There are two folders in the main trunk: *users* and *suggestions*.

- *Users*: all the users' profiles, one user per file in json format.
- *Suggestions*: all the places that occurred in the users' profiles, one place per file in json format.

Detailed data format explanation:

Users:

Each user's profile in json format is an object:

Key	What's for	Value Type
profile_questions	Yelp's official user profile, see below for the details	object
-Don't Tell Anyone Else But...	Yelp's official user profile subfield	string
- Location	Same as above	string
- My Favorite Movie	Same as above	string
- My Last Meal On Earth	Same as above	string
- My Second Favorite Website	Same as above	string
- Things I Love	Same as above	string
- Yelping Since	Same as above	string
reviews	Contains all places that the user had rated. Each element of reviews is an object, see below for the details	array of objects
- identifier	The identifier of the place. You can use this to locate the corresponding suggestion file in suggestion folder if more details about this place are needed.	string
- pid	The yelp url of this place	string (url)
- rating	The user's scalar rating for this place	number (float)
- comment	The user's review text for this place	string
- timestamp	When user gave the rating	string
user_name	User name	string
user_stats	Some statistics about the user, see below for the details	object
- fans_cnt	Number of fans	number (int)
- firsts_cnt	Number of places that the user is the first one to give review	number (int)
- friends_cnt	Number of friends	number (int)
- local_photos_cnt	Number of photos uploaded by the user	number (int)
- review_updates_cnt	Not sure about this	number (int)
- reviews_cnt	Number of reviews given by the user	number (int)

Suggestions:

Each suggestion(place) in json format is an object:

Key	What's for	Value Type
categories	Categories: one place may have multiple categories. For example, [“nightlife>bars”, “restaurants>new american”]. For each category, it shows the hierarchical category tree from most general to most specific and is split by “>”	array of strings
goodfor	Yelp's suggestion of this place	string
hours	Business hours	array
id	identifier	string
location	Location, see below for the details	object
- addressLocality	Locality of address	string
- addressRegion	Region of address	string
- postalCode	Postal Code of address	string
- streetAddress	Street of address	string
name	Name of the place	string
overallRating	Overall Rating from Yelp	number (float)
phone	Phone number	string
reviews_detail	All reviews for this place. Each of them is an object. See below for the details	array of objects
- user_id	The user ID of this review	string (url)
- user_name	The user name of this review	string
- rating	The user's scalar rating for this place	number (float)
- comment	The user's review text for this place	string
- user_review_cnt	The number of reviews given by this user in total	number (int)
- user_friends_cnt	The number of friends of this user	number (int)
total_review_number	Total review number	number (int)
url	The URL (website) of this place	string (url)

Data Splits

In order to evaluate the effectiveness of a method, it is useful to iteratively split the data into training set and testing set without overlap. There is a ready-in-hand data splits which you can download at https://s3.amazonaws.com/irj2014_yelp_data/data_splits.tar.gz.
It basically applies 10-fold cross validation: uses 90% of each user's reviewed places as training and uses the rest as testing. The data splits are stratified which means it is guaranteed that there are some positive and negative in testing set.
Explanations:

- **dlist**: training set. one line per "user:suggestion" pair
- **qlist**: testing set. one line per "user:suggestion" pair
- **judgment**: the judgment of the testing set. Ratings of 1,2,3 are judged as non-relevant. Ratings of 4 or 5 are judged as relevant. This file is intended to be used together with ireval.pl

  Please note that there is NO label(judgement) for training set. you need to find the rating and decide whether it is positive/negative