

Développement d'un indicateur d'accessibilité aux transports en commun

Présentation du projet

Notre client est Auvergne-Rhône-Alpes Énergie Environnement (AURA-EE), une agence régionale cherchant des solutions et stratégies locales de transition énergétique pour les territoires.

AURA-EE a entre autres créé TerriSTORY, un outil web permettant de visualiser divers indicateurs sur la région à partir de données en Opensource.

Notre but était d'aider à l'élaboration d'un nouvel indicateur à ajouter à TerriSTORY : l'"indicateur d'accessibilité aux transports en commun", permettant de visualiser clairement les zones qui manquent d'infrastructures et d'adapter en conséquence les politiques de transport de la région.

Étapes de la construction du projet

19 avril

Nous avons fait la réunion de lancement avec les encadrants du projet, Matthieu Denoux, Valentin Vermeulen, et Fatima Fattouh. L'équipe qui nous encadre est chargée de l'intelligence territoriale et de l'établissement d'indicateurs pour la région Auvergne Rhône-Alpes.

Il s'agissait au cours de cet entretien de mettre au clair le cahier des charges, l'enjeu du projet et les ressources que nous avons à disposition. A l'issue de la réunion, nous avons convenu de refaire un point mi-mai, le temps de pouvoir, de notre côté, regarder ces ressources et chercher ce dont nous aurons besoin pour les compléter.

10 mai

Ayant pris connaissance des documents sur lesquels nous devons nous appuyer, nous avons discuté de nos premières impressions avec nos encadrants. Nous avons déjà regardé la structure des fichiers de data que nous aurons en entrée, et nous en avons plus particulièrement discuté avec Fatima, notre référente en ce qui concernait le code. Les fichiers sont au format GTFS, un standard développé par Google spécialement pour les transports en commun. Nous avons abordé la manière dont nous devons les traiter pour les transformer en csv. Un autre sujet était l'éventuelle modification de la formule du calcul de l'indicateur, afin de le rendre plus représentatif de ce qu'on appelle "l'accessibilité aux transports en commun". Comme nous étions en stage ouvrier et avions très peu de temps pour travailler sur le projet, nous sommes tombés d'accord avec Fatima pour que la

prochaine réunion soit à la fin de la première semaine de cours, afin d'avoir eu le temps d'avancer.

31 mai

Nous avons établi une note méthodologique qui nous servirait de feuille de route, que les encadrants du projet ont validé dans la foulée. Il s'agissait, à l'issue de recherches, de décomposer le projet en plusieurs étapes, pour pouvoir se répartir les tâches mais aussi avoir une vision claire de ce que nous avons à coder.

01-02 juin

En s'appuyant sur la note méthodologique, nous avons donc écrit les premières lignes du code. Il fallait d'abord importer les bons modules, et dans un premier temps nous n'avons pas réussi à les installer correctement sur VSCode. La première version du code était donc un jupyter notebook, sur lequel les modules importés fonctionnaient comme prévu. Il fallait lire les données au format GTFS.

Après avoir transformé les fichiers à l'aide de ces modules pour en faire des csv, nous avons visualisé les données, pour regarder ce qu'elles contenaient exactement. Une fois la structure en tête, nous avons codé ce que nous avons appelé 'l'algorithme des rectangles'. Il s'agit d'associer à chaque ligne de transports un rectangle, le plus petit possible, englobant la ligne. Le rectangle en question est toujours parallèle à l'équateur, et il se présente sous la forme (lat_max, lon_max, lat_min, lon_min, trip_id). Cela permet d'éviter de parcourir l'intégralité des arrêts lorsqu'on part d'un point A, car on peut d'ores et déjà ne pas considérer les rectangles situés à une trop grande distance.

3 juin

Nous avons pu montrer notre code à Fatima, qui était satisfaite. Nous avons déterminé les prochaines étapes à suivre, les autres algorithmes pour obtenir la liste des arrêts, les fréquences, etc, fonctionnaient un peu sur le même principe que le premier donc nous pensions que ce serait rapide à coder (en pratique ça nous a posé un peu plus de problèmes que prévu...).

4-13 juin

Nous avons fait le programme qui, à partir d'un fichier gtfs, donne la liste des lignes, arrêts associés et fréquence de passage. Nous avons également résolu le problème de GTFS-kit, le module que nous n'arrivions pas à installer sur VSCode, et avons finalement obtenu un fichier .py et non .ipynb.

Lucien a contacté la métropole du Grand Lyon pour avoir accès aux données de transport de la ville de Lyon, qui étaient manquantes dans nos données mais qui sont très importantes pour la région.

Il a fallu modifier le programme des rectangles au vu de l'ajout conséquent de données.

Nous avançons sur le programme qui calcule les fréquences de passage. Il reste à cibler les données qu'il faut conserver dans le csv final exporté de ce programme.

14 juin

Nous avons eu une réunion avec Matthieu et Fatima. Nous présentons nos deux algorithmes fonctionnels à ce jour : les rectangles et celui donnant la fréquence de passage de chaque ligne à chaque arrêt. Les encadrants semblent satisfaits du timing et le but donné en fin de discussion était d'avoir une première version fonctionnelle le 22 juin, date de la prochaine réunion, afin de pouvoir commencer à cartographier.

15-21 juin

Nous avons fait le programme qui, à partir de coordonnées données et d'un rayon R, retourne la liste des arrêts situés à une distance inférieure à R, avec la fréquence de passage associée.

Afin de gagner en complexité, le choix est fait pour l'algorithme des rectangles de ne conserver que le route_id et non le trip_id, ce qui permet de diminuer drastiquement le nombre de rectangles (gain de complexité) en augmentant légèrement le nombre d'arrêts dans chaque rectangle (perte).

Nous avons également travaillé sur le programme qui, en prenant en entrée des coordonnées, renvoie l'indicateur calculé comme mentionné dans la note méthodologique. Enfin, nous avons commencé à regarder la partie cartographie, même sans avoir tous les algorithmes nécessaires pour en arriver là, afin que nous ne soyions pas bloqués. En particulier, nous arrivons à former un masque de la région Rhône-Alpes.

22 juin

Lors de cette réunion, nous avons pu présenter la version escomptée aux encadrants. Ils nous ont donné l'aval pour commencer à cartographier, et aussi à commenter le code, qui n'était à ce stade pas très clair.

29 juin

Nous avons montré la version presque finale aux encadrants, avec notamment une carte de la région colorée - mais peu précise. Nous avons également pu expliquer les difficultés rencontrées. Nous avons décidé de nous recentrer sur la présentation et la propreté du code écrit jusque là, manquant de temps pour vectoriser la fonction (et donc gagner en précision sur la carte). Ils nous ont félicité pour le travail, nous ont dit que c'était la dernière réunion et qu'ils n'étaient pas sûrs d'être là pour la présentation mais nous ont souhaité bon courage.

29 juin - 7 juillet

Nous avons terminé de mettre notre code "au propre", nous sommes occupés de la licence, de la rédaction du rapport et de la préparation de la présentation.

Sur le temps restant avant la présentation, nous parvenons à finir la vectorisation de la fonction ce qui permet d'obtenir une image de plus de deux millions de pixels. Nous nous apercevons que Grenoble n'est pas représentée : il y a un problème de lecture du fichier GTFS de la ville. Nous parvenons à corriger ce problème et à rendre notre code plus adaptatif afin qu'il puisse réagir face au manque de certaines informations.

Difficultés rencontrées

- route_id n'était pas une clef primaire, il fallait donc jongler entre les fichiers GTFS pour cibler les données pertinentes.
- Approche liste de coordonnées VS liste d'arrêts pour cartographier : suivant le facteur limitant pour la complexité, l'approche à choisir est différente. Soit la liste des points est assez petite, auquel cas le facteur limitant est le nombre d'arrêts, soit c'est l'inverse (comme pour une carte par exemple). Ceci change totalement l'approche à choisir pour le code. Nous nous sommes d'abord concentré sur la première vision (qui nous était demandée) avant de nous attaquer à la deuxième.
- Dans le calcul d'indicateur proposé dans la thèse jointe au sujet, il est fait mention d'un coefficient de fiabilité qui diffère selon le type de transport. Un train aura un coefficient de fiabilité supérieur à un bus, par exemple ; or les données dont nous disposions n'avaient pas de colonne précisant le type de transport. Nous avons donc décidé de faire comme si nous avions uniquement affaire à des bus.
- Au moment de l'ajout de la data de Lyon, il a fallu changer d'algorithme car on a multiplié les calculs par quelques dizaines : au lieu de parcourir les différentes tables en parallèle (algo rectangles initial), on a préféré joindre les différentes tables qui nous intéressaient pour ne la parcourir qu'une seule fois (algo rectangles final). On est passé d'un programme qui n'aboutissait même pas en 8h à un programme qui s'exécute en moins de 6 mins.
- Lucien et Camille ont beaucoup travaillé sur l'obtention des fréquences et des rectangles, Juliette se chargeait du calcul de l'indicateur à partir d'un csv et Matthieu s'est surtout occupé de la cartographie. Ces parties étant assez distinctes, il a fallu se synchroniser notamment en termes de noms des variables utilisées, se mettre d'accord sur les données d'entrée et de sortie de nos algorithmes, et bien les commenter pour que ce qu'on avait fait soit clair pour le reste du groupe.
- Le scaling des tableaux NumPy a posé problème, afin d'avoir plus de points pour une meilleure précision ; il a donc fallu revenir au début du code pour recréer un tableau avec plus de points
- Certains fichiers GTFS ne contenaient pas certaines données utiles au programme. Or le module de lecture GTFS utilisé est permissif (pas de message d'erreur s'il échoue), nous n'avons donc réalisé ce manque d'information que très tardivement, lorsque que l'on a remarqué que Grenoble manquait sur la carte ! Nous avons réussi à modifier le programme pour qu'il s'adapte à ce manque d'information.

Bilan : ce que ça nous a apporté

Ce projet nous a appris ce qu'était réellement un travail d'informatique en groupe. Il s'agissait du premier "gros" projet, sur le long terme, que nous devions réaliser depuis notre arrivée aux Mines. Cela nous a forcé à garder un code propre et commenté, un repo Git organisé, car dès que le projet est long et que les collaborateurs ne sont pas toujours ensemble lorsqu'ils travaillent, la communication est fortement facilitée par un repo propre. Nous avons compris l'importance de l'adaptabilité : deux personnes font un bout de code en parallèle, puis doivent les coordonner en modifiant quelques détails lors de la fusion, nous devons nous assurer d'avoir les mêmes noms de variables...

La feuille de route, au début du projet, était vitale pour mener à bien ce dernier.

Nous avons en outre échangé pour la première fois avec un "client", et nous avons compris la façon dont nous pouvions concrètement appliquer nos connaissances en informatique (et aller chercher celles que nous n'avions pas encore, par exemple en prenant en main un nouveau module) d'une manière professionnalisante.