

Investigation Into Predicting Yelp Ratings Based on Review Text

Sean Clarke

21 November 2015

Introduction

This report describes an investigation into the following question.

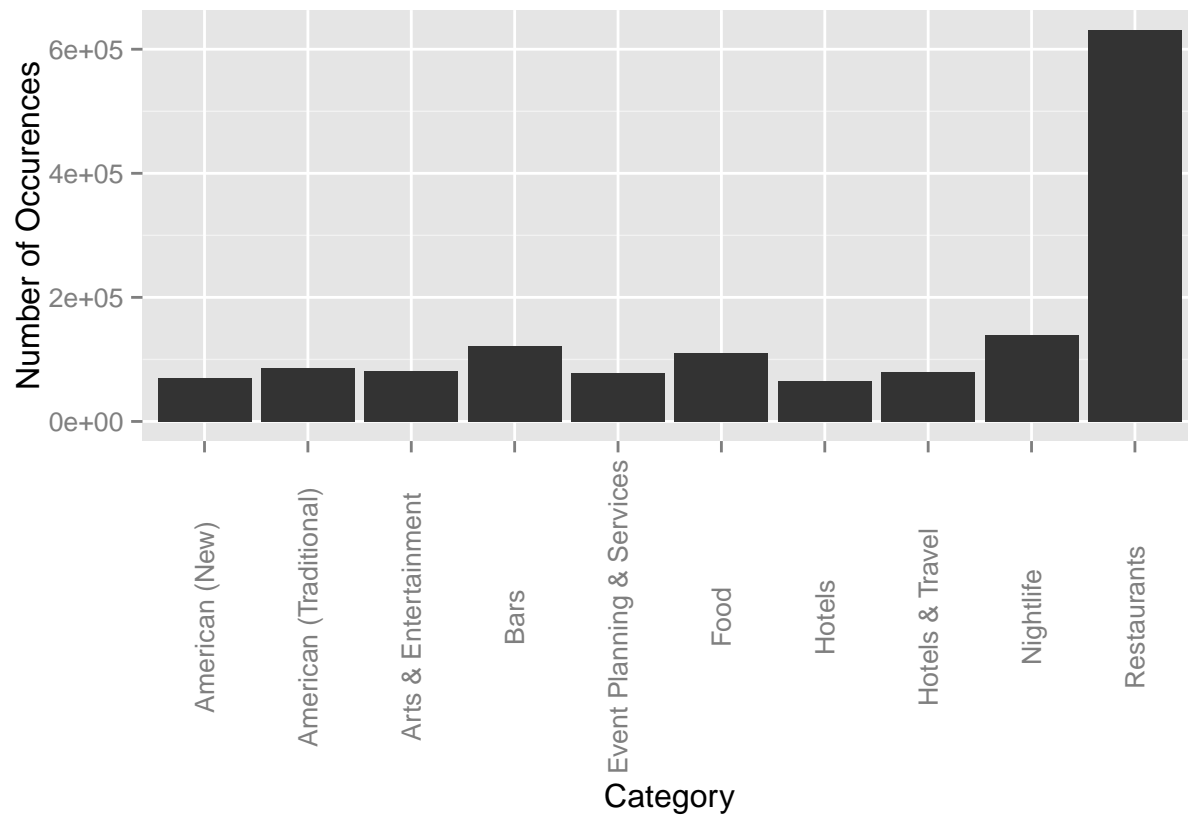
Is it possible to predict the number of stars given to a business based on analysis of the text in the review? Is the accuracy of the model affected by demographics, for example does it work less effectively in Germany. Given the fact that Spanish is increasingly becoming the language of the US, is that important in answering this question?

This question would definitely be of interest to yelp.com and their customers as it might provide a basis of judging reviewer sentiment in situations where they have comments but no ratings. It might also form the basis of generating estimated ratings from Social Media posts regarding businesses.

Methods and Data

Exploratory Analysis

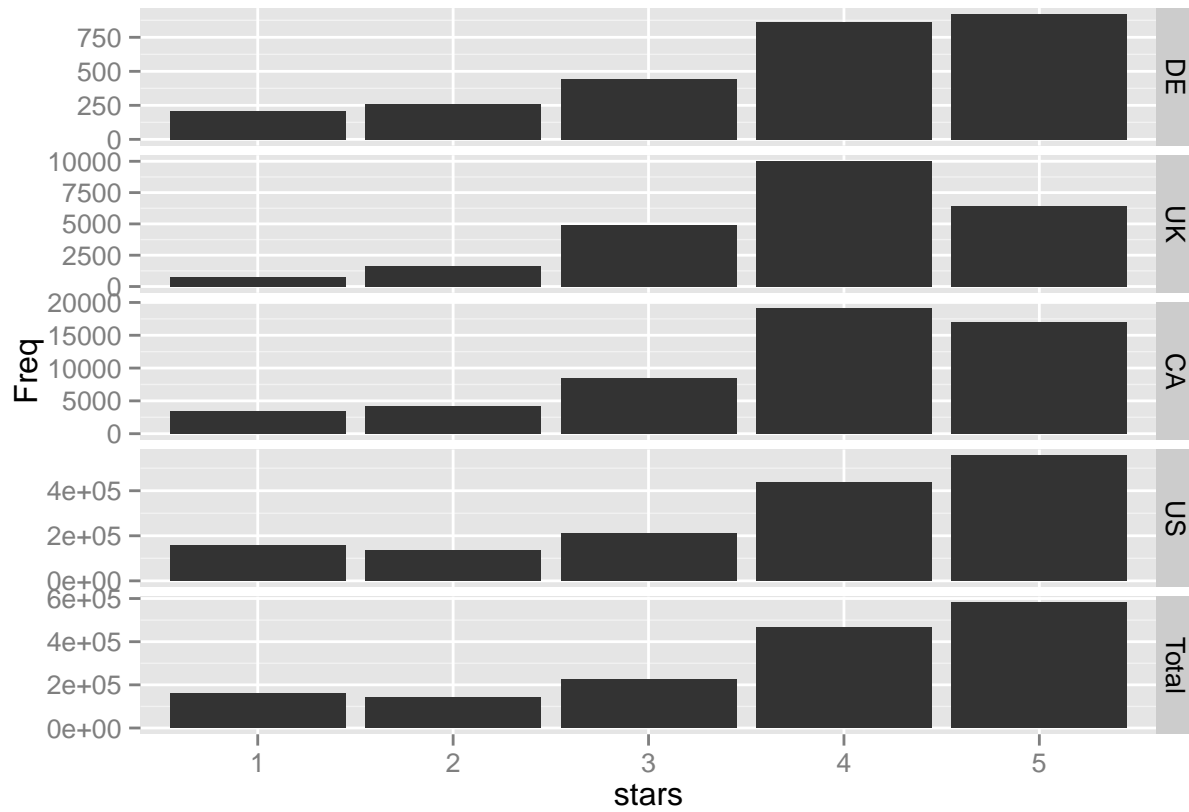
Initially we investigate the distribution of categories to see if one dominates in order to see if there was potential to limit the scope of the search to make this huge dataset more manageable. Clearly restaurants dominate all other categories so we limited the scope of this investigation to restaurants.



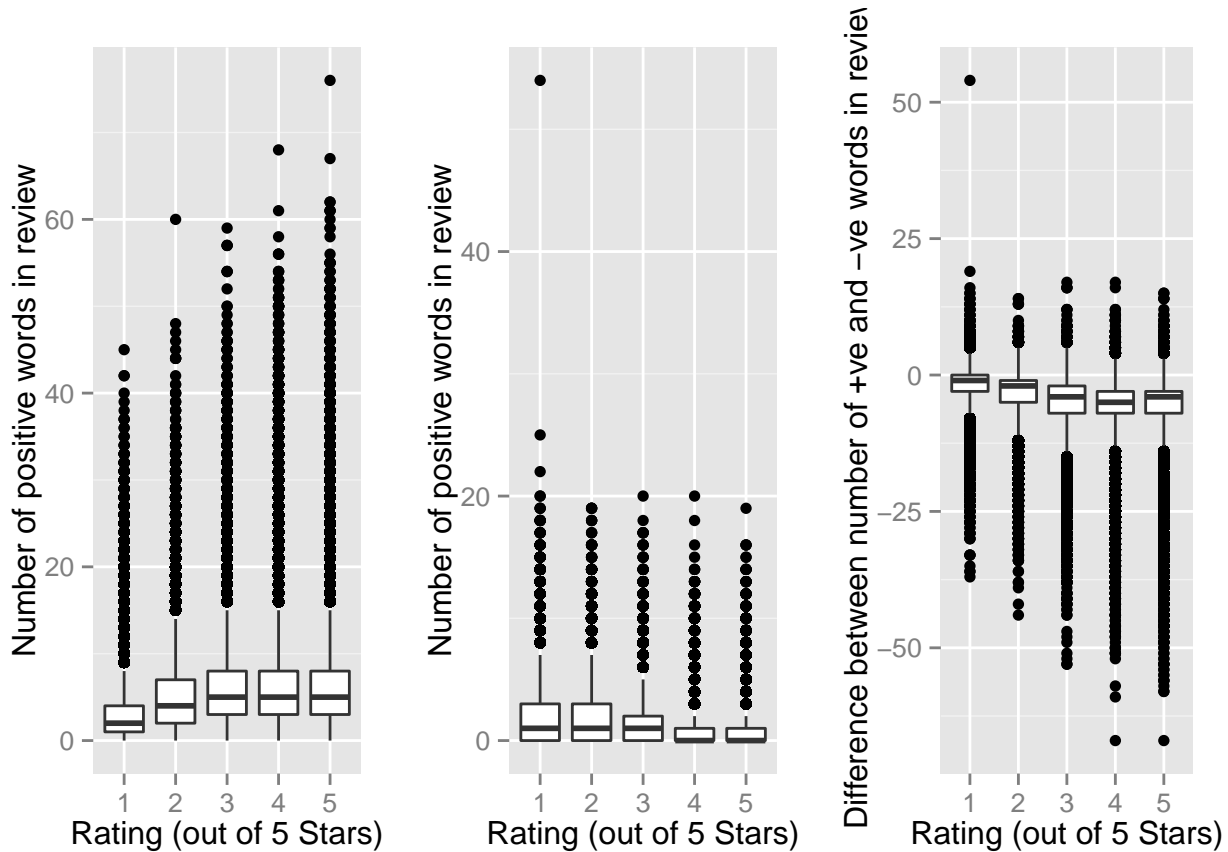
We also wanted to understand the geographical spread of the establishments in question so we extracted this from the data. This information was gleaned from the state variable in the data which we discovered wasn't entirely regular or always mapping to official state designations so we manually built a [lookup table](#) to clean this and then evaluated it.

country	length(country)
DE	2692
UK	23751
CA	52168
US	1490653

It's also informative to look at the distribution of the review scores.



Finally once the conclusion was reached that modelling using the the whole bag of words was not realistic computationally the approach used for sentiment analysis was based on the thesis that higher ratings would tend to have more positive words in them overall and that lower review scores would tend to have more negative words. The boxplot below based on the trainin data allows to investigate this idea.



On the whole, this plot seems to at least partially support this thesis. Higher ratings tend to get more positive words and lower scores tend to get more negative reviews. However there do appear to be significant numbers of outliers that do not follow this trend. Its also not clear if the trend is pronounced enough to make it possible to distinguish between for example four and five star reviews.

Preparation

All code with a ReadMe to describe processing steps is on [github](#) in order to facilitate reproducibility.

The streaming JSON files as published by Yelp were downloaded and an [ER Diagram](#) was built to relate all the quantities and the relevant keys. [R code](#) was written to pull the relevant information from the files. Some preprocessing was done including limiting the analysis to Restaurants and then the data was split into training, validation and test sets in a 60:20:20 ratio. As this was a lengthy process, the data was saved to three RDA files.

The summarised data then contains the following categories

```
## [1] "user_id"      "stars"        "date"         "text"
## [5] "business_id"  "full_address" "categories"    "city"
## [9] "name"         "longitude"    "state"         "latitude"
## [13] "country"
```

The tm Text mining package was then used to extract the Corpus from the text. We then proceeded to apply standard transforms to the corpora in order to facilitate analysis. Punctuation, numbers and whitespace was removed, all text was converted to lower case and standard English “stop-words” were removed using the facility provided by the tm framework.

The Coprora were then transformed into a TermDocument matrix in order to give a “bag of words” representation and only words that occurred in more than 1% of reviews were retained. Initially the approach was to attempt to model directly using the bag of words but it became readily apparent that attempting to model a large number of observations with such a large number of features was computationally prohibitive with the available resources.

Faced by this problem we switched tack and attempted a basic sentiment analysis of the text following [Breen et al](#) using lists of known positive and negative words originally provided by [Hui and Lu](#). Following this approach, we simplified the problem to one of modelling the dependency of the star rating on three features. These were number of positive words, number of negative words, and review length in words.

Model Development

During the training phase, multiple models were evaluated including rpart, ctree (this was used because the rpart tree that was seen, would never predict 2 or 3 star reviews), Random Forest, Naive Bayes and a Linear Model. The efficacy of these models was evaluated by comparison the Root Mean Square(RMS) error and percentage of exact matches “in Sample”.

	PctMatches	RMSValues
ctree	42.98	1.42
RF	42.87	1.45
NB	40.25	1.49
rpart	40.09	1.48
lm	23.90	1.30

Based on these results, the ctree, Random Forest and Naive Bayes models were used to predict the star ratings for the validation data set to get an estimate of the out of sample errors in an attempt to preclude potential overfitting.

	VPctMatches	VRMSValues
ctree	42.86	1.41
RF	42.64	1.44
NB	21.22	2.07

Based on these results the ctree based model was chosen to be the one that we would use to predict from the test data.

Results

Model Accuracy

The results below, show some statistic used to judge the accuracy of the model.

	1	2	3	4	5
18436	9054	6536	5725	4991	
1101	1129	971	852	486	
665	928	918	870	499	
5016	8204	14681	25057	21024	
6860	8691	21437	60815	88905	

Discussion

Explain how you interpret the results of your analysis and what the implications are for your question/problem.