Rapport :
Fouille de données sur les données personnelles Google (Google Takeout)

Fatima Souafi, Adeline Kalic, Fanny Ponce, Arnaud Duvermy et Elsa Mendes

Université Claude Bernard Lyon 1

TABLE DES MATIÈRES

1. INTRODUCTION

L'objectif de notre projet est la sensibilisation du grand public aux informations personnelles collectées par leur téléphone portable ou leur ordinateur au quotidien par Google. Cela grâce à la mise en place d'un workflow permettant d'extraire et d'analyser ces données.

- **Description du jeu de données**

Nous avons pu exporter et télécharger des données à partir du site de Google
How to download your Google data - Google Account Help .
Différents choix d'exportation sont possibles. Voici ce qui peut être exporté: **Gmail** (mails et recherches), **Chrome** (historique de recherches, autofill (ajout automatique dans questionnaires par exemple)...), les documents de **Google Drive, Youtube** (videos consultées, abonnements, historique, commentaires, playlists …), des **données GPS** (historique des positions associées à des données temporelles), historique de navigation **Google Maps** (itinéraires demandés, recherches, inférence du mode de déplacement), **Google Fit**, **Google Play** Store, Play Livres, Play Film et TV, Google **Pay** ,Google **Photos**, des enregistrements audios de l'**Assistant Google**, les **contacts** téléphoniques et **Actualités** (topics et sources de sites consultés, notifications ignorées) ...

- **Objectifs**

Dans la suite de ce rapport seront présentés les traitements informatiques envisageables pour extraire de la connaissance à partir des données recensées par Google.
Ici, nous nous concentrerons sur les données : youtube (historique), navigation WEB (historique), Gmail, GPS et Google Fit.

Afin d'extraire de l'information de ces données, nous avons utilisé différents algorithmes notamment ceux disponibles dans le package scikit learn. Le choix de ces algorithmes s'est fait en fonction du type de données à analyser et du type d'information à extraire.

- **Plan d'assurance qualité**

Afin d'assurer la reproductibilité de la fouille des données recensées par Google, plusieurs notebooks Jupyter édités à partir de python3 ont été développés. Chaque notebook permet d'étudier le contenu d'un type de données (youtube, historique de navigation, données GPS, ...).
Le choix de créer plusieurs notebook est stratégique puisqu'il permettra une meilleure lisibilité du code et de s'adapter aux besoins/envies de chaque utilisateur.
La documentation interne du projet a été réalisée directement dans les fichiers en suivant les règles de bonnes pratiques informatiques via l'utilisation de commentaires. Les utilisateurs auront des informations sur les fonctionnalités des notebooks via la mise en place d'un README. Chaque notebook a été testé avec les données de plusieurs personnes du groupe.

2. DONNÉES GPS

- **Structure des données**

Les données analysées sont celles du fichier 'Historique des positions.json'. Dans ce fichier, encodé en json, sont recensées les positions GPS recueilli par Google. Ces positions sont associées à plusieurs variables. Aucune documentation officielle n'est disponible pour le google Takeout, il a fallu s'approprier les données et comprendre le sens de ces variables.

Une interprétation des variables contenu dans le fichier
(source: Analyze Your Google Location History: Exploring Data)

Variable	LatitudeE7	LongitudeE7	accuracy	timeStampMS	activité	confidence
Explication	Latitude de l'observation	Longitude de l'observation	Estimation Google de la précision	Temps en millisecond où l'observation a été faite	Inférence de l'activité par du machine learning Google	Score de confiance de l'inférence de l'activité

On notera que ce fichier peut être très volumineux en fonction du temps depuis lequel l'utilisateur a autorisé le recueil de ses données GPS. (ici 264 Mb, depuis 2015)

- **Extraction et filtrage des données**

En premier lieu, la librairie **pandas** a permis d'extraire et de manipuler les données. Pour comprendre le score d'accuracy des positions, nous avons observé sa distribution dans les données. Au regard de nos recherches, nous considérons qu'un score d'accuracy élevé possède une accuracy inférieure à 800. Après investigation, la grande majorité des données analysées possèdent une bonne *accuracy* (< 800) (***annexe 1***).

Afin de nettoyer et de réduire la taille des données nous retirons les coordonnées GPS avec une *accuracy* supérieure à 800. Puis nous nous intéressons aux activités inférées par Google pour une majorité de position GPS. Les activités inférées sont associées à un indice de confiance, allant de 0 à 100 (**annexe 2**). Au vue des distributions, nous considérons qu'un score proche de 100 correspond à une activité avec un bon support. Afin de nettoyer et de réduire à nouveau la taille des données nous retirons les coordonnées GPS associées à une activité possédant un score inférieur à 50.

	Origine	Après filtrage sur accuracy	Après filtrage des activités
Nombre d'instance GPS	734 398	630 205	271 872

- **Normalisation et variable "weight"**

Dans l'objectif de visualiser plus tard les données, les indices "accuracy" et de "confidence" apparaissent comme des variables intéressantes pour adapter l'importance accordée aux coordonnées GPS associées. Plusieurs investigations, ont été faite pour tenter d'adapter au mieux le poids accordé au coordonnées GPS lors de visualisation :

- "accuracy" normalisée seule (non montré ici)
- "confidence" normalisée seule (non montré ici)
- "weigth" = confidence_norm / accuracy_norm (partie résultats)

Les variables accuracy et confidence ont donc été centrées réduites grâce au StandardScaler de scikit learn, et ont ensuite permis de construire la variable "weight" défini comme tel :

weight = confidence_norm / accuracy_norm

Contrairement à l'indice de "confidence", l'utilisation de l'accuracy normalisé comme indice d'importance accordé au point GPS a donné des résultats probants. Pour les visualisations ci-dessous, nous avons utilisé la variable weight qui donnait aussi des bons résultats et permettait de tenir compte des deux indices fournis par le fichier analysé.

- **Extraction de connaissances**

Sur le modèle de l'outil web open source Location History Visualizer | Heatmap qui permet de réaliser une heatmap à partir des coordonnées GPS issue du fichier '*Historique des positions.json*'. Nous

avons implémenté une application basée sur le package folium pour visualiser les densités des coordonnées GPS conservées après filtrage effectué ci-dessus. Cette application a été intégrée au notebook grâce au package jupyter-dash.

En fonction de la densité des points, les clusters de la heatmap vont être coloré différemment, ainsi on définit un gradient de couleur du rouge vers le bleu : les points chauds de forte densité en rouge et les positions moins fréquentes en bleu. L'intérêt de ce package réside aussi dans le fait que les clusters définis par la heatmap vont s'adapter au niveau de zoom désiré par l'utilisateur. Grâce à l'application implémentée et au package Folium, il est désormais très facile de fouiller efficacement les données GPS, grâce aux boutons et listes déroulantes qui permettent de les filtrer. Ce package se révèle donc comme un outil idéal pour tenter d'extraire efficacement et simplement des connaissances à partir de nos données filtrées.

Notre objectif est ainsi d'évaluer si à partir d'hypothèses, il est possible d'inférer des conclusions sur l'utilisateur.

- **Résultats**

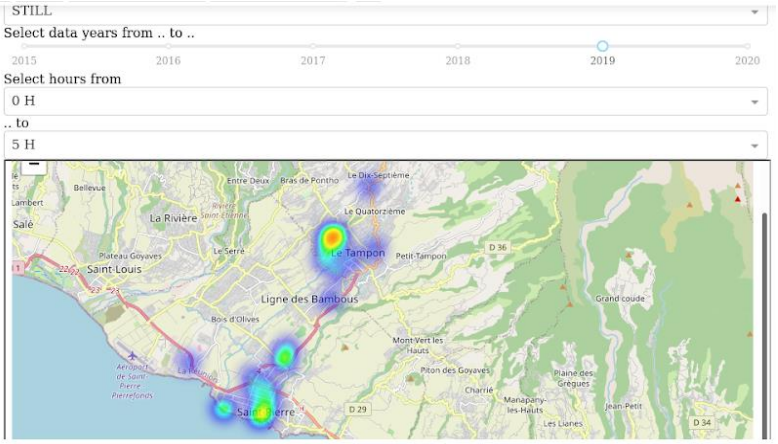
Le cas d'étude que nous étudierons ici, est l'île de la Réunion. Fréquentée pendant quasiment un an par l'un des utilisateurs de notre notebook, nous avons suffisamment de relevés GPS pour travailler convenablement. Qui plus est, nous pourrons juger de la pertinence des points relevés et des activités inférés puisque l'île présente la particularité d'être montagneuse. Le cœur de l'île est par conséquent accessible principalement par des chemins de randonnées pédestres.

Afin de juger de la qualité des activités inférées par Google et la véracité des filtres appliqués précédemment, nous avons observé la distribution des coordonnées GPS associées à différentes activités. Comme attendu, l'activité IN_VEHICULE redessine quasi parfaitement les axes principaux de l'île. En ce qui concerne l'activité ON_FOOT, la distribution est moins continue et se localise au niveau des villes principales ou encore au cœur de l'île. Ces deux activités semblent donc bien gérer par l'algorithme de machine learning Google, responsable de l'inférence des activités présumés de l'utilisateur. (***annexe 3***)

Pour autant, l'inférence de l'activité ON_BICYCLE semble plus compliquée pour l'algorithme. En effet, si le point chaud correspond à un trajet récurrent réalisé à vélo, les autres points sont vraisemblablement tous des faux positifs.
Enfin, nous nous intéressons à l'activité UNKNOWN qui reste assez obscure car les filtres appliqués aux données sur l'accuracy et la confidence n'ont étonnamment pas permis d'éliminer cette catégorie d'activités de nos données. On notera que ces 2 points correspondent à des emplacements stratégiques pour réaliser de l'autostop : on peut donc imaginer ici que l'algorithme a été perturbé par de nouvelles habitudes de conduite. (***annexe 4***)
Malgré la mise en évidence d'outliers dans nos données, les positions retenues post-filtrage retrouvées au niveau des points chauds semblent précises et correspondre au mode de vie de l'utilisateur. Nous validons par conséquent les filtres appliqués lors du nettoyage des données et l'utilisation de la variable weight pour adapter l'opacité des densités observées.

Conforté par notre approche et le contenu de nos données, nous avons ensuite tenté de retrouver la position probable du domicile de l'utilisateur.
Pour cela deux hypothèses ont été faites :

- L'utilisateur est souvent chez lui entre minuit et 5h du matin
- Une fois à la maison l'utilisateur est moins actif (activité STILL)



Visualisation des densités GPS filtrées à partir des à priori
(activité: STILL, entre minuit et 5h du matin en 2019)

Comme visible ci-dessus, les hypothèses précédentes, nous ont permis d'obtenir un point chaud qui correspond bien au domicile de l'utilisateur.

En dernière approche nous avons utilisé l'option "time series" du package folium afin de suivre heure par heure la position de l'utilisateur. L'idée de cette approche est d'affiner l'analyse afin d'étudier au plus près les habitudes de l'utilisateur. En effet, on associe pour chaque heure un point afin de suivre la position de l'utilisateur d'une heure à l'autre sur une période d'un mois (nécessité de réduire le nombre de données pour ce traitement pour éviter des crash).
Grâce à cette option, nous avons retrouvé la position du domicile, observé précédemment, mais aussi la position du lieu de travail (non montré ici) de l'utilisateur en s'intéressant aux positions récurrentes retrouvées aux horaires dites de "bureau" (du lundi au vendredi entre 9h/12h, 14h/16h). On notera que l'utilisateur n'était pas amené à beaucoup se déplacer pendant ces heures. Par manque de temps, cette option n'a pas été ajoutée à l'application mais reste disponible dans une cellule dédiée du notebook.

Au vu des résultats précédents, il est possible de retrouver les habitudes d'un utilisateur grâce à l'approche par densité du package folium et notre application. Pour le moment, notre approche nécessite un traitement des données au cas par cas. Mais il semble assez évident que Google possède les moyens financiers et humains pour mettre au point des algorithmes de machine learning capables de traiter ces données et ainsi obtenir des conclusions similaires aux nôtres.

3. DONNÉES NAVIGATION WEB

- **Structure des données**

Les données analysées sont celles du fichier 'MonActivité.html' se trouvant dans le dossier 'Takeout/Mon activité/Chrome'. Ce fichier comporte l'url consultée, la recherche effectuée ainsi que le jour, le mois et l'année correspondante. L'analyse aurait également pu se porter sur le fichier 'BrowserHistory.json'.

- **Parser le fichier html**

Dans un premier temps, le fichier 'MonActivité.html' a été parsé afin d'en extraire les informations pertinentes contenues dans des balises html. Cette étape a été réalisée à partir du fichier python 'parser_MonActivité.py'. Une fois celui-ci exécuté, nous disposons d'un fichier csv 'D_searchData.csv' permettant de commencer notre analyse dans notre notebook Jupyter.

- **Prétraitement du texte**

Nous allons principalement nous intéresser à la colonne 'Recherche' du dataframe. Celle-ci contient des phrases correspondant à la recherche effectuée sur le navigateur de recherche Google. L'étape primordiale avant l'analyse est le prétraitement du texte. **TextHero** permet de travailler rapidement et efficacement sur un ensemble de données textuelles, de plus, il a été conçu pour être utilisé avec

la librairie Pandas. Premièrement, la fonction clean() a été appliquée, elle permet d'uniformiser le texte en minuscule, de supprimer les accents, les chiffres, les ponctuations, les mots vides.

	Recherche	searchMonth	searchYear
25	Résultats pour "intelligence artificielle" - S...	mai	2020
26	Résultats pour "intelligence artificielle" - S...	mai	2020
27	Bibliothèques Universitaires	mai	2020
28	Aix-Marseille Université - Authentification	mai	2020
29	Apprentissage machine - Clé de l'intelligence ...	mai	2020

	Recherche	searchMonth	searchYear
25	resultats pour intelligence artificielle summo...	mai	2020
26	resultats pour intelligence artificielle summo...	mai	2020
27	bibliotheques universitaires	mai	2020
28	aix marseille universite authentification	mai	2020
29	apprentissage machine cle de l intelligence ar...	mai	2020

Les mots vides se réfèrent généralement aux mots les plus courants dans une langue, comme "un", "des", "le", ... Ces mots n'ont pas de signification importante et sont généralement supprimés des textes. Normalement, la méthode clean possède la fonction remove_stopwords qui permet de les supprimer. Il semble que cette fonction ne reconnaisse pas les mots vides français. NLTK (Natural Language Toolkit) en python a une liste de mots vides stockés dans 16 langues différentes, nous allons donc utiliser ce corpus de mots vides en français. Pour ce faire, nous utilisons la tokenisation, un processus de découpage d'un texte en plus petits morceaux, appelés tokens, qui dans notre cas, correspondent aux mots. Ce qui nous permet d'avoir une liste constituée de mots séparés par des virgules. Cette étape est nécessaire pour comparer la liste de mots vides de NLTK à notre liste de mots et ainsi, supprimer les mots vides.

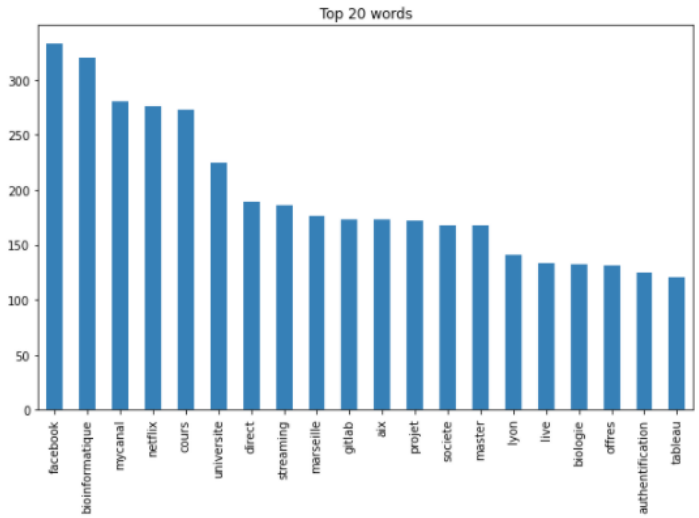
28	[aix, marseille, universite, authentification]	mai	2020
29	[apprentissage, machine, cle, de, l, intellige...	mai	2020

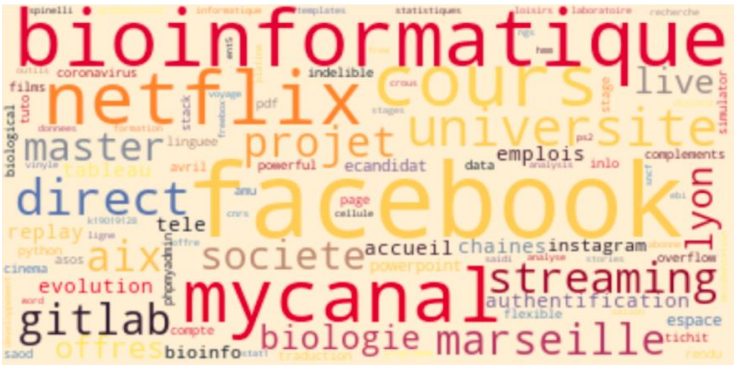
28	aix marseille universite authentification	mai	2020
29	apprentissage machine cle intelligence artific...	mai	2020

- **Résultats**

Nous pouvons visualiser par exemple les 20 mots les plus fréquemment utilisés. Ainsi, nous pouvons en déduire un profil type de l'utilisateur. Dans cet exemple, les mots 'cours', 'université', 'master' rassemblent un profil type étudiant.

Une autre visualisation intéressante est celle du wordcloud qui, selon la fréquence du mot, est représenté d'une plus grande taille si sa fréquence est importante. Ce qui pourrait être intéressant serait de filtrer une période et par rapport aux mots retrouvés en déduire l'actualité de cette période. Par exemple, la période du premier confinement liée à la pandémie de Covid-19.





4. DONNÉES D'HISTORIQUE YOUTUBE

- **Structure des données**

Les données Youtube collectées sont très diverses : nos abonnements, nos playlists, nos commentaires postées, nos historiques... Dans le cadre de ce projet nous nous sommes intéressées aux historiques de recherches effectuées et de vidéos regardées.
Ces fichiers peuvent être téléchargés avec google takeout au format html (défaut) ou json, nous avons choisi le format json pour plus de rapidité (l'extraction des données à partir d'un html est plus longue et complexe).

La composition des fichiers est variable mais les colonnes qui nous intéressent sont des données essentielles et toujours présentes :

- title : Donne la recherche effectuée ou le titre de la vidéo regardée
- titleUrl : Correspond à l'url pour effectuer la recherche ou l'url de la vidéo
- time : Date et heure de recherche ou visionnage

- **Préparation des données**

L'analyse des deux fichiers d'historique s'est effectuée de la même manière. Les données ont été extraites dans un dataframe à partir des fichiers json.

Elles sont ensuite recoupées et nettoyées. Nous ne gardons que les données d'intérêt pour l'analyse (title et time et titleUrl).
Les informations contenues dans la colonne title sont ensuite « nettoyées » pour permettre leurs comparaisons par la suite, cela de manière similaire que pour le traitement des recherches web (partie précédente) et retrait des mentions « Vous avez recherché » et « Vous avez regardées » ajoutées dans les fichiers.

Vous avez recherché bZwDgKGBc88	→	supprimé
Vous avez recherché musique pour travailler	→	musique travailler
Vous avez recherché I love you Olivia !	→	love olivia

La colonne time est traitée pour obtenir la date et l'année.

Le format initial est : 2021-01-10T13:13:59.376Z et nous récupérons 2021-01-10 et 2021

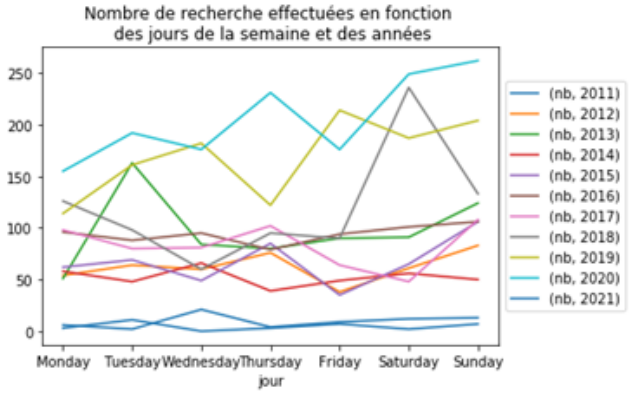
Avec l'option date time nous attribuons ensuite à chaque date son jour de la semaine correspondant.
2021-01-10 → Sunday

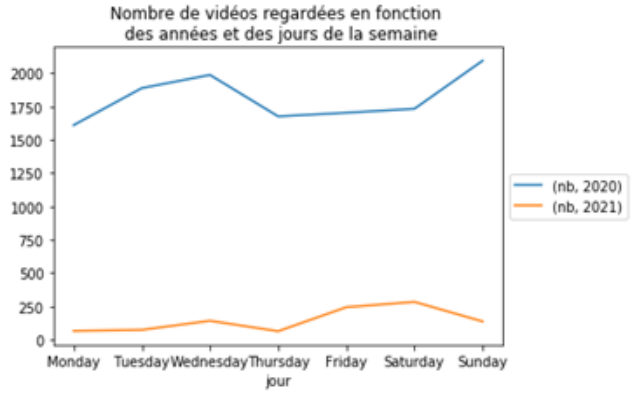
Séparément une liste est créée avec les recherches séparées mot à mot.
['same love','recette soupe', 'same world' , …] → ['same', 'love','recette', 'soupe', 'same', 'world' , …]

- **Extraction et visualisation des informations**

Avec la fonction value_count nous calculons les fréquences des recherches, des mots les plus recherchés et du nombre de fois qu'une vidéo a été regardée par l'utilisateur. Nous retournons un compte rendu de ces informations.

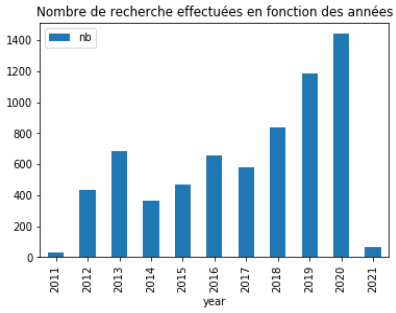
Nous réalisons ensuite un clustering des données en fonction des années, des jours et les deux, cela grâce aux fonctions group_by et value_count.





Nous avons constaté que le nombre de vidéos regardées est beaucoup plus important que le nombre de recherches effectuées, ce qui était attendu. Mais aussi que les plages temporelles des informations fournies sont très différentes dans les deux historiques. Ci-dessus par exemple, pour les recherches plage de 10 ans et pour les vidéos regardées plage de 1 année et 1 mois alors que les données ont été téléchargées au même moment.

L'évolution du nombre de recherches effectuées peut en apprendre beaucoup sur les habitudes d'une personne.



Avec l'outil wordcloud nous avons également créé un nuage avec les mots les plus utilisés dans les recherches.

Nous avons remarqué que le comptage du nombre de mots avec l'outil interne wordcould et value_countest différents. Par exemple pour nos données test, music était comptabilisé 23 ou 27 fois.

De plus, l'affichage du wordcloud ne reflète pas exactement tous les mots les plus utilisés sinon le mot épisode devrait apparaître car il était dans le top 10 des mots les plus exprimés dans les résultats fournis par wordcloud. On peut voir que les mots sont souvent couplés alors qu'une liste de mot individuelle est donnée, il doit y avoir un système de reconnaissance des mots les plus fréquemment utilisés ensemble ou cela se fait si leurs comptages sont identiques ou très proches.

La question pour l'analyse de données Youtube n'est pas de trouver des choses confidentielles mais de voir des habitudes ou des préférences. Nous avons tous une part de secret que nous n'avons pas envie de partager, notamment nos recherches Youtube.

Seriez-vous prêt à laisser l'accès à d'autres personnes à ces données ? Cela est en fait déjà le cas, youtube premièrement pour vous proposer des choix de vidéo, des pubs. Google a également un accès et utilise ces données, mais ce ne sont probablement pas les seuls.

- **Pour aller plus loin dans les historiques Youtube**

Les données extraites dans ce projet ne sont qu'une partie de ce qu'il est possible de voir, pour aller plus loin il est possible d'utiliser une API. Une API youtube YouTube Data API Overview est disponible et permet d'obtenir encore plus d'informations, par exemple de classer les vidéos par groupes (musique, cuisine, sport, série, ...). Son implémentation est difficile à mettre en place (notamment la récupération de la bonne API et de sa clé), et cela n'a pas été possible dans le délai de ce projet. Mais des outils ont déjà été développés par d'autres, donc si vous voulez aller encore plus loin sur vos données vous pouvez aller tester :

- https://github.com/Jessime/youtube_history
- https://towardsdatascience.com/explore-your-activity-on-youtube-with-r-how-to-analyze-and-visualize-your-personal-data-history-b171aca632bc

En raison de problèmes d'API, nous n'avons pas réussi à utiliser les projets proposés ci-dessus.

5. DONNÉES GMAIL

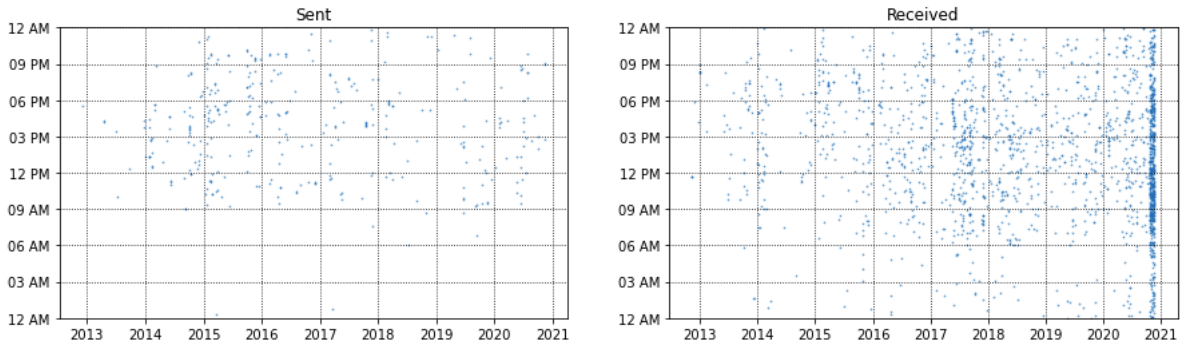
- **Structure des données**

Les données analysées sont celles du fichier "*Tous les messages, y compris ceux du dossier Spam.mbox".* Plusieurs labels y sont référencés (X-Gmail-Labels, Received, date ...) pour donner les informations de destinataires, expéditeurs, contenus… entre autres.

- **Matériels et Méthodes**

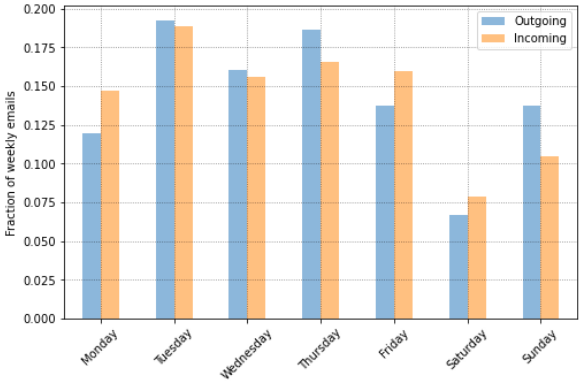
-Activités

Nos mails peuvent nous apporter des informations sur notre utilisation de la boîte mail. La liste des possibilités est variée : Qui envoie du courrier à qui (et combien/quelle fréquence) ? Quels sont les sujets de discussion les plus animés ? (**annexe 5**) Y a-t-il un moment particulier de la journée (ou de la semaine) où il y a le plus d'échanges ? Quelles sont les personnes qui s'envoient le plus de messages ?
Par exemple, nous pouvons visualiser à quels horaires nous l'utilisons le plus et même faire 1 suivi par année. De même pour les messages reçus.



Remarque: les résultats sont biaisés étant donné qu'au cours des années des mails ont été supprimés. C'est pour cela que nous voyons beaucoup plus de mails reçus pour 2020.
Nous pouvons également visualiser nos jours les plus actifs.

Ici, la personne envoie plus de mails le dimanche que le lundi. Nous pouvons aussi chercher à qui nous envoyons le plus de mail, qui nous en envoie le plus…

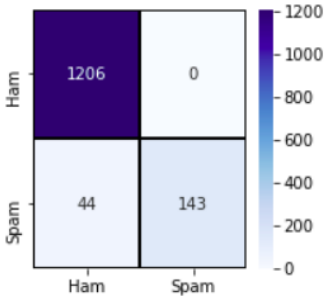


-Natural Language Processing

L'application de gestion des mails Gmail filtre et classifie vos e-mails avec la NLP en analysant le texte des e-mails qui transitent par leurs serveurs. Dans le cas de la détection de spams, cela consister à considérer comme e-mails indésirables ceux qui comportent des buzz words tels que « promotion », « offre limitée ». Pour reproduire la façon de faire de gmail, nous avons reproduit un modèle NLP qui classifie les mails en spam ou en mail légitime grâce à une base de données labellisée et certifiée.

Comment est fait le modèle ? Il y a une étape de nettoyage des données (stopwords, caractères spéciaux…). Il y a une étape importante de Vectorisation, c'est-à-dire la conversion de données textuelles en vecteurs qui permet d'accorder un poids plus important aux tokens les plus souvent retrouvés dans les spams par exemple. Il y a ensuite la création d'un modèle, et l'adaptation du modèle aux vecteurs. La dernière étape très importante est la mesure de la performance du modèle. Notre modèle se caractérise par de bons scores ainsi nous pouvons accorder une certaine confiance à nos résultats.

```
Accuracy score: 0.968413496051687
Precision score: 1.0
Recall score: 0.7647058823529411
F1 score: 0.8666666666666666
AUC: 0.9836978210551519
```



Nous avons ici 44 faux négatifs (spam mal classifiés) et 0 faux positifs (ham mal classifiés). Ainsi, nous avons réussi à reproduire un bon modèle.

C'est ainsi que google arrive à assigner des catégories à des mails. Il compare le texte de nos mails (header et corps du mail) à une base de données et met 1 tag. La reconnaissance de spam peut aussi se faire grâce à la reconnaissance des adresses mails. Ainsi, chaque mail est décrypté par google. Avec de bonnes bases de données, nous pouvons catégoriser les mails d'autres façons, c'est ainsi que google arrive à ranger des mails dans la catégorie promotion ou réseaux sociaux.

6. DONNÉES GOOGLE FIT

- **Qu'est-ce-que c'est que Google Fit?**

Google Fit est une application de suivi de santé développée par Google et disponible sur tous les systèmes. Elle permet d'accéder aux données d'activité d'autres applications et capteurs de fitness sur l'appareil sur lequel elle est installée ainsi que ceux qui lui sont connectés afin d'avoir des données plus larges.

- **Structure des données**

Les données de Google Fit sont celles relatives à l'activité physique de la personne. Elles sont regroupées dans un répertoire "Fit" dans les données Google exportées.

```
Fit
├── Activités
└── Agrégations quotidiennes
```

Les données auxquelles nous nous intéressons sont celles du répertoire "*Agrégations quotidiennes*". Il s'agit d'un ensemble de fichiers au format csv:

- plusieurs fichiers dont le nom est du type: "*année-mois-jour.csv*":
  Contient les données relatives aux :
  - heures de début et heures de fin
  - latitudes et longitudes basses et hautes
  - nombres de pas
  - distances
  - vitesses moyennes, maximales et minimales
  - …
- un seul fichier "*Résumés quotidiens.csv*": Contient les sommes des données mentionnées ci-dessus, ceci pour chaque jour. Ainsi, les heures sont remplacées par les dates.

- **Extraction des données et des connaissances**

Objectifs:

- Quels jours de la semaine l'individu est-il le plus actif ?
- A quelles heures de la journée l'individu est-il le plus actif ?
- A quels endroits se concentre son activité ?

Déroulé :

L'analyse de ces données s'est ensuite faite en deux étapes :

a) Activité hebdomadaire

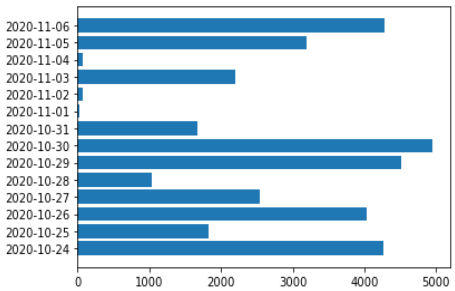
Dans cette partie, nous nous intéressons à **la distance parcourue par l'individu sur une période donnée (plusieurs jours)**. Nous utilisons donc les données récupérées du fichier "*Résumés quotidiens.csv*" que l'on aura parsé.

Nous commençons par ajouter à chaque date le jour de la semaine auquel elle correspond.
Nous filtrons ensuite les données pour ne récupérer que les données relatives aux jours pour lesquels la distance est renseignée.

Le jeu de données obtenu ressemblerait à celui sur la figure ci-dessous :

	Jours	Dates	Distances
0	Friday	2020-10-09	2860.575053
1	Saturday	2020-10-10	3373.796407
2	Sunday	2020-10-11	524.743626
3	Monday	2020-10-12	2339.617304
4	Tuesday	2020-10-13	1940.035948
...	...	...	...

Jeu de données obtenu après formatage des données

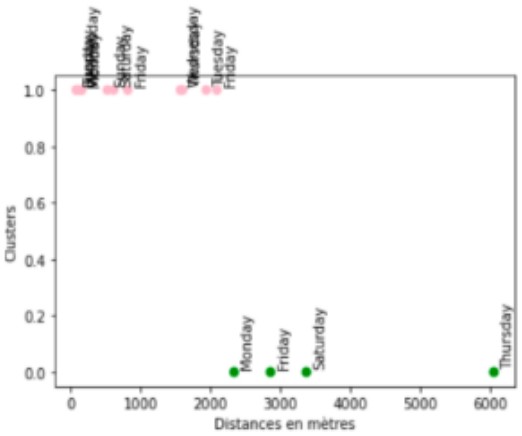


Distances parcourues en fonction des jours

=> Nous pouvons voir que la distance parcourue change d'un jour à l'autre ce qui est logique étant donné qu'un individu ne peut pas avoir exactement la même activité tous les jours.

Nous nous intéressons de plus près à **la distance parcourue en fonction des jours de la semaine**. Autrement dit, nous voudrions voir quels jours de la semaine un individu marche beaucoup, ou marche peu. Pour ce faire, nous utilisons **une méthode de clustering** basée sur **K-means.**
Nous commençons par utiliser **metrics.silouette_score()** afin de savoir le nombre de clusters que l'on devrait garder.

Nous remarquons que la différence en les valeurs obtenues est très petite (0.003), ça ne fait pas vraiment de différence.



Répartition des jours dans les clusters

Nous décidons par simplicité de garder deux clusters uniquement:

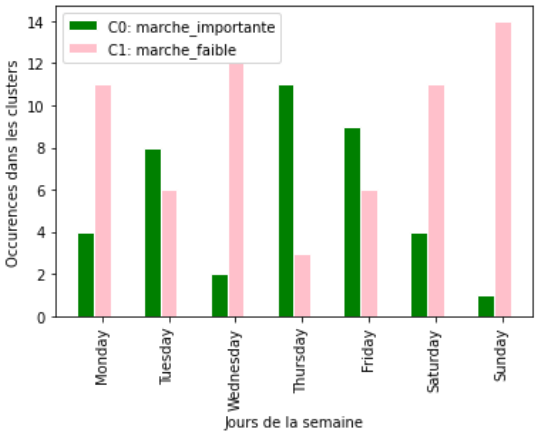
- C0: Activité importante: l'individu marche beaucoup
- C1: Activité faible: l'individu marche peu

L'algorithme nous retourne une liste contribuant à chaque jour un numéro de cluster (voir la figure ci-dessous)

=> Cette figure représente la répartition des jours dans les deux clusters, ceci pour une période de deux semaines, choisie aléatoirement. Nous avons représenté deux semaines seulement pour que ce soit plus lisible sur la figure.
Nous observons la présence de certains jours dans les deux clusters, ce qui est parfaitement normal, étant donné qu'un individu n'a pas forcément une activité similaire les mêmes jours de la semaine.

Exemple: Un individu *lambda* marche dans un parc chaque lundi en fin de journée
=> Un lundi de la semaine x, la personne va marcher en fin de journée
=> Un lundi de la semaine y, il pleut fort, la personne ne pourra pas aller faire son tour au parc

Afin d'avoir l'activité **hebdomadaire type** d'un individu, nous généralisons tout cela. Nous comptons le nombre d'occurrences de chaque jour de la semaine dans le cluster_0(*marche_importante*) et dans le cluster_1(*marche_faible*). Le résultat est représenté ci-dessous :



=> Ce graphique nous informe sur le jour de la semaine où l'activité est plus importante ainsi que ceux où l'activité est plutôt faible.

Exemple d'un utilisateur :

Sur la figure ci-dessus, nous pouvons voir que les jours où l'individu **marche le plus** sont le **mardi, jeudi et vendredi**, ce qui correspond aux jours où il travaille en dehors de chez lui.

- les mardis et les vendredis il y va en transports en commun
- les jeudis il y va à pied, d'où la forte appartenance au cluster C0

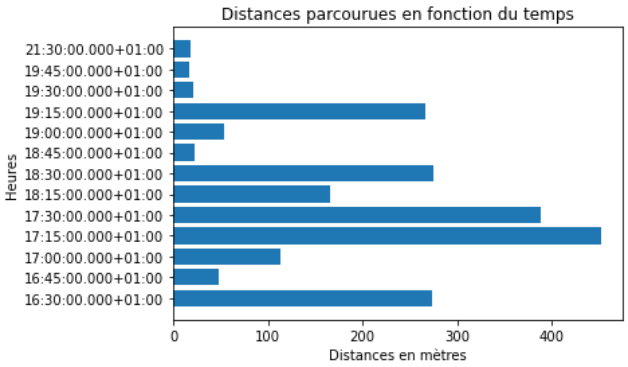
les jours où **il marche peu** sont le **lundi, mercredi, samedi et dimanche**

- les lundis et les samedis il va faire les courses
- les mercredis et les dimanches, souvent, il reste toute la journée chez lui, d'où la forte appartenance au cluster C1

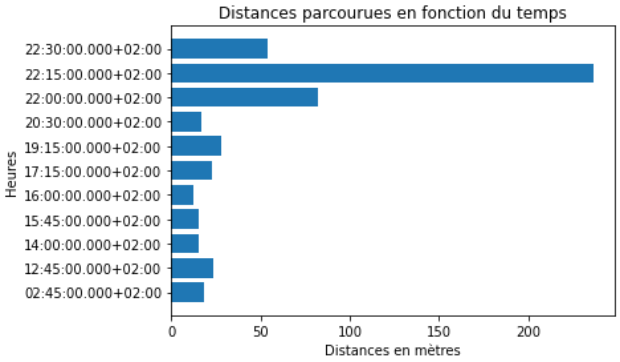
b) Activité quotidienne

Dans cette partie, nous nous intéressons aussi à **la distance pour une journée donnée**. Nous allons ici sélectionner deux jours au hasard, un correspondant à un jour du cluster_0 (marche importante) et un autre du cluster_1 (marche faible).

Nous récupérons les données de la même manière qu'à la partie précédente, et les filtrons cette fois-ci en gardant que les heures auxquelles nous avons une distance existante. Nous représentons ensuite la distance parcourue par l'individu en fonction des heures pour chacun des deux jours.



Distances parcourues pour un jour du cluster0 (jeudi)



Distances parcourues pour un jour du cluster1(dimanche)

=> Nous remarquons qu'en effet, les heures et la distance différent entre les deux jours sélectionnés. Ceci pourrait nous informer sur l'activité de l'individu. Il serait logique que les heures d'activité à certains jours de semaine soient celles où l'individu part au travail (s'il y va à pied par exemple). Quant aux jours correspondant au weekend par exemple, les horaires seraient différents et la distance aussi. L'activité pourrait être moins importante si l'on est chez soi et qu'on ne sort que pour prendre l'air ou faire des courses pas très loin.

Exemple d'un utilisateur :

Sur les deux figures ci-dessus :

- la figure de gauche: correspond à un jeudi qui est un jour du cluster_0, l'activité est importante à 17h15 et 17h30, ce qui correspond à l'heure à laquelle l'individu par au travail (garde d'enfant en périscolaire), puis à 19h15 qui correspond à l'heure à laquelle l'individu finit de travailler et rentre chez lui (ici on pourrait même dire qu'il est rentré en bus étant donné qu'il n'a pas parcouru autant de distance à l'aller.
- la figure de droite: correspond à un dimanche qui est un jour du cluster_1, l'activité est très faible toute la journée sauf tard dans la soirée où la personne a du sortir prendre l'air (le 11 octobre 2020 avant le deuxième confinement et le couvre feu)

7. DISCUSSION/CONCLUSION

Dans la société actuelle, la majorité de la population possède un téléphone portable, un ordinateur ou un accès à internet. Les frontières entre les vies privées et publiques se mélangent car nous souhaitons garder nos informations personnelles confidentielles mais n'avons jamais autant partagé d'informations sur internet, via des applications … L'utilisation de nos données après les avoir publiées, partagées ou renseignées (inscription sur un site, une application …) reste obscure.

Notre projet permet de montrer une partie des informations qui peuvent être déduites de nos habitudes, de nos recherches. Qui a accès à ces informations ? Nous, Google, les autres services concernées (ex YouTube, sites avec inscription via google, ...) Nos données sont utilisées à des fins marketing (pubs dirigés en fonction de nos activités), conseil d'achat, ... Il est habituellement demandé sur tous les supports comment on veut que nos données soient traitées (ne pas les stocker ou autre).

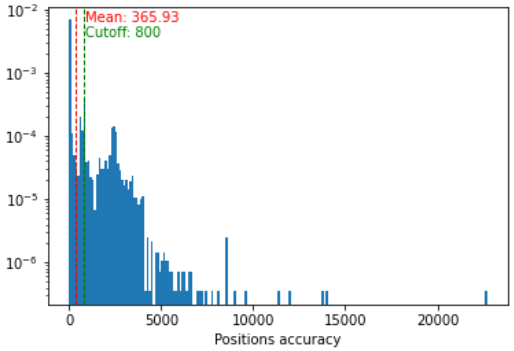
Pour plus de sécurité, supprimez votre historique des positions Google et désactivez-le afin que cela ne pose plus de problème à l'avenir. Google fournit des instructions sur la façon de procéder, et ce n'est pas trop difficile. Cela pourrait avoir un impact sur votre expérience de recherche à l'avenir, car de nombreuses fonctionnalités Google fonctionne bien, car elles se souviennent de l'endroit où vous avez été. Si vous ne voulez pas perdre toutes vos données de localisation, vous pouvez toujours garder l'historique de localisation activé pour qu'il continue à enregistrer, puis le télécharger via Google Takeout et effacer votre historique plusieurs fois par an.

La vie à l'ère de l'information consiste à faire des compromis, que vous le vouliez ou non. Il n'y a pas d'option set-it-and-forget-it sur une échelle mobile de confidentialité par rapport à la commodité, car l'échelle elle-même change constamment. En général, vous donnez probablement beaucoup plus de données que vous ne le pensez, et ce n'est pas nécessairement quelque chose que nous devrions simplement accepter. Des outils tels que le visualiser de l'historique des positions sont excellents car il ne faut que quelques minutes pour déterminer à quoi ressemblent les années de suivi, ce qui nous permet de prendre de meilleures décisions concernant notre confidentialité. Espérons que davantage de nos données seront accessibles et contrôlables par l'utilisateur via ce type d'interfaces à l'avenir. Visitez la page Google Web & Activity et cliquez sur "gérer l'activité". Pour limiter ce que Google stocke (https://myactivity.google.com/myactivity).

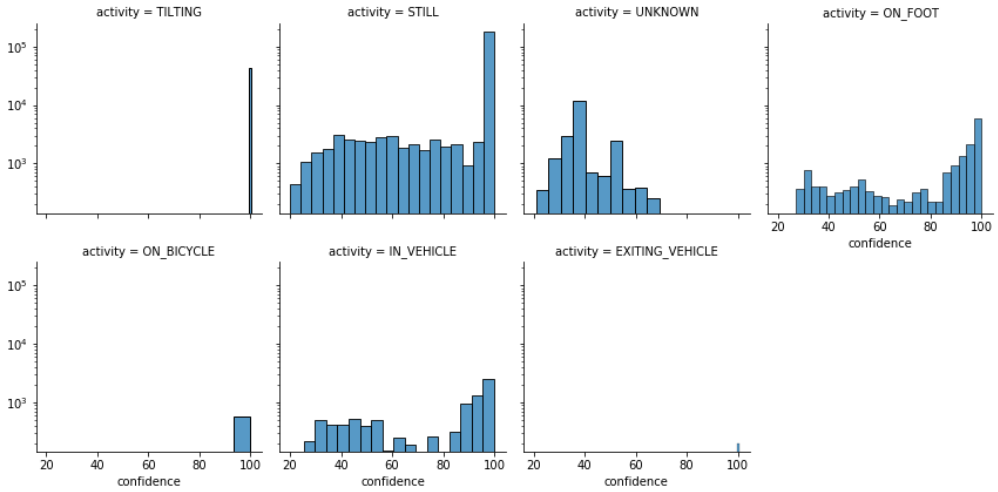
Ce projet nous a permis de prendre en compte la gêne que nous avions entre nous. Pour tester nos codes nous étions réticents à nous transmettre nos données car on ne sait pas ce qui peut s'y trouver. Alors serions-nous à l'aise pour que des inconnus aient accès à ces données ?

ANNEXES

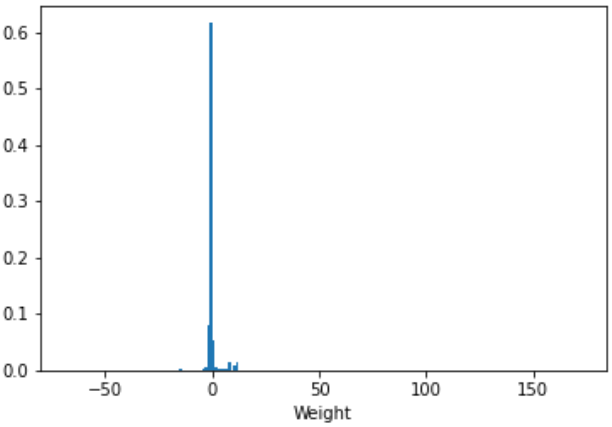
Annexe 1 : Distribution des effectifs (échelle log) d'accuracy des données



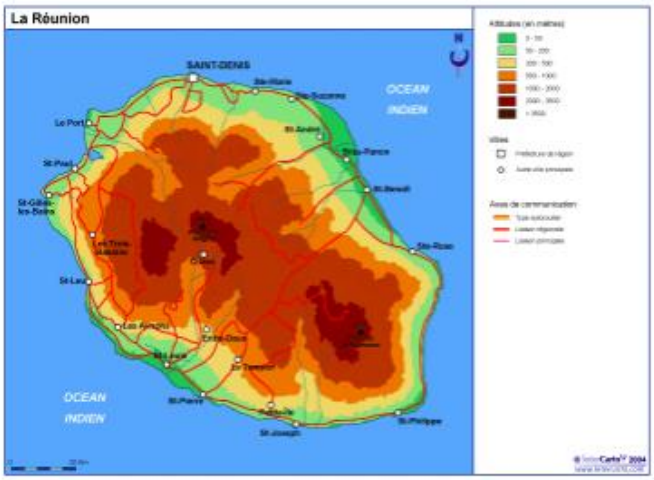
Annexe 2 : Distribution des effectifs (échelle log) de l'indice de confiance associé aux activités.



Annexe 3 : Distribution centrée réduite de la variable Weight = confidence_norm / accuracy_norm



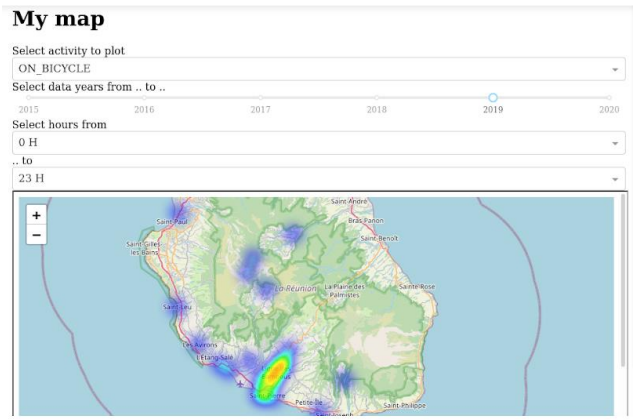
Annexes 4 :

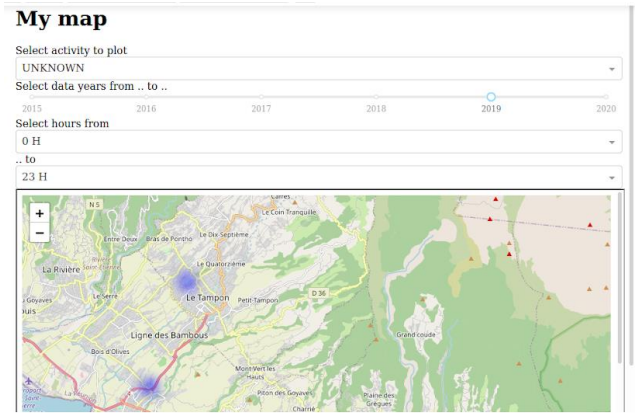


L'île de la Réunion, ses axes principaux, et ses reliefs
(Source: Carte géographique, touristique et plan de la Réunion, 974, Saint-Denis)

Visualisation des densités GPS filtrées par activités

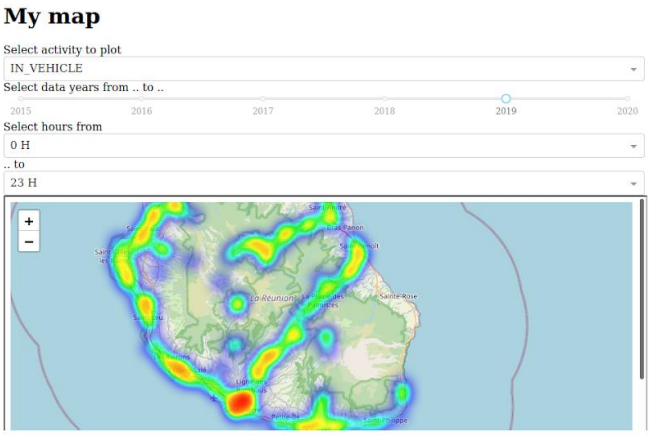
(ON_BICYCLE à gauche, UNKNOWN à droite)

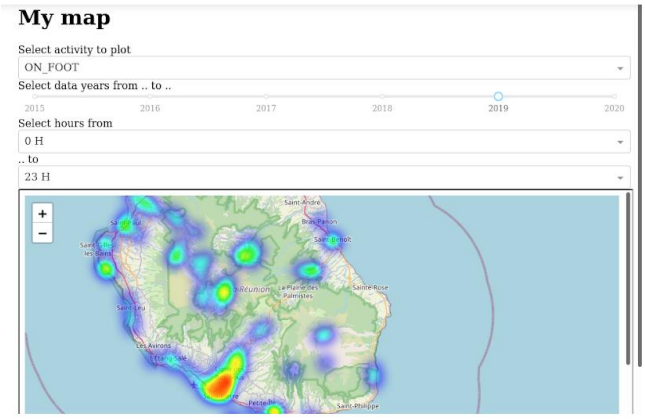




Visualisation des densités GPS filtrées par activités

(ON_FOOT à gauche, IN_VEHICULE à droite)

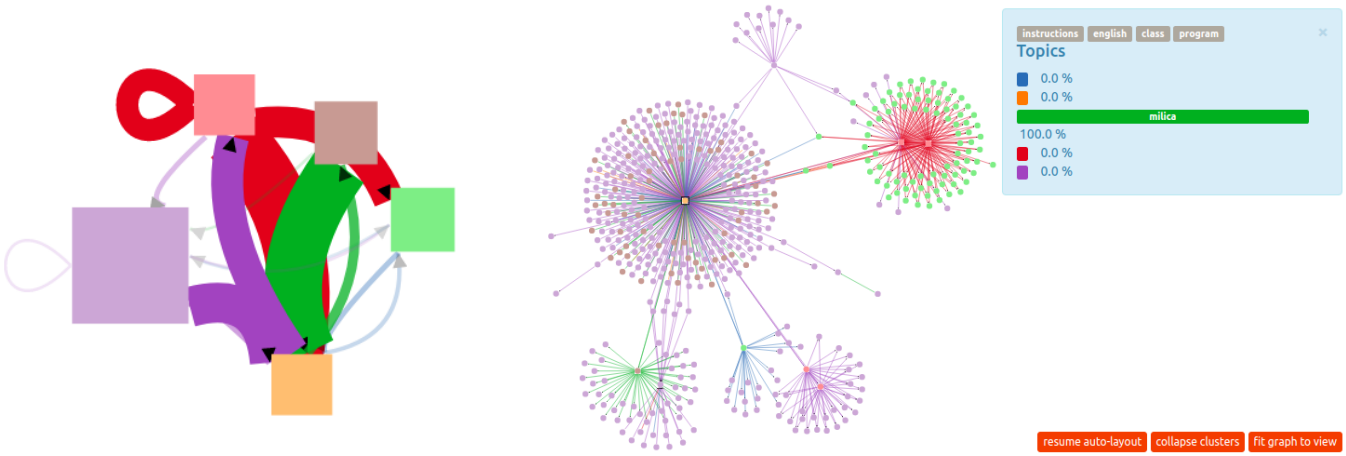




Annexe5 : Outil Linkage

Il existe déjà des façons simples d'analyser les mails. Par exemple avec le site Linkage qui combine deux types d'études : l'étude de réseau et l'analyse textuelle. Il part du principe qu'une manière d'analyser un réseau est de comprendre de quoi on parle avec qui. Il permet d'extraire ces informations :

- Les groupes de personnes qui parlent ensemble
- Les sujets dont parlent les personnes
- La structure et le contenu des relations entre les personnes



Sur la figure de gauche, il y a 5 clusters et 5 topics (qui regroupent plusieurs mots). Par exemple, on peut remarquer que les membres du cluster en rose parlent entre eux majoritairement en utilisant le topic rouge. Il suffit de cliquer sur les carrés pour avoir le nom des membres du cluster et sur les flèches pour avoir les topics et leur distribution. Sur la figure de droite, on visualise les mêmes clusters et topics mais en détaillant qui a écrit quoi à qui. Le fait de pointer sur un individu particulier (un rond) permet de chercher où se trouvent tel ou tel auteur.

Cette méthode est difficile à implémenter et coûteuse en temps. C'est pour cela que nous n'avons pas créé de code à ce sujet mais nous avons jugé intéressant de mettre cet outil en lumière.