

# Projet

---

- Recommander des activités et restaurants dans une ville
- Proposer un itinéraire piéton les reliant sur une carte
- Basé sur les préférences de l'utilisateur (budget, sportivité, nombre d'attractions souhaitées)
- Personnaliser son choix de restaurant avec une approche ML



# Choix des sites d'attractions

website name	scraping type	url normalization [1-3]	pages consistency [1-3]	Destination completeness [1-3]
cntraveler		3	2	2
time out		2	1,5	3
get your guide		2	2	3
routard		3	2	3
US news travel	unscrapable	3	3	3
lonely planet		3	3	3
earth trekker		1	1	2
guide michelin		3	3	3
bucketlisttravels		3	3	3
WorldTravelGuide		3	3	3

## Sites conservés :

- WorldTravelGuide
- BucketlistTravel
- Routard
- Lonely Planet
- CNTraveler
- Guide Michelin

## Variables récupérées:

- Nom
- Description
- Prix
- Age recommandé





# Choix des sites de restaurant

## Guide Michelin

<https://guide.michelin.com/fr/fr/{region}/{ville}/restaurants>

## Variables récupérées

- Nom du restaurant
- Description
- Type de cuisine
- Tranche de prix
- Adresse, coordonnées GPS





# Construction du code en pipeline

## Extraction

1. Extraction de 5 sites d'attraction
  - avec une méthodologie commune selon l'URL
  - Avec une gestion d'erreurs pour les urls invalides
2. Extract michelin
  - Construction d'un dataset de correspondance ville > url via webscrapping dynamique
  - Scrapping de la page de la ville et de chaque restaurant individuel

## Processing

1. Attraction
  - Fusion des attractions selon similarité du titre
  - Tri selon son occurrence, son rang global
  - Récupération de la localisation via API
2. Restaurant
  - Privilégie la nourriture durable et la labellisation bib\_gourmand
  - Sentence embedding de la colonne description avec *all-mini-lm-L6-V2*

# Construction du code en pipeline

## Clustering

1. Construction de clusters d'attractions autour des restaurants
  - Selon les filtres choisis par l'utilisateur
  - S'adaptant si les filtres pris ne donnent pas de résultat
2. Choix du meilleur cluster
  - selon la description voulu par l'utilisateur
  - Via sentence embedding avec *all-mini-lm-L6-V2* sur la description des restaurants scrappés
  - Comparaison avec cosine similarity

## Itinerary

- Utilisation de l'API *openrouteservice* pour la création d'un itinéraire optimisé
- Récupération de l'itinéraire et affichage folium
- Ajout des infos de chaque attraction et restaurants
- Affichage du temps total et de la distance total estimé