Help A Yelp

[IDS 561: Big data project]

Team: Chethan Simha -

Gaurav Purohit -

Venkatesh Iyer -

**Yelp - The Business:**

Yelp is an American multinational corporation which has built its business by publishing user reviews and ratings for local businesses, allowing user to post pictures, connect to Facebook friends etc. Yelp became a public business in 2012 and since then it has seen rise in profit and increase in the number of users who have used Yelp. Yelp recorded 135 million monthly visitors and 71 million reviews as of 2014. Recently it has come under severe criticism on the legitimacy of its reviews and processes which involve businesses paying users to post fake reviews.

**Motivation:**

Over years, the size of the users that use Yelp to express their opinions has grown exponentially. This being a good thing for Yelp, one major problem that Yelp deals with is the opinion spammers who send out false reviews and ratings. Publishing such opinion spams, negatively impacts the credibility of Yelp, generates unfair competition and gives out false information to the users.

One of the articles from by a user at Huffington post quoted -

"*I would be highly suspect in relying upon Yelp reviews in the future. I wish this were not true because it just makes it that much more difficult and confusing for customers to locate "real evidence" that is so essential to properly vet products and services."*

There are several other articles that we read conveying the message that people were dissatisfied and did not agree to the recommended post by Yelp for specific users. This problem that Yelp is dealing with became the motivation of our project. We wished to develop sophisticated algorithms that would help businesses like Yelp to identify potential opinion spammers.

**Project Objective:**

1. Filter reviews and ratings of all the users for different businesses to identify dissimilar reviews and ratings.

2. Analyzed dissimilar reviews to identify probable opinion spams (fake reviews and ratings)

**Data Source:**

**Link -** https://www.yelp.com/dataset\_challenge/dataset

**Datasets** -

1. ***Business dataset*** -

The business dataset includes basic information about the businesses mentioned. The ‘***business\_id***’ is a unique identifier for the business if it has several branches at different locations where location is specified by latitude, longitude, state and city geographical dimensions.

*{*

*'type': 'business',*

*'business\_id': (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),*

*'close': (HH:MM)*

*},*

*...*

*},*

*'attributes': {*

*(attribute\_name): (attribute\_value),*

*...*

*},*

*}*

2. ***Review dataset*** -

Review dataset contains review texts given by users for a particular business along with rating/stars along with certain information on votes that the Yelp users have casted on the 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)},*

*}*

3. ***User dataset*** -

This dataset contains characteristics about users across all the Yelp reviews, inclusive of even those business and reviews not in the other datasets.

*{*

*'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),*

*}*

Each file is composed of a single object type, one json-object per-line.

**Technical Challenges:**

The two major challenges that we faced while working on this project were -

1. Pre-processing and manipulating the unstructured raw datasets

2. Choosing an appropriate algorithm to identify potential opinion spammers

**Pre-Processing and manipulating the data**

As mentioned in the earlier part of the report, the data was in the form of a json file. There were almost over 30,000 distinct businesses and a million reviews in the dataset. The first major challenge we faced was to contemplate on what format to convert the json file to implement the algorithm. Excel or csv file was the first option we thought of and we converted the json file to csv. But when we started implementing the algorithm we released it would make more sense if we could get all the ratings and reviews pertaining to a business as a separate file. We then took a step back and converted the json files to text files. One text file was created for each business which consisted the ratings that all users had given for the business. In case of reviews, folders for each of the business were created and these folders held text files, one for each of the reviews given by the user. The python code for converting the json file to text file is mentioned in the code documents (separately submitted as a part of this project) .

The process of transforming the json to text files was the most time consuming part of the project. It was mainly because the data was very dirty. The business IDs on which we based the conversion of json to csv, had a lot of special characters in them. Many of these special characters could not be accepted as part of the text file names and the program used to execute partially before throwing an error. It took us a lot of time to remove all possible combinations of special characters and replace them with suitable substitutes. But manipulating the business IDs made it difficult to use functions like VLOOKUP between the result set csv and the business dataset csv file (json to csv conversion done earlier) to get some important data in the result set file. To resolve this issue we got rid of all the special characters present in the business ID, trimmed the resultant business ID and combined it with the user ID to get a unique key which could be used to perform VLOOKUP.

2. Choice of Algorithm

The choice of the algorithm to be used to detect probable fake reviews and ratings was single handedly the most important part of the project. We invested a large amount of time reading and understanding different algorithms which could be used to detect probable fake reviews.

**Approach:**

The preprocessed data obtained from the previous transformation from json format was distributed into two folders: **reviews dump** and **ratings dump** as shown the figure below:

**Figure 1.1** - ***Ratings Dump*** folder with an example of a text file’s content in it.

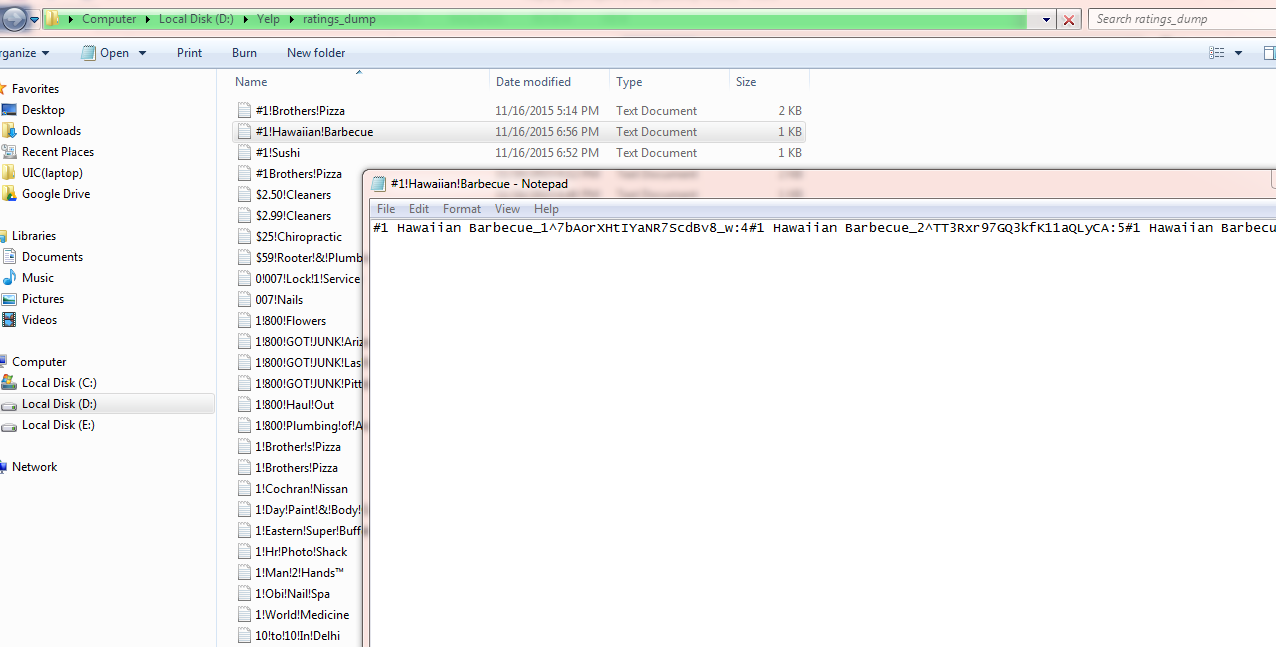


Figure 1.1 provides a snapshot of the list of text files created and shows the content of the text file **‘*#1 HawaiiaanBarbeque***’ business. Every instance of business name is followed by an underscore(\_) symbol and a number indicating the nth review for that business made by the user distinguished by the user id which is followed by ‘^’ symbol in the text.

***HawaiianBarbeque\_1^7bAorXHtIYaNR7ScdBv8\_w:4*** means this is the first review for the specified business made by the user ‘***7bAorXHtIYaNR7ScdBv8\_w***’ and this user had rated this business ***4/5 stars***.

Similarly, the rest of the users provide their ratings for the specified business. Therefore, every document in the ratings dump corpus includes a collection of ratings given by unique users for the business in question. This business is clearly identified by the text document name which is the same as the business name.

**Figure 1.2** - ***Reviews Dump*** folder.

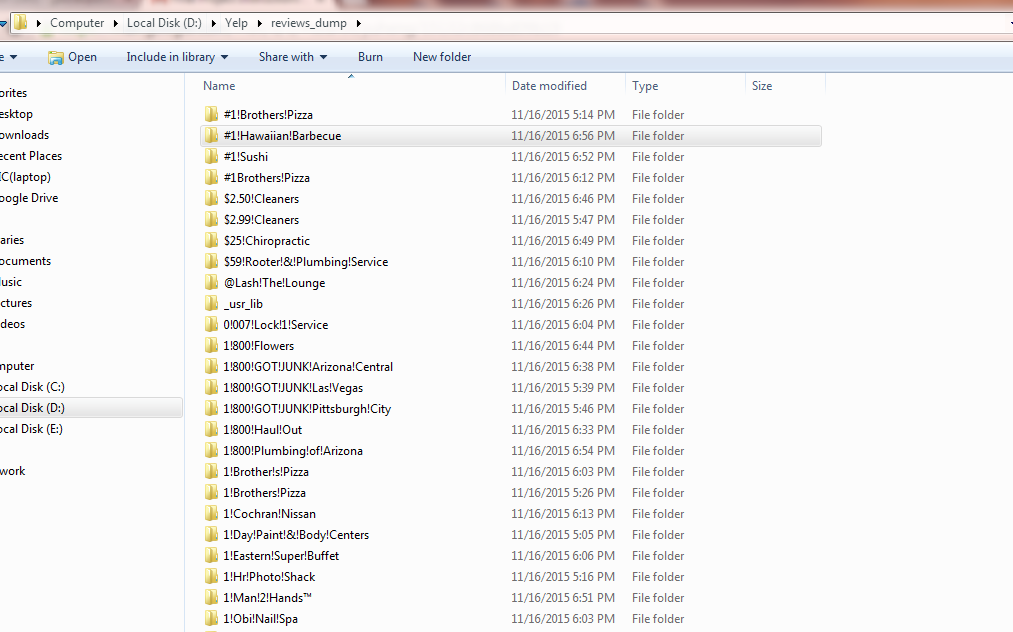


Figure 1.2 provides a snapshot of how the folders are created for each unique business and each business folder hold text documents which include the review texts for that respective business, given by unique users as shown in the figure 1.3 below.

**Figure 1.3** - Inside a **business folder** in reviews dump, displaying review contents.

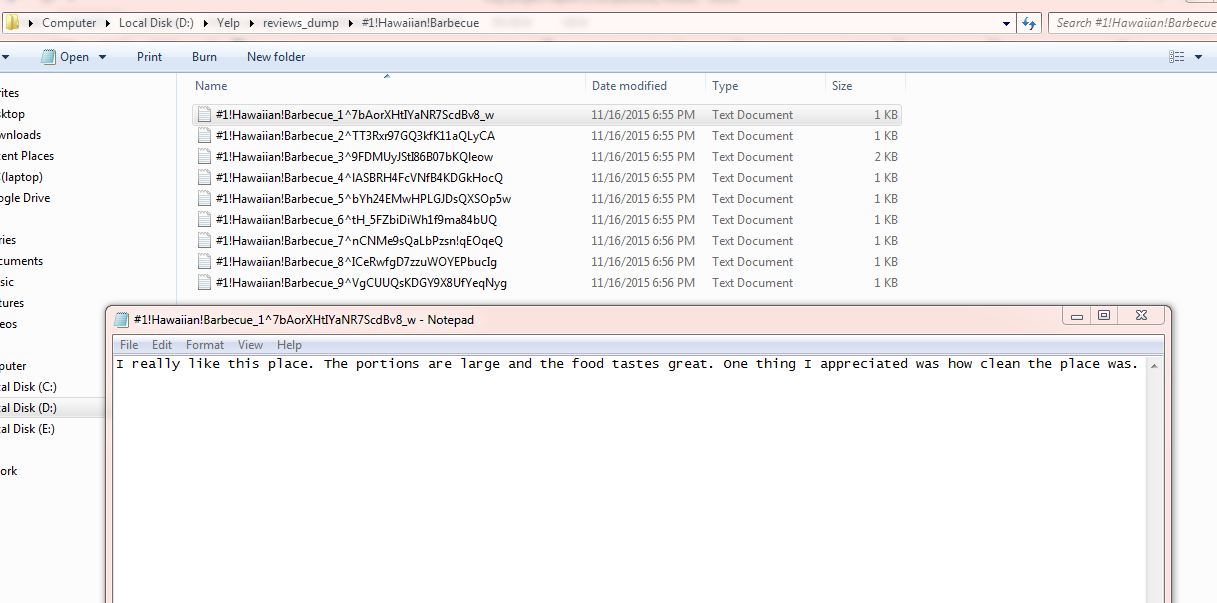


Figure 1.3 shows a snapshot of text document for business **‘*#1 HawaiiaanBarbeque***’. This document includes a review written by the user ‘***7bAorXHtIYaNR7ScdBv8\_w***’.

* Identification of dissimilar reviews was essential to the goal of this project. Therefore, the approach was to identify the ratings/stars given by users to a particular business (since multiple users will rate a particular business) and compare their ratings with respect to each other and find out their individual rating scores to identify users with low rating scores. Moreover, the review texts given by unique users to a particular business were also compared and individual user’s cosine similarity score were found out to identify users with low similarity scores.
* Individuals with low rating scores and low similarity scores were found out to have dissimilar reviews compared to the rest of the users’ reviews and ratings.

Finding out ***Rating Similarity*** scores:

Yelp maintains a consistent rating format i.e. irrespective of the type or category of the business and the users’ preferences, the businesses are always rated out of 5 stars i.e. on the scale of 1 to 5. This helped in avoiding the additional efforts needed to adjust the ratings before those could be taken into consideration.

Considering N users, where A1, A2, A3,...., An are the users who rated for a particular business then the average rating similarity score between the nth user in question and all other users is computed by the following formula:

(Rsim values lie between 0 and 1)

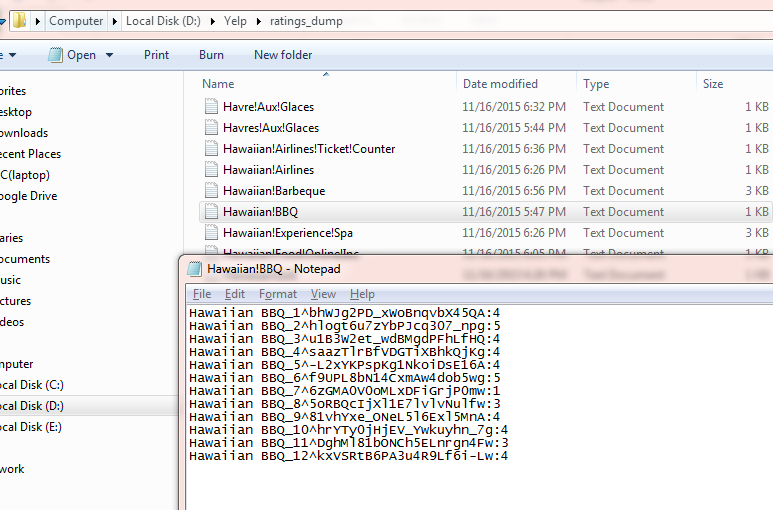
**Figure 1.4** - Snapshot of ratings for a specific business

Figure 1.4 provides a snapshot of the ratings of a business ‘Hawaiian BBQ’ given by 12 unique users where each rating is in a consistent format, rated out of 5 stars. The rating similarity scores for these 12 users is computed from the above mentioned formula and the result is as follows:

**Figure 1.5** - Computed Rating similarity scores for all the reviews of a particular business

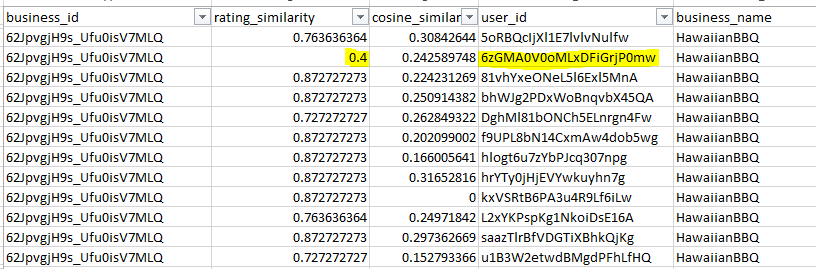


Figure 1.5 provides a snapshot of the output csv file which consists of the rating similarity scores of the reviews made by unique users on the business ‘***HawaiianBBQ’***. The user ‘***6zGMA0V0oMLxDFiGrjP0mw’*** has a dissimilar rating compared to other users, as seen from Fig. 1.4 and therefore, gets a lower rating similarity score in the Figure 1.5

Finding out ***Cosine Similarity*** scores:

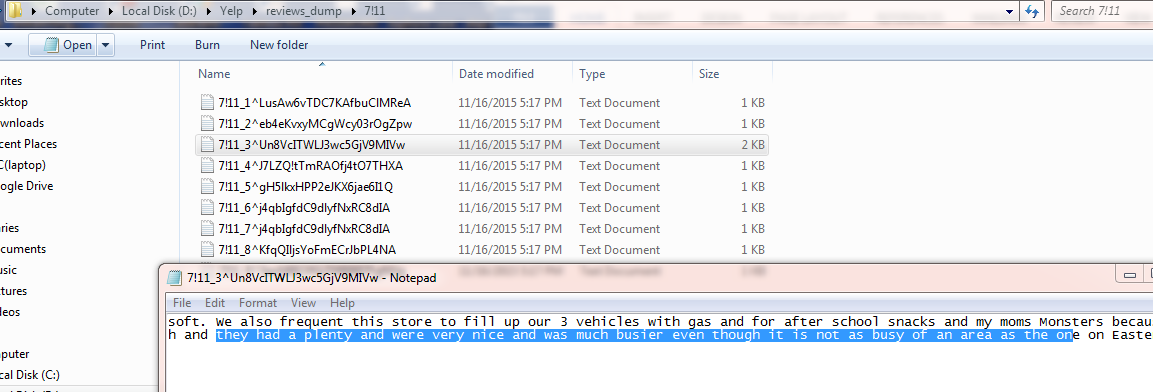
Reviews made by users on a particular business are important as these reviews depict the tone i.e. sentimentality of the review and helps in finding out whether a particular review is dissimilar compared to other reviews. Comparison between two documents involves finding the distance between two texts for which cosine based similarity has been used and it works as follows:

Consider N users and let c1,c2,c3...cn be the cosine score of user n with all other users. Then the Csimn indicates the average cosine similarity score between user n and other users for the same specified business.

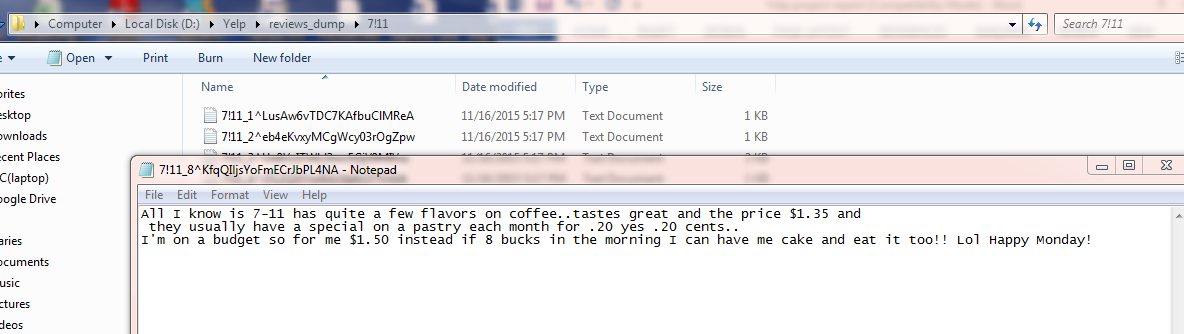
*Csimn*

(Csim lies between 0 and 1)

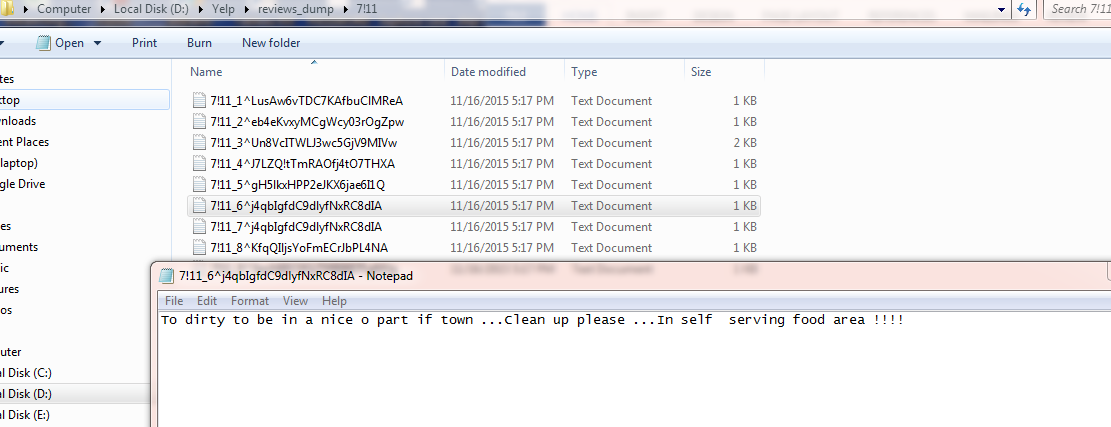
**Figure 1.6 -** Snapshot of positive review text from the reviews dump for a particular business folder



**Figure 1.7 -** Snapshot of positive review text from the reviews dump for a particular business folder



**Figure 1.8 -** Snapshot of negative review text from the reviews dump for a particular business folder



Figures 1.6 to 1.8 on the previous page provide snapshot of the review texts for the business ‘***7-11***’ given by different users. The first two reviews express positive sentiments while the last review express negative sentiments. The cosine similarity scores for these reviews can be seen in the following snapshot:

**Figure 1.9 -** Snapshot provides cosine scores for reviews for a particular business

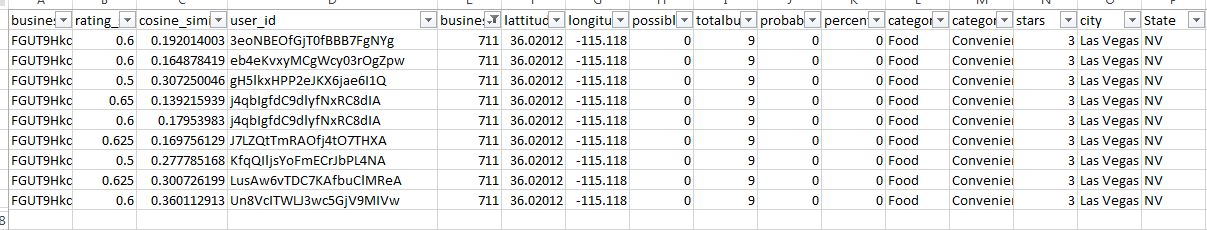


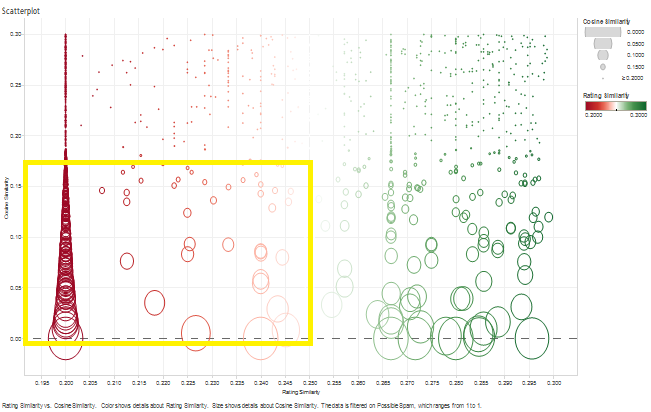
Figure 1.9 provides a snapshot of the cosine scores and it can be found that user ‘***j4qbIgfdC9dlyfNxRC8dIA’*** who provided a negative review has a low cosine score indicating dissimilarity and inconsistency of opinion compared to the rest of the reviews.

**Selecting dissimilar reviews**:

The rating and cosine similarity scores obtained have been further used to filter out the dissimilar reviews. From trial and error method along with review observations for different permutations of scores, cutoff scores of 0.3 were decided for ratings and cosine scores i.e. those reviews having rating scores and cosine scores less than 0.3 were considered and observed to be significantly dissimilar compared to the rest of the ratings and reviews.

**Observations -**

**Figure 2.1** - Scatter plot of ***Rating similarity*** v/s ***Cosine Similarity***for probable fake reviews



**X- axis -** Rating similarity score

**Y - axis** - Review Similarity score

The above scatter plot is the graphical representation of the rating similarity and review similarity scores of all the probable fake reviews given by users across all businesses. As mentioned earlier the threshold for suspecting a review as fake is 0.3 for rating similarity and 0,3 for review similarity. The color of the individual data points changes from red to green as the rating similarity scores increase. Similarly the size of the individual points decrease as the review similarity scores approaches 0.3. As described n the algorithm, the users who are the most probable candidates for writing fake reviews and ratings are the ones who have low value of rating similarity as well as review similarity. These group of people are indicated in the yellow box above.

**Figure 2.2 :** Correlation between ***Percent of fake reviews*** and ***Count of distinct businesses***

**

**Fake percent** - **12.5**

**No of distinct businesses with that fake percent** - **48**

**Fake percent** - **25**

**No of distinct businesses with that fake percent** - **128**

**Fake percent** - **33.33**

**No of distinct businesses with that fake percent** - **205**

**X-axis** - Percent fake reviews

**Y -axis** - Count of distinct businesses

As seen by the above scatterplot there is a strong positive correlation between the percentage of fake reviews and the total numbr of distinct businesses with that particular fake percentage. The statistics supposting the correlation are mentioned besides Figure 2.2 above.

**Figure 2.3: Distribution of probable fake reviews across cities**

****

The above plot is a stacked bar plot which indicates the sum of probable fake reviews for businesses in each of the cities. Each color in the bar lot represents a city with probable fake reviews. It can be seen that Arizona and Nevada are the two states which have the highest value of probable fake reviews with the maximum number of reviews in Las Vegas, Nevada (524 reviews) followed by Phoenix and Scottsdale in Arizona (269 and 128 reviews respectively).

**Recommendations/Suggestions**:

[Note: Many of the suggestions have been given by Professor ***Yuheng Hu*** during the project presentation session]

* Rating similarity scores for individual users could have had been substituted by the standard deviations of those respective scores after calculation of the mean score from all the reviews for the business in question. This approach could have been quite simpler and less computationally expensive. However, the current approach might provide more accurate results as the formula tends to normalize the results.
* Cosine based similarity compares the two texts on the basis of their presence and if so, then the how frequently the term appears in that particular text. However, an apparent shortcoming of this method is its failure to detect words with similar meaning i.e. synonyms. Future scope of this project, therefore, would be to pass these text documents through a synonym replacer algorithm or a similar algorithm which substitutes similar words for better and accurate results.
* Cosine similarity seems to get affected by the length of the texts i.e. if one of the texts in comparison is smaller in length compared to the other then the resultant accuracy might be affected. Therefore, future scope would see term frequency being replaced by normalized term frequency which divides TF by total number of terms in the text i.e. it normalizes the values in the vector before computation.