Yelp Challenge: Culture Trend

1. Introduction

This project will perform a yelp dataset challenge based on the dataset provided by yelp. The main goal is to use machine learning technologies and concepts to predict Yelp rating base on the mixture of features from dataset and generated from review text analyzing.

2. Related Work

Both predicting the rating from business information and review text analyzing have been explored. And achieved quite good result.

For feature selection, methods including univariate Feature Selection, recursive Feature Elimination, tree-Based Feature Selection are used to generate powerful features.

For prediction models, Support Vector Machine, Multinomial Logistic Regression, Multinomial Naive Bayes, Gaussian Discriminant Analysis, Decision Trees and Random Forest Classifier, Linear Regression with Regularization, Support Vector Regression are tried and achieved about 46

For review text analyzing, techniques as Latent Dirichlet Allocation (LDA), Non-negative Matrix Factorization (NMF) and models like LR - Logistic Regression, NB - Multinomial Naive Bayes and AB - AdaBoost are tried.

3. Dataset

The dataset is provided by yelp, which is a new version of iteration of its main business – business info, user info, reviews and comments on different business from users.

- Input:
  The input is the business data from yelp, which contains different features that the business have

- Ouput:
  The output is the stars(rating) of the business (rounded to half star)

3.1. Data Size

Basically, the dataset contains 2.2M reviews and 591K tips by 552K users for 77K business. 566K business attributes (e.g. hours, parking availability, ambience etc.) are given. The 552K users are connect with a social network, which is a graph 3.5M edges.

3.2. Data Fomat

The data is given in Json format and in four different types:

- **Business:**

```
{
        ”type”: ’business’,
        ”business” : (encrypted business id),
        ”name”: (business name),
        ”neighborhoods”: [(hood names)],
        ”full_address” : (localized address),
        ”city”: (city),
        ”state”: (state),
        ”latitude”: latitude,
        ”longitude”: longitude,
        ”stars”: (star rating, rounded to half-stars),
        ”review_count”: review count,
        ”categories”: [(localized category names)]
        ”open”: True / False (corresponds to
        closed, not business hours),
        ”hours”: {
        (day_of_week): {
        ”open”: (HH:MM),
        },
        ... },
},
```

- **Review:**

```
{
        ”type”: ”review”,
        ”business_id”: (encrypted business id),
        ”user_id”: (encrypted user id),
        ”stars”: (star rating, rounded to half-
        stars),
        ”text”: (review text),
        ”date”: (date, formatted like
        ”2012-03-14”),
        ”votes”: (vote type): (count),
        }
```

- **User:**

```
{
        ’type’: ’user’,
        ’user_id’: (encrypted user id),
        'name': (first name),
        'review_count': (review count),
        'average_stars': (floating point average, like 4.31),
        'votes': (vote type): (count),
        'friends': [(friend user_ids)],
        'elite': [(years_elite)],
        'yelping_since': (date, formatted like '2012-03'),
        'compliments': {
        (compliment_type): (num_compliments_of_this_type),
        ...
        },
        'fans': (num_fans),
}
```

- **Check-in:**

```
{
        'type': 'checkin',
        'business_id': (encrypted business id),
        'checkin_info': {
        '0-0': (number of checkins from 00:00 to 01:00 on all Sundays),
        '1-0': (number of checkins from 01:00 to 02:00 on all Sundays),
        ...
        '14-4': (number of checkins from 14:00 to 15:00 on all Thursdays),
        ...
        '23-6': (number of checkins from 23:00 to 00:00 on all Saturdays)
        }, # if there was no checkin for a hour-day block it will not be in the dict
}
```

- **Tip:**

```
{
        'type': 'tip',
        'text': (tip text),
        'business_id': (encrypted business id),
        'user_id': (encrypted user id),
        'date': (date, formatted like '2012-03-14'),
        'likes': (count),
}
```

3.3. Data Source

As stated in the yelp's website, those data came from 10 cities across 4 countries:

- **U.K.**: Edinburgh
- **Germany**: Karlsruhe
- **Canada**: Montreal and Waterloo
- **U.S.**: Pittsburgh, Charlotte, Urbana-Champaign, Phoenix, Las Vegas, Madison

3.4. Why The Dataset

To evaluate the relationship between culture and business trend, we need a relatively large size of data to train the machine. Yelp is a well known review website contains huge amount of not only the information about the business, but also the check-in count(which is a excellent representation of the trend during a time), and the user's rating, comments etc. With those data, we can perform different machine learning skills to find out the target. And the final result should be explicit as well as matches our common sense.

4. Methodology

4.1. Business Preference Based on Cities

We know that people from different cities have different habit. In some big cities driving tend to be a necessary traveling option for customer (e.g. cities in the Silicon Valley, Los Angeles etc.), but in some other cities it is not necessary the case (e.g. Boston and New York). Hence, in different cities, people have different attitude toward business's parking availability and traveling distance. By comparing the influence of different factors' on user's rating, we can find out which factor matters most, in different cities.

Firstly, we will sort the business based on the business type (different business should have different required features, while 24 hrs opening is a good feature for fitness center and restaurant, it is not necessary for a school).

Secondly, we will divide the dataset into two parts. Part 1 which is 80

Thirdly, we will perform different machine learning technologies to training the classifier.

Finally, we will use the classifier to predict the test data (the business's average rating)

5. Evaluation

Basically we want to use Precision, Recall, F-Score and Accuracy to evaluate our classifier.