Reflection On Prediction of Yelp Review Star Rating

Hemeng Maggie Li, Chudan Ivy Liu, Jiawen Jasmine Zhu

1. Introduction

Yelp has become a more and more popular app in our daily life as a significant influence on consumers' behavior. When we do not know where to go for dinner, we use Yelp to check out the restaurants nearby or those recommended by Yelp based on our past reviews. In this project, we hope to get our hands on some machine learning experience with the Yelp dataset. The project has two tiers. First, using a customer's review on a business to predict the star rating given by the customer, which is our major initial vision. Second, we extend the project to geographical data visualization by plotting the restaurant price range and stars rating on a map.

2. Data Source

We used a deep dataset of Yelp Dataset Challenge which is available on- line (https://www.yelp.com/dataset_challenge), which includes the data over 4, 100, 000 reviews, and a lot of businesses. Our project used 10,000 data points as the primary dataset for star rating prediction, 12,700 data points for price range visualization and 22, 893 data points for stars rating visualization.

The dataset are in json format, providing comprehensive information including restaurant business profile, review text, the star rating, location coordinates, price range and check-in information, etc.

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"
}
```

Figure 2.1 Json Attributes of Yelp Dataset

3. Progress and the Final System's Capabilities

I. Star Rating Prediction

After looking into different python libraries, the candidates used in this projects include Sklearn, Textblob, NLTK, Numpy, Pandas and Geoplotlib. We used 10,000 review texts and star rankings. For feature extraction, we included five features including:

- Word count of positive words
- Word count of negative words
- Whether there are more positive numbers
- The average positivity score per sentence
- The average polarity score per sentence

Then we marked three quarters of data as training set and the rest as the testing set for the model. After we processed the initial dataset from the json file, we trained a star rating prediction system using two different classifiers: the k-nearest neighbors algorithm and decision tree classifier with the help of sklearn library. After tunning the 'k' value for the kNN classifier and the depth for decision tree classifier through cross-validation, we made the prediction and tested these predictions. The prediction results are listed below:

```
Created and trained a KNN classifier

The total number of review we are testing is: 2500
The total number of correct predict using current KNN classifier is: 1035
```

Figure 3.1 Statistics of kNN Prediction (Before)

```
Created and trained a DT classifier

Feature Importance of DT:
[ 0.05918616  0.01252788  0.87153574  0.01230199  0.04444823]

The total number of review we are testing is: 2500
The total number of correct predict using current DT classifier is: 1056
```

Figure 3.2 Statistics of Decision Tree Prediction (Before)

To further improve the accuracy of the prediction, we sorted the restaurant reviews into three groups, favorable (reviews with 4 or 5 stars), neutral (reviews with 3 stars) and

unfavorable (reviews with 1 or 2 stars) and trained both classifiers again. The prediction accuracy results are listed below:

.

```
Created and trained a KNN classifier

The total number of review we are testing is: 2500
The total number of correct predict using current KNN classifier is: 1746
```

Figure 3.3 Statistics of Decision Tree Prediction (After)

```
Created and trained a DT classifier

Feature Importance of DT:
[ 0.03508341  0.07683718  0.7888389   0.03363398  0.06560652]

The total number of review we are testing is: 2500
The total number of correct predict using current DT classifier is: 1760
```

Figure 3.4 Statistics of Decision Tree Prediction (After)

From the statistics shown above, we could see that by classifying the ratings into only three general categories, the accuracy score of kNN classifier improved from 0.414 to 0.6984, and the accuracy score of decision tree classifier improved from 0.4224 to 0.704.

II. Geographical Data Visualization

Yelp dataset has a bunch of attributes that worth research. Data visualization helps us have a whole picture of these attributes better. We selected two attributes, star rating and price range which might be influenced by geological factors.

To make a map plot, we chose geoplotlib, a Python library of visualizing geographical data. First, we extracted the latitude, longitude, price range and star rating attributes of each business from Json into a csv file. Then we fed geoplotlib with this csv file. Geoplotlib locates the business with 'lat' and 'log' attributes. We colored the business according to its price level and star rating.

We visualized the entire dataset which contains 90,301 data points to know the distribution of these Yelp business. According to Figure 3.1, we found out the dots are clustered around several specific centers. Since the data set is pretty biased on a state level, we decided to select a city and analyze the Yelp Business around it.

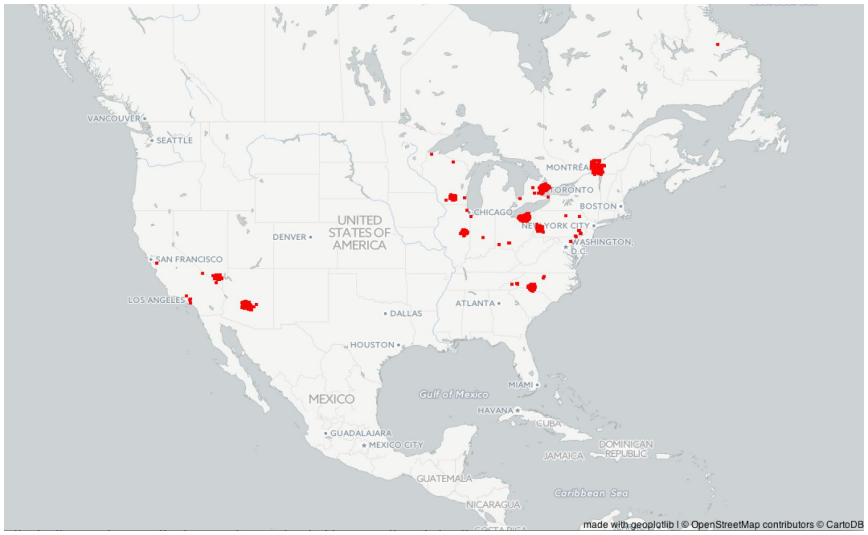

Figure 3.5 Business Distribution of Yelp Dataset

Because Las Vegas is famous for its entertainment industry and covered by Yelp data points in above figure. We chose to visualize Yelp business with city attribute 'Las Vegas' specifically.

For the price level map, there were 4275 $ business marked as black; 6806 $$ business marked as blue, 1187 $$$ business marked as green and 431 $$$$ business marked as red on the map. The map makes sense because the most expensive business are mainly located along central area of Las Vegas. $ and $$ business have reasonable prices so they have higher percentage and cover most part of the map.

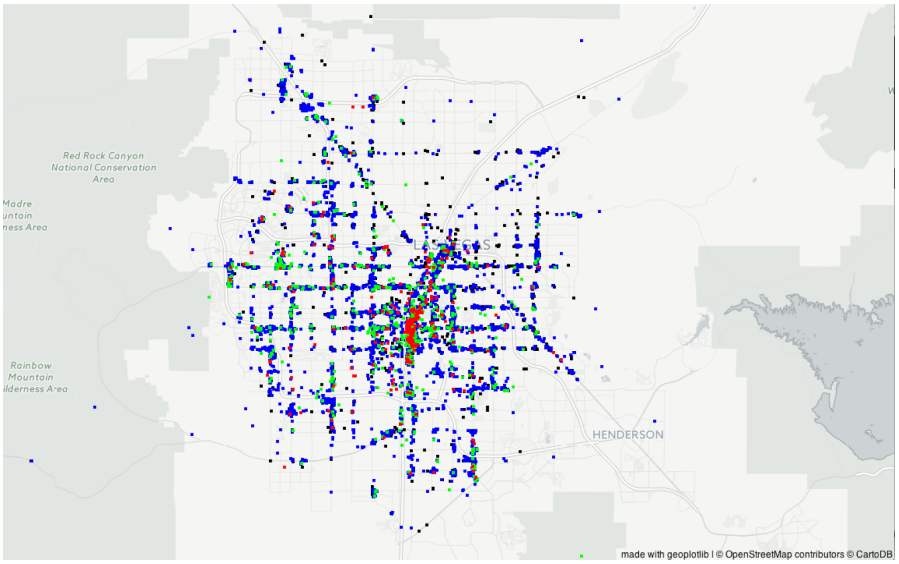

Figure 3.6 Price Range of Yelp Dataset -Las Vegas

For the star rating map, there were 2192 1-2-star business marked as blue; 8713 3-star business marked as green, 11987 4-5-star business marked as red on the map. The map surprises us a little bit because we didn't expect there are this many highly rated business. We think one reason behind is people tend to be nice rating the business nowadays. Another reason might be low-rated business become less and less popular so they have to close.

Through geoplotlib, we managed to have a geographical visualization of dataset, and have better understanding of the social ecology behind Yelp Dataset. These visualizations pass multi-dimensional information and are more persuasive that traditional graphs, such as scatterplot and pie chart. Moreover, geoplotlib allows interactivity in the visualization. Users are able to manipulate the graph in multiple ways such as zoom in and out using I and O on keyboard, checking different parts by dragging, etc.

4. Future Extension

As we finished the project, we found that we can tackle this project in many other ways. In particular, we could tackle the machine learning part of the project differently. Currently, we use Classifiers to classify the reviews into different labels or categories. As we ran the prediction, we found that Regressor could be an alternative to the Classifier, which would probably give a better result for star ratings prediction.

Since our labels (star ratings) are all numerical values (integers), using classifiers classify them into values that are only integers. But for some of the reviews that are really ambiguous between two integers, such as 2 and 3, the Classifiers would not do too well. Using Regressor instead can give us a prediction whose value of the label in between with decimals. For example, some reviews can get a prediction of 3.4 or 3.7, which the ratings still make sense. In addition, if we use regressor, each prediction contributes to the prediction of next review, which is really useful.

If we have more time and opportunity to extend this in the future, there are a lot of things that we can work on. Firstly, we can use regressors instead of classifiers to check whether regressors works well with the review predicting. Also, for our project, we only trained our predictors using 10,000 of the review from the dataset, even though the whole dataset contains about 4.1 million reviews. If we can train the model with more reviews, the performance of prediction will be improved. However, our current way of data processing takes a lot of memory space and time even for 10000 reviews. If we can work

more on this project, we can find a better way to process the data without consuming a lot of memory and time, so that we can take advantage of the huge amount of data we have.

For the data visualization part, we can definitely do a lot more. We only explored the dot plot from geoplotlib. If we have more time, we can check out more visualization, such as heat plot. In addition, instead of just visualizing the distribution of price range and star rating of businesses in the U.S., we can gather more data about demographics, such as population distribution, traffic condition, etc., and use these factors to predict how high rating businesses locate and a lot more possibilities!!!