Attributes favourable to Restaurant Success

Edwin Seah

21 November 2015

1. Introduction - Preamble and Research Questions

In the Yelp! dataset challenge, participants are provided with rich real-life data from Yelp! and asked to present and answer open-ended questions about the list of businesses, ratings and user behaviour. Large numbers of eating establishments are represented in the dataset; Yelp!'s raison d'etre and format tend to favour the review of such businesses and its use by a community of users. The project is therefore interested in the following questions:

- What characteristics does a highly rated restaurant possess that are roughly consistent across different locations (states/cities)?
- Do such restaurants obtain more positive tips that are useful to potential customers than those rated lowly?
- Is it possible to generate a reasonable and useful model that predicts their rating from these characteristics/review/tip types?

The response variable is `stars` (business rating) which has levels from 1 to 5 in 0.5 steps. An attempt at classifying the most favourable characteristics by location(state, city) is made, then the sentiment of review text is checked. For brevity, not all code is shown although the the full code is available from the Rmd document.

2. Methods

2.1. Data Sources, Cleaning and Transforming The dataset used is the academic dataset provided from the sixth Yelp! dataset challenge. The data is read into a series of three data frames (business, tip, review) using `readr::read_lines()` and `jsonlite::fromJSON()`, and flatten using `lapply(filesToRead, function(x) fromJSON(sprintf("[%s]", paste(read_lines(x), collapse=",")), flatten=TRUE))`. As the dataset is large and converting from JSON into a usable format is excruciatingly slow, the data frames are cached into .rds files and read from there.

```
filenames <- c("business", "tip", "review", "user")
vecRDS <- paste0("../data/yelp/", filenames, ".rds")
business <- readRDS(vecRDS[1]); tip <- readRDS(vecRDS[2])
review <- readRDS(vecRDS[3]); user <- readRDS(vecRDS[4])
```

For usage in sentiment analysis of review/tip texts, a combined list of positive/negative words from A.Finn(2011) `dict.afinn` and Hu,Liu (2004) `dict.huliu` is used. They are merged after scaling the former AFINN-111 (containing nuanced scores) to between -1 and 1 to match the latter, then joined into one set of positive and negative words. The enlarged set provides better coverage over a wider variety of English terms. `vec.pos` and `vec.neg` are kept as vectors of the final postive and negative words.

```
# Merge them into one single set, using only -1 or 1 to denote +/- sentiment
dict.terms <- merge(dict.huliu, dict.afinn, by.x="word", by.y="word", all=TRUE) %>%
    mutate(score=ifelse((!is.na(score.x)), score.x, ifelse((score.y>0), 1, -1))) %>%
    select(word, score)
vec.pos <- as.vector(dict.terms[dict.terms$score>0,]$word)
vec.neg <- as.vector(dict.terms[dict.terms$score<0,]$word)
```

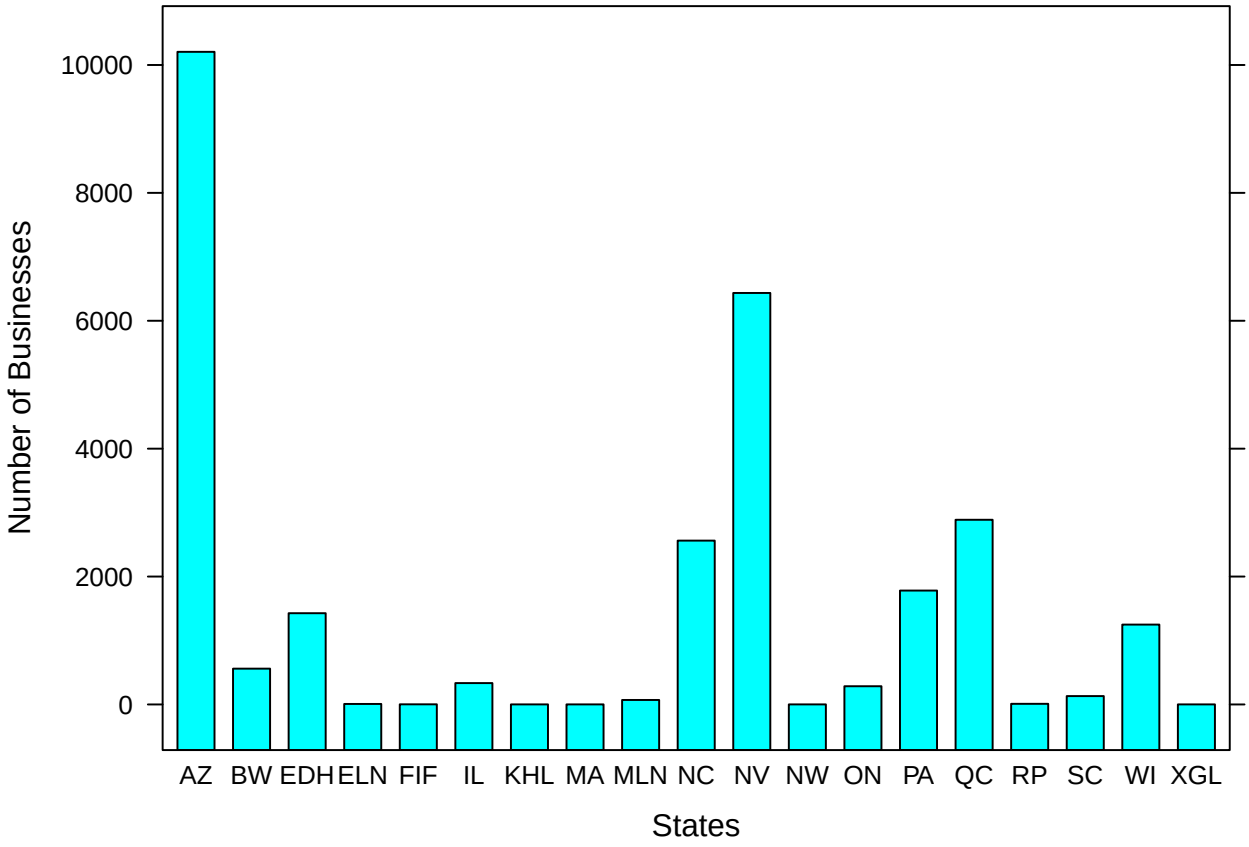
To filter for businesses which are restaurants or serve food, `grep` is run on the category column from the business dataset for "Food|Restaurants", and subsets are taken accordingly by intersecting with `business_ids` from reviews and tips. A series of transformations is applied as follows: - factorizing stars variable - flattening of nested lists - removing redundant and irrelevant variables/attributes, such as "Hair|Insurance" - turn column names R-compatible using `make.names()` - converting city names from unicode to ascii equivalent

```
# Subset the other data frames for only restaurants (subset is fastest)
b.rest <- business[grep("Food|Restaurants", business$categories),]
b.rest$categories <- sapply(b.rest$categories, toString)
b.rest <- subset(b.rest, select=-c(type, grep("Hair|Insurance", names(b.rest))))
t.rest <- subset(tip, business_id %in% b.rest$business_id, select=-c(type))
r.rest <- subset(review, business_id %in% b.rest$business_id, select=-c(type))
names(b.rest) <- make.names(names(b.rest)) # make R-compatible colnames
b.rest <- b.rest %>% mutate(stars=as.factor(stars)) # Transform stars into character
```

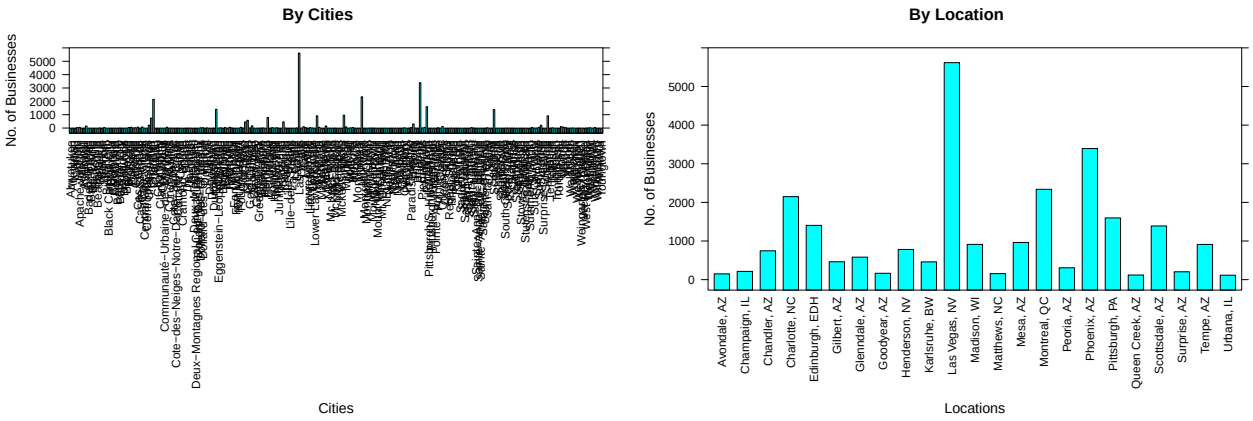
As inspired by the approach to factorization taken in (Gupta, Singh and Singh), all attributes are transformed into boolean values (T/F). Columns with free-text values are also flattened into additional factor columns ("Wi-Fi" into Wifi.paid","Wifi.free","Wifi.none"). Hours are condensed into seven T/F columns for each day of the week, marked T/F (open/close on the day). Attributes with higher than 90% of NAs are also removed, since these will not contribute much to the classification task.

The text in reviews is also grouped according to how many stars the reviewer accorded the business along with their reviews, which can the provide a baseline sentiment to use for our classification task.

2.2. EDA Checking the distribution of restaurants brings us into an issue with our research question; inconsistency in numbers of restaurants across states/cities within the dataset may derail attempts at classification. Indeed, some states are represented by only one entry.



We arrive at the following candidate cities for location-specific restaurants after restricting to states with having at least the median number of businesses amongst states, and at least the mean number amongst cities. Plotting with `lattice` yields:



Our plots show the large variance in numbers; some states/cities are actually represented by only one business, let alone individual neighbourhoods. Nevertheless, we proceed with the analysis by restricting to city-level and assume different parts of the city as synonymous to the city itself. We attempt to predict star ratings viz their business attributes and days open for the following, denoted as `B`:

```
b.loc <- b.loc[b.loc$city %in% vec.city,]
B <- select(b.rest, c(business_id, state, city))
B <- cbind(B, b.att, b.open) # combine with attributes, opening days
B <- B[B$business_id %in% b.loc$business_id,]
```

2.3. Establishing an error estimate baseline by classifying attributes/opening days Despite numerous NAs in the data, we proceed and split into train (60%), test (20%) and validation sets (20%) using a custom-defined function `createSets`. For fitting, we use a classification tree (`rpart`) and a boosted regression tree (`gbm`). An error estimate is calculated from these models.

```
set.seed(350); data <- B[,4:length(B)]; sets <- createSets(data, 0.6)
trainSet <- sets[[1]]; testSet <- sets[[2]]; validationSet <- sets[[3]]
```

```
## RPart
modRpart <- train(as.factor(stars)~., method="rpart",
                  control=rpart.control(xval=5, minsplit=15, minbucket=5, cp=0.01, maxdepth=15),
                  na.action=na.pass, data=trainSet)
## GBM with 3-fold CV
fitControl <- trainControl(method = "repeatedcv", number = 3, repeats = 3)
modGBM <- train(stars~., method="gbm", data=trainSet, na.action=na.pass,
                trControl=fitControl, verbose=FALSE)
```

2.4. Sentiment scoring review text Using the positive and negative word lists combined from A.Finn(2011) and Hu,Liu (2004), we grade the sentiment for restaurant reviews and generate a score capturing the nominal sentiment per star grouping. Review text is transformed with `tm` into lower-case with newlines, stopwords, punctuation and numbers removed, but opt to user a custom function instead of a corpus from `tm`:

```
# Returns score of +/- words from vector "txt"
# using + and - word vectors "pos" and "neg"
library(tm)
scoreReviewText <- function(txt, pos, neg) {
    t <- tolower(txt); t <- removeWords(t, words = stopwords("en"))
    t <- gsub("\\.", " ", t); t <- gsub("\\\n", "", t)
    t <- removePunctuation(t); t <- removeNumbers(t)
    s <- lapply(strsplit(t, " ", fixed=TRUE), function(x) ifelse(!duplicated(x), x, ""))
    sapply(s, function(x) sum(pos %in% x)-sum(neg %in% x))
}
```

3. Results

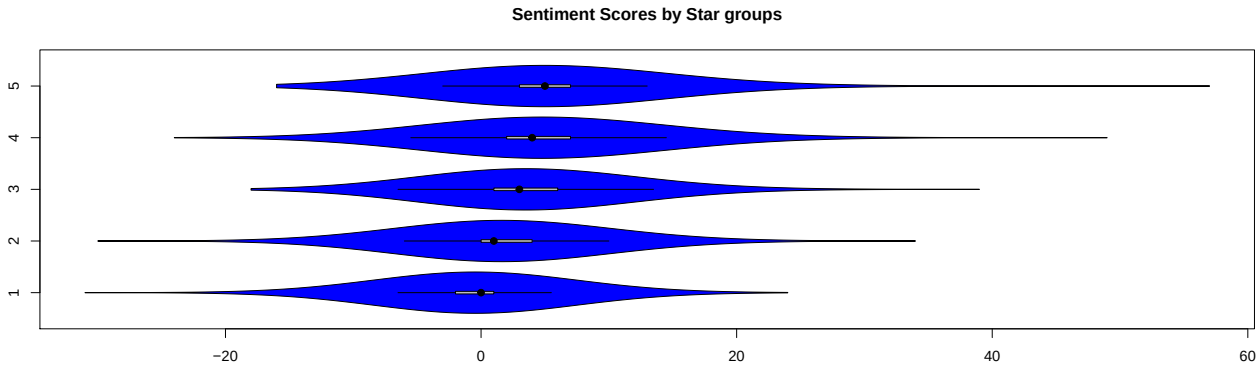
Classifying using attributes and opening days, the classification error estimates (calculated with a user-defined function `metrics(model, newdata)`) from both methods indicate clearly that a coin-flip works better. Our results from omitting opening days also seem to indicate opening days hardly matter either in influencing the outcome, at least based on the error estimates:

```
modE <- rbind(Rpart=metrics(modRpart, testSet)[[3]],
                  GBM=metrics(modGBM, testSet)[[3]])
modEAtt <- rbind(Rpart=metrics(modRpartAtt, testSet)[[3]],
                     GBM=metrics(modGBMAtt, testSet)[[3]])
modE <- cbind(modE, modEAtt)
colnames(modE) <- c("ErrEst(%).all", "ErrEst(%).attributesOnly") ; modE
```

```
##       ErrEst(%).all ErrEst(%).attributesOnly
## Rpart          71.5                     70.6
## GBM            69.5                     68.2
```

Plotting the results indicates that reviews do have a tendency to match the ratings given, although the variability is rather large. This suggests the word lists used may be insufficient to cover the breadth of sentiment, but says nothing about whether there are many shill reviews within the lot.

```
library(vioplot)
vioplot::vioplot(score1s, score2s, score3s, score4s, score5s, col="blue", horizontal=TRUE,
                 rectCol="gray", colMed="black")
title("Sentiment Scores by Star groups")
```



4. Discussion - Results and Follow-ups

The results from the attempt at classification was rather dismal, characterised by high error estimates. Our classification task was hampered by large numbers of NAs within the attributes. The fragmented nature of the data was ill-suited to the question, or rather, vice versa. Insufficient data was usable to train any model based simply on a combination of characteristics. The disappointing conclusion may be that classification by attribute is the wrong approach to use in this case.

On a positive note, sentiment scores for review text are distinctly banded, implying text is more valuable in this dataset than business characteristics, which have little use for prediction tasks.

```
##   stars Min. 1st Qu. Median    Mean 3rd Qu. Max.
## 1     1  -31      -2      0 -0.4275       1   24
## 2     2  -30       0      1  1.7730       4   34
## 3     3  -18       1      3  3.8210       6   39
## 4     4  -24       2      4  5.1930       7   49
## 5     5  -16       3      5  5.4410       7   57
```

Perhaps restaurants already know that beyond simple local discovery, attributes and taking time to enhance the listing is not nearly as important as concentrating on their core function; GOOD FOOD. The result is reflected in the review texts; good food draws effusive reviews and higher ratings that tend to accumulate over time.

Generalising the approach to non-food related businesses may not help us find a good model of attributes either. Approaches which focus on the review text itself should work better than business characteristics; after all, the reviews are continuously updated with newer reviews/ratings and grows, while business attributes are inherently more static. Given that there sentiment scores look like they fall into bands, albeit overlapping, making use of reviews upvoted by users may be helpful in providing a scaling factor to use, or simply to predict ratings.

More interesting follow-ups would include investigation into applying further NLP techniques on the text data, along with additional qualitative factorization methods on the user-contributed corpus. Perhaps an investigation into multiple ngram tokenization of phrases will provide better quality sentiment scores, or provide a proxy voting mechanism as additional explanatory variables, rather than just one-word sentiment alone.

5. Project Repo and References

- All files and full code used are available from the Github Project Repository
- Yelp Dataset used
- Collective Factorization for Relational Data: An Evaluation on the Yelp Datasets Nitish Gupta, Indian Institute of Technology, Kanpur and Sameer Singh, University of Washington.
- Finn Arup Nielsen, "A new ANEW: evaluation of a word list for sentiment analysis in microblogs", Proceedings of the ESWC2011 Workshop on 'Making Sense of Microposts': Big things come in small packages. Volume 718 in CEUR Workshop Proceedings: 93-98. 2011 May. Matthew Rowe, Milan Stankovic, Aba-Sah Dadzie, Mariann Hardey (editors). Links: Paper and Word list
- Minqing Hu and Bing Liu. "Mining and Summarizing Customer Reviews." Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD-2004), Aug 22-25, 2004, Seattle, Washington, USA. Links: Paper, Page, Word list