

Développement Logiciel - Projet

Mohammed El-Khmissi
Gueladio Niasse
Quentin Fontana

Décembre 2021

1 Introduction

Dans le cadre du cours de développement logiciel, il nous a été donné pour projet de créer un package qui permettrait d'utiliser Map interactive des autoroutes du Sud de la France. Cette Map aurait pour intérêt de renseigner le trajet le moins cher en terme de coût de péages entre une entrée et une sortie choisi par un utilisateur.

2 Présentation du package deloqv

2.1 La Map

La Map est l'objectif central de notre projet. C'est en effet son aspect le plus pragmatique.

Elle permet à un utilisateur qui souhaiterait se déplacer entre une des villes suivantes :

Vendargues	Montpellier est	Montpellier sud	Montpellier ouest
St-Jean-de-Vedas	Sete	Agde Pezenas	Beziers ouest
Narbonne est	Narbonne sud	Sigean	Leucate
Perpignan nord	Perpignan sud	Le Boulou (peage sys ferme)	Sesquieres
Frontiere Espagnole	Lezignan	Carcassone est	Carcassone ouest
Bram	Castelnoudary	Villefranche-de-Lauragais	Nailloux
Mazeres-Saverdun	Pamiers nord	Pamiers sud	Montgiscard
Le palays	Montaudran	Lasbordes	Soupetrad
La Roseraie	La Croix Daurade	Borderouge	Les lzards

... de visualiser son voyage et d'être informé du trajet à effectuer pour que le coût des péages lui soit minimum. Voyons tout de suite un exemple.

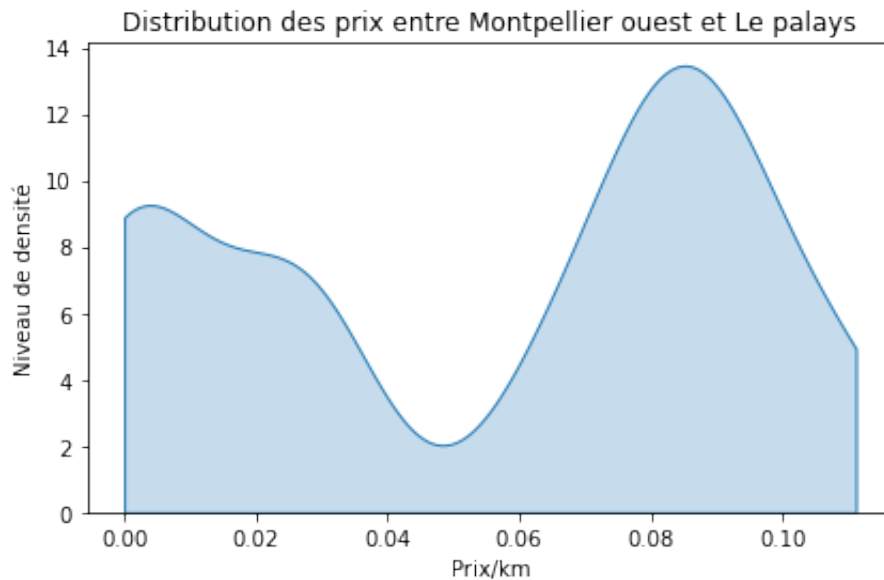
Alfonse habite dans l'Ouest Montpellierain et souhaite rendre visite à son ami qui vit à proximité de la sortie "Le Palays" dans la banlieue de Toulouse. Il aimerait néanmoins que son trajet lui revienne le moins cher possible et puisque le voyage sera long, il prévoit de s'arrêter deux fois en cours de route. En utilisant la Map que nous avons développée, Alfonso va pouvoir déterminer où sortir afin que ses deux arrêts lui permettent d'optimiser le coût de son voyage. En effet une fois que les caractéristiques de son trajet auront été saisies (nom gare de départ, nom gare d'arrivée, nombre d'arrêts), la Map lui renverra le message suivant :

'Si vous sortez aux gares : (Beziers ouest,Lezignan), votre trajet vous coutera 20.3 €. Vous aurez ainsi fait une économie de 3.4 € sur le coût des péages.'

2.2 Distributions des Prix

Le deuxième aspect de notre package est de rendre compte de la distribution des prix entre les différentes gares de péages listées précédemment. Cette partie de notre travail est destinée à un utilisateur curieux de connaître la politique des prix mise en place par l'entreprise ASF (propriétaire des autoroutes).

En choisissant une gare d'entrée et une gare de sortie, notre programme va renvoyer les différentes fréquences de prix par kilomètres. En reprenant l'exemple précédent, voici ce que renverrait notre programme :



Alfonse constaterait alors que le prix par kilomètres est le plus souvent compris entre 0.08€/ km et 0.1€/km.

3 Etapes de la réalisation du package deloqv

Nous allons dans cette section expliquer comment nous avons procédé pour réaliser notre travail.

3.1 Etape 1 - Data Cleaning

La première étape a été de traiter les données qui nous ont été fournies afin que nous puissions les utiliser à nos fins. Nous avons tout d'abord transformé les données de la page 3 du pdf suivant en format `.csv` afin de lui utiliser via le logiciel `VScode` :

`https://public-content.vinci-autoroutes.com
/PDF/Tarifs-peage-asf-vf/ASF-C1-TARIFS-WEB-2021-maillage-vf.pdf`

Nous avons ensuite supprimé les lignes et les colonnes qui nous étaient inutiles, c'est à dire les lignes (respectivement les colonnes) qui faisaient référence à des gares de péages qui ne constituaient pas des "sorties" possibles. Par exemple, le *Peage de Montpellier St-Jean*.

Nous avons répété cette procédure avec le fichier :

`gares-peage-2019.csv`

disponible à l'URL suivant : `https://www.data.gouv.fr/en/datasets/gares-de-peage-du-reseau-routier-national-concede/`. Nous avons ainsi supprimé les gares de péages qui ne faisaient pas partie de notre champ d'analyse (les gares de péages qui ne sont pas proposées dans le fichier des prix) afin de pouvoir l'utiliser en

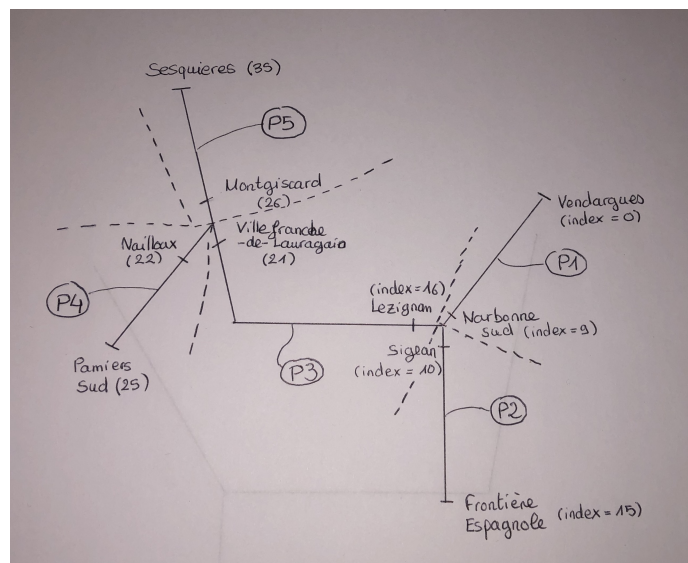
parallèle avec le fichier des prix. Nous avons de plus supprimé toutes les colonnes qui nous étaient inutiles afin de ne garder au final que les colonnes qui indiquaient pour chaque gare leur nom, la section d'autoroute sur laquelle elle était située et leurs coordonnées X et Y.

Une fois notre nettoyage de données réalisé (voir le code : `data_cleaning.py`), nous avons sauvegardé nos résultats sous forme de dataframes dans les fichiers : `prices_clean.csv` et `coordonnees_clean.csv`.

3.2 Etape 2 - Algorithme du chemin le moins cher

Cette étape fut une des plus difficile à mettre en oeuvre. Notre objectif était le suivant : calculer le coût de péage le moins élevé entre un point de départ et d'arrivée en fonction du nombre d'arrêt qu'il était possible de faire.

Dans un premier temps, nous devons trouver le moyen de compter le nombre de gares présentes sur le trajet entre notre gare de départ et d'arrivée. Pour ce faire, nous avons d'abord partitionné les gares en fonction de la portion de route sur laquelle elles étaient situées. Voir le partitionnement en 5 ci-dessous.



Nous avons ensuite créé 5 sous-dataframes correspondant à notre partitionnement. L'idée était la suivante : en réinitialisant les indexs des gares et en les inversant pour la partition P1, nous pouvions compter le nombre de gares en sommant l'index de la gare de départ et celui de la gare d'arrivée. Grâce à la fonction :

```
nb_gares_sur_trajet(i,j)
```

nous étions en mesure de savoir de combien de gares était constitué n'importe quel trajet.

Deuxièmement, nous voulions une fonction qui nous permette de renvoyer le dataframe constitué des gares présentes sur notre trajet ordonnées par rapport au départ et à l'arrivée et dont les indexs allaient de 0 (index de la gare de départ) à $n = \text{nombre de gares sur le trajet} + 1$ (index de la gare d'arrivée). C'est ainsi dans cette perspective que nous avons développé la fonction composée :

```
Le_Trajet(tab_algo(i,j))
```

qui prend comme arguments les indexs respectifs des gare de départ et d'arrivée de notre dataframe initial :

```
coordonnees_clean.csv
```

Enfin, nous avons créé la fonction qui nous permettrait de calculer le trajet à parcourir pour qu'il revienne le moins cher en fonction du nombre d'arrêt qui était accepté. Pour cela nous avons commencé par calculer puis lister les arrangements possibles des indexs pour un trajet, puis nous avons fractionner les tuples d'indexs en couples d'indexs afin de créer des sous-trajets, pour finalement sommer le prix relatif à chaque couple d'indexs et ainsi déterminer lequel était le moins cher. Illustrons notre démarche en reprenant notre exemple.

Alfonse part de 'Montpellier Ouest' (index = 3) et veut se rendre à 'Le Palays' (index = 27). On a :

```
nb_gares_sur_trajet(3,27) = 11
```

Il y'a donc 13 gares au total dans le dataframe en comptant la gare de départ et la gare d'arrivée. Alfonso a décidé de plus de s'arrêter 2 fois, ce qui revient à dire que son trajet va être décomposé en 3 sous-trajets. L'algorithme utilise la fonction :

```
suite_de_trajet(3,13)
```

qui renvoie la liste des suites de couples suivantes :

```
[[ (0, 1), (1, 2), (2, 12)],
  [(0, 1), (1, 3), (3, 12)],
  [(0, 1), (1, 4), (4, 12)],
  :
  [(0, 10), (10, 11), (11, 12)]]
```

Une fois cette liste de couple générée, la fonction `chemin_moins_cher(3,27,2)` va calculer les prix de chaque sous-trajets possibles décrit par la liste des couples d'indexs ci-dessus, les sommera puis les comparera entre eux pour déterminer laquelle est le moins cher. On obtiendra ensuite comme vu précédemment :

'Si vous sortez aux gares : (Beziers ouest,Lezignan), votre trajet vous coutera 20.3 €. Vous aurez ainsi fait une economie de 3.4 € sur le coût des péages.'

3.3 Etape 3 - Création de la Map

La dernière étape de notre projet fût la création de la Map qui permettrait une utilisation pragmatique de notre package.

Pour ce faire, nous avons utilisé le package python suivant :

```
openrouteservice
```

pour lequel nous avons dû créer un compte afin qu'il nous renvoi une clé de connexion :

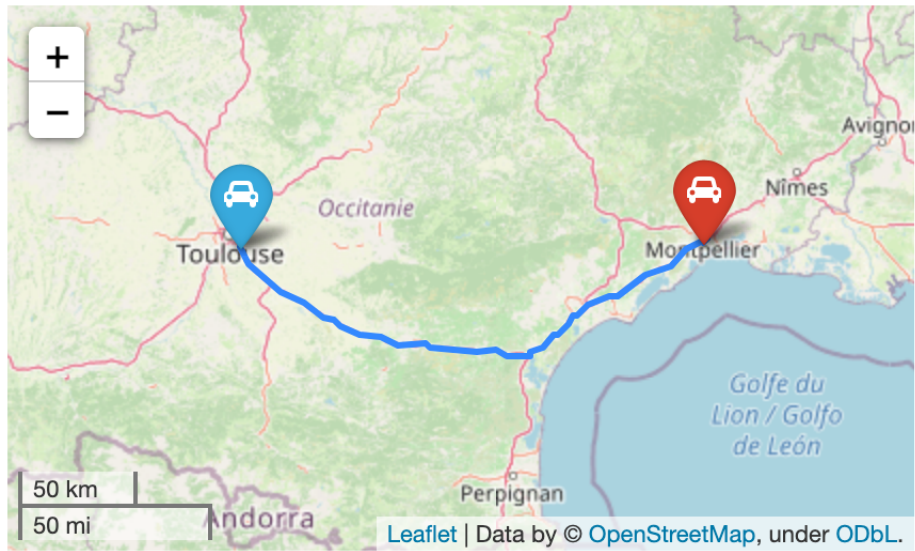
```
clé = 5b3ce3597851110001cf62486f5564a064e34f3895221e5a0d9a2405
```

nécessaire pour son utilisation. En utilisant les dataframes `prices_clean.csv` et `coordonnees_clean.csv` nous avons défini une classe `map` comprenant les fonction `index` prenant comme seul paramètre le nom d'une gare de péage et qui renvoi son index, et la fonction `road(DEPART,ARRIVEE)` qui nous permet de convertir les coordonnées "X" et "Y" de chaque gare de péage en format GPS afin d'afficher le trajet entre la gare de départ et d'arrivée ainsi que son prix et la distance entre les deux gares.

Alfonse visualisera son trajet telle que vu ci-dessous.

DEPART ▼

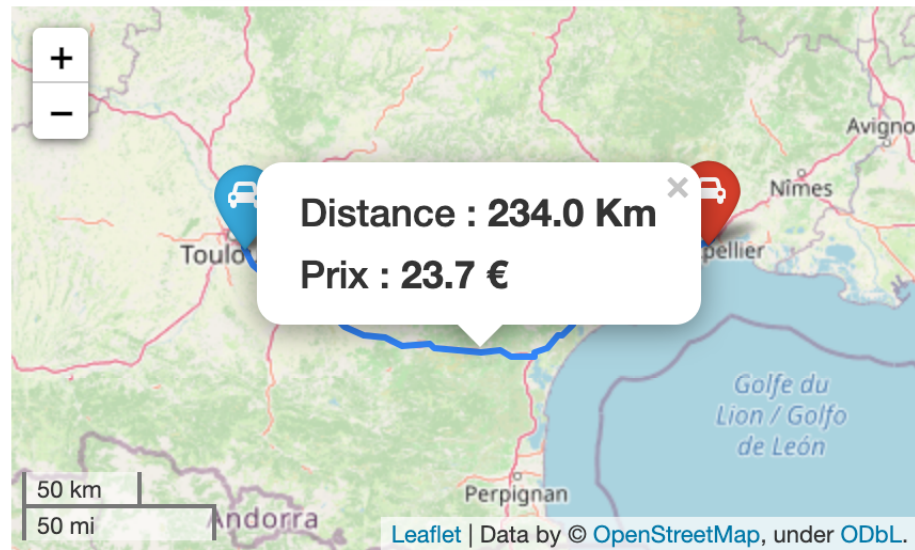
ARRIVEE ▼



En cliquant sur le chemin en bleu.

DEPART ▼

ARRIVEE ▼



Et en utilisant l'algorithme :

Depart	3	▼
arrivee	27	▼
Nb_sortie	2	▼

Si vous sortez aux gares :
(Beziers ouest,Lezignan), votre trajet vous coutera 20.3 €.
Vous aurez ainsi fait une economie de 3.4 € sur le coût des