Yelp Challenge: Rating Prediction

Yuanjian Lai LAI.YUA@HUSKY.NEU.EDU
Northeastern University, 360 Huntington Ave, Boston, MA 02110

Zhikai Ding DING.ZHIKA@HUSKY.NEU.EDU
Northeastern University, 360 Huntington Ave, Boston, MA 02110

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. Novelty

Yelp is the hottest business review website among the world. It basically helps people (especially for those that have no experience on the business) to have direct information from peers, and therefore customers can make better decisions. Based on Yelp, a new pattern of consumption has been created. However, while customers benefit from Yelp, business owner, on the other hand, can also utilize this mediate platform to optimize itself. When the business owners decide to launch a new business, they may want to have knowledge about what features they should focus on (or be inclined to) to enhance the average user experience. Based on this assumption, we believe machine learning is a good way to create the solutions.

3. 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% accuray. (Kyle Carbon, 2015)

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. (Rakesh Chada, 2015)

Here what we want to do is to combine the prediction from review text and the features of business itself, most of the previous work focused on either review text(mainly sentimental analyze) or business features (such as parking, child-friendly, wheelchair-friendly). A paper from Standford mentioned their way of combining these two by extracting features from review-text and mix them with features directly from dataset, but there is still plenty of work for optimizing.

Based on their efforts, our work will focus on investigating the generation of new features based on review text and user data that may hold predictive power. In particular, combining analyzing review text to develop new features or stronger models for sentiment analysis. In addition to sentiment, user data could also be analyzed to gather other features about the business that can be improved. We will also try some pre-processing method such as sorting businesses into categories before running predictions to see if there is difference.

4. 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)

4.1. 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.

4.2. 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),
}
```

4.3. 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

4.4. 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.

5. Methodology

6. Sort Type

There are various tracks for businesses(i.e. restaurant, barber shop), it is apparent that businesses in different tracks share very different features. While parking maybe a fundamental feature for supermarkets, it is not necessarily the same for a cafe store. Mixing all the businesses together regardless their business type will no doubt introduce much noise into our training data. Hence, we may want to sort the business by their categories and train different models for each of the category. A very naive way is using unsupervised clustering to sort the data. On the other hand, notice that we have much information from the dataset itself (business, category and type for the business data), we might utilize those data to sort it as well. We will find out which is better during the project.

6.1. Models

We will try the tested models on the dataset, and improve the accuracy by adding more features depending on their type. If time allows, we will try more complicated models like Neural Networks or Adaboost (Elkouri, 2015). The contents below are mainly from wikipedia. (WIKI, 2016)

- Multinomial Logistic Regression

  multinomial logistic regression is a classification method that generalizes logistic regression to multiclass problems, i.e. with more than two possible discrete outcomes.[1] That is, it is a model that is used to predict the probabilities of the different possible outcomes of a categorically distributed dependent variable, given a set of independent variables (which may be real-valued, binary-valued, categorical-valued, etc.).

- Multinomial Naive Bayes

  naive Bayes classifiers are a family of simple probabilistic classifiers based on applying Bayes' theorem with strong (naive) independence assumptions between the features.

- Gaussian Discriminant Analysis

  Gaussian Discriminant Analysis(Also called Linear discriminant analysis (LDA)) is a generalization of Fisher's linear discriminant, a method used in statistics, pattern recognition and machine learning to find a linear combination of features that characterizes or separates two or more classes of objects or events. The resulting combination may be used as a linear classifier, or, more commonly, for dimensionality reduction before later classification.

- Decision Trees and Random Forest Classifier

  Random forests is a notion of the general technique of random decision forests[1][2] that are an ensemble learning method for classification, regression and other tasks, that operate by constructing a multitude of decision trees at training time and outputting the class that is the mode of the classes (classification) or mean prediction (regression) of the individual trees. Random decision forests correct for decision trees' habit of overfitting to their training set.

- Linear Regression with Regularization

  linear regression is an approach for modeling the relationship between a scalar dependent variable y and one or more explanatory variables (or independent variables) denoted X. The case of one explanatory variable is called simple linear regression. For more than one explanatory variable, the process is called multiple linear regression

- Support Vector Machine

  support vector machines (SVMs, also support vector networks[1]) are supervised learning models with associated learning algorithms that analyze data used for classification and regression analysis. Given a set of training examples, each marked for belonging to one of two categories, an SVM training algorithm builds a model that assigns new examples into one category or the other, making it a non-probabilistic binary linear classifier.

- Support Vector Regression.

  The model produced by support vector classification (as described above) depends only on a subset of the training data, because the cost function for building the model does not care about training points that lie beyond the margin. Analogously, the model produced by SVR depends only on a subset of the training data, because the cost function for building the model ignores any training data close to the model prediction.

6.2. Workflow

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% of the dataset will be used as the training data. And the remaining part 2 will be used to test. The test and training data should be well-distributed in cities.

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)

7. Evaluation

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

References

Prasanth Kyle Carbon, Kacyn Fujii. Applications of machine learning to predict yelp ratings. Technical report, Stanford University Stanford, CA, 2015.

Rakesh Chada Rakesh Chada. Data mining yelp data - predicting rating stars from review text. Technical report, Stony Brook University, 2015.

Andrew Elkouri. Predicting the sentiment polarity and rating of yelp reviews. Technical report, Dept. of Mathematics, Statistics and Physics Wichita State University, 2015.

WIKI, 2016. URL https://en.wikipedia.org.