



Géocalized Web Usage Mining

26.01.2022

Réalisé par :
Ismael MEBROUKI

Ali AZZoug

Formation :
Polytech Info 5A

Professeurs :
Nicolas Durand
Mohamed Quafafou

Sommaire

1 Introduction	2
1.1 Objet du document	2
1.2 Responsabilités	2
1.3 Outils Utilisés	2
2 Objectif du projet	3
2.1 Présentation générale	3
2.2 Cahier des charges	3
3 Présentation des données	4
3.1 Les fichiers Logs	4
3.2 Analyse exploratoire des données	5
4 Choix des solutions techniques	8
4.1 Principe du Géocodage	8
4.2 Solution de géocodage.	9
4.3 Choix de la technologie d'application web.	10
4.4 Comparaison des services de cartographie.	10
4.5 Solution de carte retenue.	15
5 Spécifications de l'application réalisée	16
5.1 Une interface web intuitive	16
5.2 Analyses statistiques sur les données.	17
5.3 Visualisation sur la carte	18
5.4 Connaître le nombre de vues d'un site spécifique	18
5.5 Pages les plus consultées par un utilisateur	19
5.6 Suivi du parcours utilisateur	20
5.7 Fonctionnalités annexes.	21
6 Conclusion	22

1 Introduction

1.1 Objet du document

Ce document sert à présenter le projet de fin d'étude : "Geocalized Web Usage Mining"

Il présente les différentes fonctionnalités de l'application proposée.

Il contient les explications des choix techniques des différents outils utilisés, le déroulement et la réalisation de ce projet.

1.2 Responsabilités

La constitution de ce rapport est de la responsabilité des membres du groupe de projet. Chaque membre du groupe s'engage à relire et approuver chaque nouvelle version de ce dernier.

1.3 Outils Utilisés

L'application produite a été développée en python avec le framework flask.

L'exploration et l'analyse des données préalable a été réalisée avec l'outil Jupyter Notebook.

L'interface utilisateur est sous la forme d'une page web, utilisant html, css.

La carte de visualisation a été intégrée à l'application avec un script javascript.

Le stockage et le partage des différentes versions de l'application ont été réalisées avec Github.

2 Objectif du projet

2.1 Présentation générale

Depuis le début de la pandémie de la Covid19 en 2020, on a de nouveau connu une très forte croissance du monde digital, tant dans le nombre de services disponibles : sites webs, applications ou logiciels que dans le nombre d'utilisateurs de ces services.

Cette croissance a généré de très grandes masses de données relatives aux traces d'usage du Web par les internautes. Ces traces d'usages sont enregistrées par les serveurs dans des fichiers journaux, appelés Log Files.

Ces données sont des ressources précieuses pour les propriétaires des sites web qui auront la capacité de mieux comprendre leurs visiteurs afin de mieux répondre à leurs attentes.

Le web usage mining est un domaine informatique qui s'intéresse au processus d'extraction des connaissances à partir des données d'usage sur internet.

Dans ce projet on va analyser un jeu de données d'un serveur pour en extraire les informations sur le trafic des sites, mais également en établissant une analyse de la localisation géographique des sources de trafic.

2.2 Cahier des charges

Le projet consiste à développer une application web d'analyse de fichiers logs (fichiers journaux), par exemple, de serveurs Web. L'originalité est d'introduire la localisation de différents éléments analysés afin de visualiser les résultats sur une carte.

Les données sources sont extraites d'un ancien projet réalisé en Master.

Le projet doit répondre aux contraintes suivantes :

- 1) définition de l'architecture générale de l'application web,
- 2) prétraitement des logs et extraction des motifs (à l'aide d'outils existants),
- 3) géoréférencement des données,
- 4) visualisation des résultats (motifs) sur une carte,
- 5) mise en place de filtres pour une meilleure visualisation des résultats.

3 Présentation des données

3.1 Les fichiers Logs

Les fichiers logs d'un serveur sont un aperçu brut et non filtrés de l'ensemble du trafic sur un site. Chaque fois qu'un navigateur ou un agent utilisateur demande une ressource (pages, images, fichier javascript, etc.) à votre serveur, le serveur ajoute une ligne dans le journal.

Les fichiers log contiennent donc des informations qui peuvent être d'une grande valeur pour l'établissement qui les détient.

Ils permettent :

- d'améliorer la **fiabilité** du système, parce que les erreurs ou les requêtes lentes sont identifiables.
- de maintenir la **sécurité** des environnements de cloud computing puisqu'ils enregistrent les tentatives de connexion, les échecs d'authentification ou les surcharges inattendues du serveur qui peuvent indiquer une **cyberattaque**.
- d'améliorer les prises de **décision commerciale** en analysant le trafic on peut identifier les pages les plus fréquentées et donc celles qui intéressent les visiteurs.

Dans les fichiers logs on peut avoir des informations sur l'évolution du référencement d'un site : on peut savoir si les bots des moteurs de recherche analysent notre site, on pourrait voir apparaître une chaîne d'agent utilisateur qui inclut Googlebot ou BingBot si Google ou Bing demandent une ressource.

Exemple d'une donnée log.

Voici un exemple de ligne stocké dans un fichier log :

11.222.333.44 -- [11/Dec/2018:11:01:28 -0600] "GET /blog/page-address.htm HTTP/1.1" 200 182 "Mozilla/5.0
Chrome/60.0.3112.113"

Le 11 décembre 2018, un utilisateur de Google Chrome a essayé de charger le site

<https://www.portent.com/blog/page-address.htm>. Le '200' signifie que le serveur a trouvé le fichier. Page-address.htm pèse 182 octets. L'adresse IP du client (ou du logiciel qui a demandé le fichier) était – 11.222.333.44.

3.2 Analyse exploratoire des données

Avant de commencer le développement d'une solution à notre projet, nous allons faire une analyse préalable des jeux de données dont on dispose.

Pour explorer nos données on a décidé d'utiliser Jupyter Notebook car c'est un outil qui permet de compiler en une seule page plusieurs codes sources et d'afficher des rapides directement dans la page.

Mais on a surtout utilisé la bibliothèque Panda de Python pour analyser les fichiers textuelles et les transformer rapidement en données faciles à traiter par la machine.



Dans notre jeu de données on dispose de 2 fichiers, le 1er se nomme : da-11 (fichier texte)

On a transformé les contenus du fichier texte en tableau.

Visualisation du fichier da-11 avec IP anonyme

```
Intrée [26]: import pandas as pd
url = "./da-11-16.ipntld.log"

df=pd.read_csv(url, sep=" ", encoding="utf-8")
df.columns=['IP', 'Client Identity', 'ID', 'Time', 'GMT', 'HTTP request', 'Status Code', 'Data(Bytes)', 'From']
df.head(5)
```

Out[26]:	IP	Client Identity	ID	Time	GMT	HTTP request	Status Code	Data(Bytes)	From
	ip1664.com	-	-	[16/Nov/2005:00:00:43 -0500]		GET /gpspubs/sigkdd-kdd99-panel.html HTTP/1.0	200	14199	
	ip1115.unr	-	-	[16/Nov/2005:00:01:00 -0500]		GET /news/99/n23/i12.html HTTP/1.1	200	3171	http://discount-blah1.professional-doctor.c
	ip2283.unr	-	-	[16/Nov/2005:00:01:02 -0500]		GET /dmcourse/data_mining_course/assignments/a...	200	8090	http://www.google.com/sear hi=en&q=use+of+c
	ip2283.unr	-	-	[16/Nov/2005:00:01:03 -0500]		GET /dmcourse/dm.css HTTP/1.1	200	155	http://www.kdnuggets.com/dmcourse/data_mining
	ip1389.net	-	-	[16/Nov/2005:00:02:46 -0500]		GET /gpspubs/kdd99-est-ben-lift/sld021.htm HTT...	200	1385	http://www.google.com/sear hs=JnE&hl=en&lr=

Ce fichier a le même format qu'un fichier log, à la seule exception que les adresses IP ont été anonymisées pour des raisons de confidentialités.

Mais malgré qu'il soit anonymisé ce fichier nous donne la structure des colonnes d'un fichier log.

A partir du tableau et avec les bibliothèques Python, on a isolé les colonnes pour les répertorier :

```
(0, 'IP')
(1, 'Client Identity')
(2, 'ID')
(3, 'Time')
(4, 'GMT')
(5, 'HTTP request')
(6, 'Status Code')
(7, 'Data(Bytes)')
(8, 'From Url')
(9, 'SE')
```

Il existe 10 colonnes (numérotées ici de 0 à 9) :

- IP : L'adresse IP du système qui envoie une requête au serveur
- Client Identity : le nom d'utilisateur (ou le nom du système client)
- ID : le login HTTP (en cas de connexion par mot de passe)
- Time : La date et l'heure de la requête
- GMT : Le fuseau horaire
- Http request : la méthode utilisée dans la requête (GET, POST, etc.) et le nom de la ressource Web demandée (l'URL de la page demandée),
- Status Code : le statut de la requête i.e. le résultat de la requête (succès, échec, erreur, etc.),
- Data : la taille de la page demandée en octets.
- From Url : La page web de provenance
- SE : le navigateur et le système d'exploitation utilisé par le client.

Beaucoup de données sont manquantes dans ce fichier. Certaines sont incomplètes pour des raisons évidentes, comme la page de provenance qui peut être vide si la requête est directe. Mais il y a aussi des colonnes qui sont complètement vides : Client Identity et ID

En dehors de ce fichier on a besoin de données exploitables pour effectuer nos traitements, on va maintenant regarder le 2e fichier : apres.txt

On procède de la même manière pour transformer le fichier texte en tableau sur Jupyter.

Visualisation du fichier apres.txt

```
url = "./apres.txt"
import pandas
full_df2=pandas.read_csv(url, sep=" ", encoding="utf-8")
full_df2.columns =['Date', 'Heure', 'Client Identity', 'IP','Visited Site', 'Status Code','Data(Bytes)']
full_df2
```

	Date	Heure	Client Identity	IP	Visited Site	Status Code	Data(Bytes)
0	2015-11-16	22:34:49	usep-pinchinades	68.180.229.29	associations.paysdaixassociations.org	500	12562.0
1	2015-11-16	22:35:03	association-regionale-pour-le-developpement-de...	68.180.228.251	agneaux-50.ville.mygaloo.fr	301	171.0
2	2015-11-16	22:35:05	association-regionale-pour-le-developpement-de...	68.180.228.251	agneaux-50.ville.mygaloo.fr	500	1968.0
3	2015-11-16	22:35:14	ascair	68.180.230.237	paris-75.ville.mygaloo.fr	301	156.0
4	2015-11-16	22:35:16	ascair	68.180.230.237	paris-75.ville.mygaloo.fr	500	1968.0
...
360400	2016-03-09	08:19:28	el-flamenco-vive-2	68.180.229.29	associations.paysdaixassociations.org	302	421.0
360401	2016-03-09	08:20:49	jack-daniel-acoustic-2	195.221.156.31	amiens-80.ville.mygaloo.fr	301	31.0
360402	2016-03-09	08:20:49	jack-daniel-acoustic-2	195.221.156.31	amiens-80.ville.mygaloo.fr	200	750.0
360403	2016-03-09	08:21:06	ecole-de-musique-sonat-aix-sonat-aix	208.115.111.66	associations.paysdaixassociations.org	301	343.0
360404	2016-03-09	08:22:23	usep-raimu-2	208.115.111.66	associations.paysdaixassociations.org	404	3234.0

Le premier constat qu'on peut faire est la taille du fichier, il contient plus 360 000 lignes de données, ça va avoir un impact sur le choix des solutions dans notre projet.

Dans ce fichier les données sont plus compactes et complètes.

La répartition des colonnes est différentes :

La date et l'heure sont premier suivi par le nom du client, l'ip, la page web visitée, le code état et la quantité de données.

La différence majeure que va nous apporter ce fichier est la présence d'adresses IP réelles, cela va nous permettre de faire une analyse sur la localisation du client à partir de son IP.

On va voir cela dans la partie suivante dédiée à la solution choisie pour convertir les IP en adresses physique reconnaissables.

4 Choix des solutions techniques

4.1 Principe du Géocodage

Le géocodage est l'identification d'une position géographique avec ses deux points cardinaux : latitude et longitude.

A partir d'une adresse IP il est possible de connaître des informations sur l'origine du réseau qui fournit la connexion au client à partir de ses informations il est possible de connaître le lieu à partir duquel le système client se trouve.

On peut ainsi en déduire le pays du client et également la localité (ville, commune ...).

Ainsi on a également le géocodage du client.

N.B. Il y a évidemment un risque d'erreur avec cette méthode, l'utilisation d'un VPN par exemple donnera des informations erronées.

On a utilisé un web service de conversion IP / géocodage pour avoir les données suivantes sur un échantillon de nos données

	IP	lat	lon	City
0	86.192.138.5	50.2922	2.78040	Arras
1	188.143.232.21	59.8761	30.43390	St Petersburg
2	136.243.36.93	49.0976	12.48690	Falkenstein
3	188.143.232.34	59.8761	30.43390	St Petersburg
4	176.148.125.134	48.9683	2.24820	Sannois
5	68.180.229.90	47.1988	-119.84260	Quincy
6	188.143.232.22	59.8761	30.43390	St Petersburg
7	90.51.246.245	43.6323	3.89350	Castelnau-le-Lez
8	68.180.230.177	47.1988	-119.84260	Quincy
9	68.180.229.32	47.1988	-119.84260	Quincy
10	68.180.230.237	47.1988	-119.84260	Quincy
11	136.243.36.93	49.0976	12.48690	Falkenstein
12	37.162.182.205	45.4642	9.18998	Milan
13	188.143.232.11	59.8761	30.43390	St Petersburg
14	188.143.232.62	59.8761	30.43390	St Petersburg
15	89.92.73.17	48.8871	2.27450	Neuilly-sur-Seine
16	68.180.230.237	47.1988	-119.84260	Quincy
17	188.143.232.24	59.8761	30.43390	St Petersburg
18	136.243.36.82	49.0976	12.48690	Falkenstein
19	82.226.89.14	48.9097	2.44640	Bobigny

Dans la partie suivante on va voir de quel web service il s'agit.

4.2 Solution de géocodage.



Le web service IP API nous permet de connaître le géocodage d'une adresse IP. Cela signifie qu'on peut connaître la latitude et la longitude correspondante à chaque adresse IP puis d'en extraire une adresse concrète.

LATITUDE	LONGITUDE
43.29833984375	5.383220195770264

IP Location: France (FR)

CITY, STATE	ZIP CODE
Marseille 01 (Provence-...)	13001

LANGUAGE	TIME ZONE
French (FR)	CET

CALLING CODE	CURRENCY
+33	Euro (€)

LOCATION CONNECTION SECURITY

Dans l'exemple ci-dessus, depuis le site ipapi.com , on a analysé l'IP 89.156.240.127

On obtient les coordonnées (43.298 , 5.383) correspondantes à un utilisateur en France à Marseille.

Dans notre projet, on va faire des requêtes api pour extraire les géocodes et les localisations. On peut tester ce fonctionnement directement depuis la page table des IP.

Comme dans l'exemple ci-dessous on voit que l'IP **193.248.56.75** est situé à Paris (lat 48.8323 , lon 2.4075).

Chercher une localisation spécifique :

193.248.56.75

Afficher

Cette adresse IP : 193.248.56.75 est localisé à Paris aux coordonnées suivantes :

latitude : 48.8323

longitude : 2.4075

Parfois la localisation est assez approximative, mais elle permet toutefois de connaître la ville correspondante à une adresse IP avec une grande justesse ce qui est suffisant pour analyser nos données.

4.3 Choix de la technologie d'application web.

Le développement de l'application web se fera avec le framework Flask.

Flask est un micro framework open-source de développement web en Python. Il est classé comme microframework car il est très léger.

Nous avons choisi d'utiliser Flask car ils reposent entièrement sur les bibliothèques et les outils de Python qui sont les plus efficaces pour traiter des données brutes.

Les anciens étudiants de Master ont réalisés leur précédent projet en utilisant Apache Tomcat mais le choix n'était pas le plus pertinent car ils leur fallu consacrer une grande partie de leur projet et de leur temps au développement d'une solution pour extraire, séparer et réécrire les données du fichier textuel en données facilement manipulable par la machine.

Or en utilisant les outils déjà intégrés sur Python et avec la bibliothèque Panda de python on effectue du traitement de données rapidement avec sans développer d'outils intermédiaire.

Flask permet également de générer des pages web légères, donc rapides et qui consomment peu de mémoires.

L'autre raison qui nous a poussé à choisir Flask sont nos compétences avec cet outil qu'on a manipulé dans un autre projet à Polytech.



4.4 Comparaison des services de cartographie.

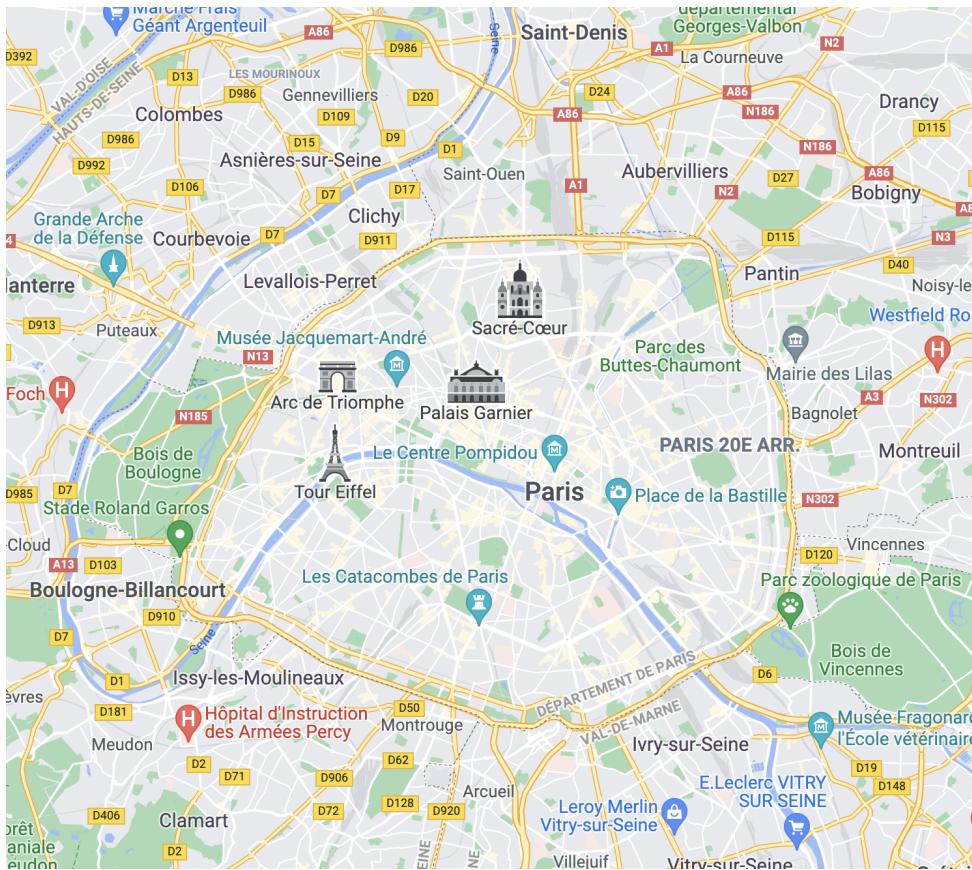
Il existe de nombreux services de carte, et chacune a ses propres particularités, avantages et inconvénients.

Nous avons sélectionné les 3 services les plus populaires que sont **Google Maps**, **OpenStreetMaps** et **MapBox**. Nous allons les analyser en détails pour décider de la solution à retenir pour notre projet.

Pour chaque solution on va voir :

- une description de l'outil
- les entreprises ou services qui l'utilisent
- les avantages
- les inconvénients

4.4.1 Google Maps

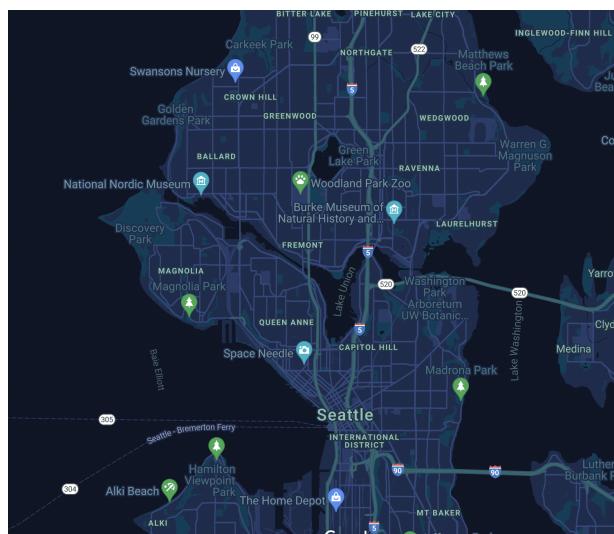
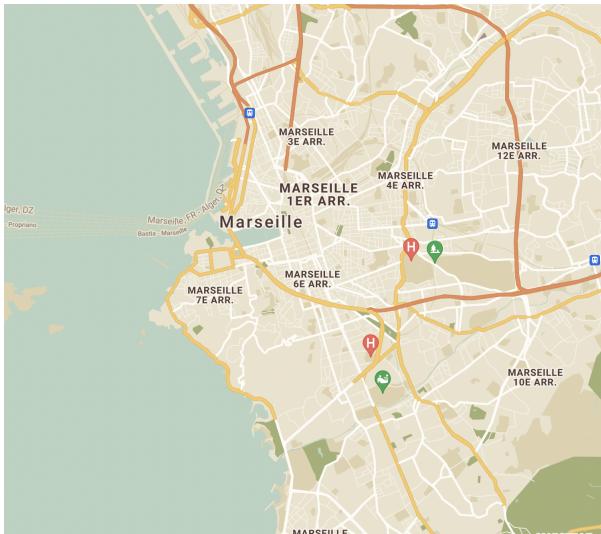


Google Maps est le plus célèbre SDK d'intégration de carte dans une application. Il est connu pour son énorme base de données de lieux et de routes, et il dispose de tous les outils de Google Maps : images, informations sur les lieux, itinéraires.

Ce SDK est identique à celui utilisé pour un usage public, cela signifie que les utilisateurs seront totalement à l'aise avec l'ergonomie de cette carte puisqu'ils ont l'habitude de l'utiliser dans le site de Google.

L'apparence de la carte est personnalisable et peut être choisie par le concepteur, 6 thèmes sont disponibles avec la possibilité de choisir la densité des informations à afficher.

Beaucoup de grandes entreprises l'utilisent comme Uber, Bolt et Tripadvisor.



Voici par exemple ci-dessus 2 exemples de thèmes qu'on peut utiliser avec l'API Google Maps

Avantages :

- Un certain nombre de requêtes est gratuit.
- Une Base de donnée très riche de lieux et de routes
- Géocodage (transformation des coordonnées latitude/longitude en adresses).
- La possibilité d'avoir une vue sur la rue.
- Une personnalisation visuelle.

Inconvénient :

- Un temps de chargement important des pages
- Une consommation importante de la mémoire qui peut même diminuer rapidement la batterie de l'appareil.
- Un coût très important des appels API (qui peut atteindre 14\$ Les 1000)

4.4.2 OpenStreetMap



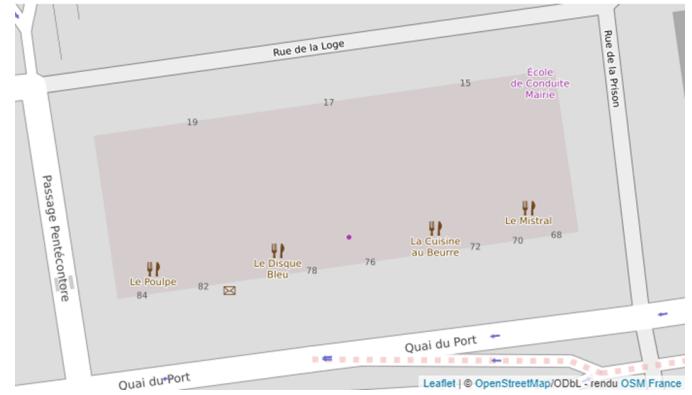
OpenStreetMap est une carte gratuite il s'agit d'un projet open source proposé par des bénévoles du monde entier qui contribue à enrichir la carte avec des lieux et des routes régulièrement.

Le service de cartographie ainsi que la base de données sont entièrement publics.

Il n'est pas possible d'interroger l'API sur les données de navigation en temps réel pour suivre un itinéraire. Le service se limite à l'affichage de cartes et leur édition (pour afficher des marqueurs par exemple comme dans notre projet).

Beaucoup de services l'utilisent comme : Moovit, OpenTouchMap

OpenStreetMap propose 20 niveaux de zoom de 1 à 20 : le niveau 1 correspond à la carte planétaire, le niveau 20 permet de voir les détails disponibles sur une zone d'une dizaine de mètres ce qui est pratique pour voir en détails certains lieux disponibles (établissements, restaurants ...).



Avantages :

- Une base de données open source
- Une API gratuite

Inconvénient :

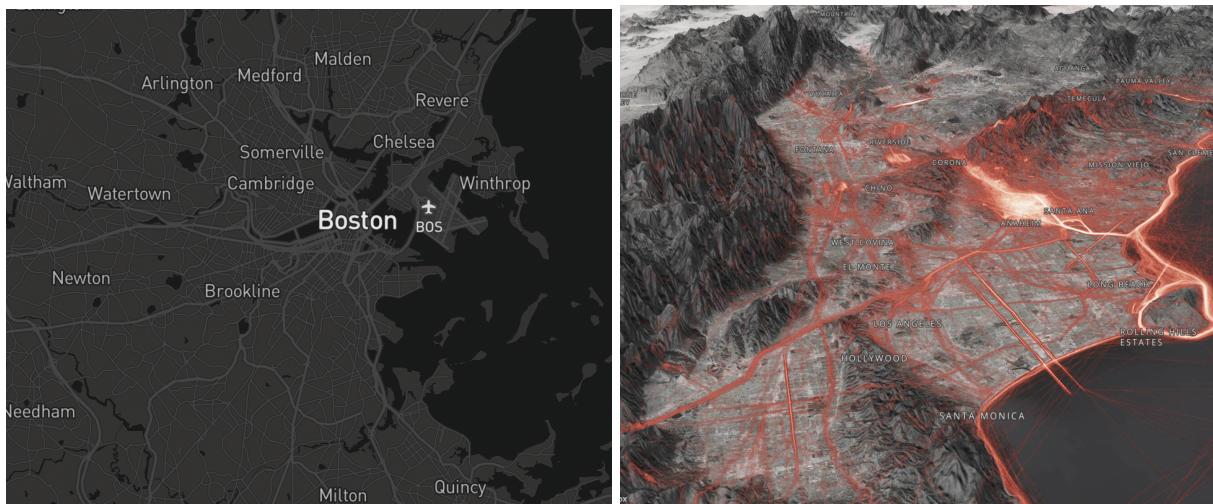
- Destiné principalement à des fins d'édition de cartes.
- Les requêtes excessives d'API peuvent entraîner un ralentissement
- Les noms de localités sont uniquement dans les langues d'origine (non traduites)
- Peut ne pas s'intégrer avec certains outils.

4.4.3 MapBox



MapBox est un service de cartographie flexible et personnalisable qui peut être facilement intégré dans les applications web ou mobiles. Il permet aux développeurs de styliser entièrement leur carte et il donne accès à des informations de localisations assez complètes. Le service est payant mais il propose un niveau de gratuité assez intéressant qui peut convenir aux applications avec un petit volume d'utilisation.

Le niveau de personnalisation des designs est très avancé et très complet ce qui permet par exemple des rendus visuels en reliefs.



MapBox est surtout connue car elle est utilisée par des réseaux sociaux comme Facebook ou SnapChat.

Avantages :

- Gratuit jusqu'à un certain volume d'appels
- Facile à intégrer et une documentation très complète
- La personnalisation des cartes.
- Des cartes visuellement agréables pour les utilisateurs.

Inconvénients :

- Un service payant (4\$ les 1000 appels API)
- Orienté principalement pour les grands projets.
- Dispose de moins de données sur les lieux et les établissements que Google Maps.

4.5 Solution de carte retenue.

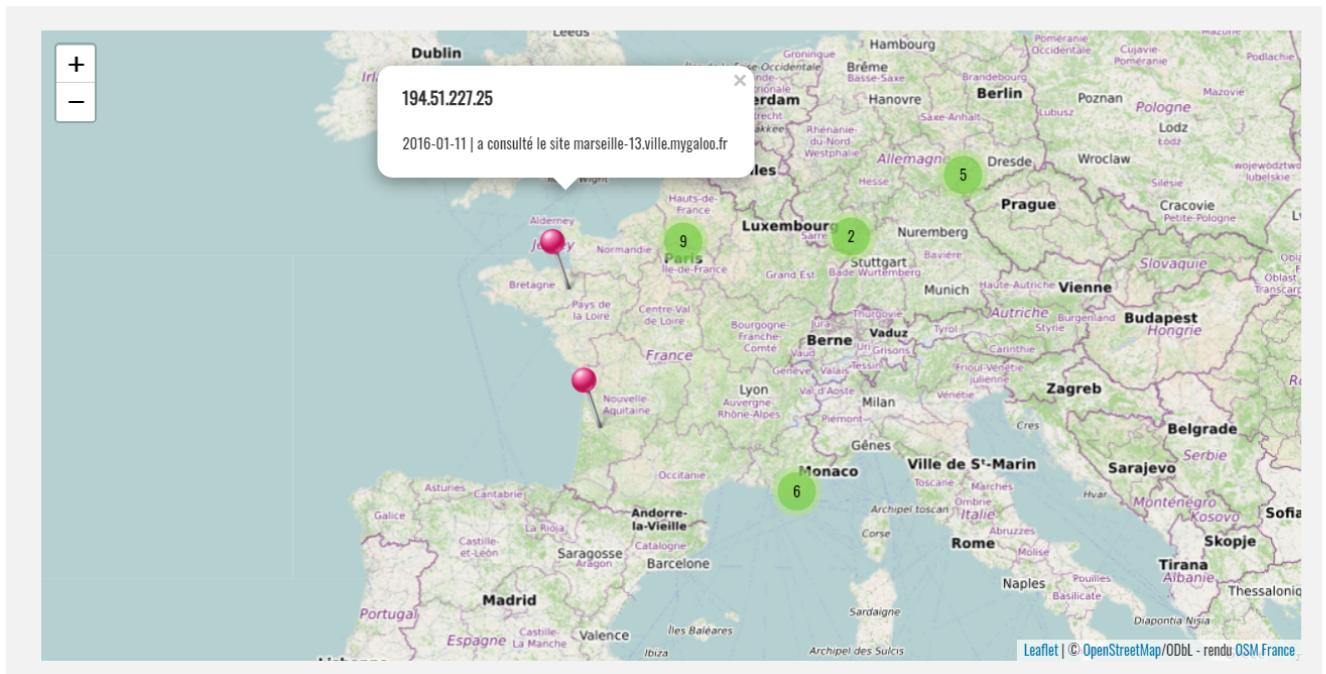
On a finalement choisi d'utiliser OpenStreetMap.

Ce service dispose des outils suffisants pour notre application web :

- ajout de marqueurs
- personnalisation des informations affichées sur la fenêtre pop up d'un marqueur.
- affichage de clusters pour assembler les marqueurs trop poche, ce qui permet de ne pas encombrer l'affichage

Mais la raison principale qui nous a fait préférer ce service est sa gratuité. Notre base de données dispose d'énormément d'éléments, si on avait utilisé Google Maps ou MapBox on aurait rapidement dépasser le forfait gratuit.

L'utilisation de OpenStreetMap avec beaucoup de données peut entraîner des ralentissements ou le non fonctionnement de certains éléments de la carte (marqueurs ...) mais cela nous permet de rester dans le cadre de notre projet universitaire.



5 Spécifications de l'application réalisée

5.1 Une interface web intuitive

L'application finale est sous la forme d'un site web, elle est hébergé à cette adresses :

<http://cazeq.pythonanywhere.com/>

Elle dispose d'une organisation des pages et d'une interface selon les normes des sites webs les plus réponduis :

Une page d'Accueil, un menu de navigation et les pages de contenus.

Bienvenue sur Geolocalized Web Data Mining !



Projet de fin d'étude réalisé pour POLYTECH Marseille

Réalisé par : Ali Azzoug et Ismael Mébrouki

Professeurs : Nicolas Durand et Mohamed Quafafou

Accueil

Statistiques

Carte

Base de données

Table des IP

La page d'accueil donne une présentation rapide du site et le menu permet de naviguer entre les 5 pages disponibles.

5.2 Analyses statistiques sur les données.

La page statistiques met en avant les éléments les plus récurrents dans le fichier log.

Les visiteurs les plus actifs (utilisateurs ou systèmes), les sites les plus visités, les pages les plus actifs des sites webs et les jours avec le plus de fréquentation.

Clients les plus actifs :

	Nombre de pages visitées
IP	
146.185.234.48	51034
208.115.111.66	20311
68.180.229.29	13870
208.115.113.83	11761
188.143.232.11	11517
188.143.232.24	11272

Sites les plus visités :

	Sites les plus visités
Visited Site	
associations.paysdaixassociations.org	61608
paris-75.ville.mygaloo.fr	13709
marseille-13.ville.mygaloo.fr	8544
actu.dignelesbains.fr	7957
brest-29.ville.mygaloo.fr	4994
courbevoie-92.ville.mygaloo.fr	4170

Pages les plus consultés (tout sites confondus) :

	Pages les plus consultées
Consulted Page	
association-des-victimes-du-stalor-et-du-cholstat-2	569
association-soleil-13127	515
amistad-de-la-salsa	302
galoo	274
comite-de-jumelage-de-la-ville-de-houe-helair_civhha	210

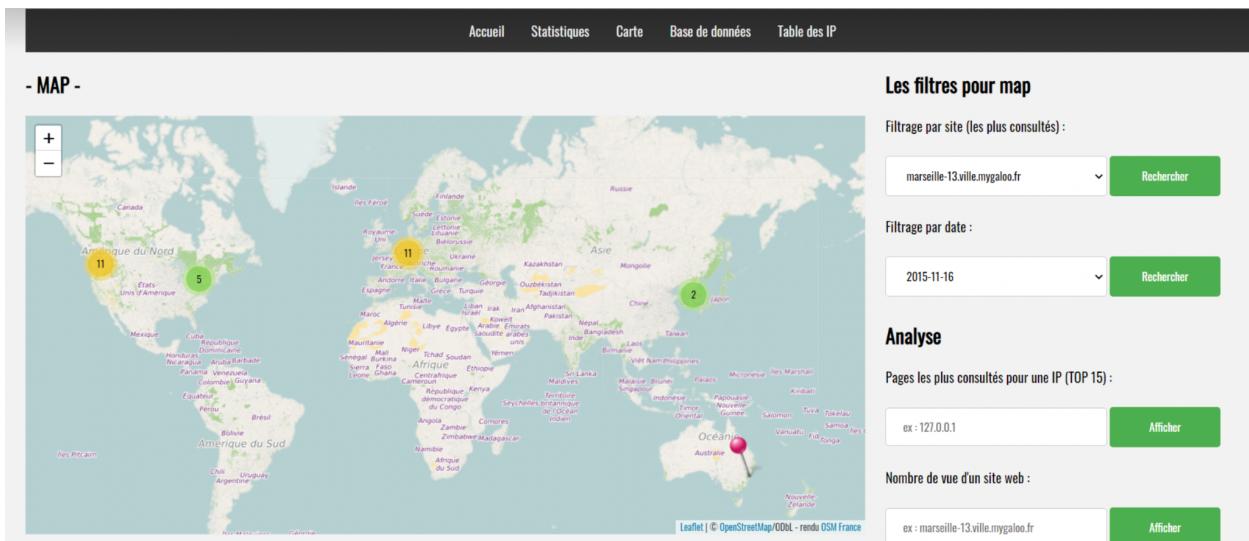
Jours avec le plus de visites :

	Nombre de visites selon le jour
Date	
2016-02-01	16152
2016-02-03	13737
2016-01-31	11434
2016-02-10	10502
2016-02-02	7781
2015-12-09	7532

5.3 Visualisation sur la carte

Sur la page Carte, il est possible de voir la localisation des visiteurs enregistrés dans le fichier log, et il est également possible de les filtrer soit par site web visité, soit par date de visite.

Les résultats apparaissent directement sur la carte les plus proches sont rassemblées en cluster, le nombre et la couleur du cluster indique leur densité.



5.4 Connaître le nombre de vues d'un site spécifique

Dans la page Map il est possible de vérifier le nombre de vues uniques d'un site enregistré dans le fichier log. Cette analyse est cependant limitée aux sites les plus important en termes de trafic.



5.5 Pages les plus consultées par un utilisateur

Il est possible de voir quelles pages ont été les plus visitées par un utilisateur en entrant l'adresse IP de l'utilisateur qu'on veut analyser, un tableau apparaît avec la liste des sites visitées classés par nombre de vues de cet utilisateur

Analyse	
Pages les plus consultés pour une IP (TOP 15) :	
5.9.112.6	Afficher
Consulted Page	Nombre de fois que la page a été consulté
boxing-club-ben-dhaou-bcbd-2	10
ata-artistes-techniciens-associes-compagnie-petia-vaillant	7
pet-j-videau-st-loubes	6
espace-farina	4
l-eco-logique-stephanie-olivier-	

5.6 Suivi du parcours utilisateur

Une chose intéressante lorsqu'on effectue une analyse des données pourrait être d'analyser un utilisateur particulier, ça peut être un client qui a passé commande sur un site ecommerce pour comprendre ses motivations, le suivi du tunnel d'utilisation imaginé par le créateur du site, ou le suivi d'un visiteur malveillant.

Pour cela, on a créé une solution qui permet le suivi d'un utilisateur.

Il suffit de rentrer l'adresse IP, un tableau s'affiche alors avec l'ensemble des interactions de cette utilisateur classé chronologiquement.

IP tracking :							
						Afficher	
	Date	Heure	Consulted Page	IP	Visited Site	Status Code	Data(Bytes)
1343	2015-11-17	10:05:34	elections-samuel-serre-association-financiere	5.9.112.6	saint-gilles-30.ville.mygaloo.fr	301.0	15.0
1344	2015-11-17	10:05:34	elections-samuel-serre-association-financiere	5.9.112.6	saint-gilles-30.ville.mygaloo.fr	500.0	406.0
4381	2015-11-17	19:25:57	association-style-et-toiles	5.9.112.6	lambersart-59.ville.mygaloo.fr	301.0	1031.0
4382	2015-11-17	19:26:00	association-style-et-toiles	5.9.112.6	lambersart-59.ville.mygaloo.fr	200.0	2406.0
5752	2015-11-17	23:46:59	assemblee-chretienne-de-la-parole-parlee	5.9.112.6	saint-brice-sous-foret-95.ville.mygaloo.fr	301.0	31.0
	2015-		assemhlee-chretienne-		saint-hrice-sous-foret-		

5.7 Fonctionnalités annexes.

La page base de donnée donne un aperçu des éléments contenus dans le fichier log affichés sous la forme d'un tableau, en raison du volume très important des données on se limite seulement à 1000 valeurs.

Tableau contenant 1000 valeurs de la base de données :

	Date	Heure	Consulted Page	IP	Visited Site	Status Code	Data(Bytes)
329546	2016-02-22	05:39:42	rebirth13.le-korigan-live-club-live-club	194.187.168.218	aix-en-provence-13.ville.mygaloo.fr	404	46.0
151016	2015-12-27	14:02:25	agiras	80.12.39.78	lormont-33.ville.mygaloo.fr	200	265.0
301224	2016-02-10	13:32:46	dph-diffusion	188.143.232.26	thorigny-sur-oreuse-89.ville.mygaloo.fr	404	1062.0
216916	2016-01-24	21:02:55	la-chapelle-des-chesnelierres-2	78.242.14.198	chateaubourg-35.ville.mygaloo.fr	301	46.0
56121	2015-12-03	06:11:18	operation-parisis-propre-opp-2	146.185.234.48	franconville-95.ville.mygaloo.fr	500	218.0
118124	2015-12-15	13:37:17	les-arts-de-vivre-arts-et-pedagogie-au-service-du-mieux-etre-2	109.26.209.86	saint-hilaire-de-riez-85.ville.mygaloo.fr	301	31.0
3742	2015-11-17	18:48:49	compagnie-les-archers-du-born	8.37.71.17	mimizan-40.ville.mygaloo.fr	200	734.0
202259	2016-01-17	04:39:56	fans-de-gospel-2	136.243.17.161	port-de-bouc-13.ville.mygaloo.fr	301	31.0
272521	2016-02-11	11:00:01	l'association_des_accidentes_de_la_vie_fnath_cortinna_du_csr	196.243.36.92	associations_novedavassociations_nra	301	15.0

La page table des IP permet de voir les correspondances entre les adresses IP et leur géocodage ainsi que la localisation de certaines adresses IP contenues dans nos données.

Mais il est surtout possible de rechercher directement la localisation de n'importe quelle adresse IP directement dans la barre de recherche, on obtient alors la latitude, longitude et la ville correspondante à une IP.

Chercher une localisation spécifique :

89.156.221.127

Afficher

Cette adresse IP : 89.156.221.127 est localisé à Marseille aux coordonnées suivantes :
 Latitude : 43.2947
 Longitude : 5.4331

Tableau contenant des exemples d'adresse IP converties :

	IP	lat	lon	City
0	86.201.234.47	43.6046	1.445100	Toulouse
1	80.12.59.127	43.1167	-0.066700	Poueyferre
2	78.226.184.199	43.3736	5.354700	Marseille
3	90.8.34.84	41.9199	8.742400	Ajaccio
4	207.189.232.72	60.7180	-135.047500	Whitchorse

6 Conclusion

Dans ce projet on a pu découvrir les différents aspects du web usage mining, on a effectué l'exploration, le traitement et l'analyse d'un fichier log d'un serveur web et établi des visualisations de son contenu sur une carte.

On a construit et déployé une application web de notre projet et utilisé différentes API et web services pour mettre en place l'application finale. On a su comparer différentes ressources existantes pour choisir celles qui sont les plus adaptées aux besoins de notre système.

On a effectué une analyse détaillée des différentes solutions de carte et de géolocalisation qui nous seront très utiles dans nos projets professionnels futurs.

Malgré la complexité de ce projet on a pu construire une solution pour analyser les données d'usage d'un site. D'autres évolutions sont possibles : le diagnostic des performances des sites, l'analyse de l'origine du trafic, ou même d'étendre le traitement à n'importe quelle serveur.