Sentiment Analysis, Data Analytics, Recommendation Algorithm on Yelp

Team: Ashutosh Bhargave (Individual – Software Project)

1. Introduction:

Technology has spread its wings in every field. Be it a retail sector, health care or local businesses. The local businesses are diving into the Technology horizons today. The scenario for the local business is quite different than the big business. The most important difference is in the area of marketing. The difference is because of

A] Differences on Budget Constraints

B] Differences on Staffing

C] Differences on Strategy and Creativity.

Most of the small businesses become giants because of good reviews in the market. And when it comes down to reviews who could beat yelp.com. Today because of yelp.com many of the local businesses have tasted the fruits of success. Yelp.com has 135 million monthly visitors and 71 million reviews. My project deals with three big problems in different areas,

A] **Sentiment Analysis on Yelp Reviews** :

The marketplace for all the consumer products and businesses has moved to the Internet. Because of the shift of users to the online world, the shopping experience has changed a lot. The concept of mouth publicity is decreasing day by day. Prior to yelp or Amazon and still today certain parts of businesses relies on the surveys to know about the user sentiments about the product. Most of the times the efforts carried out for conducting the survey has large amount of monetary issues associated with it.

Yelp has made business reviews and rating open to all the consumers. So, today anybody can go and post on Yelp.com about their experience of using a particular product. Anybody sitting at different place can make use of these reviews and take decision about buying a certain product or visiting a certain place. But how can these reviews be useful to the companies or manufacturers of the product? With the increase in user-generated content, efforts have been made to understand the information in the correct context, and develop methods to determine the intent of the author. Understanding what online users think of its business can help a company or businessmen develop a particular feature and improve upon the ones which they lack at. With the help of this the company can market its product as well as mange its online reputation. This could also help the company to find out where it stands in comparison with their competitors on certain product feature. For example, X business can find out that compared to products of similar range by Z Company, we have better location but we lack in battery life so we should concentrate more on that.

The intent of this project is to study this large problem of knowing the positive and negative attitude of users towards particular business. Sentiment analysis attempts to determine which features of text are indicative of its context and build systems to take advantage of these features. Some work involving sentiment analysis in review or opinion mining has been done relatively recently. With the use of different machine learning systems Pang and Lee (2002) classified a large number of movie reviews. Although Naïve Bayes did not perform the best out of their strategies, it did well compared to the baseline provided by human-generated classifying words.

B] **Recommendation system** :

Recommendation in typical English means to give advice or guidance. In our general life we make many decisions based on recommendations itself. Someone recommends us to try some new outfit brand and we do try out that brand mainly because we know that the person who recommended us with the brand has same taste as ours. Recommendations always make it easy for us to make decisions. It filters things for us to make choice from.

Now-a-days, people have turned to online shopping and review sites. In our traditional system we relied on our friends and relatives for the recommendation but today, not every friend has gone to every place and there is not always a case that we like the similar things which our friends have liked. Also, If there is some new place which none of the friend circle nor you have tried before online reviews and recommendation help us. Passing on recommendations from person to person is useful but it also has a disadvantage. The person who is recommending something may not be fully aware of all the options involved in a particular thing. I am trying to solve the recommendation problem based on the user ratings and the similarity measures. I have written an algorithm that finds top 5 business recommendations for the particular user.

C] **Restaurant Data Analytics/Location Mining :**

Technology is reaching its peak and today we are able to live a more comfortable life just due to technical advent. We can order food online, we can even view the restaurants which are near to our location. Recently I read an article online "The End of Food Poisoning?" while looking about the yelp dataset. After reading a lot about this I came to know that most of the times government officials go for the inspection of particular restaurant. Health departments typically send clipboard-toting inspectors to restaurants to check for things like improper food-storage temperatures and evidence of rodents. The worst violators are shut down, but the rest stay open, and each local government makes its own decisions about whether and how to inform citizens about inspection scores. During this process most of the times the inspectors find many restaurant in good condition and there is actually no need of inspection to be done at that time. This involves many employees and it leads to waste of money and time. I have tried to solve this problem with the help of yelp reviews and the ratings received by particular restaurant. If this algorithm implemented on a large scale in every city, government officials could concentrate on only those restaurants which are actually unhygienic. This would save a lot of money and same time could be utilized for other constructive work.

This solution could also help the yelp users on large. A person who has already visited a restaurant would give his reviews of the restaurant. He rates the restaurant according to its hygienic conditions, quality, ambience and overall cleanliness. All such reviews would be collected and an important decision of whether the restaurant is clean or not can be made. A person who is new to certain place would not need to go in the restaurant and find out for himself if the restaurant is clean or dirty. What that person would do is open yelp.com and search all restaurants near to him. After that, he could also see if the restaurant is dirty. If the reviews of other people give a result that the restaurant is dirty, he would opt not to go in that restaurant. This was one of the best text and location mining problems I have dealt with recently

2. Dataset :

The dataset for released by Yelp contained five different files **business, review, user, check-in, tip.**

Out of these I was interested in business and review file for all the three experiments. The business related file contained all the information about the business, But I was interested in the following factors,

business_id	*name*	*full_address*	*city*	*state*	*latitude*	*longitude*	*stars*

The review file also contained had lots of information but I was more interested in following data about the reviews,

type	*business_id*	*user_id*	*stars*	*text*

3. Methodology :

A] Sentiment Analysis on Yelp Reviews:

There are large amount of reviews on Yelp which facilitated us to do a supervised machine learning. I implemented Naïve Bayes classifier and Support Vector Machine classifier. For implementing this classification techniques I selected following featuresets.

1. Featureset

Bag of Words: In bag of words, a text (such as a sentence or a document) is represented as the bag (multiset) of its words, disregarding grammar and even word order but keeping multiplicity. A bag of words feature vector consists of all of the words in the article as independent features. In these experiments, all of the words were added to a list and only the top 2000 most frequently occurring words were kept. Each of the words in this list was then compared to the words in the review list, and a dictionary was generated that mapped each of the features to either True or False, denoting whether the feature appeared in the review. This is known as a binary feature vector.

Bigrams: The bag of words feature model, ignores some of the relationships between words that affect their meanings in the context of the article. For example, the phrase ‘bad quality’, has a different meaning than ‘bad’ and ‘quality’ appearing independently. These relationships between words can be captured in the by including bigrams. Using the ‘bigram’ function provided by NLTK, all of the bigrams in each of the articles are saved and sorted by frequency. The top 200 of them are included in the feature vectors for each of the articles

2. Classification

Naive Bayes:

Naive Bayes classifiers are linear classifiers that are known for being simple yet very efficient. The probabilistic model of naive Bayes classifiers is based on Bayes’ theorem, and the adjective naïve comes from the assumption that the features in a dataset are mutually independent. In practice, the independence assumption is often violated, but naive Bayes classifiers still tend to perform very well under this unrealistic assumption. Especially for small sample sizes, naive Bayes classifiers can outperform the more powerful alternative. The conditional probability can be decomposed as

$$\textbf{Posterior} = \frac{\textbf{prior * likelihood}}{\textbf{Evidence}}$$

This can be formally written as,

$$\hat{y} = \underset{k \in \{1,\ldots,K\}}{\operatorname{argmax}} \ p(C_k) \prod_{i=1}^{n} p(x_i | C_k).$$

Support Vector Machines (SVM):

SVM is a supervised machine learning algorithm which does some extremely complex data transformations, then figures out how to separate your data based on the labels or outputs you've defined. It uses a technique called the kernel trick to transform your data and then based on these transformations it finds an optimal boundary between the possible outputs. The benefit is that you can capture much more complex relationships between your data points without having to perform difficult transformations on your own. The downside is that the training time is much longer as it's much more computationally intensive

3. Experiments

I formulated emotion detection as the obvious binary classification problem. The dataset required pre-processing as the positive bias in the training data available from the reviews presented a big challenge. There were many reviews with a rating of 4 to 5 stars out of 5 stars which has a high positive bias but I also needed to consider a product with sufficient number of reviews to be able to train our model efficiently.

I used the stars rating on Yelp as our labels and converted it to values of zeros and ones to generate a classification problem of binary values. The ratings above 3 were considered positive and labelled with 1 rating, others (1, 2 and 3 rating) were labelled as 0. After all the rearrangement, the dataset had a balance of positive reviews. So, after the pre-processing steps, the training dataset had approximately 60% ratio of positive to negative reviews.

I performed initial feasibility experiments using two different classifier types, Naive Bayes and Support Vector Machines. These initial experiments were based only on careful word unigram features from review texts. Performance as measured on the development set ranged from Naive Bayes at 67.0% accuracy to Support Vector Machines at 74.0% accuracy. I performed experiments on different combinations of feature sets and came up with different results. I used 5-fold cross validation to check our accuracy over the entire training set.

B] **Recommendation system on Yelp Reviews:**

A collaborative filtering algorithm usually works by searching a large group of people and finding a smaller set with tastes similar to ours. It looks at other things they like and combines them to create a ranked list of suggestions for us to decide upon. The steps were as follows,

> 1] For the purpose of creating a recommendation engine, the prerequisite is to create a dictionary which would include the name of the user and the score (say on a scale of 5 or 10 or any decided metric) given by the user for a particular item or any real world entity.
>
> 2] The very next step in recommendation engine after creation of dictionary is to determine how similar people are in their tastes. For this purpose, similarity score should be calculated. There are various algorithms which could be used to calculate similarity score. Various algorithms which could be used are : a) Euclidean distance score: It takes the items that people have ranked in common and uses them as axes for a chart. b) Pearson correlation score: The correlation coefficient is a measure of how well two sets of data fit on a straight line. This similarity score ranges from 0 to 1 where 0 means there is no similarity between two users and 1 means that the two users are completely similar. I have implemented the algorithm using the pearson’s correlation score.
>
> 3] After calculating the similarity score, the next step is to rank the users. This is nothing but a function that scores everyone against a given user and finds the closest match. This helps us get an ordered list of people with similar tastes to the specified user.

All the above performed steps would help us to create a recommendation list. This could be done by considering choice of a user who is close in similarity scale to the other user. I have scored the items by producing a weighted score that ranks the users. Take the votes of all the other users and multiply how similar they are to a particular user by the score they gave to each item.

The above procedure explains how we should find a user similar to other and also it gives an overview on how to create a recommendation list for a particular user.

C] Restaurant Data Analytics/Location Mining:

All the reviews of previous visitors of the restaurant have been collected on Yelp. My task here was to find all the restaurants which are dirty and unhygienic in different cities and plot them on map. These maps could be useful to quickly find out the locations of different dirty restaurant in a particular city. The algorithm was implemented in the following steps,

1] First from a set of 1.6 million rows or review set only those reviews were filtered which were related to Restaurants. For this words such as “restaurant”, “lunch”, “Brunch”, “Dinner”, “dinner” etc. were used to filter the reviews.

2] After filtering these reviews, I filtered out those reviews which are have reviews related to bad hygiene and dirty places. For this I used NLTK [Natural Language Toolkit] to find words related to dirty and filtered reviews based on that. For example the words were, dirty', 'soiled', 'grimy', 'grubby', 'filthy', 'mucky', 'stained', 'unwashed' etc. While filtering these reviews only those reviews were considered whose rating was below 3 on a scale of 5.

3] After finding a particular review to be bad about the restaurant, I took the business id and traversed in business related file to find the other important information related to particular restuaurant like the location, name of business, the longitude and latitude information.

4] After finding a list of different dirty restaurants in a city I plotted all the locations with the help of longitude and latitude data on the map. I took help of R statistical software. I used ggmap and other packages for implementation of the same.

4. Results :

A] Sentiment Analysis on Yelp Reviews:

Feature sets:

I tried different settings of feature sets on Naïve Bayes Classifier. I first used all the review to collect all the words and used 500 most frequently used words as our feature set by applying Bag of Words. This experiment on Yelp text is considered as baseline algorithm which gave me 59% accuracy. Then, I analysed different features. As shown in Figure 1, the accuracy of features was slightly better to 67%. The accuracy decreased when I used bigrams as our features. I expected improvement but it was disappointing to observe that using Bigrams on the dataset did not performed as expected and the accuracy was decreased considerably. Even combining both of them as feature sets didn’t performed better.

Baseline	59%
Unigrams	67%
Bigrams	60%
Text + Bigrams	65%

Figure 1

2. Cross Validation Results

We then used Support Vector Machines classifier and trained our classifier using Bag of Words unigrams only as our feature set. The mean accuracy observed after 10-fold cross validation using SVM was 76.8%.

The accuracy of individual folds were-

[0.6083871, 0.67354839, 0.73741935, 0.84967742, 0.80580645, 0.76064516, 0.86870968, 0.841909385, 0.80847896, 0.73623377]

SVM Evaluation Report

	Precision	Recall	F1-score	Support
Pos	80%	0.46	0.58	1909
Neg	73%	0.93	0.82	1187
Avg/total	76%	0.75	0.73	3096

3. Most Important Features

Figure 3 shows a list of the most important keywords that were used by our classifier for the classification. We can observe words like love is positive and words like waste, reset, error, not good indicates negative review.

Most Informative Features

Feature = Bag of Words	Feature = Bigrams	Feature =Bag of words + Bigrams
love	(great !)	(big fuss)
waste	(love size)	(pictures)
reset	(disappointed)	(crowded location)
freezing	(excellent mp)	(good choose)
time	(after bad)	(first time)

Figure 3

B] **Recommendation system on Yelp Reviews:**

The recommendation system gives the list of top 5 recommendations for a particular user. But how to check whether a particular recommendation is close to correctness? For this I used one of the goodness of fir tests.

Goodness of Fit : Goodness-of-fit tests compare observed cell counts to expected cell counts. The expected cell counts are computed under the assumption that the null hypothesis is correct; the null hypothesis is rejected if the discrepancy between observed and expected cell counts is sufficiently large. There are various ways to measure the discrepancy between observed and expected counts; perhaps the best known is *Pearson's chi-squared statistic*, which measures squared errors in relation to expected counts.

$$X^2 = \sum_{j=1}^{k} \frac{(o_j - e_j)^2}{e_j}$$

In the above equation, Oj is the vector of observed values and ej is the vector of expected values. I compared it the with the data from the test file and I got a good significance probability value of **0.69**. I tested this for a set of 20 different users.

C] Restaurant Data Analytics/Location Mining:

I have plotted different graphs to find out the different locations of dirty restaurants. The results are mentioned below,

Most dirty restaurant in Phoenix city:

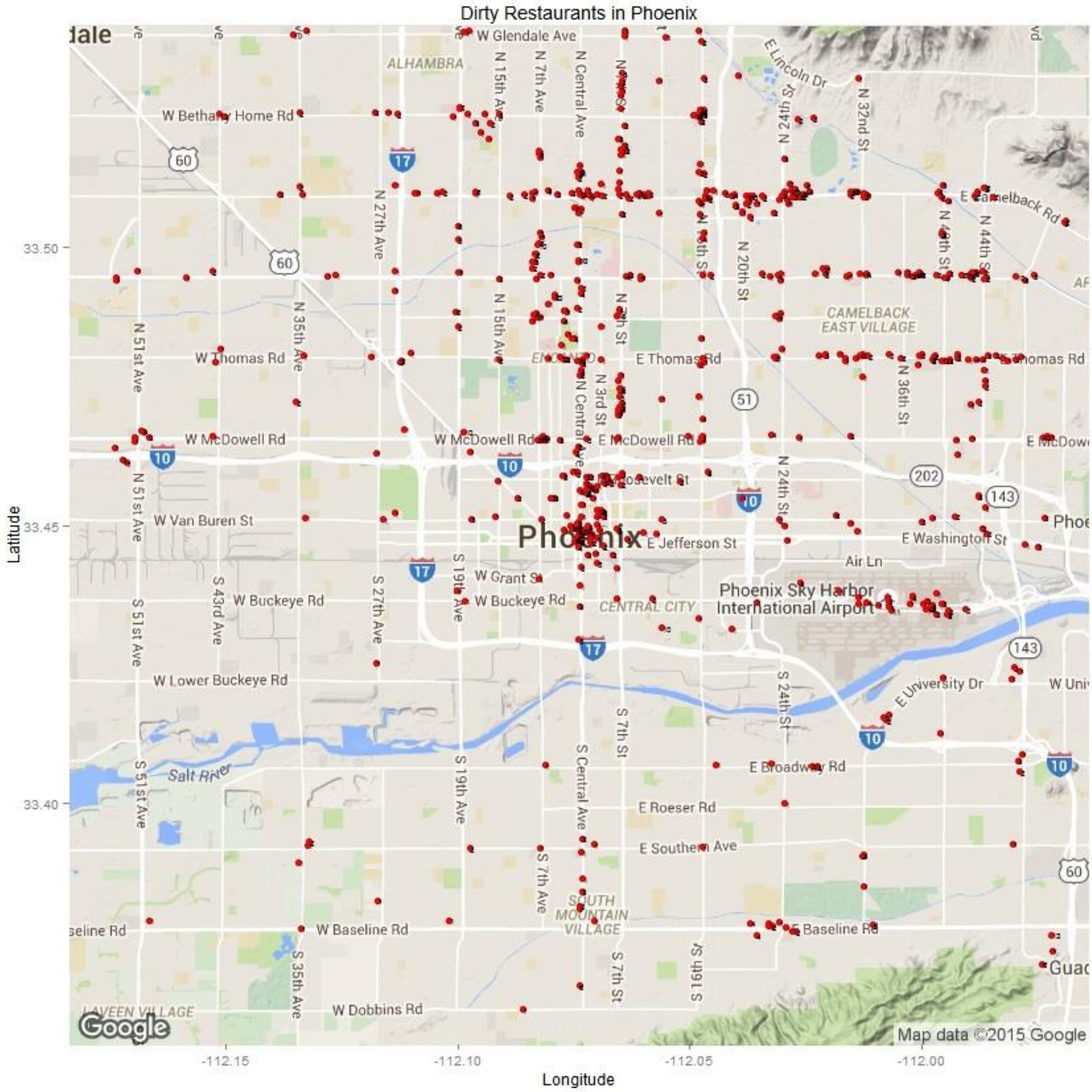

Figure 4 : Dirty Restaurants in Phoenix

This map has been plotted in R with the help of ggplot library.

This map is a good representation of unhygienic places in the city of Phoenix. The red dots in the graph are the locations of the restaurants in Phoenix city.

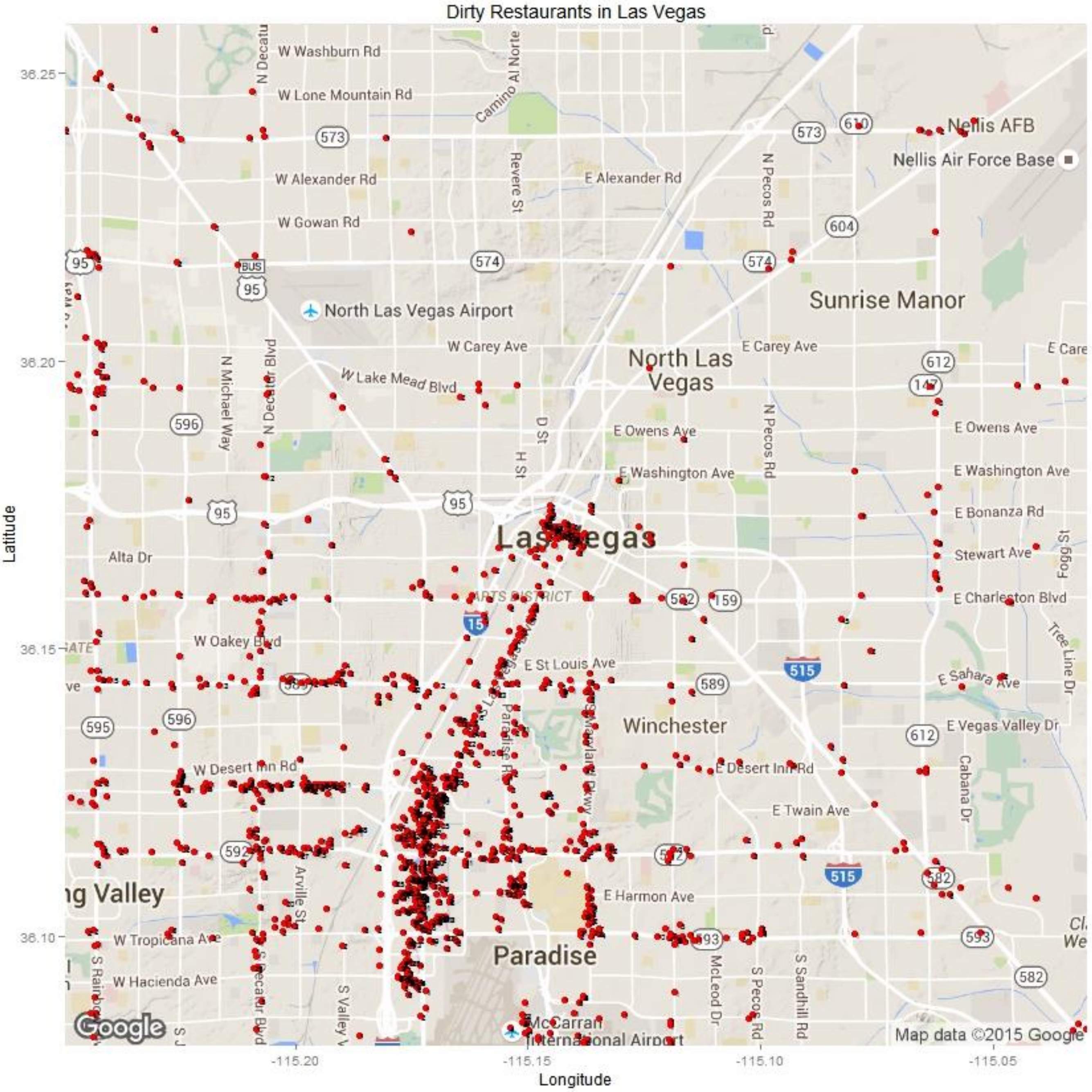

Figure 5: Dirty Restaurant Locations in Las Vegas (R implementation)

This implementation could be very helpful to the health inspectors and also to the users. There is a good scope of improvement for this project. I think we can plot the visualizations in real time on a graph. Also, the above implementations could be done on distributed processing frameworks like Hadoop. For real time analytics Apache spark could also be used.

5. Reproducibility

A] **Sentiment Analysis on Yelp Reviews:**

For both the classifiers I have used Canopy as Python Deployment environment. I used Natural Language toolkit (NLTK) as a text processing package which is a very robust and widely used in Natural Language Processing. Different packages and statistical tools used were Scikit Learn, CSV reader, svm, random etc.

The Naïve Bayes and SVM code can be imported in canopy and the dataset should be kept on the correct path as the one mentioned in code.

In future work, this work could be extended to make our prediction system more accurate and achieve better accuracy. I can use the concept of Sentiwordnet and POS tags to improvise our model. I want to use this model to better predict results on more ecommerce websites and then extend it to a wider category of commodities from electronics to other range of products as well. The dataset is available on github's same folder. The CSV file named "yelp_training_set_review.csv". The file has been cleaned and it has five columns namely review text, business id, votes, ratings and the rating converted into 0's and 1's and approximately forty thousand reviews out of which twenty thousand were used for training.

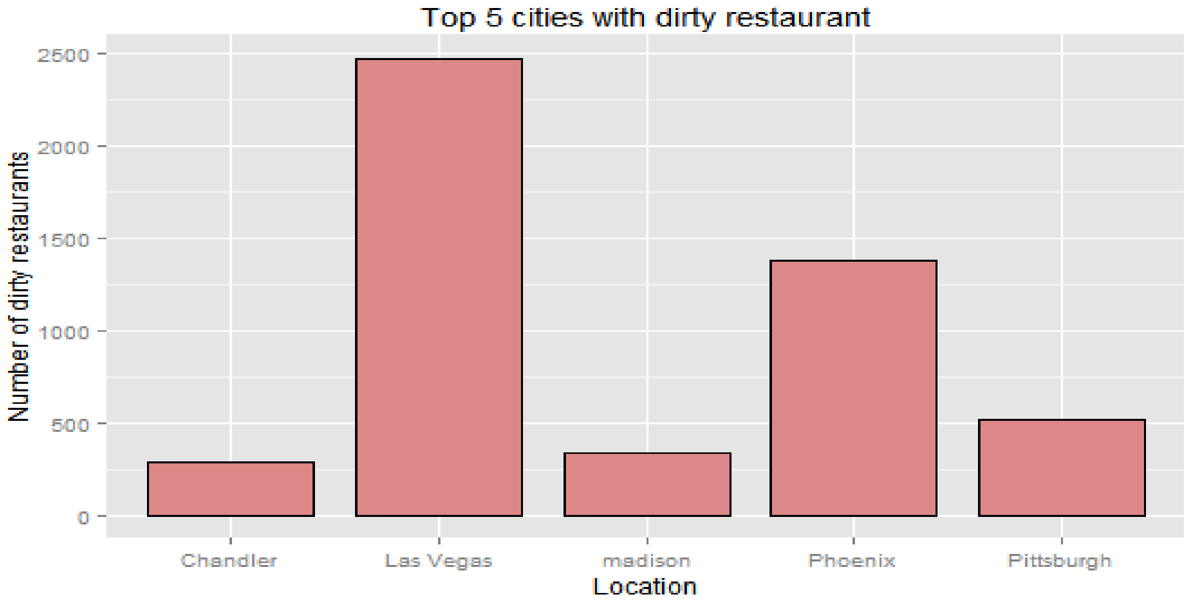

Figure 6: Top 5 cities with maximum dirty restaurants

B] **Recommendation system on Yelp Reviews:**

For both the classifiers I have used Canopy as Python Deployment environment. I used Numpy for different calculations. NumPy is an extension to the Python programming language, adding support for large, multi-dimensional arrays and matrices, along with a large library of high-level mathematical functions to operate on these arrays. I made use of PearsonR package in Stats library. PearsonR Calculates a Pearson correlation coefficient and the p-value for testing non-correlation. The Pearson correlation coefficient measures the linear relationship between two datasets. Instead of CSV files two JSON files are used as dataset in this part. Files named "yelp_training_set_business.json" and "yelp_training_set_review.json" have been made available on githubs on same location. The file "yelp_training_set_review" has been compressed and pushed into an zip folder because of github's file size limit. The business file has columns like business_id, location, city, longitude latitude etc.

C] Restaurant Data Analytics/Location Mining:

For problem I have used Canopy as Python Deployment environment. I used Natural Language toolkit (NLTK) as a text processing package which is a very robust and widely used in Natural Language Processing. I have checked the stopwords using Stopwords package in NLTK. The corpus of stopwords, that is, high-frequency words like *the, to and also* that we sometimes want to filter out of a document before further processing. Stopwords usually have little lexical content, and their presence in a text fails to distinguish it from other texts.

This part of project requires Python 2.7 version. For plotting the maps, R language has been used. The ggplot2, maps libraries are used. For this part of project I have dealt with 1.6 million reviews. The data is available to be downloaded on yelp data set challenge web page. The data is in JSON format and I have changed the data into CSV format using JSON to CSV converter code. This part of project in future could be done on Hadoop because to deal with 1.6 million rows in Windows operating systems take lot of time for implementation.

Code is available on github at : https://github.com/ashubhargave/sentiment_recomendation_location_mining

6. Conclusion

The results of these experiments were successful. The classifiers managed to accurately tag a great amount of user-generated data much further past the random baseline of 50%.Naive Bayes performed good on smaller dataset. But, as we increased the size of dataset, SVM outperformed Naïve Bayes in our classification. Our best classifier after parameter tuning performed at the mean accuracy of 76 % in 10-fold cross validation. The implementation of dirty restaurants part was really interesting and I could learn a lot about plotting longitudinal data on graphs/Maps in language like R. Users and government organizations could use this data for lots of different concrete purposes.

7. References:

John Blitzer and Hal Daumé, Domain Adaptation, Paper available at: http://john.blitzer.com/talks/icmltutorial_2010.pdf

Hal Daume III, Frustratingly Easy Domain Adaptation, Paper available at: http://www.umiacs.umd.edu/~hal/docs/daume07easyadapt.pdf

John D. Burger and John Henderson and George Kim and Guido Zarrella. Discriminating Gender on Twitter ShlomoArgamon, Moshe Koppel, James W.Pennebaker, andJonathan Schler. 2007. Mining the blogosphere: Age,gender, and the varieties of self-expression.

John D. Burger and John C. Henderson. 2006. An explorationof observable features related to blogger age. In ComputationalApproaches to Analyzing Weblogs

Analyzing Weblogs: Papers from the2006 AAAI Spring Symposium. AAAI Press.Chris Callison-Burch and Mark Dredze. 2010. Creating speechand language data with Amazon's Mechanical Turk.

In Proceedingsof the NAACL HLT 2010 Workshop on CreatingSpeech and Language Data with Amazon's Mechanical Turk,CSLDAMT'10. Association for Computational Linguistics.

Minqing Hu and Bing Liu. 2004. Mining and summarizing customer rviews. In Proceedings of the Tenth ACM SIGKDD International Conference on Knowl-edge Discovery and Data Mining, KDD '04, pages 168–177, New York, NY, USA. ACM

Bo Pang, Lillian Lee, and Shivakumar Vaithyanathan. 2002. Thumbs up?: Sentiment classification using machine learning techniques. In Proceedings of the ACL-02 Conference on Empirical Methods in Natural La nguage Processing - Volume 10, EMNLP '02, pages 79–86, Stroudsburg, PA, USA. Association for Computational Linguistics.

Yelp Dataset challenge, http://www.yelp.com/dataset_challenge