

## Workshop

---

### Data

#### yelp\_academic\_dataset\_business.json

```
{
  "business_id": "encrypted business id",
  "name": "business name",
  "neighborhood": "hood name",
  "address": "full address",
  "city": "city",
  "state": "state -- if applicable --",
  "postal code": "postal code",
  "latitude": latitude,
  "longitude": longitude,
  "stars": star rating, rounded to half-stars,
  "review_count": number of reviews,
  "is_open": 0/1 (closed/open),
  "attributes": ["an array of strings: each array element is an attribute"],
  "categories": ["an array of strings of business categories"],
  "hours": ["an array of strings of business hours"],
  "type": "business"
}
```

#### yelp\_academic\_dataset\_review.json

```
{
  "review_id": "encrypted review id",
  "user_id": "encrypted user id",
  "business_id": "encrypted business id",
  "stars": star rating, rounded to half-stars,
  "date": "date formatted like 2009-12-19",
  "text": "review text",
  "useful": number of useful votes received,
  "funny": number of funny votes received,
  "cool": number of cool review votes received,
  "type": "review"
}
```

#### yelp\_academic\_dataset\_user.json

```
{
  "user_id": "encrypted user id",
```

```

    "name": "first name",
    "review_count": number of reviews,
    "yelping_since": date formatted like "2009-12-19",
    "friends": ["an array of encrypted ids of friends"],
    "useful": "number of useful votes sent by the user",
    "funny": "number of funny votes sent by the user",
    "cool": "number of cool votes sent by the user",
    "fans": "number of fans the user has",
    "elite": ["an array of years the user was elite"],
    "average_stars": floating point average like 4.31,
    "compliment_hot": number of hot compliments received by the
user,
    "compliment_more": number of more compliments received by the
user,
    "compliment_profile": number of profile compliments received
by the user,
    "compliment_cute": number of cute compliments received by the
user,
    "compliment_list": number of list compliments received by the
user,
    "compliment_note": number of note compliments received by the
user,
    "compliment_plain": number of plain compliments received by
the user,
    "compliment_cool": number of cool compliments received by the
user,
    "compliment_funny": number of funny compliments received by
the user,
    "compliment_writer": number of writer compliments received by
the user,
    "compliment_photos": number of photo compliments received by
the user,
    "type": "user"
}

```

### yelp\_academic\_dataset\_checkin.json

```

{
    "time": ["an array of check ins with the format day-
hour: number of check ins from hour to hour+1"],
    "business_id": "encrypted business id",
    "type": "checkin"
}

```

### yelp\_academic\_dataset\_tip.json

```
{
  "text": "text of the tip",
  "date": "date formatted like 2009-12-19",
  "likes": compliment count,
  "business_id": "encrypted business id",
  "user_id": "encrypted user id",
  "type": "tip"
}
```

Idées :

- Déterminer la recette des restaurants pour trouver les caractéristiques d'un bon restaurant -> compliqué
- Déterminer l'humeur, la situation sociale du client en fonction du restaurant (nom, etc ...) qu'il fréquente, de la météo
- l'ambiance du quartier en fonction des mots clés dans catégorie
- Déterminer si le client est prêt à faire des km pour avoir un bon restaurant
- Déterminer la probabilité qu'un client aille au restaurant en sachant qu'un certain nombre de ses amis y soient déjà allé + ses commentaires, ses conseils ...
- Savoir si le goût est le même ou différent entre amis : restaurants fréquentés, commentaires postés, conseils donnés
- Chercher pourquoi un restaurant est fermé : trouver les caractéristiques

Choix: La météo influence t'elle la notation des clients ?

Données utilisées :

- User -> Id, average stars,
- review -> Business, Stars, user, date, type
- Business -> city, longitude & latitude
- New dataset -> météo -> date, city, temps

To do:

- Créer le dataset contenant les données météorologiques : chercher l'api, et créer le fichier sous le format « json »
- Rechercher les «rating» de l'utilisateurs X qui sont inférieur à sa moyenne
- Mettre en concordance avec la météo du jour recherchée
- Catégoriser par type de business

- Prédire les futurs notation des clients
- 

-Idée:

-Déterminer la probabilité qu'un client aille au business en fonction du rating du client.

-Prédiction future rating

-Données utilisées:

-User : id, friends

-Review: user\_id, stars, business\_id

- business : id, stars

Influence des amis, qui écouter ?

To Do :

- Create graphFrames

Pour chaque business :

- Récupérer la date + id\_business, analyser si ses amis sont déjà allés auparavant.

- Si oui, calculer la moyenne rating de tous ses amis.

- Si Moyenne rating >3 => Bon business => **Personne influençable**

**On peut prédire son future rating**

- Sinon mauvais => **N'ecoute pas ses amis**

- Analyser son rating

- Si non, voir si ses amis sont y allés après lui.

Fin pour, Savoir si une personne est influençable ou non