MIS510 Web Computing and Mining - Individual Assignment 2

Ming Chen

April 15, 2016

> Note: This document is generated using R markdown. Please contact me if you have trouble viewing it.

Introduction

This document summarized the part of results finished on my behalf for our group project. The structure of the document is organized as following. First, the project scope and problem statement is described again. Next, the key technologies that used to achieve our project goal are introduced. Last, the results for this stage are shown including extracted features and models performance comparison. The current work are available on the Project Github Repo.

Group Project Description

After ther mid-presentation, our project scope is modified a bit to be more centratlized. The goal of this project is to develop an advanced Yelp restaurant recommendation system that is empowered by the utilization of additional publicly accessible features other than traditional Yelp business features. We aimed at including two parts of features to power our recommendation system: Yelp restaurant nearby location features and quantified Yelp review features based on sentiment analysis. All of these two parts of features are processed and aggregated at the business-level. By taking advantage of the ability of content-based recommendation system, we will be able to transfer these features along with traditional business features into Users' preferences over these features (which will be introduced by the other teammate).

The focus of this document will focus on the work of location features extraction and sentiment analysis of Yelp reviews.

Methodology

Crawling Google Map Place Data

The Yelp restaurant nearby location features are crawled using Google Map Place Web Service API (noted as Google API). For a given pair of longitude and latitude, the Google API can return the nearby places information of different categoires (e.g., restaurant, food, theaters, bus stations) within a certian radius.

However, there are some limitations for using the Google API:

1. The maximum requests allowed per day is 150,000.
2. Each request can only query google places for one category.
3. For each request, it only return 20 results in one page. The maximum number of places return for a given request is 3 page, i.e., 60 places. Each page counts once of 150,000 maximum requests.

After pre-liminary data reduction, we have a total of 5667 Yelp restaurants as our targeting businesses in the Phoenix Metropolitan area. In order to crawl all the nearby google places information under the constraints of Google API limitations, I used the strategy described as following.

1. Identified a total of 15 interested categories from the whole list of google place categories.
2. Loop over the Yelp restaurants, for each restaurant, request the nearby google places for each category within 500 meters.
3. If additonal pages exist for a single request, loop over all the pages until reach the maximum 60 results.
4. Adding expotentially backoff between two tries of requests if necessary to avoid error due to high traffic.
5. Append the corresponding Yelp business id to each returned google palce infomation (json format), and store into MongoDB NoSQL database.

6. If the number of requests exceed the maximum limit of 150,000 within 24 hours, terminate the program and start from where it left the next day.

The sample google place data processed is shown as below. (Click here to see sample data).

```
{
    "_id" : ObjectId("56ef2aad84bb9d274bd9441e"),
    "rating" : 4.3,
    "name" : "Ichi Ban Japanese Restaurant & Sushi",
    "reference" : "CoQBeAAAAFahBQ0eAdkbsKpSUb5bsyjdV1JyYjV5JmB3b60y4u7f2hDzfDsmnIxJuONFUKvdB6MXmrnySj8h
    "price_level" : 2,
    "geometry" : {
        "location" : {
            "lat" : 33.4793622,
            "lng" : -112.0732746
        }
    },
    "opening_hours" : {
        "weekday_text" : [ ],
        "open_now" : false
    },
    "place_id" : "ChIJ9VgZuF0SK4cRZp9HH6C7V7U",
    "vicinity" : "2815 North Central Avenue, Phoenix",
    "place_type" : "restaurant",
    "yelp_id" : "x5Mv61CnZLohZWxfCVCPTQ",
    "scope" : "GOOGLE",
    "id" : "ca28c49d74a27b4c7c58197957ea8ce957620d6a",
    "types" : [
        "restaurant",
        "food",
        "point_of_interest",
        "establishment"
    ],
    "icon" : "https://maps.gstatic.com/mapfiles/place_api/icons/restaurant-71.png"
}
```

The aggregation of data is performed by querying MongoDB using Python. For each identified category, I calculated the count and Google place rating information(min, avg, max) for a given Yelp restaurant. Some category such as bus station may not have rating information and thus the corresponding rating data are ignored.

Yelp review sentiment analysis

Another important step for getting more features to power recommendation system is through the sentiment analysis. In order to get more quantitative information from the review text. We used two steps to extract our feature list. The first step is to build a review text classifier to predict a binary label. The second step is to apply the VADER algorithm to get polarity scores (i.e., positive, negative, and neutral).

Review classifier

The goal of building review classifier is to predict what perspective the reviewers are talking about. In particular, are they talking about the food, service, ambiance, or price (i.e., four categories). The content of review can be very arbitrary and such a classifier can help use to better understand the quantitative information behind the review.

To achieve this objective, I have figured it out two alternative solutions. The first one can be noted as word-based classifier and the second is a n-gram based classifier.

- **Word-based classifier**
  The similar idea of a word-based classifier was initially proposed by Gupta et al. (2009). The method starts by extracting the top N (e.g., N = 1000) highest frequency nouns occurred in the most representative or all review texts. Next, these nouns need to be mannually labeled to match one of the category: food, service, ambiance, and price. Once the nound words are correctly labeled, we can construct a training dataset by using the *hypernyms* of nouns as features (i.e., independent variables). The hypernyms for a noun can be obtained by back-trace the path from the noun (node) to the root in the WordNet Hierarchy tree. This can be implemented using Python's nltk package NLTK WordNet Hierarchy. The underlying reason support this method is that noun words (for example, pizza and burger) of the same label (for example, food) share lots of common hypernyms. Last, we can train common machine learning algorithms such as Support Vector Machine (SVM) or Neural Networks (NN) to predict the label given a noun word. The model results then can be further applied to the review text to identify the label of a sentence or paragraph of review.
- **N-gram based classifier**
  The second more straightforward method is to use the most frequent n-grams as features to predict the label of a single review. This method starts by extracting the top N (e.g., N = 1000) n-grams (e.g., unigram, bigram, trigram) from the review text. Next, each review for building the training dataset is mannually given a label indicating what category is the review talking about. Once the training dataset is built, we can train a SVM or NN to predict the label. When predicting for a new review, we can similarily construct a feature vector by matching the review to those n-grams as dummy features and use the feature vector to feed the trained model.

Both solutions are applicable but have their own pros and cons. The first solution take much less time to label words mannually (top N words). However, the relationship between the noun words and outcome label is indirect. Moreover, it is hard to distinguish a review when there are several noun words in this review falling into several different categories (For example, a review both contain the wrods "pizza", "queue", "deal"). In particular, the review in reality are more complex than expected so it would be hard to apply this method. The second solution is much more straightforward and the n-grams and outcoming label are directly related. However, it would take much more resources (i.e., time and labor) to build the training dataset.

In this project, I used the second solution but taking advantage of an existing training dataset mannully labeled by Sajnani et. al. The training dataset can be downloaded from Yelp Dataset for Classifier for .arff format or from the Project Github Repo for converted .csv format.

Support Vector Machine

The Support Vector Machine (SVM) is used to build the review classifier. SVM is a well-known supervised learning model for classification and regression problems. Theoratically, the SVM can be considered as modified version of logistic regression in which the hyperplane (i.e, decision boundary) is chosen to maximize the distance from it to the nearest data point. That is why it is also known as large margin classifiers. The complete cost function for SVM is given by:

$$\min_{\theta} C \sum_{i=1}^{m}[y^{(i)} cost_1(\theta^T x^{(i)})] + (1 - y^{(i)} cost_0(\theta^T x^{(i)})] + \frac{1}{2}\sum_{i=1}^{n}\theta_j^2$$

where $cost_1(\theta^T x)$ and $cost_0(\theta^T x)$ are cost functions for $y = 1$ and $y = 0$ correspondingly which can be treated as a linear approximattion of hyppothesis function from logistic regression. The cost functions were shown in the below. C is the cost coefficient or known as regularization term to avoid overfitting.

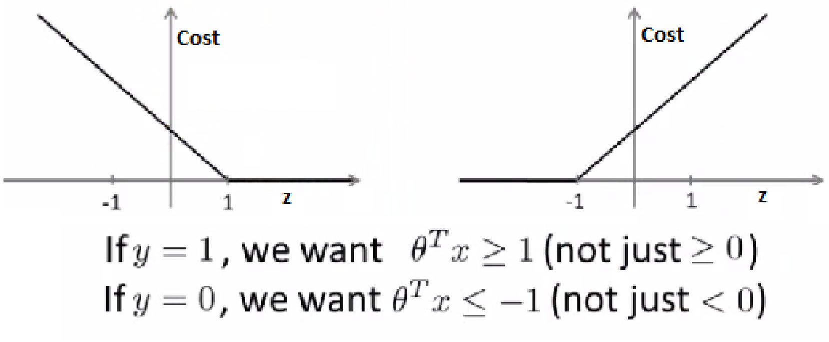


Figure 1. cost functions for SVM

Another important parameter for SMV is the kernel function. The kernel function is defined as a combination of derived feature vector to control the decision boundary. The linear kernel function can only identify a linear decision boundary and other functions such as polynomial, Gaussian (or called RBF) and sigmoid kernels can be used to identify non-linear decision boundary. In this project, I used the Python's scikit-learn package to implement SVM for classification over both linear and non-linear Gaussian kernels.

Vader Score

The VADER (Valence Aware Dictionary for Sentiment Reasoning) is a rule-based model for general sentiment analysis of NLP proposed by Hutto adn Gilbert (2014). The VADER is constructed through a human-centered approach by collecing and examining all lexical features of existing well-established & human validated sentiment lexicons such as LIWC, ANEW, GI. Next, supplementary lexical features commonly used to express sentiment in social media text are integrated to obtain over 9,000 lexical feature candidates. A wisdom-of-crowd (WotC) approach is applied to acquire valid point estimate for the sentiment valence for each lexical feature. A total of 10 qualified raters were hired to giuve a score to all the 9,000 features from -4 (extremely negative)to +4 (extremely positive). The results of these scores are used to consturct rule-based models.

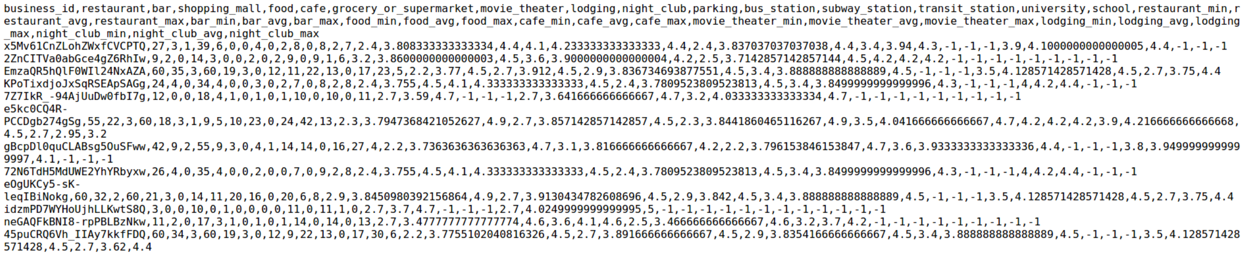
Compared to traditional sentiment lexicons based methods such as Linguistic Inquiry and Word Count (LIWC) and machine learning based methods such as Naive Bayes (NB) classifier, VADER has the following advantages (Hutto and Gilbert, 2014). - workds well on social media style text, yet readily generalize to multiple domains] - require no training data, but is constructed from a generalizable, valence-based, human-curated gold standard sentiment lexicon - fast enough to apply on online streaming data - does not suffer from speed-performance trade-off

These advantages play vital roles when analyzing massive Yelp reviews that are collected from social media platform.

Results

Yelp Restaurant Nearby Location Features

The processed Yelp restaurant nearby location features can be viewed at Github Repo. Click here to see processed location features. The following showed sample rows for the processed location features.



```
business_id,restaurant,bar,shopping_mall,food,cafe,grocery_or_supermarket,movie_theater,lodging,night_club,parking,bus_station,subway_station,transit_station,university,school,restaurant_min,r
estaurant_avg,restaurant_max,bar_min,bar_avg,bar_max,food_min,food_avg,food_max,cafe_min,cafe_avg,cafe_max,movie_theater_min,movie_theater_avg,movie_theater_max,lodging_min,lodging_avg,lodging
_max,night_club_min,night_club_avg,night_club_max
x5Mv61CnZLohZWxfCVCPTQ,27,3,1,39,6,0,0,4,0,2,8,0,8,2,7,2.4,3.808333333333334,4.4,4.1,4.233333333333333,4.4,2.4,3.837037037037038,4.4,3.4,3.94,4.3,-1,-1,-1,3.9,4.1000000000000005,4.4,-1,-1,-1
2ZnCITVa0abGce4gZ6RhIw,9,2,0,14,3,0,0,2,0,2,9,0,9,1,6,3.2,3.8600000000000003,4.5,3.6,3.9000000000000004,4.2,2.5,3.7142857142857144,4.5,4.2,4.2,4.2,-1,-1,-1,-1,-1,-1,-1,-1,-1
EmzaQR5hQlF0WIl24NxAZA,60,35,3,60,19,3,0,12,11,22,13,0,17,23,5,2.2,3.77,4.5,2.7,3.912,4.5,2.9,3.836734693877551,4.5,3.4,3.888888888888889,4.5,-1,-1,-1,3.5,4.128571428571428,4.5,2.7,3.75,4.4
KPoTixdjoJxSqRSEApSAGg,24,4,0,34,4,0,0,3,0,2,7,0,8,2,8,2.4,3.755,4.5,4.1,4.333333333333333,4.5,2.4,3.7809523809523813,4.5,3.4,3.8499999999999996,4.3,-1,-1,-1,4,4.2,4.4,-1,-1,-1
7Z7IkR_-94AjUuDw0fbI7g,12,0,0,18,4,1,0,1,0,1,10,0,10,0,11,2.7,3.59,4.7,-1,-1,-1,2.7,3.641666666666667,4.7,3.2,4.033333333333334,4.7,-1,-1,-1,-1,-1,-1,-1,-1,-1
e5kc0CQ4R-
PCCDgb274gSg,55,22,3,60,18,3,1,9,5,10,23,0,24,42,13,2.3,3.7947368421052627,4.9,2.7,3.857142857142857,4.5,2.3,3.8441860465116267,4.9,3.5,4.041666666666667,4.7,4.2,4.2,4.2,3.9,4.216666666666668,
4.5,2.7,2.95,3.2
gBcpDl0quCLABsg5OuSFww,42,9,2,55,9,3,0,4,1,14,14,0,16,27,4,2.2,3.7363636363636363,4.7,3.1,3.816666666666667,4.2,2.2,3.796153846153847,4.7,3.6,3.9333333333333336,4.4,-1,-1,-1,3.8,3.949999999999
9997,4.1,-1,-1,-1
72N6TdH5MdUWE2YhYRbyxw,26,4,0,35,4,0,0,2,0,0,7,0,9,2,8,2.4,3.755,4.5,4.1,4.333333333333333,4.5,2.4,3.7809523809523813,4.5,3.4,3.8499999999999996,4.3,-1,-1,-1,4,4.2,4.4,-1,-1,-1
eOgUKCy5-sK-
leqIBiNokg,60,32,2,60,21,3,0,14,11,20,16,0,20,6,8,2.9,3.8450980392156864,4.9,2.7,3.9130434782608696,4.5,2.9,3.842,4.5,3.4,3.888888888888889,4.5,-1,-1,-1,3.5,4.128571428571428,4.5,2.7,3.75,4.4
idzmPD7WYHoUjhLLKwtS8Q,3,0,0,10,0,1,0,0,0,0,11,0,11,1,0,2.7,3.7,4.7,-1,-1,-1,2.7,4.0249999999999995,5,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1
neGAQFkBNI8-rpPBLBzNkw,11,2,0,17,3,1,0,1,0,1,14,0,14,0,13,2.7,3.4777777777777774,4.6,3.6,4.1,4.6,2.5,3.466666666666667,4.6,3.2,3.7,4.2,-1,-1,-1,-1,-1,-1,-1,-1,-1
45puCRQ6Vh_IIAy7kkfFDQ,60,34,3,60,19,3,0,12,9,22,13,0,17,30,6,2.2,3.7755102040816326,4.5,2.7,3.891666666666667,4.5,2.9,3.8354166666666667,4.5,3.4,3.888888888888889,4.5,-1,-1,-1,3.5,4.128571428
571428,4.5,2.7,3.62,4.4
```

Figure 2. Processed sample location features

Sentiment Analysis of Yelp Review

In the project, I built four SVM classifiers for the review text to predict four labels: IsFoodGood, IsServiceGood, IsAmbianceGood, and IsPriceGood. The feature vector is composed of 668 most frequent unigrams, bigrams and trigrams. The following parameter grid was identified to search optimal settings.

- kernel: {linear, rbf}
- C: {1, 10}
- gamma for rbf kernel: {0.001, 0.0001}

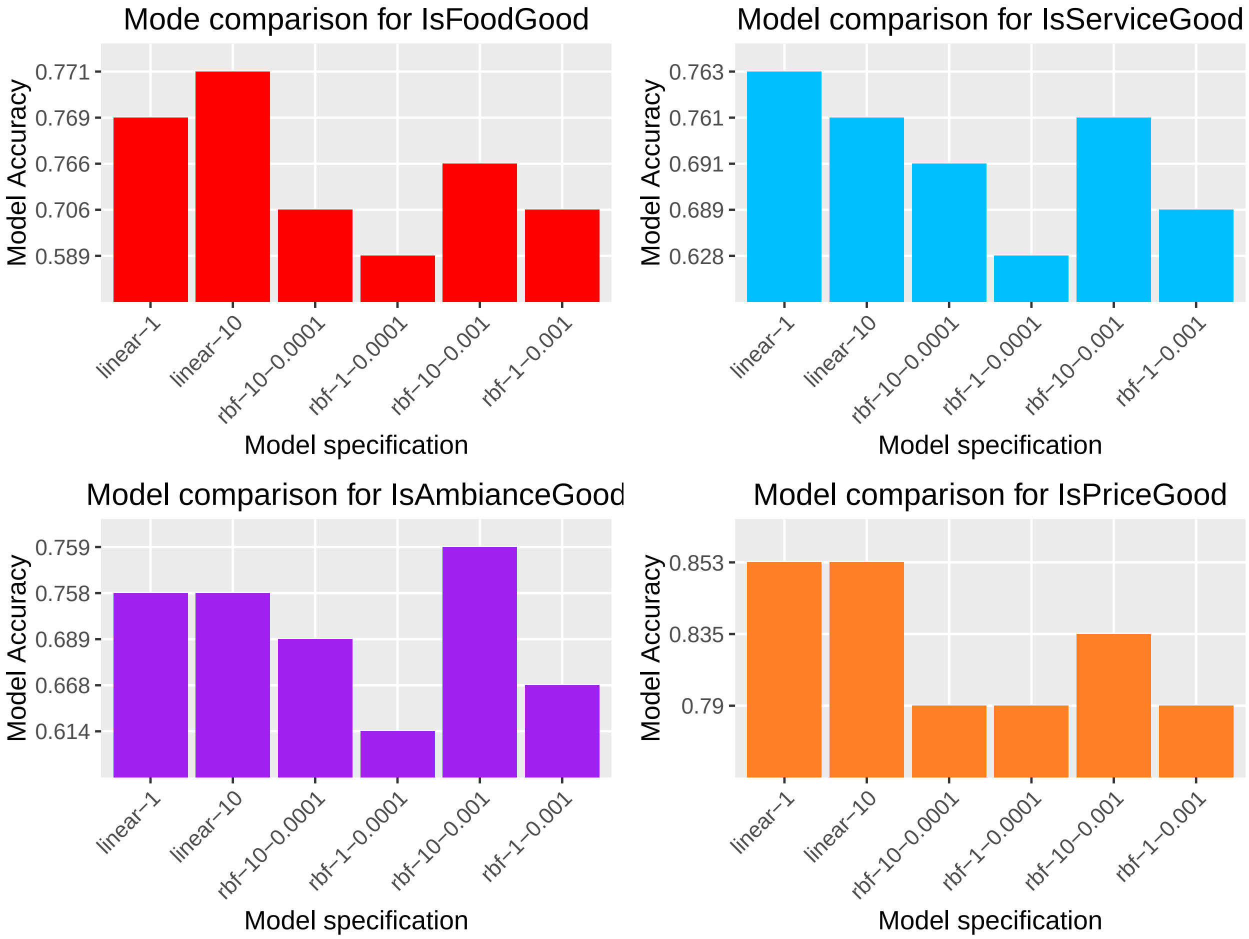
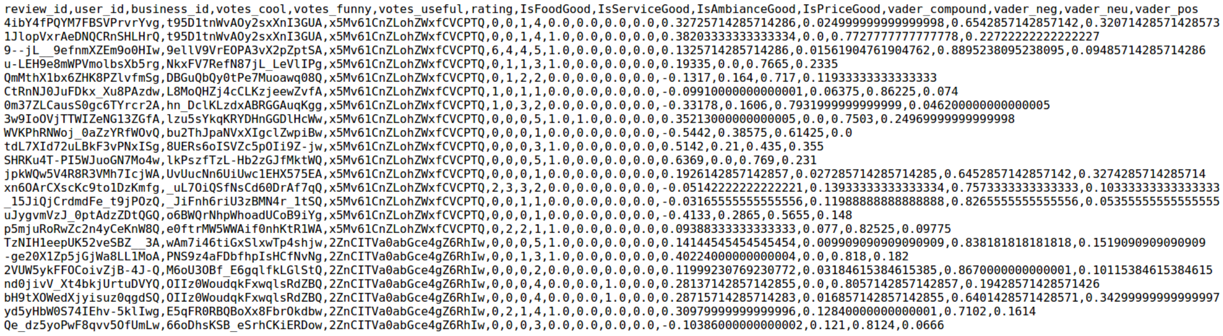


Figure 3. SVM model comparison over different parameter settings

The processed Yelp review sentiment analysis features can be viewed at Github Repo. CLieck here to see processed review features. The following showed the sample rows for the processed review features. In the feature vector, four binary variables (i.e., IsFoodGood, IsServiceGood, IsAmbianceGood, and IsPriceGood) are predicted by the developed SVM review classifier. The afterward four features (i.e., vader_compound, vader_neg, vader_neu, and vader_pos) are calculated from VADER algorithm.



```
review_id,user_id,business_id,votes_cool,votes_funny,votes_useful,rating,IsFoodGood,IsServiceGood,IsAmbianceGood,IsPriceGood,vader_compound,vader_neg,vader_neu,vader_pos
4ibY4fPQYM7FBSVPrvrYvg,t95D1tnWvAOy2sxXnI3GUA,x5Mv61CnZLohZWxfCVCPTQ,0,0,1,4,0.0,0.0,0.0,0.0,0.32725714285714286,0.024999999999999998,0.6542857142857142,0.32071428571428573
1JlopVxrAeDNQCRnSHLHrQ,t95D1tnWvAOy2sxXnI3GUA,x5Mv61CnZLohZWxfCVCPTQ,0,0,1,4,1.0,0.0,0.0,0.0,0.38203333333333334,0.0,0.7727777777777778,0.22722222222222227
9--jL__9efnmXZEm9o0HIw,9ellV9VrEOPA3vX2pZptSA,x5Mv61CnZLohZWxfCVCPTQ,6,4,4,5,1.0,0.0,0.0,0.0,0.1325714285714286,0.01561904761904762,0.8895238095238095,0.09485714285714286
u-LEH9e8mWPVmolbsXb5rg,NkxFV7RefN87jL_LeVlIPg,x5Mv61CnZLohZWxfCVCPTQ,0,1,1,3,1.0,0.0,0.0,0.0,0.19335,0.0,0.7665,0.2335
QmMthX1bx6ZHK8PZlvfmSg,DBGuQbQy0tPe7Muoawq08Q,x5Mv61CnZLohZWxfCVCPTQ,0,1,2,2,0.0,0.0,0.0,0.0,-0.1317,0.164,0.717,0.11933333333333333
CtRnNJ0JuFDkx_Xu8PAzdw,L8MoQHZj4cCLKzjeewZvfA,x5Mv61CnZLohZWxfCVCPTQ,1,0,1,1,0.0,0.0,0.0,0.0,-0.09910000000000001,0.06375,0.86225,0.074
0m37ZLCausS0gc6TYrcr2A,hn_DclKLzdxABRGGAuqKgg,x5Mv61CnZLohZWxfCVCPTQ,1,0,3,2,0.0,0.0,0.0,0.0,-0.33178,0.1606,0.7931999999999999,0.046200000000000005
3w9IoOVjTTWIZeNG13ZGfA,lzu5sYkqKRYDHnGGDlHcWw,x5Mv61CnZLohZWxfCVCPTQ,0,0,0,5,1.0,1.0,0.0,0.0,0.35213000000000005,0.0,0.7503,0.24969999999999998
WVKPhRNWoj_0aZzYRfWOvQ,bu2ThJpaNVxXIgclZwpiBw,x5Mv61CnZLohZWxfCVCPTQ,0,0,0,1,0.0,0.0,0.0,0.0,-0.5442,0.38575,0.61425,0.0
tdL7XId72uLBkF3vPNxISg,8UERs6oISVZc5pOIi9Z-jw,x5Mv61CnZLohZWxfCVCPTQ,0,0,0,3,1.0,0.0,0.0,0.0,0.5142,0.21,0.435,0.355
SHRKu4T-PI5WJuoGN7Mo4w,lkPszfTzL-Hb2zGJfMktWQ,x5Mv61CnZLohZWxfCVCPTQ,0,0,0,5,1.0,0.0,0.0,0.0,0.6369,0.0,0.769,0.231
jpkWQw5V4R8R3VMh7IcjWA,UvUucNn6UiUwc1EHX575EA,x5Mv61CnZLohZWxfCVCPTQ,0,0,0,1,0.0,0.0,0.0,0.0,0.1926142857142857,0.027285714285714285,0.6452857142857142,0.3274285714285714
xn6OArCXscKc9to1DzKmfg,_uL7OiQSfNsCd60DrAf7qQ,x5Mv61CnZLohZWxfCVCPTQ,2,3,3,2,0.0,0.0,0.0,0.0,-0.05142222222222221,0.13933333333333334,0.7573333333333333,0.10333333333333333
_15JiQjCrdmdFe_t9jPOzQ,_JiFnh6riU3zBMN4r_1tSQ,x5Mv61CnZLohZWxfCVCPTQ,0,0,1,1,0.0,0.0,0.0,0.0,-0.03165555555555556,0.1198888888888888,0.8265555555555556,0.05355555555555555
uJygvmVzJ_0ptAdzZDtQGQ,o6BWQrNhpWhoadUCoB9iYg,x5Mv61CnZLohZWxfCVCPTQ,0,0,0,1,0.0,0.0,0.0,0.0,-0.4133,0.2865,0.5655,0.148
p5mjuRoRwZc2n4yCeKnW8Q,e0ftrMW5WWAif0nhKtR1WA,x5Mv61CnZLohZWxfCVCPTQ,0,2,2,1,1.0,0.0,0.0,0.0,0.09388333333333333,0.077,0.82525,0.09775
TzNIH1eepUK52veSBZ__3A,wAm7i46tiGxSlxwTp4shjw,2ZnCITVa0abGce4gZ6RhIw,0,0,0,5,1.0,0.0,0.0,0.0,0.1414545454545454,0.009909090909090909,0.838181818181818,0.1519090909090909
-ge20X1Zp5jGjWa8LL1MoA,PNS9z4aFDbfhpIsHCfNvNg,2ZnCITVa0abGce4gZ6RhIw,0,0,1,3,1.0,0.0,0.0,0.0,0.40224000000000004,0.0,0.818,0.182
2VUW5ykFFOCoivZjB-4J-Q,M6oU3OBf_E6gqlfkLGlStQ,2ZnCITVa0abGce4gZ6RhIw,0,0,0,2,0.0,0.0,0.0,0.0,0.11999230769230772,0.03184615384615385,0.8670000000000001,0.10115384615384615
nd0jivV_Xt4bkjUrtuDVYQ,OIIz0WoudqkFxwqlsRdZBQ,2ZnCITVa0abGce4gZ6RhIw,0,0,0,4,0.0,0.0,1.0,0.0,0.28137142857142855,0.0,0.8057142857142857,0.19428571428571426
bH9tXOWedXjyisuz0qgdSQ,OIIz0WoudqkFxwqlsRdZBQ,2ZnCITVa0abGce4gZ6RhIw,0,0,0,4,0.0,0.0,1.0,0.0,0.28715714285714283,0.016857142857142855,0.6401428571428571,0.34299999999999997
yd5yHbW0S74IEhv-5klIwg,E5qFR0RBQBoXx8FbrOkdbw,2ZnCITVa0abGce4gZ6RhIw,0,2,1,4,1.0,0.0,0.0,0.0,0.30979999999999996,0.12840000000000001,0.7102,0.1614
Qe_dz5yoPwF8qvv5OfUmLw,66oDhsKSB_eSrhCKiERDow,2ZnCITVa0abGce4gZ6RhIw,0,0,0,3,0.0,0.0,0.0,0.0,-0.10386000000000002,0.121,0.8124,0.0666
```

Figure 3. Processed sample review sentiment features

The model selection is performed based on the mean absoluate error (MAE) measure. The MAE measure is calculated based on 5-fold cross-validation. This enire model evaluation and validation process is implemented using Python Scikit-learn package.

Reference

1. Google Map Place Web Service API. https://developers.google.com/places/web-service/. Avaible by April 2016.
2. Hutto, C. J., & Gilbert, E. (2014, May). Vader: A parsimonious rule-based model for sentiment analysis of social media text. In Eighth International AAAI Conference on Weblogs and Social Media.
3. Gupta A., Tenneti, T., Gupta, A. (2009). Sentiment based Summarization of Restaurant Reviews. http://nlp.stanford.edu/courses/cs224n/2009/fp/9.pdf.
4. Sajnani H., Saini, V., Kumar, K., Gabreilova, E., Choudary, P., Lopes, C. Yelp review classifier dataset. http://www.ics.uci.edu/~vpsaini/, Available by April 2016.
5. Scikit-learn SVM. http://scikit-learn.org/stable/modules/svm.html#svm-kernels.