Predicting the next class of elite Yelp reviewers and a sentiment analysis on reviews

Renee Sikes
Parker Jamison
Kim Tricker

December 12, 2017

Abstract

Yelp is arguably the biggest and fastest growing hub for crowd-sourced business reviews. While the importance of each reviewer individually varies on a vast scale, there is no arguing that the community as a whole is responsible for the social change in consumerism that we are seeing today. Our project recognizes the significance of this community of reviewers, and pays particular attention to those users that Yelp classifies as elite. Our goal is to predict who will be the next class of elite users, for Yelp to extend their elite designation to those worthy of the title and for the community to benefit from an increased quantity of high quality elite reviews. To execute this, we created a clustering model which analyzes all of the users (both elite and otherwise) with attributes such as the age of their Yelp account, the number of cool, funny, and useful votes they get, and their complement scores. After clustering these users, we found the cluster that contained the most elite users, calculated each user's distance from the cluster center, and found those users who did not have the elite flag but shared characteristics of other elite users. We produced a list of 20 users who we predict would be the next generation of elite. Our secondary goal is to dissect the content of reviews using a Term Frequency - Inverse Document Frequency vectorizer to predict the star rating of a given review based on the negative and positive words in that review. We executed a sentiment analysis on all reviews and extracted a group of positive words and a group of negative words. We then looked for these words in each review and counted the number of positive and negative words that appeared in our baseline, plotted these against the star rating of the review, and found the ratio of positive and negative words correlates well with the average star rating for the business. However, there are some pockets of reviews for which the counts of positive and negative words do not align with the star rating, which could warrant further examination to ensure the review is not fraudulent.

Introduction

The impact of online reviews cannot be understated in a society increasingly influenced by social media. Yelp, one of the most distinguished hosts of crowd-sourced business reviews, flourishes in an increasingly media-centric society, where consumers have at their disposal more information than they need to make nearly every purchasing decision they have. With this near-perfect information, the consumer becomes infinitely more powerful. The ease of access to the Internet and crowd-sourced review platforms like Yelp has contributed to their rise. No longer do businesses have the benefit of

masking their flaws before the unassuming customer; instead, they are held to a higher standard to deliver a quality product or else be at the mercy of their customers' brutally honest reviews.

Among Yelp reviewers a class divide exists: those with the elite designation and those without. The elite reviewers are those who write detail-oriented reviews, provide quality tips, boast a full personal profile, and are active in the Yelp community[1]. These are trusted reviewers whose special badges designate to consumers and reviewers alike that their reviews are more reliable. Naturally, these characters of elite reviewers are what many strive to achieve. Not only does Yelp benefit from having more elite reviewers, because their community becomes more trustworthy, but it also does the community good to have more elite reviewers, because the community benefits from larger quantities of reliable reviews.

The purpose of our project is to identify who among the community of reviewers are next in line to become elite. The current process for selecting elite users is subjective: an individual nominates a candidate for elite status, and that candidate is vetted through a panel at the Yelp headquarters in San Francisco, California[2]. To expedite this outdated and largely manual process, our goal is to improve the selection of the elite users with a clustering algorithm that can predict who among the community of reviewers might be worthy of the elite status based on the current pool of elite users. This will serve to benefit Yelp as a company, who will no longer need to dedicate resources to in-depth audits of users who wish to become elite.

The secondary purpose of our project is to perform a sentiment analysis on the words used in a review. Our goal is to be able to identify the negative and positive words used in a review to determine if these fall in line with the associated star rating. We believe that this could be a step towards identifying fraudulent reviews, as it would be able to flag reviews whose text does not align with its star rating, and can then be further investigated by a Yelp for indications of fraud. We also believe it would be advantageous of Yelp to look into reviews of this nature because they could be skewing the overall star rating of a business in a direction contradictory to the true intention of the review.

[1] What is Yelp's Elite Squad? Yelp. https://www.yelp-support.com/article/What-is-Yelps-Elite-Squad?l=en_US. Accessed 7 December 2017.

[2] Patterson, Brian. "What businesses need to know about the elite Yelp program." 10 February 2017. https://marketingland.com/businesses-need-know-yelp-elite-program-202793. Accessed 7 December 2017.

Dataset

The dataset comes from Yelp. Designed for educational use, it consists of 4.7 million reviews across 156,000 businesses made by 1.1 million users. It is drawn from 12 metropolitan areas and includes business attributes such as hours, bike parking, good for kids, and other business-specific attributes. Importantly for our project, it contains information about Elite reviewers, their number of years in the elite status, and their network of friends. It also contains information about the reviews each of them have written and how useful, funny, and cool these reviews were rated to be.

Merged tables

We initially received the dataset in two formats: JSON and MySQL. After some experimentation with JSON in Python, it was evident that we did not have the computing power required to load and manipulate the raw data. We then procured a MySQL database server and proceeded to load the MySQL data there. We performed our initial table joins and filtering in MySQL to produce a set of comma separated value files that we were able to load via Python in our Jupyter notebooks.

Name	Engine	Version	Row Format	Rows	Avg Row Length	Data Length	Max Data Length	Index Length	Data Free
attribute	InnoDB	10	Compact	1156441	92	101.6 MiB	0.0 bytes	85.8 MiB	4.0 MiB
business	InnoDB	10	Compact	155289	138	20.6 MiB	0.0 bytes	0.0 bytes	4.0 MiB
category	InnoDB	10	Compact	537431	67	34.6 MiB	0.0 bytes	38.6 MiB	4.0 MiB
checkin	InnoDB	10	Compact	3668434	71	251.8 MiB	0.0 bytes	277.0 MiB	4.0 MiB
elite_years	InnoDB	10	Compact	180271	61	10.5 MiB	0.0 bytes	13.5 MiB	4.0 MiB
friend	InnoDB	10	Compact	36301380	77	2.6 GiB	0.0 bytes	2.5 GiB	4.0 MiB
hours	InnoDB	10	Compact	731966	73	51.6 MiB	0.0 bytes	53.7 MiB	6.0 MiB
photo	InnoDB	10	Compact	194242	100	18.6 MiB	0.0 bytes	17.6 MiB	4.0 MiB
review	InnoDB	10	Compact	4220446	859	3.4 GiB	0.0 bytes	766.0 MiB	5.0 MiB
stg_attribute_pivoted	InnoDB	10	Compact	138224	57	7.5 MiB	0.0 bytes	0.0 bytes	6.0 MiB
text_analysis_data_nc_sc	InnoDB	10	Compact	251010	871	208.7 MiB	0.0 bytes	0.0 bytes	7.0 MiB
tip	InnoDB	10	Compact	1018640	156	151.7 MiB	0.0 bytes	139.5 MiB	4.0 MiB
user	InnoDB	10	Compact	1069200	135	137.8 MiB	0.0 bytes	0.0 bytes	4.0 MiB
user_review_stats	InnoDB	10	Compact	1170712	134	150.7 MiB	0.0 bytes	0.0 bytes	5.0 MiB

Figure 1 MySQL tables and their descriptions

Transformed variables

The dataset in its raw form was not conveniently formatted for machine learning use. There were several data manipulation steps that we had to take to convert it to a usable input for our models.

One of the transformations we executed early on was converting the category table to a more usable format. The category table was originally formatted with *business_id* as the key and one line for each category into which that business falls. Because some of the categories are superfluous and

more granular than we need them to be, we found the categories with the highest count of businesses in them with the assumption that these are the most general categories (for example, a business with the categories American (Traditional) and Burgers would also have the more general category Restaurants). Then we counted the number of reviews in each of these general categories by business user. Although the purpose of this was to be used in the clustering analysis in the hopes of finding groups of users who tend to review similar categories (e.g. food critics or hotel reviewers), it ended up not contributing to the final results so we ultimately decided to remove them.

We also created a few descriptive variables pertaining to a user's elite status that we used in the clustering analysis. We merged the elite_years table with the users table and created a boolean identifying all of the users that came out of the inner join as 1 for elite users. Those that were not contained in the elite_years table were marked with a 0. We then calculated the number of years that they had been elite from the elite_years table, which has a row for each year that a user was elite. This was a simple count on the number of times that a user appeared in the elite_years table. Due to time constraints we were not able to determine if these years were consecutive. A final calculated column we performed on the users table was determining the user's account age in days. The table contains the variable *yelping_since*, which indicates when a user first joined Yelp. We subtracted the *yelping_since* date from the maximum review date (2017-07-31) to calculate the user's account age.

We also transformed some variables that summarized user statistics, such as *NbrReviews*, *NbrUsefulReviews, NbrFunnyReviews, NbrCoolReviews, AvgStarsPerReview,* and *NbrStarsTotal.*

For the sentiment analysis, we created a boolean indicating a positive review as one that was given a star rating greater than 3. We used this as the target variable in the logistic regression in our sentiment analysis model.

Models and Results

Clustering

The goal of our clustering model is to identify a cluster of elite users, and then find the closest cluster to this elite group who would be the best candidates for attaining the elite distinction. We began with a dataset that contained information about each user, such as their number of fans, how many times they had been voted funny, cool, or useful, their compliment statistics, whether or not they are elite, and their account age. We did not include the flag identifying an elite user with the intention of revisiting this after clustering. We converted the dataset into a numpy matrix and then began

exploring k-means clustering models. Note that in this initial step we did not do any feature extracting, with the idea that we would execute that step after an initial clustering analysis.

We found that three clusters yielded the best results for the model by elite to non-elite user ratio after several trials with different values of *k* up to 100. When the number of clusters increased, the quality and size of the possible predictor clusters diminished. We plotted those clusters, shown in Figure 2.

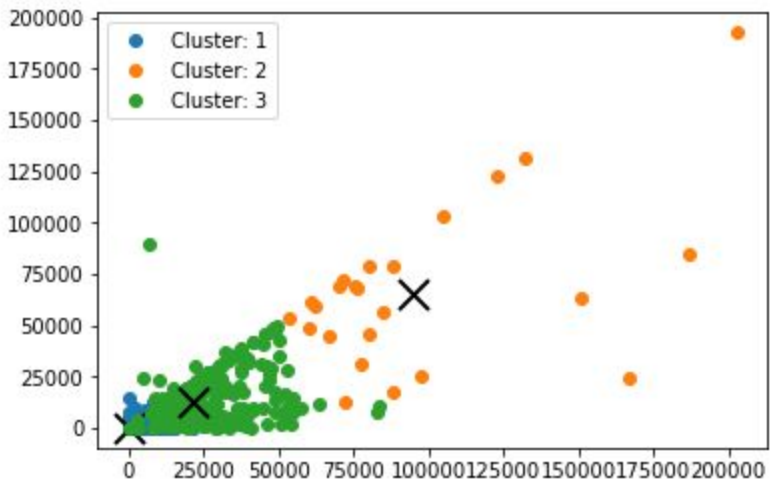

Figure 2 Initial clusters. k = 3. Cluster centers marked with an X

After finding the clustering to be weak and seeing potential room for improvement, we performed feature selection analysis to see if reducing the dimensionality would improve results. We employed the Principle Component Analysis tool from scikit-learn and identified the explained variance ratios. After transforming the data into the three principle components, we applied k-means to explore the results and compared these to the results from the clustering that did not use PCA. Figure 3 shows the plot of this cluster analysis.

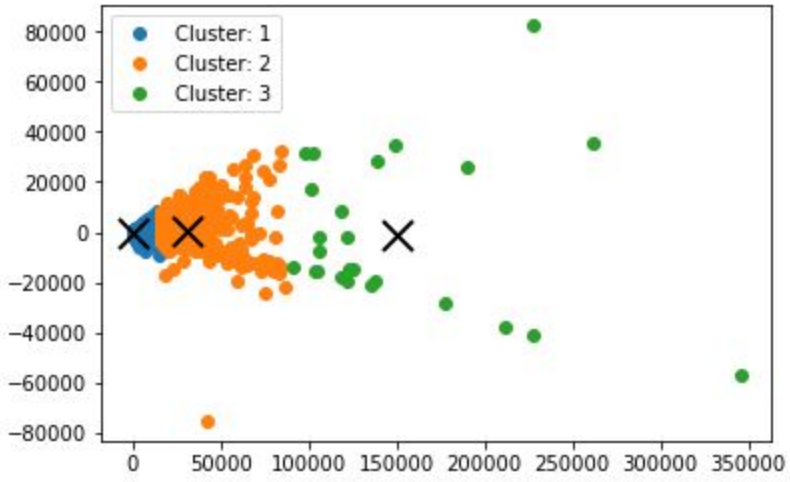

Figure 3 Cluster analysis after PCA was performed

We then created a data frame that included only the PCA variables and reset the index to align the PCA data with the user IDs. Then we merged the PCA data back to the original data frame and recreated the PCA variables with the new index for clustering analysis. We re-ran the cluster analysis on this newly created dataset and found that as we intended it produced the same results and we were able to move forward in the analysis.

We pulled the cluster numbers from the analysis and added them to the dataset so that each ID had its corresponding cluster number. We then merged the output with a table we created earlier that identified with a flag whether or not a user was elite. We grouped by cluster and elite user to determine which cluster was most appropriate to be considered the *predictor cluster*, or the cluster of users that we predict will be the next class of elite. Figure 4 shows the breakdown of each cluster.

```
Out[55]: Cluster  elite_user_flag
         0        0                1127215
                  1                  55670
         1        0                      2
                  1                     23
         2        0                     24
                  1                    428
         dtype: int64
```

Figure 4 Cluster results

It is important to note that the predictor cluster number changes with each iteration of the model, and this should be considered when other users are running the model.

We calculated the distance from the cluster center for each user, transformed x to the cluster-distance space, and added the cluster distances to the data frame. Then we merged the descriptor variables back into the data frame and began to explore the clusters.

Cluster 3 was identified by our model as the predictor cluster because the number of elite users was much greater than the non-elite users for this cluster. We plotted the elite users against the non-elite users in Cluster 3, shown in Figure 5.

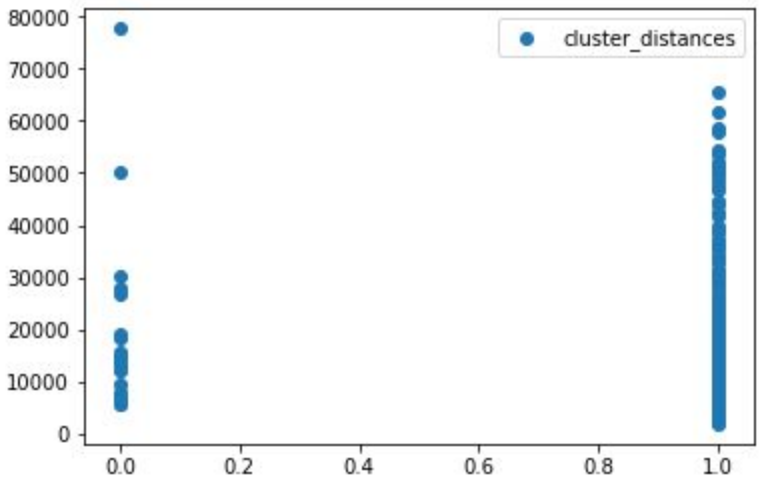

Figure 5 Elite users vs. Non-elite users in the predictor cluster

To reach our end goal of finding the next class of elite users, we ranked the non-elite users by their distance from the cluster center. This gave us the top 20 non-elite users who were most likely to be the next generation to bear the elite badge. Please see Appendix B for the list of those 20 users.

Text Mining

The goal of our text mining model is to determine if the count of negative and positive words in a review are indicative of the associated star rating. The analysis was performed on the full set of review data, which included 11,882 businesses. Figure 6 shows a preview of the sample dataset. Please see Appendix A for the variable definitions.

	id	stars	date	text	useful	funny	cool	business_id	user_id	name
0	7DB3B7ZZgLEepBNSa3qbgg	5	7/17/2017 0:00	Last week, I attended an event that I'll never...	1	1	1	--7zmmkVg-IMGaXbuVd0SQ	leXAScJxQ_Un9qrgmDEzpA	Primal Brewery
1	80foVUsSJRWkd94BOn1-hw	4	11/25/2015 0:00	I'll be honest, I don't get a chance to ventur...	8	2	7	--7zmmkVg-IMGaXbuVd0SQ	E43QxgV87lj6KxMCHcijKw	Primal Brewery
2	8EwOM89d64owYwW4ONi5dg	5	10/5/2015 0:00	Love this place. Second time there, and loved...	0	0	0	--7zmmkVg-IMGaXbuVd0SQ	G_nqFjq8tdurrJYQmNgQQw	Primal Brewery
3	94sO3Zl_UdlI8T_NAobzgg	2	4/2/2017 0:00	Wanted to sit outside and enjoy the beautiful ...	0	0	0	--7zmmkVg-IMGaXbuVd0SQ	YtbsDTnjb-o75V0HFHtzCg	Primal Brewery
4	AqABeZEFE5aRV0x3IZw8ug	5	10/7/2015 0:00	My husband and I took our friend here and we h...	2	0	1	--7zmmkVg-IMGaXbuVd0SQ	XQqNG5R6xX-gDnwNFNE20g	Primal Brewery

Figure 6 Review Data Sample dataset

Our first step in the sentiment analysis was to create a baseline word set of positive and negative terms using the scikit-learn feature extractor tool *tfidfvectorizer*. To do this, we began by defining a star rating of 3 as neutral, and created a boolean feature indicating a review as positive if was rated better than 3 stars. Our independent variable *x* was defined as the *text* variable, and the dependent variable *y* was defined as the *positively_rated* boolean variable.

Next, we vectorized the review text, turning all of the non-numeric tokens into terms. We then created between one and three n-gram words to account for negative phrases (e.g. "did not like" or "was not happy"). After transforming the terms into a document matrix, we applied a logistic regression with the goal of predicting whether a term is positive or negative. The coefficients resulting from this regression were indicators of a term's sentiment: those with low coefficients were negative terms, while those with high coefficients were positive terms. We retained 2000 terms with the smallest coefficients and 2000 terms with the largest coefficients to create the set of words which we would use as a baseline for the sentiment analysis of each business. Samples of the negative and positive terms are shown below in Figures 7 and 8.

```
Smallest Coefs:
[u'worst' u'bland' u'horrible' u'terrible' u'rude' u'mediocre'
 u'disappointing' u'poor' u'awful' u'overpriced' u'disappointed' u'ok'
 u'told' u'dry' u'meh' u'worse' u'cold' u'disgusting' u'money'
 u'unfortunately' u'tasteless' u'dirty' u'slow' u'disappointment' u'okay'
 u'sorry' u'waste' u'left' u'gross' u'asked' u'bad' u'unprofessional'
 u'stale' u'flavorless' u'lack' u'poorly' u'soggy' u'tasted' u'average'
 u'sad' u'tasted like' u'maybe' u'ridiculous' u'shame' u'barely' u'paid'
 u'lacking' u'management' u'charged' u'guess' u'zero' u'said' u'customers'
 u'ordered' u'salty' u'paying' u'better' u'lacked' u'avoid' u'greasy'
 u'overcooked' u'subpar' u'nasty' u'won t' u'sucks' u'attitude'
 u'undercooked' u'minutes' u'waited' u'pay' u'joke' u'burnt' u'ruined'
 u'unacceptable' u'frozen' u'beware' u'sick' u'mess' u'edible' u'supposed'
 u'manager' u'sadly' u'hoping' u'refused' u'downhill' u'completely'
 u'potential' u'ignored' u'par' u'yuck' u'fine' u't great' u'watery'
 u'inedible' u'sub par' u'average best' u'mediocre best' u'tiny' u'clearly'
 u'filthy' u'business' u'slowest' u'excited' u'response' u'wanted like'
 u'hopes' u'apparently' u'unless' u'food ok' u'waitress' u'taste' u'used'
 u'mushy' u'anymore' u'food' u'returning' u'expensive' u'customer' u'blah'
```

Figure 7 Negatively Associated Words

```
Largest Coefs:
[u'great' u'delicious' u'amazing' u'awesome' u'excellent' u'love' u'best'
 u'perfect' u'definitely' u'fantastic' u'friendly' u'wonderful' u'favorite'
 u'loved' u'good' u'enjoyed' u'helpful' u'happy' u'perfectly'
 u'highly recommend' u'outstanding' u'nice' u'glad' u'fresh' u'yummy'
 u'tasty' u'easy' u'reasonable' u'little' u't wait' u'thank'
 u't disappointed' u'professional' u'clean' u'highly' u'charlotte'
 u'pleased' u'notch' u'solid' u'quick' u'yum' u'incredible' u'fast' u'fun'
 u'comfortable' u'really good' u'quickly' u'spot' u'knowledgeable'
 u'perfection' u'satisfied' u'fabulous' u'won t disappointed' u'attentive'
 u'phenomenal' u'beautiful' u'super' u'recommend' u'helped' u'delish'
 u'beat' u'bit' u'die' u'exceptional' u'complaint' u'gem' u'efficient'
 u'downside' u'flavorful' u't wrong' u'unique' u'right' u'just right'
 u'huge' u'pleasantly' u'free' u'love place' u'affordable' u'courteous'
 u'disappoint' u'superb' u'exactly' u'heaven' u'lol' u'favorites' u'honest'
 u'pleasant' u'thanks' u'pleasantly surprised' u'able' u'reasonably priced'
 u'appreciate' u've' u'took time' u'great job' u'definitely recommend'
 u'try' u'reasonably' u'complaints' u'enjoy' u'great place'
 u'accommodating' u'appreciated' u'thorough' u'variety' u'good food'
```

Figure 8 Positively Associated Words

Our next step was to find the 50 most positive and negative words for each business and then count the number of those words that are in the baseline group we established. We repeated the steps from above, but this time using a multinomial logistic regression so we could predict the star rating instead of the negative or positive word association. After reviewing the SKLearn documentation, the *newton-cg solver* was selected because it supports multiclass problems, while not requiring the features to all be on the same scale[3]. We again set up the independent variable *x* as the review text, but this time set the dependent variable *y* as the star rating. We counted the number of positive and negative words per business that appeared in the baseline list we created and formed a new table based on the output, a sample of which is shown in Figure 9.

[3] 3.2.4.1.5 sklearn.linear_model.LogisticRegressionCV. scikit-learn.org. http://scikit-learn.org/stable/modules/generated/sklearn.linear_model.LogisticRegressionCV.html. Accessed 10 December 2017.

name	count_bad	count_good	stars
--7zmmkVg-IMGaXbuVd0SQ	6	3	3.904762
--EX4rRznJrltyn-34Jz1w	5	4	4.000000
--KCl2FvVQpvjzmZSPyviA	7	3	2.916667
--U98MNlDym2cLn36BBPgQ	11	3	2.750000
--cZ6Hhc9F7VkKXxHMVZSQ	2	2	3.913333
--j-kaNMCo1-DYzddCsA5Q	11	14	3.666667
--j-kaNMCo1-DYzddCsA5Q	11	14	3.666667
-0QtTRrAMn6DKLZNef3Ojg	8	4	2.333333
-2HjuT4yjLZ3b5f_abD87Q	8	2	3.375000
-2pQf1ceDZyE2ReCNbj-3A	8	3	3.440000
-2pmn-oTJeybmDrL-ojwrw	11	2	4.166667
-2uUrtgM5fiOaCpQEnPv0g	10	0	2.750000
-33_OPx1aKM22qxioPgJ_Q	5	2	3.000000
-4cA4Kt0UKzGYimrZsYuQg	3	4	2.666667
-5FndHRwNg-xhPsctrhTbA	3	1	2.875000
-5KBZ3UmQzW_PkrjHGMuUg	13	3	2.850000
-5L8zOxibac-vBrsYtxXbQ	5	5	3.572917
-5XuRAfrjEiMN77J4gMQZQ	11	6	3.714286
-5y5m3v_5ZHwlXwsp5K59Q	5	2	3.529412
-6e0liTvH5EoB4HuncuQgA	2	1	3.666667

Figure 9 Summarized Sentiment Analysis Extract

Figure 10 shows a plot of the count of the positive words on the x-axis compared to the negative words on the y-axis. The color of the marker is dependent on the average star rating for the business, rounded to the next whole number.

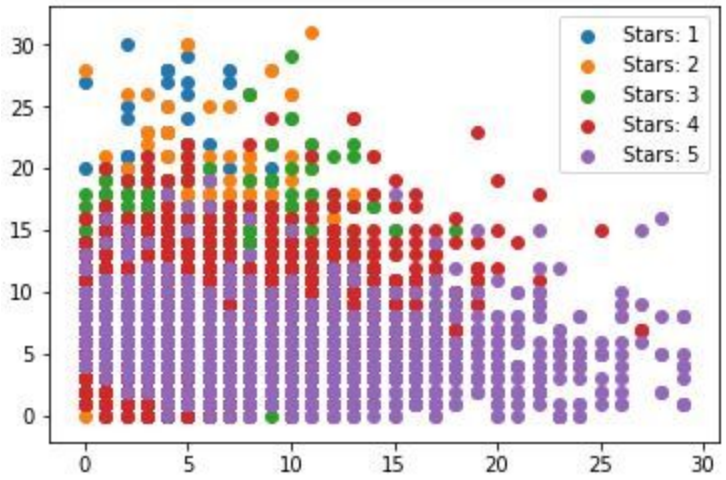

Figure 10 Plot of Positive Words (x-axis) by Negative Words (y-axis) Denoted by Average Business Star Rating

The majority of the reviews appear to be in alignment with the average star rating for the business. For example, in Figure 6, the bulk of the purple dots, representing 5 star ratings are clustered at the bottom of the chart, which shows a lower number a negative words; although, that number of positive words does span from 0 to 30. Most of the blue dots, representing 1 star ratings appear at the top left of the plot, representing a high number of negative words and a low number of positive words.

Perhaps of more interest are the reviews whose average star rating do not appear in line with the number of positive and negative terms in the text. For example, in the bottom left of the plot, near the origin there are a few 4 star (red) dots that have very few matches on the positive or negative word lists. The content of these reviews could be further examined to determine whether the star rating seems appropriate.

This association is further emphasized in Figure 11, which plots the ratio of positive to negative words on the y-axis to the average star rating of the business on the x-axis. Although the is some variance in the ratios the overall trend shows a higher ratio of positive to negative terms on the 5 star ratings and a smaller ratio on the 1 star ratings. However, there are some points that stick out that could represent a misleading star rating based on review context. For example, the point at approximately (1.3, 11) shows the ratio of positive to negative words is higher than would be expected for a low rated business.

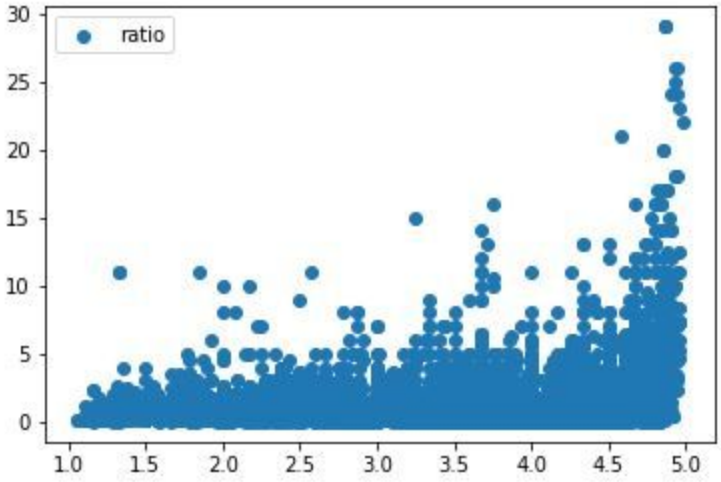

Figure 11 Plot of Ratio of Positive Words (x-axis)/ Negative Words (y-axis) Across Star Rating

Conclusion

Selecting the next group of elite reviewers on Yelp is not currently formulaic, but we believe that with our clustering model we can make strides towards making it that way. Finding those users who share characteristics similar to the elite reviewers through machine learning models can lead to a reduction in subjectivity, the elimination of bias, and a more fair practice of promoting users. We believe that given more time and resources, we could improve this model to more accurately analyze and predict each year's new class of users given the elite badge, and replace an abstract practice with data-driven decision making.

Similarly, we believe our text analysis model can be a stepping stone towards the reduction of user error and potentially fraud. Yelp could apply the model to the text of their reviews and verify that a user's review aligns with their star rating. It could also flag instances that exist where these two parameters do not align, and with continued research and development we believe this could be the first step towards identifying fraudulent reviews.

Appendix A

Variable Definitions

Variable	Definition
id	Review ID
stars	Star rating (1-5)
date	Date of review
text	Review text
useful	Number of times the review was voted useful
funny	Number of times the review was voted funny
cool	Number of times the review was voted cool
business_id	Business ID

user_id	User ID
name	Business name
category	Business category

Appendix B

Top 20 users predicted to become Yelp elites in the future from cluster 3:

id	cluster_distances	user_name
bXwZ86LHDjW2iEAQcPPDtQ	5808.082968	William
9MF-l8xgyWN9dm2-r2mF9w	5811.560938	Jim
6LvSDK7CKaSQqoltfPgRVA	6559.445718	Mika
5ZsumqlAUtKqBGAhGD2Gdg	7249.045524	serbelle
wTcAXp4eV7OowBmc2uxzcg	7994.991750	Scott
EifD5YZX7FsVMHieTkX7xQ	9607.388023	Peaches
OdwpAeVVjsRzmdSdQtBcoQ	12256.504858	Michelle
w2l2UY6NsxNGfr3rSdz7vQ	12299.498204	Val
RqR1bGDgdOPyHDnl73LeFw	12423.530794	Marc
Ug0ujy9-jd0Oc2nk3KlotA	12597.526500	Messer
e-Bh0XDCOgYMaC-BmdxkAQ	13072.617269	D.wight
csWwDC-ALUmvAdWLCTZ5-w	13822.981135	Spicy
7lP8KOkPNncenRib2QVsLA	14169.040374	Jo Anne
r1DaKxM0KcblWAGpld-FEQ	14801.078708	Giannina
lBZMvu2dTKy-j23qCJgVWQ	14825.583464	Sue Ellen
_K9sKlA4fVkWl4hyGSpoPA	15396.020009	Robert
lwvh3Q9mt8vefdSymdmaSA	15490.100112	Mic
jxjbrQOlVbRjyYWQpRHK6w	18316.521193	Marvin
SDytcPY5fiuuDbTkPE8GyA	19025.948065	Pinky And The
FZNRzY6m67fhlnNE8XglTQ	26609.124877	S. Alicia