Capstone Project

Albion Dervishi

November 19, 2015

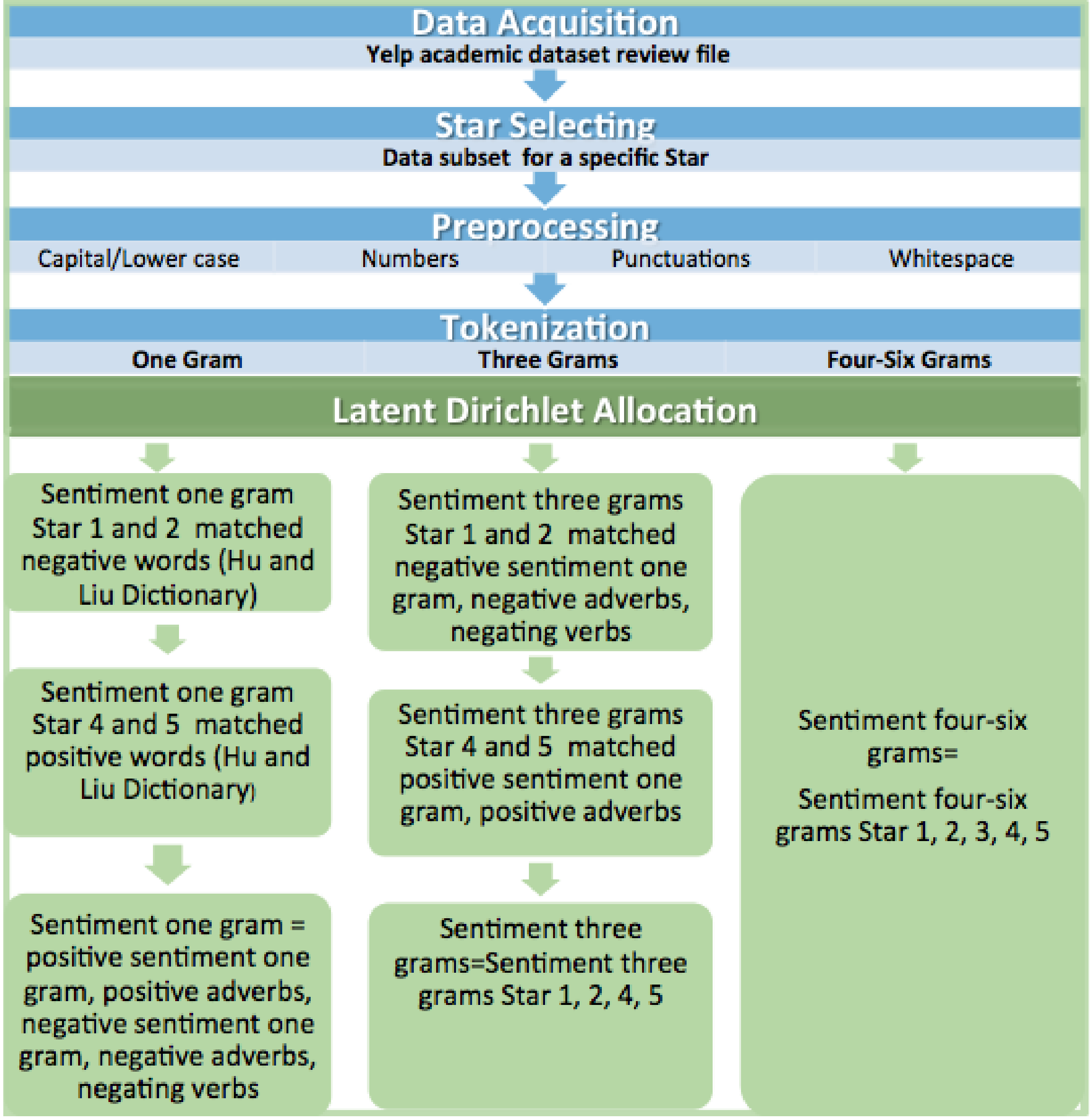
Introduction

This is final product of the Coursera Data Science capstone project for the fall 2015 session. The goal of the project is to build a model for prediction star review from its text alone, “Write your tip, we rate for you”.

Consumer’s reviews from web pages and social medias have tremendous impact for the business success also in the performance of the employee work effectiveness. Customers look for a complete and satisfactory experience regarding their performance, quality of service, and many other features. Customers reviews content are more informative than star review although star rating is a valuable way for quick business review.

In this project we implemented NLP (Natural language processing) techniques, Latent Dirichlet Allocation for topic modeling, and external resources to build an algorithm in R environment for star rate prediction. The algorithm will train a 1.6 million reviews and will be able to make predictions of star review based from its text alone

Main steps of “prediction star review from its text alone” Yelp dataset challenge/capstone project.

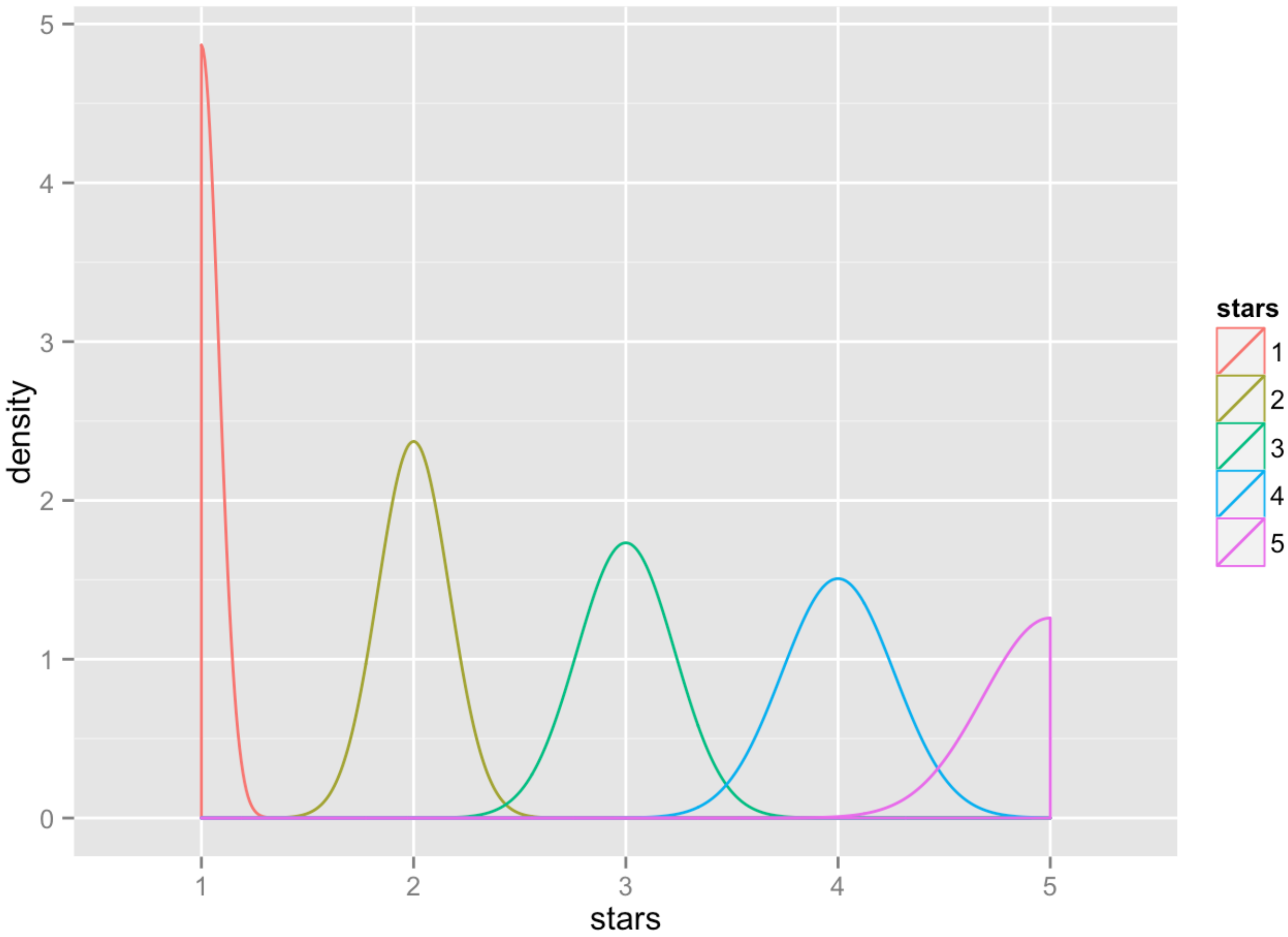

Data Acquisition
Yelp academic dataset review file
Star Selecting
Data subset for a specific Star
Preprocessing
Capital/Lower case
Numbers
Punctuations
Whitespace
Tokenization
One Gram
Three Grams
Four-Six Grams
Latent Dirichlet Allocation
Sentiment one gram Star 1 and 2 matched negative words (Hu and Liu Dictionary)
Sentiment one gram Star 4 and 5 matched positive words (Hu and Liu Dictionary)
Sentiment one gram = positive sentiment one gram, positive adverbs, negative sentiment one gram, negative adverbs, negating verbs
Sentiment three grams Star 1 and 2 matched negative sentiment one gram, negative adverbs, negating verbs
Sentiment three grams Star 4 and 5 matched positive sentiment one gram, positive adverbs
Sentiment three grams=Sentiment three grams Star 1, 2, 4, 5
Sentiment four-six grams=
Sentiment four-six grams Star 1, 2, 3, 4, 5

Methods and Data

The dataset is part of the Yelp Dataset Challenge and the specific dataset used in this project which corresponds to Round 6 of their challenge. The dataset consists of a set of JSON files that include business information, reviews, tips, user information etc. We have extracted data from yelp academic dataset review file and includes field in review text, star review and review id.

```
## 'data.frame':    1569264 obs. of  8 variables:
##  $ votes      :'data.frame': 1569264 obs. of  3 variables:
##   ..$ funny : int  0 0 0 0 0 0 0 0 0 0 ...
##   ..$ useful: int  2 2 1 0 2 0 0 0 1 0 ...
##   ..$ cool  : int  1 0 1 0 1 0 0 0 0 0 ...
##  $ user_id    : chr  "Xqd0DzHaiyRqVH3WRG7hzg" "H1kH6QZV7Le4zqTRNxoZow" "zvJCcrp
m2yOZrxKffwGQLA" "KBLW4wJA_fwoWmMhiHRVOA" ...
##  $ review_id  : chr  "15SdjuK7DmYqUAj6rjGowg" "RF6UnRTtG7tWMcrO2GEoAg" "-TsVN23
0RCkLYKBeLsuz7A" "dNocEAyUucjT371NNND41Q" ...
##  $ stars      : int  5 2 4 4 4 1 5 5 1 5 ...
##  $ date       : chr  "2007-05-17" "2010-03-22" "2012-02-14" "2012-03-02" ...
##  $ text       : chr  "dr. goldberg offers everything i look for in a general pr
actitioner.  he's nice and easy to talk to without being patronizing; "| __truncat
ed__ "Unfortunately, the frustration of being Dr. Goldberg's patient is a repeat o
f the experience I've had with so many other doctor"| __truncated__ "Dr. Goldberg
has been my doctor for years and I like him.  I've found his office to be fairly e
fficient.  Today I actually got "| __truncated__ "Been going to Dr. Goldberg for o
ver 10 years. I think I was one of his 1st patients when he started at MHMG. He's
been great ov"| __truncated__ ...
##  $ type       : chr  "review" "review" "review" "review" ...
##  $ business_id: chr  "vcNAWiLM4dR7D2nwwJ7nCA" "vcNAWiLM4dR7D2nwwJ7nCA" "vcNAWiL
M4dR7D2nwwJ7nCA" "vcNAWiLM4dR7D2nwwJ7nCA" ...
```

This plot let us to distinguish the stars density in the general data.



Latent Dirichlet Allocation

Once the text was cleaned up, DocumentTermMatrix was generated by applying 1 gram, 3 grams and 4-6 grams tokenization on the review dataset, and removing all the sparse terms. We run a Latent Dirichlet Allocation (LDA) using the document-term frequencies matrix of star review language model as input. We obtain most frequent words used for each topic in specific star review. In general, a LDA topic model discerns topics within a relied text review.

The topics are β1:K, where each β_k is a dispersion over the vocabulary The topic percentages for the d theta document are θ_d, where $\theta_{d,k}$ is the topic proportion for topic k in document d. The topic assignments for the d theta document are z_d, where $z_{d,n}$ is the topic assignment for the n theta word in document d. Result observed words for document d are w_d, where $w_{w,n}$, n is the n theta word in document d, which is an element from the fixed vocabulary.

$$p(\beta_{1:K}, \theta_{1:D}, z_{1:D}, w_{1:D}) \\ = \prod_{i=1}^{K} p(\beta_i) \prod_{d=1}^{D} p(\theta_d) \left(\prod_{n=1}^{N} p(z_{d,n} \mid \theta_d) p(w_{d,n} \mid \beta_{1:K}, z_{d,n}) \right).$$

Rating system

Data were trained in specifc words-star for each star data group: 1 gram, 3 grams and 4-6 grams.

Data model sentiment 1 gram

During our exploratory analysis we have distinguished that, there is important difference of containing positive words between 1:2 stars and 4:5 stars. In this project we applied selection of the negative words for star 1 and 2 and positive words for star 4 and 5. Dictionary provided by Hu and Liu (2004) and Liu et al. (2005) were used for the positive and negative words selection. In addition to this, were enhanced number of words by adding negative and positive adverbs, negative adverbs rated manually between 1 and 2 vice versa positive adverbs rated between 4 and 5.

In rating system special attention were paid in the negating words, it was hard to get good results with natural words-star extracted rates. We created a list with negating words and contractions and rated -3 as exclusion from natural rates 1:5.

```
##          Word Freq Rates    Word.1 Freq.1 Rates.1
## 1      is not    1    -3     isn't      1      -3
## 2     are not    1    -3    aren't      1      -3
## 3     was not    1    -3    wasn't      1      -3
## 4    were not    1    -3   weren't      1      -3
## 5    have not    1    -3   haven't      1      -3
## 6     has not    1    -3    hasn't      1      -3
## 7     had not    1    -3    hadn't      1      -3
## 8    will not    1    -3     won't      1      -3
## 9   would not    1    -3  wouldn't      1      -3
## 10     do not    1    -3     don't      1      -3
## 11   does not    1    -3   doesn't      1      -3
## 12    did not    1    -3    didn't      1      -3
## 13     cannot    1    -3     can't      1      -3
## 14  could not    1    -3  couldn't      1      -3
## 15 should not    1    -3 shouldn't      1      -3
## 16  might not    1    -3  mightn't      1      -3
## 17   must not    1    -3   mustn't      1      -3
```

Data model sentiment 3 grams

Formation of sentiment 3 grams data model we realized with collection of vector sentiment 3 grams words were matched with sentiment 1 gram words for a specific star. For example: sentiment 3 grams star 4 words relied on matching words of the sentiment 1 gram star 4.

Data model sentiment 4-6 grams

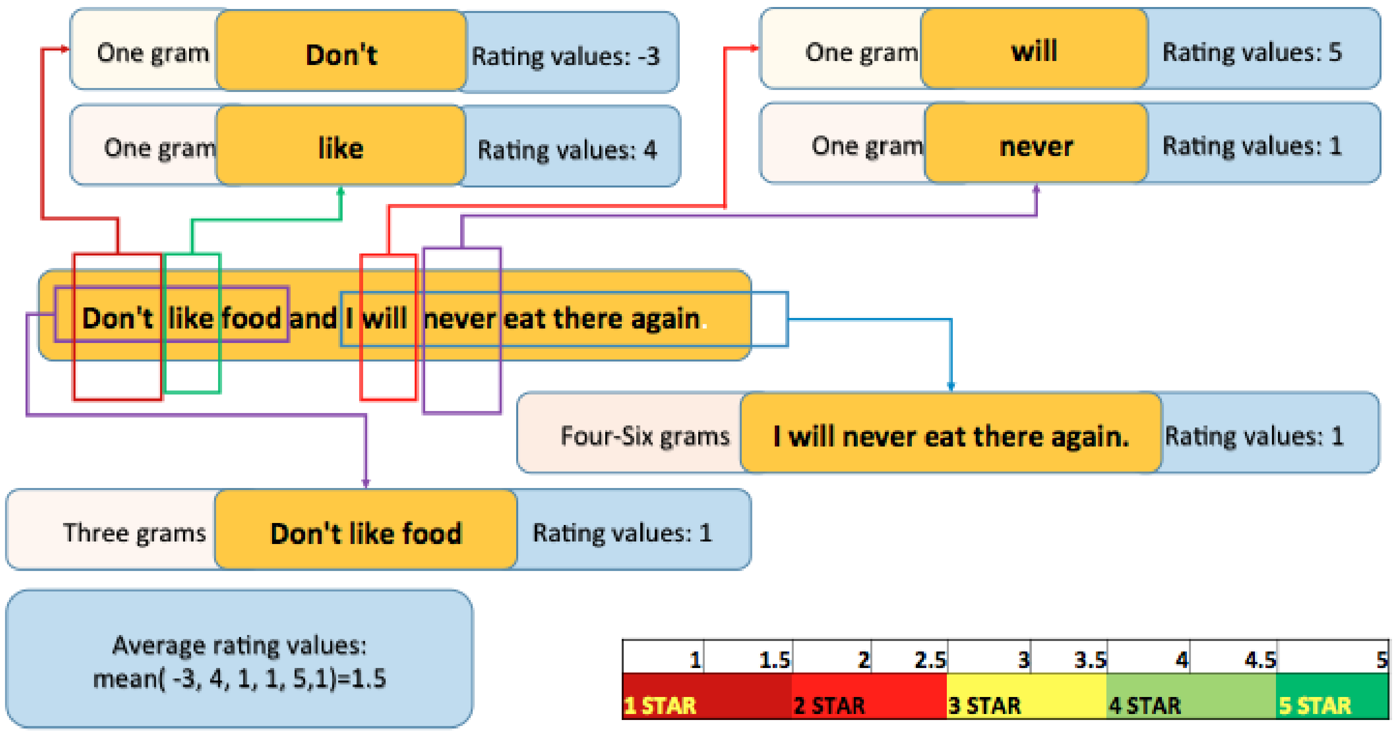
Generation of this group data derives direct from LDA topic model selection and contains sentiment words vector from all stars review.

Algorithm / Application

In this project we also developed online Shiny application with user-friendly interface. For prediction star review from its text alone we implement 1 gram, 3 grams, 5-6 grams algorithms.

The main tasks in the Shiny app:

1. Detecting the 1, 3 and 4-6 words of users text input.
2. Each detected words have rating system based on extracted star.
3. Calculating mean of the detected words rates, which it corresponds with star rate.



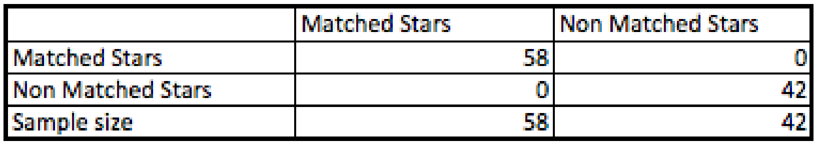
Results

Testing our application.

Does this application predict accurate? We took 100-testing sample of review texts from different resources. Each review text has a specific star assigned, copy review text and paste in our application and receiving the predicted star. We calculate the matching rate between real stars and test stars.The matched are assigned with 1 and non-matched with 0.

```
##   Real.Star Test.Star matching
## 1         5         5        1
## 2         2         2        1
## 3         4         4        1
## 4         4         3        0
## 5         4         4        1
## 6         1         2        0
```

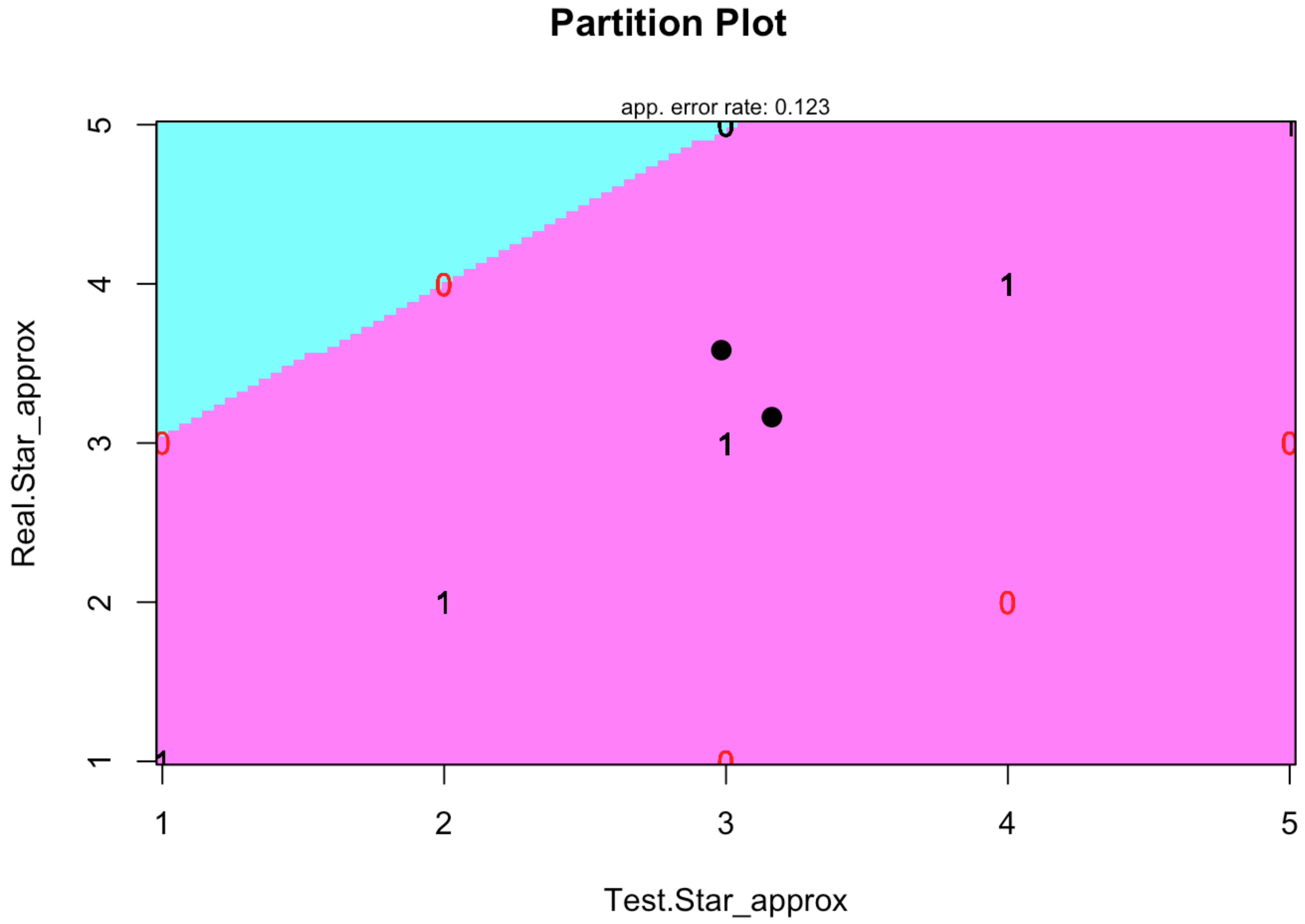
Predicted accuracy in this sample for matched predicted test star and real star is: 58%.



	Matched Stars	Non Matched Stars
Matched Stars	58	0
Non Matched Stars	0	42
Sample size	58	42

The app is designed to predict the star review based on the sentiment positive or negative words. So, to increase accuracy of the star prediction we created a function that allows us to approximate +1, -1 of the real star review. For example: Real star is 5, predicted star should be 5 or 4 but not 1, 2 and 3…

Predicted accuracy in this sample for approximation test star and real star is: 88%. Visualization of observations based on classification method Linear Discriminant Analysis we used function partimat{klaR}. Important feature of this package is classification borders are displayed and the apparent error rates are given in each title.



Conclusion

We think this topic is still very interesting subject and very challenging work. Results does not apply only in business, but also in many other fields such as healthcare, data mining, Neuro-linguistic programming (NLP), neurobiology etc. Accuracy in our application for star prediction based on its text alone is not as accurate as we want to be. Reasons for this are in the both sides, one side is that customer did not use proper words for star rate another side is application's algorithm. We have worked very hard and spent enormous time in calculation and came out with some important outcomes, but we are staying keen in improving this application still.