire un $Y_{prédit}$ et de pouvoir le comparer au $Y_{attendu}$. Cette méthode est à la base de tout modèle informatique, le modèle connaissant le résultat a obtenir, il peut se corriger (modifier les fonctions internes) si sa prédiction n'est pas exacte pendant la phase d'entraînement.

Pour l'optimisation, le jeu de données total sera divisé en un jeu de données d'entraînement de 185 238 lignes correspondant à 70 % et en un jeu de test de 79389 lignes. Cette fraction de 70 % pour l'entraînement et 30 % pour l'auto-correction du modèle est très classique en apprentissage profond. La première optimisation a été effectuée sur la taille du *batch* et la *learning rate*. Le temps de calcul étant très dépendant du nombre d'*epochs*, nous avons décidé de le laisser fixe à 5 pour le moment, et de tester ce paramètre dans un second temps.

On voit qu'il y a trois groupes selon les paramètres du modèle LSTM (figure 16). Nous souhaitons sélectionner les paramètres qui maximisent la précision du modèle. calculée sur les 30 % sortis du total des données disponibles. Le modèle avec la meilleure précision (*accuracy = 0.234*) est celui avec une *learning rate* à 0.001 et un *batch* à 8. Il possède plus de 1 % de précision en plus, ce qui est important sur un tel jeu de données.

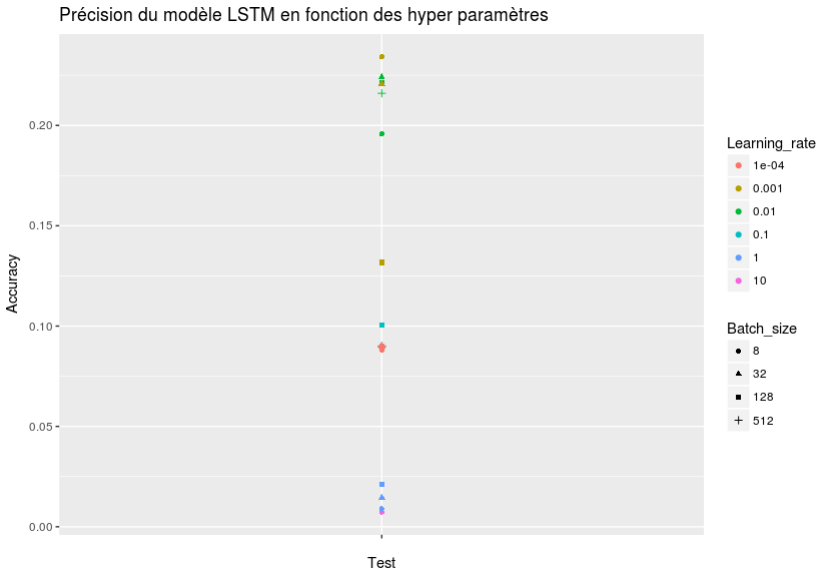

FIGURE 16 – Comparaison de la précision du modèle avec différents hyper-paramètres
Nombre d'*epochs* fixé à 5

Dans un second temps, l'influence du nombre d'epochs sur la précision a été étudiée (figure 17). On remarque également une forte influence de la combinaison des paramètres *batch* et *epochs*, la précision variant de 20.67 % pour une combinaison de 25 epochs et de 32 batch à 23.20 % pour 5 epochs et 8 batchs. Cette dernière combinaison de paramètres a donc été retenue (*accuracy = 0.232*).

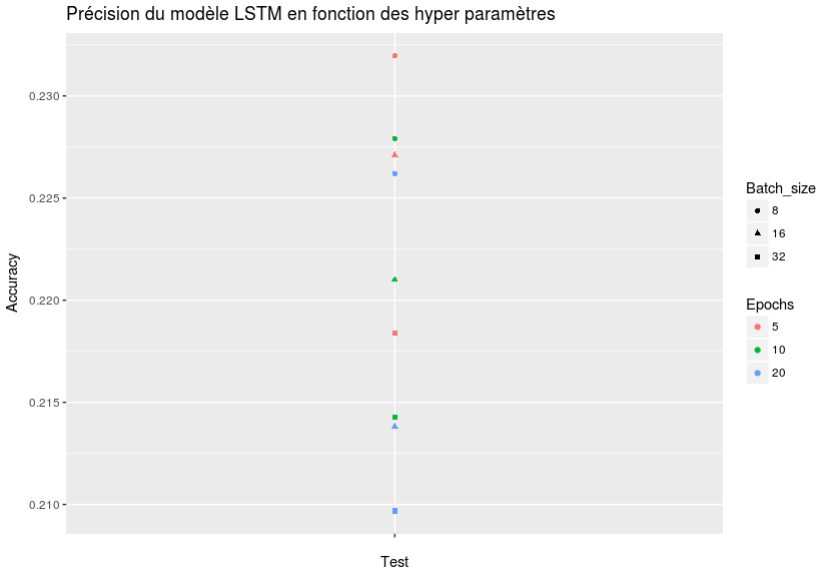

FIGURE 17 – Comparaison de la précision du modèle avec différentes combinaisons de paramètres

Learning Rate fixée à 0.001.

Le dernier paramètre important, et qui sera testé est la *température* du modèle LSTM. Ce paramètre permet de ne pas figer le modèle dans une boucle infinie. Par exemple, lorsque l'on génère du texte, cela évite la répétition des mots précédant très retrouvés dans le corpus d'entraînement. Nous allons tester une température de 0.1, qui donne un modèle plutôt stable mais avec souvent de nombreuses répétitions et une température à 1 qui donne un modèle moins classique dans ces prédictions. Ces deux valeurs seront validées ultérieurement, lors de la comparaison des modèles. Ainsi, deux modèles LSTM seront entraînés, chacun avec une température spécifique.

3.3 Analyse de la précision des modèles

3.3.1 Précision automatique des modèles

La précision est couramment utilisée pour valider un modèle. Pour le jeu de données Test, composé de 79 389 lignes, il est comparé la prédiction contenue dans $Y_{attendu}$ et dans $Y_{prédit}$. Le tout est multiplié par 100 pour avoir le pourcentage de réussite du modèle. La précision d'un modèle totalement aléatoire est aussi effectuée afin de comparer nos résultats au hasard. Il prend aléatoirement et sans pondération une page parmi les 22 737 pages du site existantes et présentes dans notre jeu de données.

Modèle statistique La précision des modèles statistiques (tableau 2) est de 21.68 % pour le modèle prenant le chemin de plus court près la page choisie. et de 13.92 % pour le modèle statistique pondérant le choix de la prédiction par le poids des arrêtes. Cette différence est cohérente car la prédiction la plus probable est la plus présente dans notre jeu de données. En créant un modèle rigide, on peut prévoir une page sur cinq. Néanmoins, on risque de prédire souvent une même page et donc ne pas effectuer une prédiction utile à l'utilisateur.

Modèle d'apprentissage profond La précision des deux modèles LSTM (tableau 2) en précision automatique est de 15.61 % pour le modèle le plus plastique avec une température égale à 1 et de 24.65% pour le modèle avec une température de 0.1, plus rigide. Il est donc observé là encore que par validation automatique, le modèle rigide est le plus précis.

Nom du modèle	Paramètres	Précision
Graph	Statistiques	13.92
Graph	le plus probable	21.68
LSTM	température=1	15.61
LSTM	**température=0.1**	**24.65**
Aléatoire	Tirage aléatoire parmi 25 000pages	0.003

TABLE 2 – Précision obtenue par validation automatique

Comparaison des modèles La précision des modèles semble faible, moins de 25 %. Ils sont néanmoins bien meilleurs qu'un modèle aléatoire, la précision de ce dernier étant inférieur à 0.005 (tableau 2). Les modèles plastiques semblent être meilleurs d'après ce critère. Or, le modèle risque de moins se tromper car il prédit plus souvent un groupe de pages sur-représentées dans notre jeu de données. Il n'apporte pas forcément une page qui serait cohérente de proposer à l'utilisateur. Par exemple, certaines pages catégories du site sont sur-représentées car elles sont fréquentées par de nombreux utilisateurs. Un

autre critère n'est pas pris en compte en étudiant la précision automatique. La structure particulière du site internet n'est pas utilisée lors de la validation. En effet, une page prédite est considérée comme fausse si elle n'est pas exactement la page attendue. Ce qui n'est pas forcément vrai au regard de nos données. Si le modèle prédit que la page vue est une page outil comme */easyrnaseq-tool* et que la page dans y est */deseq-tool*, elle sera dite mal prédite. Or ces deux outils appartiennent à la même catégorie et permettent la même analyse bioinformatique. Le modèle pense s'être trompé alors que sa prédiction est logique. Pour pallier à cette différence, deux autres validations ont été effectuées. La première est une validation manuelle, la deuxième une validation automatique mais en prenant compte la structure.

3.3.2 Précision par validation manuelle

L'idée du deuxième indicateur va être de prendre en compte cela. On va effectuer une validation manuelle ou la prédiction sera notée par des spécialistes : 0 si elle est incohérente, 1 si elle est cohérente et 2 si elle est meilleure que les autres (Hersh et al., 1994). Cela devrait permettre de comparer les modèles.

Une interface de validation humaine (figure 18) a alors été mise en place pour permettre la notation par des scientifiques et non de manière automatique. Le programme de validation va envoyer une séquence de 3 pages à chacun des 5 modèles, et l'expérimentateur devra noter des prédictions avec un score compris entre 0 et 2 (0 correspond à une mauvaise prédiction, 1 à un bonne prédiction et 2 à la meilleure des 5 prédictions). Cette validation a été effectuée par 3 bio-informaticiens de l'entreprise. Ils ont effectué une centaine de validations chacun ce qui permet de faire les statistiques sur 301 résultats.

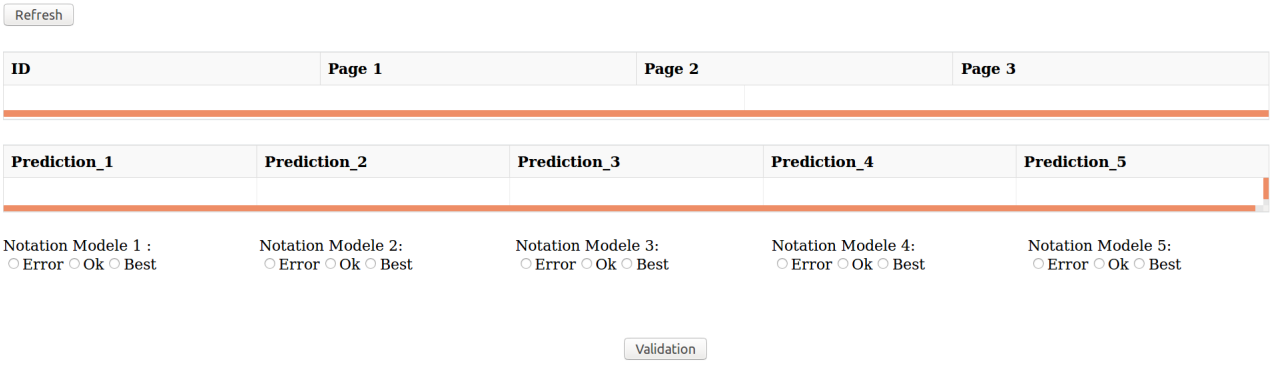

FIGURE 18 – Interface de notation manuel des modèles

Modèle statistique Sur le tableau 3, la précision du modèle graphe utilisant le poids des arrêtes afin de prédire les pages a une précision supérieure de 37,54 % contre 33.22 %

pour le modèle rigide. Par validation manuelle, la variété des prédictions semble permettre un meilleur modèle.

Modèle d'apprentissage profond Pour ces deux modèles, la précision est beaucoup plus faible, autour de 10 % (tableau 3). A l'inverse, le modèle rigide est un peu meilleur que le modèle plastique, il y a 1 % de différence de précision.

Nom du modèle	Paramètres	Précision
Graph	**Statistiques**	**37.54**
Graph	le plus probable	33,22
LSTM	température=1	10.30
LSTM	température=0.1	11.30
Aléatoire	Tirage aléatoire parmi 25 000pages	4.32

TABLE 3 – Précision finale des différents modèles par validation manuelle

Comparaison des modèles Le modèle le plus simple de type graphe est par validation manuelle jugé meilleur que les modèles LSTM. La différence est de plus de 20 %, ce qui est assez étonnant. On pourrait penser que le modèle LSTM aurait mieux prédit car il prenait en compte les 3 dernières pages vues. Deux biais peuvent exister dans cette validation. Un biais lié à l'humain qui valide, et un autre lié au faible nombre de données validées.

3.3.3 Précision en prenant en compte la structure du site internet

La validation manuelle a permis de se faire une première idée de la précision. Mais cela prend du temps, et par conséquent peu de prédictions ont pu être vérifiées (moins de 1 %). Une validation automatique prenant en compte la structure du site a alors été effectuée. L'algorithme compare $Y_{attendu}$ et $Y_{prédit}$. Lorsqu'ils sont différents, trois cas de figures apparaissent :

- $Y_{attendu}$ et $Y_{prédit}$ sont des pages outils : on regarde si les outils ont la même catégorie parent
- $Y_{attendu}$ et $Y_{prédit}$ sont des pages catégories : on regarde si les outils ont la même catégorie parent
- L'un des Y est un outil et l'autre une catégorie : on regarde si la catégorie de l'outil est égal à la catégorie de l'autre Y.

Il affecte donc un score de 1 quand les pages sont les mêmes ou quand elles sont contenues dans la même catégorie.

Modèle statistique Les deux modèles Graphe donnent une précision au tour de 32 % (tableau 4). Ces précisions sont de même ordre de grandeur que la validation manuelle. On peut noter qu'il n'y a pas de différence entre nos deux modèles.

Modèle d'apprentissage profond Les modèles LSTM ont une bien meilleure précision que lors de la validation manuelle. Le sous jeu de donnée apportait donc un biais. Le modèle plastique est moins précis (23,59 %) que le modèle rigide (29.39 %). Ce dernier est donc proche des 30 % de précision, ce qui montre bien qu'il est essentiel de prendre en compte la structure du site internet pour cette validation.

Nom du modèle	Paramètres	Précision
Graph	**Statistiques**	**32,48**
Graph	le plus probable	32,09
LSTM	température=1	23,59
LSTM	température=0.1	29,39
Aléatoire	Tirage aléatoire parmi 25 000pages	5,02

TABLE 4 – Précision par validation automatique prenant en compte la structure du site internet

Comparaison des modèles Enfin, les modèles Graphe sont meilleurs que les modèles LSTM. Cela parait étonnant car le modèle LSTM prenant en compte plus d'informations, il devrait être plus performant. Néanmoins, la dernière validation montre que cet écart est faible et inférieur à 3 %.

Quelques pistes restent à aborder afin d'obtenir un modèle plus performant. Il faudrait prédire seulement des pages outils, afin de proposer une page réellement utile à l'utilisateur. Par exemple, prédire la page d'accueil du site n'a aucun sens. Cette dernière était déjà exclue des prédictions, mais certaines pages catégories pouvaient encore être prédites. Une deuxième piste d'amélioration pourrait être de réduire le jeu de données sur les dernières semaines, afin d'obtenir moins de liens, mais peut être des liens plus pertinents car produit récemment par l'utilisateur.

Conclusion

Les objectifs de ce stage étaient d'exploiter une source de données stockées sous forme de fichiers sur des serveurs. Une étape de structuration et de filtration a donc permis de récupérer des données utilisables, puis la création d'une base de donnée a permis le stockage optimisé de celles-ci.

Ensuite, une visualisation interactive a été mise en place afin de permettre à l'équipe marketing de l'entreprise d'avoir accès à ces données et de les exploiter quotidiennement. Cela permet aussi de comprendre au niveau du domaine de la bioinformatique les intérêts des utilisateurs et d'utiliser ces informations afin d'écrire des articles sur le blog qui soient en adéquation avec les intérêts des visiteurs. Une étude statistique a aussi permis de valoriser les données et de montrer que les visites étaient plus nombreuses que ce que Googleanalytics détectait. Ces données pourront être utilisées lors des prochains rendez vous avec les investisseurs.

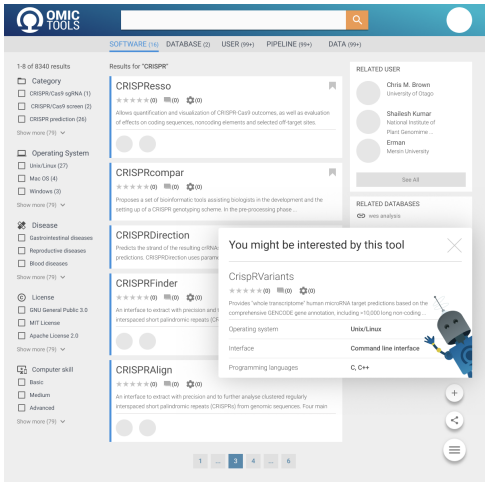

FIGURE 19 – Pop-up s'affichant sur le site

Enfin, ce travail devrait permettre de créer sur le site internet une fenêtre de type *pop-up* qui apparaîtra pour suggérer une page à l'utilisateur. Le design de cette fenêtre (figure 19) a été faite par Ulrich Moutoussamy, graphiste d'OmicX, et son implémentation sur le site internet pourra être réalisé par l'équipe de développement web. Lors de l'écriture de ce rapport et au vu de la précision, l'implémentation n'a pas encore été décidée mais elle est réalisable. Il reste quelques pistes d'amélioration à étudier avant de le mettre en ligne. Une librairie python sera construite avec les fonctions de prédiction et les modèles associés. Une procédure automatique de mise à jour devra extraire chaque semaine les nouvelles

données de log de la base et les intégrer au modèle le lundi matin. Ainsi, l'activité des utilisateurs servira à améliorer la prédiction toutes les semaines.

Ce travail permet donc de valoriser des données non exploitées par de nombreuses entreprises. Aujourd'hui, les données non exploitées sont nombreuses, coûteuses alors qu'elles pourraient être valorisées et donc apportées de la plus-value à une entreprise. Par exemple, les fichiers clients d'une petite entreprise sont rarement découpés et groupés par type d'acheteur, or cette segmentation est souvent utile afin de cibler les utilisateurs d'un produit en particulier.

Bibliographie

Références

Abadi, M., Barham, P., Chen, J., Chen, Z., Davis, A., Dean, J., Devin, M., Ghemawat, S., Irving, G., Isard, M., et al. (2016). Tensorflow : a system for large-scale machine learning. In *OSDI*, volume 16, pages 265–283.

Bender, B. (2018). Introducing Google Marketing Platform.

Chande, S. (2015). Google Analytics - Case study. page 11.

Chiu, J. P. C. and Nichols, E. (2017). Named Entity Recognition with Bidirectional LSTM-CNNs. page 14.

Chung, W. and Paynter, J. (2002). Privacy Issues on the Internet. *th Hawaii International Conference on System Sciences*, page 9.

CNIL (2017). Solutions pour les cookies de mesure d'audience.

Coronel, C., Morris, S., and Rob, P. (2012). *Database Systems : Design, Implementation, and Management*.

Cox, J. (2017). The world's most valuable brands revealed.

Dash (2015). Collaborative data science. *Plotly Technologies Inc.*

De villepin, D. (2006). Décret n 2006-358 du 24 mars 2006 relatif à la conservation des données des communications électroniques.

Dongre, V. and Raikwal, J. (2015). An Improved User Browsing Behavior Prediction Using Web Log Analysis. 4(5) :5.

Gartner (2017). 8.4 Billion Connected "Things" Will Be in Use in 2017, Up 31 Percent From 2016.

Google (2018). How Google uses cookies.

Hagberg, A., Schult, D., and Swart, P. (2018). NetworkX Reference.

Hersh, W., Buckley, C., Leone, T. J., and Hickam, D. (1994). OHSUMED : An Interactive Retrieval Evaluation and New Large Test Collection for Research. In *SIGIR '94*, pages 192–201. Springer London, London.

Hochreiter, S. and Schmidhuber, J. (1997). LONG SHORT-TERM MEMORY. *Neural Computation*.

Kingma, D. P. and Lei, J. (2015). Adam : A Method for Stochastic Optimization. page 15.

Knowlton, P. (2018). About Ghostery.

Meunier, S. (2018). RGPD : Que sont les données personnelles sensibles ?

Perrin, H., Denorme, M., Grosjean, J., Pichon, F., Darmoni, S., Desfeux, A., and Gonzalez, B. J. (2017). OMICtools : a community-driven search engine for biological data analysis. page 11.

Pham, V., Bluche, T., Kermorvant, C., and Louradour, J. (2014). Dropout Improves Recurrent Neural Networks for Handwriting Recognition. pages 285–290. IEEE.

Reimers, N. (2017). Optimal Hyperparameters for Deep LSTM-Networks for Sequence Labeling Tasks.

Annexes

ANNEXE 1 : Filtre utilisé pour nettoyer les données

Type	Indicateur
Affichage sur le site	.php
Affichage sur le site	.jpg
Affichage sur le site	.js
Affichage sur le site	.gif
Affichage sur le site	.svg
Affichage sur le site	.xml
Affichage sur le site	.css
Affichage sur le site	.png
Affichage sur le site	.eot
Affichage sur le site	.html
Robot	Slurp
Robot	spider
Robot	bot
Robot	crawler
Robot	oncrawl
Robot	pingbreak
Robot	scrapy
Robot	baidu
Robot	facebookexternalhit
Robot	Wtrace
Robot	ltx71
Informations internes à l'entreprise	IP de l'entreprise (non communiqué)
Informations internes à l'entreprise	IP des prestataires (non communiqué)
Indicateurs de développement web	favicon
Indicateurs de développement web	staging
Indicateurs de développement web	admin
Indicateurs de développement web	.api