

A decorative graphic on the left side of the slide consisting of two overlapping parallelograms. The front one is blue and the back one is a light greenish-blue. They are positioned diagonally, with the blue one partially covering the green one.

# Etude de cas Data Science

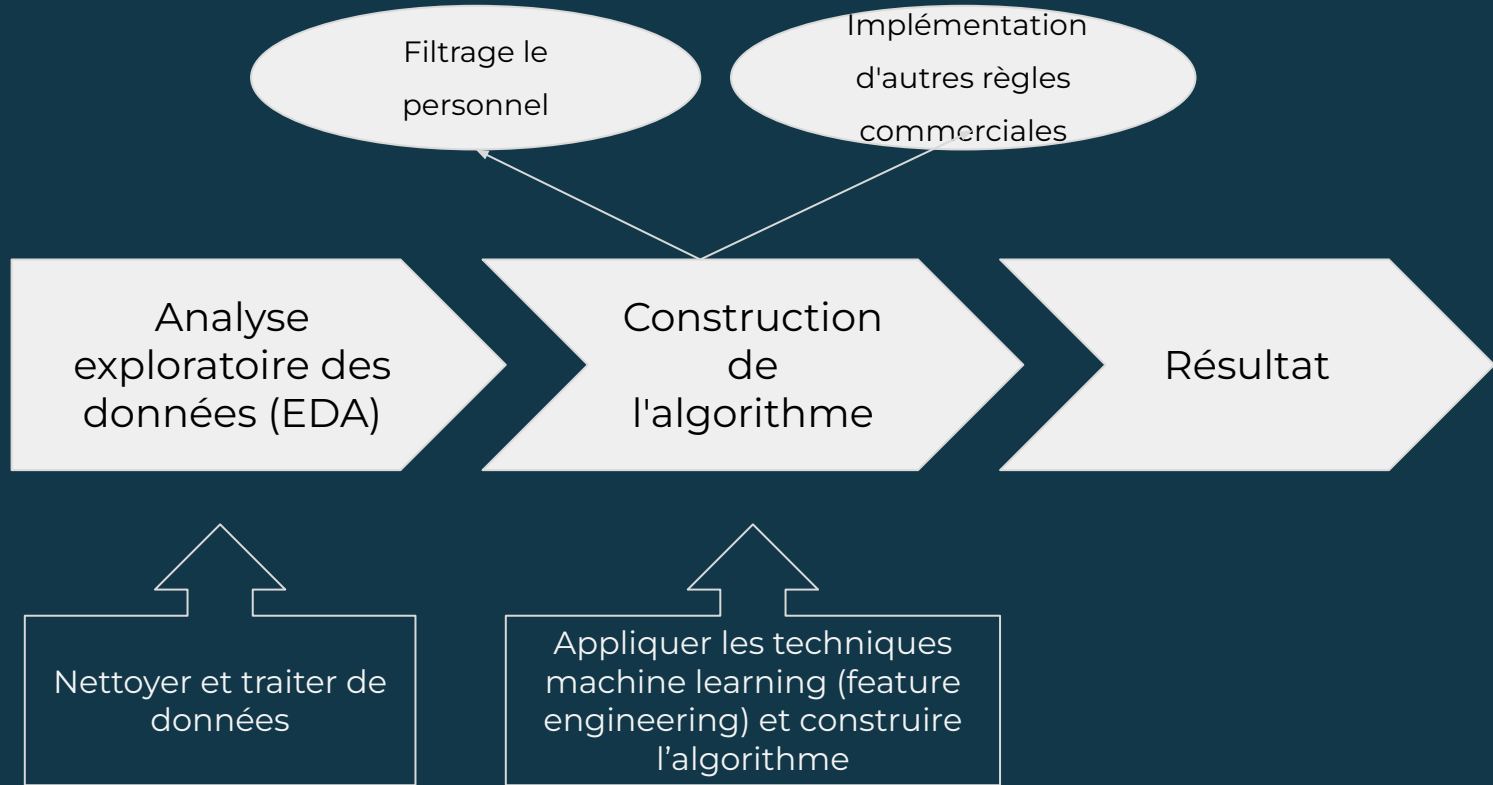
Dong-Pha Pham

## Problématique

Nous avons à disposition un jeu de données qui contient des identifiants anonymisés de 4 centres commerciaux français ainsi que des identifiants de téléphones dont les propriétaires ont visité ces centres commerciaux avec la date et l'heure des différents pings observés au cours de la visite

L'objectif de l'étude est, à partir de ces données, de déterminer les horaires d'ouverture de chacun des centres pour chaque jour de la semaine. Autrement dit, nous proposons un algorithme qui permettra de déterminer les horaires d'ouverture d'un centre ne faisant pas partie de ce dataset à partir de ses données de fréquentation

Note : Pour mieux comprendre le travail, il faut de consulter le notebook en même temps pour comprendre comment traiter les données et construire des algorithmes



## Etape 1 : Analyse exploratoire des données (EDA)

Ici, nous utilisons EDA pour explorer, analyser et visualiser le dataset

## Etape 2 : Construction de l'algorithme


En résumé notre algorithme, nous déterminons les heures d'ouverture et de fermeture d'un centre commercial comprend ces petites étapes :

- Étape 1 : Extraire les informations de la date et de l'heure des visites
- Étape 2 : Filtrer les visites de non-clients (le personnel du centre)
- Étape 3 : Mettre en œuvre d'autres règles commerciales, notamment : le pourcentage minimum de clients à servir pour répondre aux besoins des magasins ou les préférences des heures d'ouverture et de fermeture

## Etape 1 : Analyse exploratoire des données (EDA)


	shopping_center_id	device_local_date	device_hash_id
0	b43e9e4f-acd1-4941-874d-e0c5650ab91e	2019-09-14 10:00:25	6fdffac307
1	b43e9e4f-acd1-4941-874d-e0c5650ab91e	2019-09-14 17:13:15	386141ebd8
2	b43e9e4f-acd1-4941-874d-e0c5650ab91e	2019-09-14 9:07:06	b06242b848
3	b43e9e4f-acd1-4941-874d-e0c5650ab91e	2019-09-14 17:14:49	c13cc52e82
4	599cb959-11ef-49aa-9eb3-e6c17b4ea6ba	2019-09-14 10:17:35	f339ddf999

```
shopping_center_id
599cb959-11ef-49aa-9eb3-e6c17b4ea6ba    39827
cb2d5bb6-c372-4a51-8231-4ffa288a0c28    15639
b43e9e4f-acd1-4941-874d-e0c5650ab91e    15080
0cd35523-1eca-4f09-ab0d-0b506ae9d986    11292
dtype: int64
```



Nous visualisons sur le notebook que le dataset contient 3 colonnes et 81 838 entrées :

- shopping\_center\_id : représente les identifiants hachés de 4 centres commerciaux français
- device\_local\_data : représente la date et l'heure des différents pings observés lors de la visite du 2019-09-01 au 2019-09-17
- device\_hash\_id : représente les identifiants hachés de 5702 téléphones dont les propriétaires ont visité ces centres commerciaux




Pour explorer davantage la date et l'heure, nous allons convertir la colonne "device\_local\_date" (objets actuels) en type datetime pandas, puis extraire plus d'informations (telles que le jour de la semaine, la date, l'heure, l'heure) de cette colonne

Nous normalisons les données horaires à une demi-heure. Exemple : 10h36 devient 10h30, 9h04 devient 9h00, etc

	shopping_center_id	device_hash_id	device_local_datetime
0	b43e9e4f-acd1-4941-874d-e0c5650ab91e	6fdffac307	2019-09-14 10:00:25
1	b43e9e4f-acd1-4941-874d-e0c5650ab91e	386141ebd8	2019-09-14 17:13:15
2	b43e9e4f-acd1-4941-874d-e0c5650ab91e	b06242b848	2019-09-14 09:07:06
3	b43e9e4f-acd1-4941-874d-e0c5650ab91e	c13cc52e82	2019-09-14 17:14:49
4	599cb959-11ef-49aa-9eb3-e6c17b4ea6ba	f339ddf999	2019-09-14 10:17:35



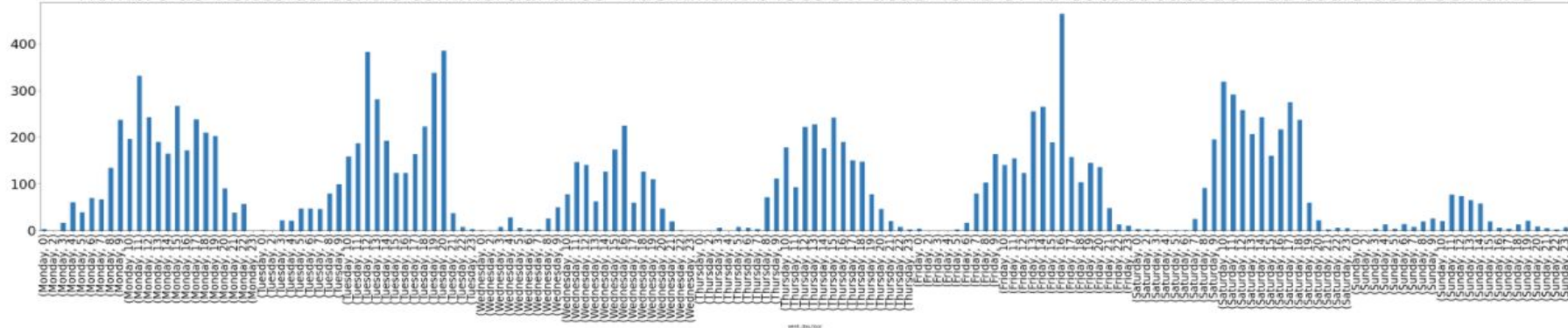


Nous créons de nouvelles variables extraites de la date qui nous servirons plus tard

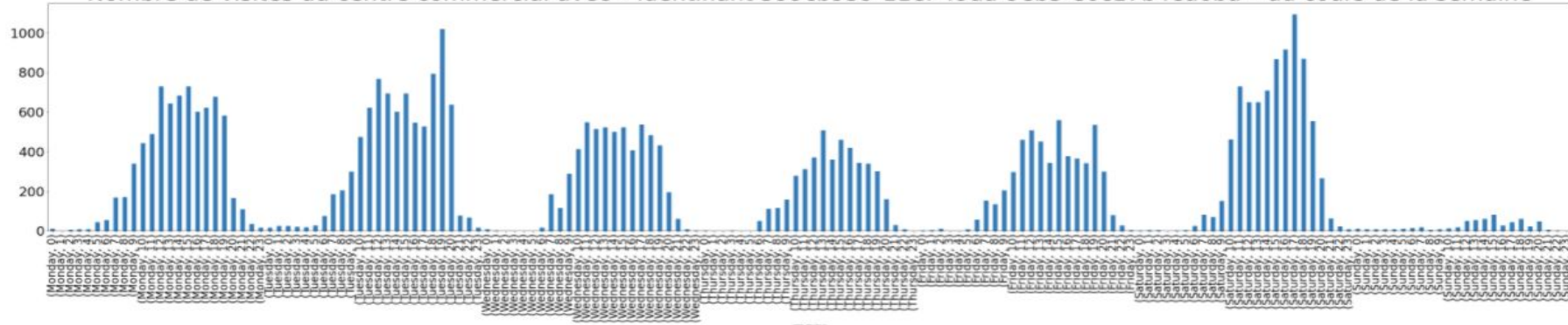
	shopping_center_id	device_hash_id	device_local_datetime	week_day	date	hour	time_rounded
0	b43e9e4f-acd1-4941-874d-e0c5650ab91e	6fdffac307	2019-09-14 10:00:25	Saturday	2019-09-14	10	10:00:00
1	b43e9e4f-acd1-4941-874d-e0c5650ab91e	386141ebd8	2019-09-14 17:13:15	Saturday	2019-09-14	17	17:00:00
2	b43e9e4f-acd1-4941-874d-e0c5650ab91e	b06242b848	2019-09-14 09:07:06	Saturday	2019-09-14	9	09:00:00
3	b43e9e4f-acd1-4941-874d-e0c5650ab91e	c13cc52e82	2019-09-14 17:14:49	Saturday	2019-09-14	17	17:00:00
4	599cb959-11ef-49aa-9eb3-e6c17b4ea6ba	f339ddf999	2019-09-14 10:17:35	Saturday	2019-09-14	10	10:30:00

Maintenant, nous visualisons le nombre de visites de chaque centre commercial à différentes heures de la semaine

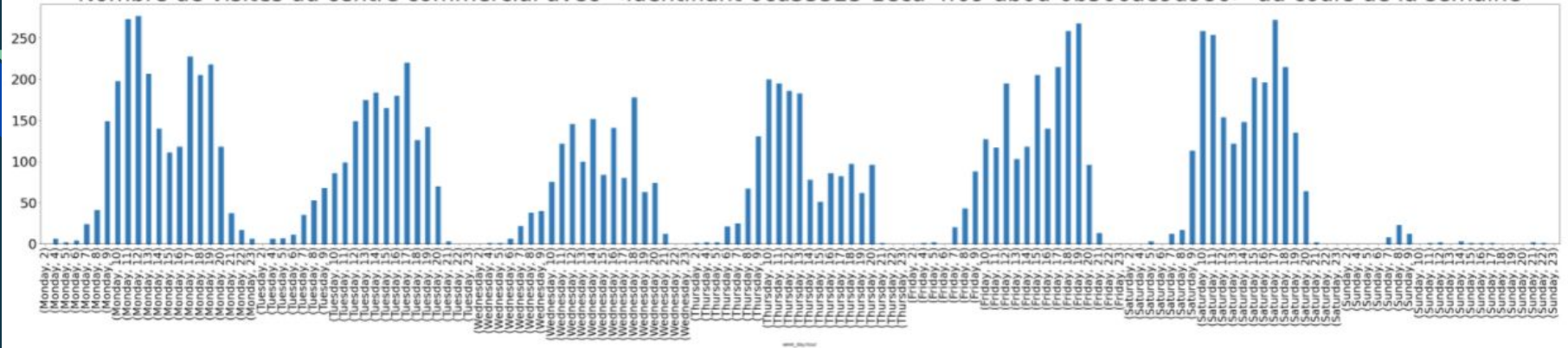
Nombre de visites du centre commercial avec <identifiant b43e9e4f-acd1-4941-874d-e0c5650ab91e> au cours de la semaine



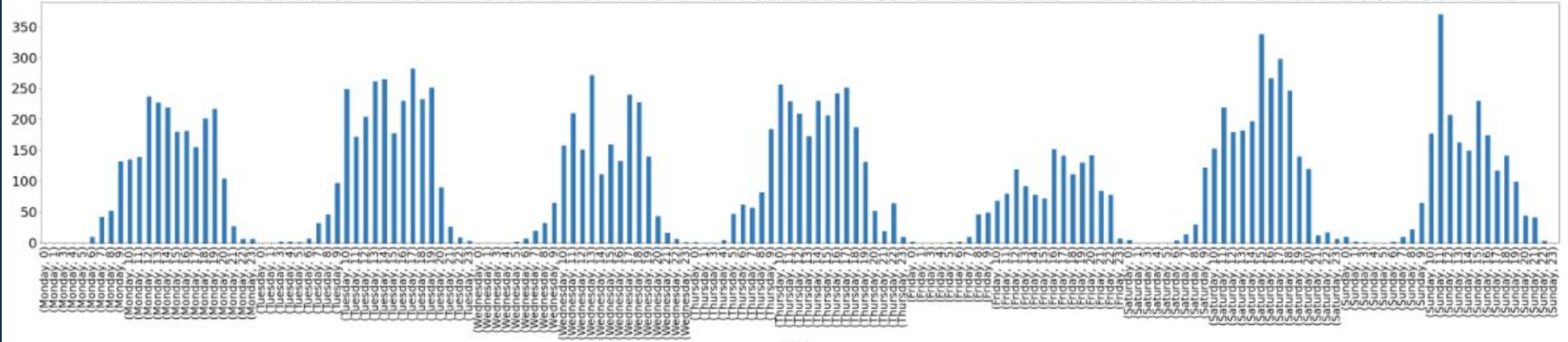
Nombre de visites du centre commercial avec <identifiant 599cb959-11ef-49aa-9eb3-e6c17b4ea6ba> au cours de la semaine




Nombre de visites du centre commercial avec <identifiant 0cd35523-1eca-4f09-ab0d-0b506ae9d986> au cours de la semaine



Nombre de visites du centre commercial avec <identifiant cb2d5bb6-c372-4a51-8231-4ffa288a0c28> au cours de la semaine






Selon ces visualisations, d'une part, il est évident que la majeure partie du trafic de l'appareil provient de 8h à 21h tous les jours de la semaine **sauf le dimanche**. Le dimanche, les répartitions de visites varient d'une boutique à l'autre. D'autre part, pendant les heures de fermeture, il y a une présence de visiteurs. Nous considérons donc ces visiteurs comme **le personnel**

Nous savons que, dans tout centre commercial, il y a un personnel qui est présent en dehors des heures d'ouverture du centre commercial. Exemple : les agents de propreté, les vigiles, les vendeurs, etc. Nous essayons de les identifier afin de les retirer du dataset pour ne pas les confondre avec les clients

## Etape 2 : Construction de l'algorithme

### Etape 2.1 : Filtrage le personnel

- L'idée de cette étape est de savoir comment détecter que `device_hash_id` est présent au moins `X jours` sur les `17 jours` d'historique dont nous disposons. X sera un paramètre que nous pourrons modifier si nécessaire
- Pour mieux filtrer le personnel, nous supprimerons également tout `device_hash_id` qui a passé plus de `Y heures en une journée` dans le centre commercial. Donc, nous considérons :
- Toute personne présente dans le centre commercial plus de `X jours` ou présente dans le centre commercial plus de `Y heures dans une journée` en tant que le personnel



Dans notre cas, nous définirons `day_threshold = 4` et `hour_threshold = 6`, ce qui signifie que nous considérons tous les appareils qui visitent un centre commercial plus de 4 jours, ou plus de 6 heures par jour, comme étant les appareils du personnel de ce centre, il faut donc les supprimer

## Etape 2.2: Implémentation d'autres règles commerciales

Comme observé, la plupart du trafic vient de (8h à 21h). Cependant, le trafic est faible tôt le matin (7h00-8h00) et tard le soir (20h00-22h00), alors que le coût d'ouverture pendant ces intervalles peut être plus élevé que d'habitude. Par conséquent, pour optimiser le profit, certains centres commerciaux peuvent préférer ouvrir après X heures du matin, ou fermer avant Y heures du soir, pour économiser le coût d'exploitation tout en servant la plupart (disons, 95 %) des clients



Nous allons écrire une autre fonction pour implémenter ces règles commerciales

Cette fonction consiste à implémenter d'autres règles commerciales

- @param: `df` (Dataframe) : jeu de données
- @param: `percent_of_cus_to_serve` (float) : le pourcentage minimum de clients que les centres commerciaux aimeraient servir
- @param: `open_hour_prefer` and `close_hour_prefer` (datetime.time) : l'heure d'ouverture et de fermeture de préférence




Si le pourcentage réel de client servi dans l'intervalle de préférence  $\geq$  le pourcentage minimum de clients que les centres commerciaux aimeraient servir (`percent_of_cus_to_serve`), nous sélectionnons cet intervalle comme l'heure d'ouverture et de fermeture. Sinon, nous sélectionnons l'heure minimale et maximale des visites réelles comme l'heure d'ouverture et de fermeture

Dans notre cas, nous définissons `percent_of_cus_to_serve = 95 %`, `open_hour_prefer = 8h00 am`, `close_hour_prefer = 22h00 pm`

	shopping_center_id	week_day	open_hour	close_hour
0	0cd35523-1eca-4f09-ab0d-0b506ae9d986	Monday	08:30:00	22:00:00
1	0cd35523-1eca-4f09-ab0d-0b506ae9d986	Tuesday	09:00:00	21:00:00
2	0cd35523-1eca-4f09-ab0d-0b506ae9d986	Wednesday	08:30:00	21:00:00
3	0cd35523-1eca-4f09-ab0d-0b506ae9d986	Thursday	08:30:00	21:00:00
4	0cd35523-1eca-4f09-ab0d-0b506ae9d986	Friday	09:00:00	21:30:00
5	0cd35523-1eca-4f09-ab0d-0b506ae9d986	Saturday	09:00:00	21:00:00
6	0cd35523-1eca-4f09-ab0d-0b506ae9d986	Sunday	07:30:00	22:30:00
7	599cb959-11ef-49aa-9eb3-e6c17b4ea6ba	Monday	08:00:00	22:00:00
8	599cb959-11ef-49aa-9eb3-e6c17b4ea6ba	Tuesday	08:00:00	21:30:00
9	599cb959-11ef-49aa-9eb3-e6c17b4ea6ba	Wednesday	08:00:00	21:30:00
10	599cb959-11ef-49aa-9eb3-e6c17b4ea6ba	Thursday	08:00:00	21:00:00
11	599cb959-11ef-49aa-9eb3-e6c17b4ea6ba	Friday	08:30:00	22:00:00
12	599cb959-11ef-49aa-9eb3-e6c17b4ea6ba	Saturday	08:30:00	22:00:00





En résumé notre algorithme, nous déterminons les heures d'ouverture et de fermeture d'un centre commercial comprend ces petites étapes :

Étape 1 : Extraire les informations de la date et de l'heure des visites

Étape 2 : Filtrer les visiteurs qui ne sont pas des clients (les personnels des centres)

Étape 3 : Mettre en œuvre d'autres règles commerciales, notamment : le pourcentage minimum de clients à servir pour répondre aux besoins des magasins ou les préférences des heures d'ouverture et de fermeture

# Résultat

## Centre 1

	shopping_center_id	week_day	open_hour	close_hour
0	0cd35523-1eca-4f09-ab0d-0b506ae9d986	Monday	08:30:00	22:00:00
1	0cd35523-1eca-4f09-ab0d-0b506ae9d986	Tuesday	09:00:00	21:00:00
2	0cd35523-1eca-4f09-ab0d-0b506ae9d986	Wednesday	08:30:00	21:00:00
3	0cd35523-1eca-4f09-ab0d-0b506ae9d986	Thursday	08:30:00	21:00:00
4	0cd35523-1eca-4f09-ab0d-0b506ae9d986	Friday	09:00:00	21:30:00
5	0cd35523-1eca-4f09-ab0d-0b506ae9d986	Saturday	09:00:00	21:00:00
6	0cd35523-1eca-4f09-ab0d-0b506ae9d986	Sunday	07:30:00	22:30:00

## Centre 2

	shopping_center_id	week_day	open_hour	close_hour
7	599cb959-11ef-49aa-9eb3-e6c17b4ea6ba	Monday	08:00:00	22:00:00
8	599cb959-11ef-49aa-9eb3-e6c17b4ea6ba	Tuesday	08:00:00	21:30:00
9	599cb959-11ef-49aa-9eb3-e6c17b4ea6ba	Wednesday	08:00:00	21:30:00
10	599cb959-11ef-49aa-9eb3-e6c17b4ea6ba	Thursday	08:00:00	21:00:00
11	599cb959-11ef-49aa-9eb3-e6c17b4ea6ba	Friday	08:30:00	22:00:00
12	599cb959-11ef-49aa-9eb3-e6c17b4ea6ba	Saturday	08:30:00	22:00:00
13	599cb959-11ef-49aa-9eb3-e6c17b4ea6ba	Sunday	08:00:00	22:00:00

## Centre 3

	shopping_center_id	week_day	open_hour	close_hour
14	b43e9e4f-acd1-4941-874d-e0c5650ab91e	Monday	08:00:00	22:00:00
15	b43e9e4f-acd1-4941-874d-e0c5650ab91e	Tuesday	08:00:00	21:30:00
16	b43e9e4f-acd1-4941-874d-e0c5650ab91e	Wednesday	08:00:00	22:00:00
17	b43e9e4f-acd1-4941-874d-e0c5650ab91e	Thursday	08:30:00	21:00:00
18	b43e9e4f-acd1-4941-874d-e0c5650ab91e	Friday	08:00:00	22:00:00
19	b43e9e4f-acd1-4941-874d-e0c5650ab91e	Saturday	08:00:00	22:00:00
20	b43e9e4f-acd1-4941-874d-e0c5650ab91e	Sunday	09:00:00	21:30:00

## Centre 4

	shopping_center_id	week_day	open_hour	close_hour
21	cb2d5bb6-c372-4a51-8231-4ffa288a0c28	Monday	08:30:00	22:00:00
22	cb2d5bb6-c372-4a51-8231-4ffa288a0c28	Tuesday	08:00:00	22:00:00
23	cb2d5bb6-c372-4a51-8231-4ffa288a0c28	Wednesday	08:00:00	21:30:00
24	cb2d5bb6-c372-4a51-8231-4ffa288a0c28	Thursday	09:00:00	22:00:00
25	cb2d5bb6-c372-4a51-8231-4ffa288a0c28	Friday	08:00:00	22:00:00
26	cb2d5bb6-c372-4a51-8231-4ffa288a0c28	Saturday	08:30:00	22:00:00
27	cb2d5bb6-c372-4a51-8231-4ffa288a0c28	Sunday	08:30:00	22:00:00



## Commentaires

En regardant les 4 tableaux ci-dessus, nous pouvons déterminer les heures d'ouverture de chacun des 4 centres commerciaux pour chaque jour de la semaine


Actuellement, les visites du client sont détectées par certaines règles de bon sens, alors peut-être qu'ils ne sont pas précis à 100%. Si le centre commercial exige que tout son personnel enregistre ses appareils, cela conduit à l'ensemble de données contenant une colonne supplémentaire classant si le trafic provient des clients ou du personnel. Cela nous aidera à filtrer avec précision le trafic du personnel




## Recommandation

Dans certaines situations particulières (par exemple la situation de Covid 19), certains centres commerciaux peuvent vouloir limiter le nombre de clients visitant à tout moment de la journée (Ce nombre soit dynamique en fonction de la taille des centre commerciaux). L'algorithme doit également être modifié pour refléter cette règles commerciales.

Pour améliorer cet ensemble de données, nous devons ajouter d'autres types de données. Exemple :

- 
- Le comportement de consommation des clients. Par exemple, si nous observons qu'il y a beaucoup de clients ont besoin de faire leurs courses, d'acheter, de manger ou de prendre un café, ... etc à l'heure d'ouverture de 8h00, nous pouvons envisager d'ouvrir le centre avant 8h00
  - L'identifiant de client. Exemple : Nous envisageons d'utiliser des caméras pour suivre les clients (suivi) et observer le moment où ils entrent dans le centre commercial au lieu d'utiliser leur identifiant de téléphone (ping). En effet, d'un point de vue personnel, il y a beaucoup des clients qui entrent dans le centre sans utiliser le wifi ou sans enregistrer leur identifiant ou quelque temps plus tard lorsqu'ils entreront dans le centre, ils accéderont au wifi central. Cela affecte l'exactitude des données car nous utilisons l'identité du téléphone du client pour déterminer quand le client est entré dans le centre



Une fois que nous aurons un ensemble de données "très précises", nous pourrions sélectionner les "feature engineering" qui affectent directement ou indirectement les heures d'ouverture du centre. Par conséquent, nous allons construire un algorithme pour déterminer les heures d'ouverture avec une plus grande précision