Homework 6: Prediction of rating from Yelp review text

Grover, Karan & Arora, Pragya & Ghai, Piyush

{grover.120, arora.170, ghai.8}@osu.edu

November 30, 2016

1 PROBLEM STATEMENT

1.1 ABOUT YELP

Yelp [1] a famous website as well as a mobile app which publishes crowd sourced reviews about food joints and businesses. It also has a division which handles online reservations for restaurants. **Yelp Dataset Challenge**[2] is a publicly open contest sponsored by Yelp, in which the participants are challenged to use Yelp's data in an innovative way.

1.2 OUR MISSION

The Yelp dataset downloaded from Yelp dataset challenge website is huge and consists of over 2.7M reviews by roughly 687k users for over 86k businesses [2]. For this project, we chose to predict a review's rating based on the review text. The rating will be done on a scale of 1-5, where 1 stands for awful and 5 stands for excellent. We built multiple models and accessed which models would fit our use case the best. This is explained in more depth in the later sections of this report.

1.3 ABOUT THE DATASET

The Yelp Dataset consists of several files in JSON format of the data. The main files as per our use-case were the *yelp_academic_dataset_business.json* & *yelp_academic_dataset_review.json*. The two data files were **2.13 GB** & **73.6MB** in size respectively. The data representation in the for acadmic_dataset_business is as follows :

```
1  {
2      "type": "business",
3      "business_id": (encrypted business id),
4      "name": (business name),
5      "neighborhoods": [(hood names)],
6      "full_address": (localized address),
7      "city": (city),
8      "state": (state),
9      "latitude": latitude,
10     "longitude": longitude,
11     "stars": (star rating, rounded to half-stars),
12     "review_count": review count,
13     "categories": [(localized category names)]
14     "open": True / False (corresponds to closed, not business hours),
15     "hours": {
16         (day_of_week): {
17             "open": (HH:MM),
18             "close": (HH:MM)
19         },
20         ...
21     },
22     "attributes": {
23         (attribute_name): (attribute_value),
24     },
25  }
```

The data representation in the for acadmic_dataset_review is as follows :

```
1  {    "type": "review",
2       "business_id": (encrypted business id),
3       "user_id": (encrypted user id),
4       "stars": (star rating, rounded to half-stars),
5       "text": (review text),
6       "date": (date, formatted like "2012-03-14"),
7       "votes": {(vote type): (count)},
8  }
```

The given datasets were first converted and exported into csv formats using a simple python script. The python script is a part of the **PreProcessing1.py** python file.

2 CODING CONTRIBUTION

2.1 DATA TRANSFORMATION PRE-PROCESSING

Since the files were in JSON format, they were first converted to CSV format. After converting to CSV format. After converting to CSV format, the business and review data frames were joined on *business_id* for restaurant establishments. From the combined dataset, we got over 1.6M reviews for restaurants alone. The dataset now consisted of just two columns : *Review Text & Star Rating.*

```
1  {
2          len(resto_review_data)
3          1630712
4  }
```

The length of reviews was tweaked based on our observations from Exploratory Dataset Analysis. Since a very small review might not contribute meaningfully to our models, we decided to limit the minimum characters in a review to 50, while the maximum character limit was fixed to 500 (this was kept low, because at higher ranges, the dataset was still too huge and required more computing power to run than was possible on our MacBooks). This further reduced the number of records to a little over 900k.

```
1  {
2          resto_review = reduceReviewBasedOnLength(resto_review_data=
                resto_review_data, minReviewLen=50, maxReviewLen=500)
3  len(resto_review)
4          910340
5  }
```

2.2 CLEANING

Since we are dealing with real world text reviews, the data available to us will contain plenty of punctation words as well common English stopwords. We followed a standard process for text mining to further cleanse and process the review dataset. In this, we first converted all the reviews to **lowercase**. We then removed **numbers** & **punctuations**. After this, we **tokenized** the text using the NLTK library's tokenizer. Post this, we removed the **stopwords** and we also performed **stemming** of the words. The cleaned dataset was thus used as an input to all our models trained.

2.3 Creating Training and Testing Corpus

We used **70-30** sampling to create a training and testing corpus from the cleaned text reviews. In order to ensure sufficient representation of all the star label values in our training corpus, we split the *original corpus on the basis of the star labels*, thus creating 5 corpora, one for each star label. The 70-30 sampling was then carried out on each of the 5 corpora and the training and test files thus created were then combined to create one big training and testing corpus. The number of reviews in training corpus for each star label is given in Table 2.1.

Table 2.1: Star rating distribution in the training corpora

Dataset	Count
Rating 1	58739
Rating 2	47154
Rating 3	72107
Rating 4	174334
Rating 5	284902

References

[1] ggplot Library is used in this assignment to plot most of the graphs in this assgnment `http://ggplot2.org/`.

[2] ggplot Library is used in this assignment to plot most of the graphs in this assgnment `http://ggplot2.org/`.

[3] ggplot Library is used in this assignment to plot most of the graphs in this assgnment `http://ggplot2.org/`.