

Building a tool to assess business performance based on customer reviews

Jonathan Leslie

November 22, 2015

Introduction

With the increasing popularity of on-line crowd-sourced review databases such as Yelp, both consumers and business owners have access to a wealth of customer review data that was not possible a decade ago. For the business owner, this data represents a valuable resource, providing insight into customer satisfaction and business performance. While objective assessments such as star ratings and tick-boxes may give some idea as to customer satisfaction, they do little to provide real insight into exactly how a given business is meeting the expectations of its customers, or as the case may be, falling short.

To fill this gap, customers are encouraged to write reviews that appear as blocks of text. These give the consumer a chance to elaborate on his or her experience with more specific descriptions and they offer a wealth of information for the business owner. Unfortunately, reading customer reviews can be a long, painstaking process. For the business owner wanting to draw general conclusions from hundreds of such documents, the task can be arduous and prohibitively time-consuming.

This project is a feasibility study aimed at creating an application that can help streamline this process, making it easier for business owners to gain more insight into the satisfaction of their customers from Yelp reviews. By submitting a query by Yelp business ID, the application will return several pieces of information. The first is a plot of average star ratings (averaged monthly) over time, showing the general trend in the business' customer satisfaction. The second is list of words that are associated with the business' positive and negative reviews, using text mining algorithms and a user-defined threshold star rating (to distinguish "positive" from "negative" reviews). If, for example, a restaurant owner discovers that the word "wait" is often found in the negative reviews, he or she might infer that customers are often critical of how long they must wait for either seating or food/drink service. The goal of this project is to determine if such text-mining algorithms can be tuned to provide useful information to the business owner.

Methods and Data

Sourcecode

The R code used for this study can be found in this report's Github repository: <https://github.com/jleslie17/CapstoneProject> ##Loading Data The data for this study came from the Yelp Dataset Challenge (http://www.yelp.com/dataset_challenge). This dataset includes five JSON files corresponding to several aspect of Yelp review data and should be unzipped in the working directory. For this study, I used only the files pertaining to business IDs and reviews. I loaded the files of interest as below and saved them as rds files for subsequent analyses.

Subsetting the data into a shortlist

Beuase this is a feasibility study, I used only a subset of the data. This application could easily be expanded to include all businesses in the data set. It should be noted, however, that with fewer reviews, data describing customer satisfaction (see section below) may become noisy. For this reason, I restricted my feasibility study to a shortlist of businesses consisting of only restaurants and only those restaurants that had 100 or more reviews in the Yelp data set.

Examining changes in performance over time

To examine the change in customer satisfaction over time, I wrote a script that gathered review data for a given business (using the `business_id` field of the business data file, assigned to the variable `BID` (Business ID)). This used only the `business_id`, `review_id`, `date`, `stars` and `text` variables of the review data file. Reviews were gathered by month using the `as.yearmon()` command and average monthly star ratings were calculated. The first six lines of an example data frame are shown below.

##		business_id	YearMonth	AvStarsPerMonth
## 1	2WHP5nhS1rFszfRBKe6fWQ	2014-02-01	4.888889	
## 2	2WHP5nhS1rFszfRBKe6fWQ	2014-03-01	4.318182	
## 3	2WHP5nhS1rFszfRBKe6fWQ	2014-04-01	4.058824	
## 4	2WHP5nhS1rFszfRBKe6fWQ	2014-05-01	4.000000	
## 5	2WHP5nhS1rFszfRBKe6fWQ	2014-06-01	4.187500	
## 6	2WHP5nhS1rFszfRBKe6fWQ	2014-07-01	4.571429	

Examining review texts

To perform text-mining, I used the `tm` package. The program generated a corpus for all reviews in each pool (good and bad). The corpora were preprocessed to remove whitespace, transform all words to lowercase, remove stopwords (such as ‘the’) and stemmed to retrieve word radicals (free of suffixes). Punctuation was also removed. Document-term matrices were generated for each corpus, and mean frequency (how often each term appeared in a corpus) was calculated for each term. The terms were then sorted in order of decreasing frequency and the top 20 were printed as output.

Results

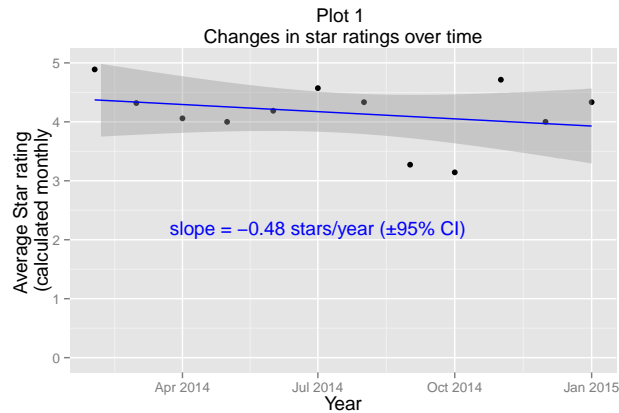
Sample business

For the remainder of this report, I have shown the data for a single example restaurant with the Yelp business id `2WHP5nhS1rFszfRBKe6fWQ`. Because this application is meant to function as a tool for the business owner to investigate performance, I reasoned that in this context, an analysis of an individual business would be more useful than a general analysis of customer tastes. The following sections describe the steps in performing the analysis for this sample business.

Examining changes in performance over time

A central tenant of business is to “know your audience”. While a business may have met the demands of its audience well in the past, consumer taste changes quickly, and it can be hard to stay on top of that as a business owner. To assess how well a business is meeting the demands of its public, I wanted to examine ratings over time.

To visualise this trend, the program plots average stars over time and draws a linear regression line. The slope of this line can be thought of as the trend in customer satisfaction.



In this example, the change in star rating (stars per year) for business 2WHP5nhS1rFszfRBKe6fWQ is: **-0.48**

As we can see from the plot, the reviews have, on average, gotten worse with time. The owner of this restaurant might want to know why. The next section deals with ways to help answer that question.

Generating word lists for good and bad reviews

To help shed light on factors that might be contributing to customer satisfaction (either for the good or the bad), I wanted to employ text-mining techniques on the texts that accompany each review. I hypothesize that such an analysis might reveal trends associated with poor customer satisfaction.

Since each business owner may want to decide what star rating differentiates between a “good” and a “bad” review, I allow this threshold to be set. In this case, it is set with the StarsThresh variable. In this example, StarsThresh is set to 3.5, so reviews receiving 1, 2 or 3 stars will be pooled in the “bad” group, and the rest will be considered “good”.

```
StarsThresh <- 3.5
```

The text mining algorithm generates a so-called corpus for each group, with the text from each review within the groups functioning as a distinct document. The algorithm goes on to apply preprocessing techniques as described under the Methods and Data section above to yield final corpora to be used for subsequent analyses.

The good and bad corpora were used to generate document-term matrices, which were further used to produce final lists of frequently used terms. The 20 most commonly-used terms (minus stopwords such as “the” and “an”) for each group are listed below:

```
[1] "Keywords associated with good reviews:"
```

seafood	good	great	food	place	lobster	high	fresh	tide
0.66	0.63	0.62	0.61	0.56	0.48	0.41	0.40	0.36
tri	back	bar	happi	nice	love	order	will	best
0.36	0.35	0.33	0.31	0.31	0.29	0.29	0.28	0.27
friend	hour							
0.27	0.27							

```
[1] "Keywords associated with bad reviews:"
```

food	order	place	good	servic	fish	just	high	nice
0.94	0.84	0.72	0.56	0.53	0.44	0.44	0.41	0.41
one	seafood	bar	dinner	menu	pot	steam	tast	went
0.41	0.41	0.38	0.38	0.38	0.38	0.38	0.38	0.34
back	better							
0.31	0.31							

Word usage visualization

To better visualize word usage, I took advantage of the wordcloud package to produce a wordcloud for each group (good and bad reviews). This process included converting each document-term matrix into a data frame and employing the wordcloud() function as shown below.

```
mGood <- as.matrix(DTMGood)
mBad <- as.matrix(DTMBad)

vGood <- sort(colSums(mGood), decreasing = T)
vBad <- sort(colSums(mBad), decreasing = T)
dGood <- data.frame(word = names(vGood), freq=vGood)
dBad <- data.frame(word = names(vBad), freq=vBad)

wordcloud(dGood$word, dGood$freq, scale=c(2.5,0.5),
  max.words=50, random.order=FALSE,
  rot.per=0.35, use.r.layout=FALSE,
  colors=brewer.pal(8, 'Dark2'))
wordcloud(dBad$word, dBad$freq, scale=c(2.5,0.5),
  max.words=50, random.order=FALSE,
  rot.per=0.35, use.r.layout=FALSE,
  colors=brewer.pal(8, 'Dark2'))
```



Word clouds for “good” reviews (left) and “bad” reviews (right).

These word clouds help to give a better visual appreciation for what topics have often been discussed in customer reviews, and may provide useful information to the business owner.

Discussion

The goal of this project was to explore ways in which business owners might be able to make better use of customer review data. Specifically, I was interested in (1) assessing changes in customer satisfaction over time and (2) using text-mining approaches to extract useful information from customer reviews. This report is a feasibility study, aimed at determining if my program achieves these goals.

In the first case, I believe the algorithm was largely successful. Using linear regression in concert with appropriate grouping and averaging parameters, this program produces a clear picture of trends in average customer satisfaction over time. The robustness of this analysis is dependent on several factors, including the number of reviews gathered, the variation in stars awarded and the period over which reviews are averaged. I restricted this study to only those businesses that had more than 100 reviews, hoping that this would make the algorithm less susceptible to these confounding factors. For businesses having fewer reviews, the period over which star ratings were averaged would probably have to be adjusted in order to smooth out noise. Nonetheless, for this example business (and many others not shown), the algorithm produces a nice visualization of temporal trends in customer satisfaction.

For the second part of the analysis I employed text-mining techniques to extract meaningful keywords from review texts. When analyzing word clouds one must use caution. For instance, in this example, many words are shared between good reviews and bad ones. Words such as “food”, “place” and “good” show up in both groups with similar frequencies and therefore don’t tell use much about customer satisfaction. Other words are more telling. For example, in the positive reviews, we see that the words “lobster”, “fresh”, and “seafood” appear with a high frequency. In contrast, the words “order”, “servic” and “menu” show strongly in the negative reviews. One might infer, then, that when customers of this business are happy with their experience, the quality of the food is something they notice. When they are unhappy, however, the service seems to be where the business is falling short.

In the case of this example, the program worked reasonably well. Text mining is a complicated endeavour, and it is rare that the output gives a black-and-white picture of the trends in term usage. I suspect that this algorithm could be tuned to be more robust, and in a final application, it may be wise to make these tuning parameters variable that can be adjusted by the user. Ultimately, whether or not such text-mining applications are useful to the business owner may largely depend on the individual.