# 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"

}

—————————————————————————————————————————————-

IDEA

* Determine the probability that an user go to a business if him friends have good rating this buisiness.
* Class influencable and not influencable people
* Predict the future rating of a user based on him friends rating on the same business

Data will be used

* **User** : id, friends
* **Review** : user\_id, stars, business\_id

To Do

1. Create a graphFrames with all friends of users
2. For each business
   * From business table Get the id buisness and the date
3. For each users :
   * Join user & review : id\_user,id\_business,date,stars
   * Joint users & friends: id\_user, id\_friend
   * Join friends\_id & review: id\_friend, id\_business\_friend,
   * date\_business\_friens,stars\_friend
4. Delete all rows with id\_business != id\_business\_friend
5. Delete all rows where date\_business\_friend > date
6. Group by business\_id : Do average of stars\_friend,
7. Compute the mean friends stars (Mstars)
   * + - * if Mstars > 3 => influençable people
         * else, no influençable people

Présentation:

Extensions: Compute a page rank on friends graph to get the five best friends