Using ElasticSearch on Yelp Data

Matthias Funke mafux777@gmx.de

21 Nov 2015

Using ElasticSearch and R to analyse text in reviews

Introduction

The question I decided to analyse was: There are several binary attributes linked to the businesses, e.g. "Live Music". However, only a small percentage of the businesses have that attribute checked (about 800 - 1.3%). By doing text analysis of the reviews, I wanted to determine:

a) Can we statistically determine that the attribute is correctly set?
b) Can we improve the accuracy of the attribute by parsing / searching the reviews? I used the R package "elastic" with ElasticSearch in the background to do some of the heavy lifting.

Methods

ElasticSearch is an open-source search engine. It creates a reverse index of documents and allows to perform relatively sophisticated queries on the documents. A document in the context of this project could be a business or a review. The following steps were performed for the analysis to take place:

- Install ElasticSearch (https://www.elastic.co/downloads/elasticsearch)
- Install elastic R package (install.packages("elastic"))
- Ingest JSON file provided with businesses and reviews (Business.R and Review.R)
- Create a suitable mapping of data types to facilitate creating the index in ElasticSearch (rev.JSON)
- Ingest the reviews into ElasticSearch (elastic.R)
- Do a Search for "live music" or "band" in all reviews. ElasticSearch returns 16378 hits with a certain score (ES_score). More about the scoring later.
- Add all scores by business_id. 4186 business had a score greater than 0. Normalize using number of reviews per business.
- Calculate *max*, *mean* of the *ES_score*
- Compare and contrast the results with the attribute "Music-live" by showing a histogram of the *ES_score* by attribute (different colors for True and False)

Scoring in ElasticSearch - the Statistical Model Used - TF/IDF

The first thing you'd expect a search engine to do is return a list of all documents matching your search string (in this case, "live music" OR band). You'd expect the most relevant search results to be on top of the list. The sort order of the list is in fact determined by the score. The document with the highest score is the first in the list. The score used for my analysis is the "term frequency / inverse document frequency". Note that this is what I consider a *statistical model* in the context of evaluating this exercise.

More info here:

https://www.elastic.co/guide/en/elasticsearch/guide/current/scoring-theory.html

Term Frequency If our term "live music" appears several times in the document, that must mean that particular document is more relevant.

Inverse Document Frequency This score depends on the search term itself. Of two searches with different search terms, a hit for a term that appears infrequently in the entire index (all documents) will have a higher score. Since we are only searching for one particular string "live music" OR band, this score is not relevant.

Field Length Norm Imagine two reviews, one simply saying "Place with great live music" and another one going on and on about the food, and then saying "by the way, the place has great live music". Which one seems more relevant? The shorter the field length, the higher score.

Results

Recall that of the roughly 60,000 businesses, 813 have the attribute for live music set to "true", 1907 to "false", and the overwhelming majority of 58,464 have not set the attribute (showing as *NA*). I could have restricted the text analysis to only those businesses which can be reasonably expected to have live music, but I chose not to. ElasticSearch can cope with 1.5M reviews and performs the search in seconds once the more time consuming index building is completed.

We compare and contrast the attribute settings to the statistical parameters Max, Min, Mean, SD of the *ES_score*.

As mentioned earlier, I added up all the review scores by *business_id*, and called that *ES_score* for the business. Since different business have a different number of reviews, I then divided the *ES_score* by the variable *no_reviews*. This is based on the assumption that a place with live music would have a percentage x of customers mentioning it in the review, no matter whether that place receives a large or small overall number of reviews.

Let's now review the results:

```
biz_with_ES_score[,list(N=.N, max=max(ES_score), mean=mean(ES_score), min=min(ES_score),
sd=sd(ES_score)), by=attributes.Music.live]
```

```
##   X attributes.Music.live     N       max        mean min          sd
## 1 1                    NA 58464 0.5941232 0.001046862   0 0.008658254
## 2 2                 FALSE  1907 0.3043548 0.003886252   0 0.016103957
## 3 3                  TRUE   813 0.5734039 0.044372846   0 0.056577998
```

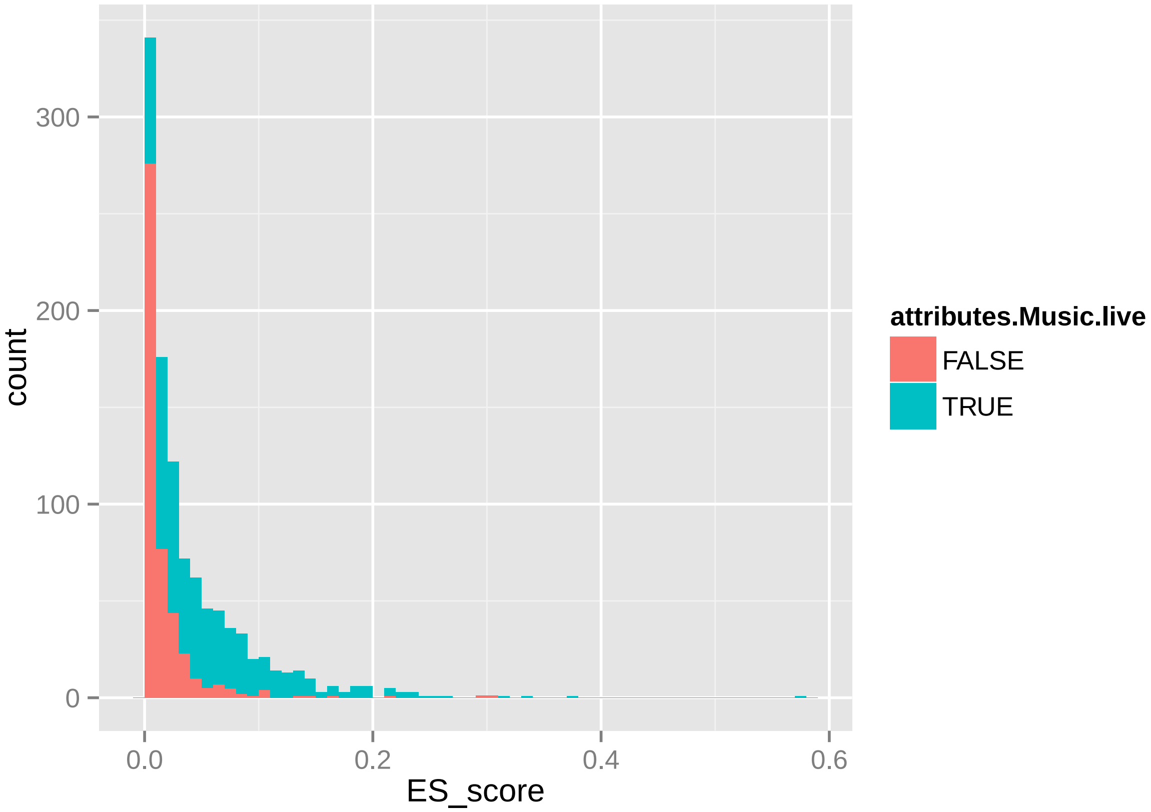
Note how the mean of the ES_score for "T" is roughly 10 times higher than for businesses with Live Music=F.

It's a bit surprising that there are 204 businesses with the attribute set to TRUE, but an ES_score of 0, i.e. no mention in any review of "live music" or a band. I reviewed a sample of the reviews in that category and concluded that they were mis-classified, since nothing in the review suggested live music by any stretch of imagination. So I would call the 204 businesses "false positives".

On the other hand, there are 3117 businesses with an ES_score > 0 and no attribute set for live music at all. These could be called "false negatives". If we increase the threshold by 1/100 at a time, e.g. requiring an ES_score of at least .05, we would still obtain 307 businesses where live music was mentioned in several reviews.

By combining both points mentioned above, we would increase the number of accurate records by about 500, which is a significant improvement given that we started off with about 800.

Let's look at a histogram of the ES_score for the businesses which have the attribute set (T/F) and have ES_score>0:



We can see clearly that the reviews which have the attribute live music set to true have much higher *ES_score*. With low ES_scores, between .01 and 0.03, the attribute setting seems more random or arbitrary. We would not expect so many businesses to be marked with the live music attribute although their scores are so low. The discussion section explains why.

Discussion

Yelp says on its web page: “Where does Yelp get its /business attribute/ information (e.g., how expensive the business is, whether it’s good for kids, and suggested attire)?

These subjective attributes are voted on by users who have reviewed the business. They can change over time as more people review the business and cast more votes. The more objective attributes that we show in the business listing (whether the business accepts credit cards or is wheelchair accessible) can be set by the business owner if he/she has signed up for a free Business Account."

So now we have an explanation of the discrepancies between the Elastic Search Score for “live music or band” and the setting of the attribute. The process is not the same. I’d argue that the proposed method of scoring is better, since it is more scientific, more easily reproduced, and generates higher number of business with the desired attribute.

The recommendation for Yelp would be to review the businesses which seem to have the Live Music attribute inappropriately set, e.g. send a message to all businesses with a low score and the live attribute set to T, so that the attribute can be reviewed (in the case of self-attribution) or to review the voting mechanism to better understand why such a high number of businesses was voted to have live music but the reviews don’t mention it.

One thing I learnt in this exercise that it is incredibly challenging for a business like Yelp to deal with the massive amount of text that gets uploaded to their web site every day. No-one can expect a human to read all this text. So automated text analysis like the one presented in this paper is a must.

Known Issues with this Research

I did not filter out non-English language reviews. They have the potential to skew the numbers slightly. I performed the search with ElasticSearch over all reviews, including businesses where live music would never be played. This makes the mean score for that category (attribute.Music.live==NA) less meaningful.

Source code

https://github.com/mafux777/Yelp_ElasticSearch/