Chicago Health Score: The integration of Chicago Food Inspections and Yelp Reviews

Introduction

Through the analysis of the City of Chicago Food Inspection and Yelp reviews data we can create a business opportunity by providing accurate information about the quality of the service as well as insights about the conditions on how food providers do business in the City of Chicago.

This research will use different data mining techniques to clean, separate and cluster the food inspection and Yelp data in order to find a correlation between negative Yelp reviews including terms that can be related to customers getting sick or having a bad experience due to bad conditions of the restaurant. If it is possible this research will determine patterns, key words and behaviors that can be translate into a health score for Chicago Restaurants.

About the Chicago Food Inspections Data

The Chicago Department of Public Health has a Food Protection Program, where a field inspector visits different business that handle food following a determine form and then the data is reviewed by a State of Illinois Licensed Environmental Health Practitioner. This data is updated every Friday, and goes all the way back until 2010, when the program started.

There are different ways to download the data, we are working with its Excel and cvs format. The data is compose of string, number, address, dates, gps location fields. All but one of the columns on this data is structure, and are divided in 17 columns and there are 100,730 since the last updated Friday, March 6, 2015. With a size of approximately 70Mb.

Column Name	Column Description	Data Type
inspection_id	Number to identify inspection	Number
dba_name	Doing Business as	Plain Text (String)
aka_name	Also Know as	Plain Text (String)
license_	Dpt of Business Affairs unique number	Number
facility_type	Type of business	Plain Text (String)

Risk	Risk to public health	Plain Text (String)
Address	Street address of the business	Plain Text (String)
City	City where the business is located	Plain Text (String)
State	State where the business is located	Plain Text (String)
Zip	Business Zip code	Number
inspection_date	Date of the inspection	Data & Time
inspection_type	There are 7 different type of inspections	Plain Text (String)
Results	Result of the inspection	Plain Text (String)
Violations	Health code violated by the business	Plain Text (String)
Latitude	latitude of the business	Number
Longitude	longitude of the business	Number
Location	latitude and longitude of the business	Location

Description of the Data

Inspection type

- **Canvass:** Random check of the business.
- **Consultation:** When a business is going to open for the first time.
- **Complain:** When a complain about the business is filed.
- **License:** When a business is obtaining a particular license.
- **Suspect food poisoning:** When a business is suspected of making a person ill.
- **Task-force inspection:** Bar or tavern inspection.
- **Re-inspection:** After one of the previous inspections is performed.

Results

- **Pass:** When no critical violations are found.
- **Pass with conditions:** When a critical violation is found but its corrected during the inspection.
- **Fail:** A critical violation that couldnt be fix during the inspection.
- **Others:** Business not located, out of business, no entry.

Risk (To affect public health)

- **Risk 1:** High
- **Risk 2:** Medium
- **Risk 3:** Low

Chicago Health Score: The integration of Chicago Food Inspections and Yelp Reviews
Jose Luis Rodriguez Orjuela
COMP300

About the Yelp data parameters

Yelp API returns a json-object per line. Every object contains a 'type' field, where we can determine if is a business, a user, or a review.

Business Objects

Basic information about local businesses. We can determine the 'business_id' of restaurants in the Chicago Food Inspection data to fetch more information about a particular business. The fields are as follows:

```
{
  'type': 'business',
  'business_id': (a unique identifier for this business),
  'name': (the full business name),
  'neighborhoods': (a list of neighborhood names, might be empty),
  'full_address': (localized address),
  'city': (city),
  'state': (state),
  'latitude': (latitude),
  'longitude': (longitude),
  'stars': (star rating, rounded to half-stars),
  'review_count': (review count),
  'photo_url': (photo url),
  'categories': [(localized category names)]
  'open': (is the business still open for business?),
  'schools': (nearby universities),
  'url': (yelp url)
}
```

Review Objects

Contain the review text, the star rating, and information on votes Yelp users have cast on the review. The business_id is crucial in this part to associate this review with others of the same business.

```
{
  'type': 'review',
  'business_id': (the identifier of the reviewed business),
  'user_id': (the identifier of the authoring user),
  'stars': (star rating, integer 1-5),
  'text': (review text),
  'date': (date, formatted like '2011-04-19'),
  'votes': {
    'useful': (count of useful votes),
    'funny': (count of funny votes),
    'cool': (count of cool votes)
  }
}
```

User Objects

Contain aggregate information about a single user across all of Yelp (including businesses and reviews). For further research the User Object can determine the weight of the review, as users that are more active in Yelp will be different to those that just create an account to make a complain about a particular restaurant.

```
{
  'type': 'user',
  'user_id': (unique user identifier),
  'name': (first name, last initial, like 'Matt J.'),
  'review_count': (review count),
  'average_stars': (floating point average, like 4.31),
  'votes': {
    'useful': (count of useful votes across all reviews),
    'funny': (count of funny votes across all reviews),
    'cool': (count of cool votes across all reviews)
  }
}
```