Yelp Dataset - Proposal of modifying rating system

1. Indroduction:

Yelp is having a million of review dataset of restaurants. Sometimes when we search for the restaurant we will get same starts for more than on restaurants, which leads to confusion for the customer. Goal of the analysis is to bring new rating syatem which give more accurate systems.

First method is create new rating which gives more weight to those who have reviewed more restaurants of the same cuisine. Let take we have 12 restaurants which is showing the same rating and some has visited and reviewed all the 12 restaurants, then their opinion should be given significantly more weight. Even someone who has been visited 3 out of 12 should be given more weight than someone who has just attended one.

2. Required Environment

```
library(rjson)
library(jsonlite)
library(data.table)
library(dplyr)
library(knitr)
```

3. Yelp Dataset

The main file “review” consist of text and star rating of each user review. The user business are identified through a unique user_id and business_id. The date of the review is also included.
More details on the each user can be found in user file. The actual dataset is in JSON.

```
setwd("/Users/anushreeshivarudrappa/Desktop/Yelp/yelp_dataset_challenge_academic_dataset")

yelp_review <- fread("yelp_academic_dataset_review.csv")
head(yelp_review)
yelp_review$V1 <- NULL

yelp_user <- fread("yelp_academic_dataset_user.csv")
yelp_user$V1 <- NULL

yelp_business <- stream_in(file
("/Users/anushreeshivarudrappa/Desktop/Yelp/yelp_dataset_challenge_academic_dataset/yelp_academic_datase
yelp_new <- yelp_business[,c("business_id","city","name", "categories", "review_count",
                             "stars")]
categories <- yelp_new$categories
yelp_business <- fread("yelp_academic_dataset_business.csv")
yelp_business$V1 <- NULL
yelp_business$categories <- categories
yelp_business$stars <- as.integer(yelp_business$stars)
names(yelp_business) <- c("business_id", "city", "business_name", "categories",
                          "review_count","Avg_stars")
```

** JOIN the dataframe **

```
y <- merge(yelp_review,yelp_user,by.x = "user_id", by.y = "user_id")
yelp <- merge(y,yelp_business, by.x = "business_id", by.y = "business_id")
# After join process "--" is added to user_id and business_id so removing it
yelp$user_id <- sub("--", "", yelp$user_id)
yelp$business_id <- sub("--", "", yelp$business_id)
```

4. Analysis Method: Giving more weight to multiple reviewers of a cuisine.

Analyze the dataset first to find multiple reviews there for cuisine. If there are very few the adding weight to their opinions may ultimately have little impact on overall rating.

A. Lets first look at Indian cuisine.

Add as "is_indian" column to the table based on whether the word "Indian" appear in "categories"

```
# Add "is_indian" field for any review that has "Indian" in "categories"
yelp$is_indian <- grepl("Indian", yelp$categories)

# filter data frame with Indian restaurants
yelp_Indian <- yelp[yelp$is_indian == T]
# After above join process "-" is added to user_id and business_id so removing it
yelp_Indian$business_id <- sub("-", "", yelp_Indian$business_id)
yelp_Indian$user_id <- sub("-", "", yelp_Indian$user_id)
head(yelp_Indian)
```

```
##              business_id                user_id stars         name
## 1: XUMQ8i1DFLahHSfbev10A  VYJMsseTmBBKyLM0j-YSg     5         Erik
## 2: XUMQ8i1DFLahHSfbev10A  1k0Qp2lGLvylQTYX_IgOw     2       Vikram
## 3: XUMQ8i1DFLahHSfbev10A D5G8KP_W0STCrdBwY4PtSQ     5           Lu
## 4: XUMQ8i1DFLahHSfbev10A Eugjl8_d69EwWT8X84UduQ     4           SM
## 5: XUMQ8i1DFLahHSfbev10A FihTWq8q5EU32Oc4vbh3fw     5 WhiteFeather
## 6: XUMQ8i1DFLahHSfbev10A  JMbiCAlDGPEUdP_l_Il1g     5      Natalie
##        city         business_name         categories review_count
## 1: Montréal Restaurant Tibetan Om Indian,Restaurants           14
## 2: Montréal Restaurant Tibetan Om Indian,Restaurants           14
## 3: Montréal Restaurant Tibetan Om Indian,Restaurants           14
## 4: Montréal Restaurant Tibetan Om Indian,Restaurants           14
## 5: Montréal Restaurant Tibetan Om Indian,Restaurants           14
## 6: Montréal Restaurant Tibetan Om Indian,Restaurants           14
##    Avg_stars is_indian
## 1:         4      TRUE
## 2:         4      TRUE
## 3:         4      TRUE
## 4:         4      TRUE
## 5:         4      TRUE
## 6:         4      TRUE
```

** Generate a summary of number of reviews of that cuisine done by each reviewer** Use group__by and summaries commands from dplyr to create a table of # of reviews of Indian restaurants each user has done.

```
review_Indian_count <- yelp_Indian %>% group_by(user_id) %>% summarise(tot_rev = sum(is_indian))
```

** Print the table, show the total # of entries, and find the avg # of reviews per user**

```
table(review_Indian_count$tot_rev)
```

```
##
##     1     2     3     4     5     6     7     8     9    10    11    12
## 11321  1528   449   192    94    63    34    22    12    16     6     3
##    13    14    16    17    19    21    23    24    29    30    34
##     3     3     5     4     2     1     1     1     1     1     1
```

```
count(review_Indian_count)
```

```
## Source: local data table [1 x 1]
##
##       n
##   (int)
## 1 13763
```

```
mean(review_Indian_count$tot_rev)
```

```
## [1] 1.347962
```

This yield result of 13763 total reviews, with 11321 doing just one review, 1528 doing 2 review. more than 10% of users have done multiple review of indian cuisine. Will use these reviews to improve rating system.

B. Use similar method on different cuisine

CHINESE

```
yelp$is_Chinese <- grepl("Chinese", yelp$categories)
yelp_Chinese <- yelp[yelp$is_Chinese == T]
review_Chinese_count <- yelp_Chinese %>% group_by(user_id) %>% summarise(tot_rev = n())
table(review_Chinese_count$tot_rev)
```

```
##
##     1     2     3     4     5     6     7     8     9    10    11    12
## 36281  6085  1960   896   486   260   199   133    85    60    48    36
##    13    14    15    16    17    18    19    20    21    22    23    24
##    28    34    13    19     9    12     9    14     2     7     5     6
##    25    26    27    28    29    30    31    32    33    34    36    37
##     4     4     1     1     4     4     1     3     2     1     1     2
##    38    39    41    42    43    46    47    48    52    54
##     1     3     1     1     2     1     1     1     2     1
```

```r
count(review_Chinese_count)
```

```
## Source: local data table [1 x 1]
##
##       n
##   (int)
## 1 46729
```

```r
mean(review_Chinese_count$tot_rev)
```

```
## [1] 1.514177
```

MEXICAN

```r
yelp$is_Mexican <- grepl("Mexican", yelp$categories)
yelp_Mexican <- yelp[yelp$is_Mexican == T]
review_Mexican_count <- yelp_Mexican %>% group_by(user_id) %>% summarise(tot_rev = n())
table(review_Mexican_count$tot_rev)
```

```
##
##     1     2     3     4     5     6     7     8     9    10    11    12
## 58187 11343  4153  1930  1077   640   454   297   230   168   150   100
##    13    14    15    16    17    18    19    20    21    22    23    24
##    79    79    64    48    42    46    29    34    26    15     9    24
##    25    26    27    28    29    30    31    32    33    34    35    36
##    12     9    14     8    14     1    10     5     4     5     1     4
##    37    38    39    40    41    43    44    46    49    50    51    52
##     1     1     3     2     1     3     2     6     1     3     1     1
##    53    54    56    61    68    69    70    71    73    74    76    77
##     2     1     1     2     1     1     1     1     2     2     1     1
##    89   119   147
##     1     1     1
```

```r
count(review_Mexican_count)
```

```
## Source: local data table [1 x 1]
##
##       n
##   (int)
## 1 79355
```

```r
mean(review_Mexican_count$tot_rev)
```

```
## [1] 1.710957
```

ITALIAN

```r
yelp$is_Italian <- grepl("Italian", yelp$categories)
yelp_Italian <- yelp[yelp$is_Italian == T]
review_Italian_count <- yelp_Italian %>% group_by(user_id) %>% summarise(tot_rev = n())
table(review_Italian_count$tot_rev)
```

```
##
##     1     2     3     4     5     6     7     8     9    10    11    12
## 54714  9468  3289  1474   783   497   359   228   166   137    93    83
##    13    14    15    16    17    18    19    20    21    22    23    24
##    60    64    39    42    19    22    24     9    16     6     6    13
##    25    26    27    28    29    30    31    32    33    34    35    36
##     7     6     3     7     5     7     4     4     4     5     1     1
##    37    38    39    41    45    47    48    53    54    61    63    65
##     2     2     2     1     2     4     3     1     1     1     1     1
##    69    70    71    75    79    81    96
##     1     1     2     1     1     1     1
```

```
count(review_Italian_count)
```

```
## Source: local data table [1 x 1]
##
##       n
##   (int)
## 1 71694
```

```
mean(review_Italian_count$tot_rev)
```

```
## [1] 1.589017
```

GREEK

```
yelp$is_Greek <- grepl("Greek", yelp$categories)
yelp_Greek <- yelp[yelp$is_Greek == T]
review_Greek_count <- yelp_Greek %>% group_by(user_id) %>% summarise(tot_rev = n())
table(review_Greek_count$tot_rev)
```

```
##
##     1     2     3     4     5     6     7     8     9    10    11    12
## 12620  1592   437   148    80    40    21    18     6     8     5     4
##    13    14    15    16    18    20
##     3     2     2     1     1     2
```

```
count(review_Greek_count)
```

```
## Source: local data table [1 x 1]
##
##       n
##   (int)
## 1 14990
```

```
mean(review_Greek_count$tot_rev)
```

```
## [1] 1.27058
```

FRENCH

```r
yelp$is_French <- grepl("French", yelp$categories)
yelp_French <- yelp[yelp$is_French == T]
review_French_count <- yelp_French %>% group_by(user_id) %>% summarise(tot_rev = n())
table(review_French_count$tot_rev)
```

```
##
##     1     2     3     4     5     6     7     8     9    10    11    12
## 19135  2591   669   252   128    66    42    25    14    16    11     8
##    13    14    15    16    17    18    19    20    22    23    32    48
##     5     2     5     1     3     1     2     1     2     2     1     1
```

```r
count(review_French_count)
```

```
## Source: local data table [1 x 1]
##
##       n
##   (int)
## 1 22983
```

```r
mean(review_French_count$tot_rev)
```

```
## [1] 1.298612
```

THAI

```r
yelp$is_Thai <- grepl("Thai", yelp$categories)
yelp_Thai <- yelp[yelp$is_Thai == T]
review_Thai_count <- yelp_Thai %>% group_by(user_id) %>% summarise(tot_rev = n())
table(review_Thai_count$tot_rev)
```

```
##
##     1     2     3     4     5     6     7     8     9    10    11    12
## 23102  3347  1085   406   193   111    82    52    51    22    16    15
##    13    14    15    16    17    18    19    20    21    24    25    26
##     5    10     4     2     7     2     3     4     1     1     1     1
##    29    30    31    34    36    39    79
##     1     1     3     1     1     1     1
```

```r
count(review_Thai_count)
```

```
## Source: local data table [1 x 1]
##
##       n
##   (int)
## 1 28532
```

```r
mean(review_Thai_count$tot_rev)
```

```
## [1] 1.379364
```

MEDITERRANEAN

```
yelp$is_Mediterranean <- grepl("Mediterranean", yelp$categories)
yelp_Mediterranean <- yelp[yelp$is_Mediterranean == T]
review_Mediterranean_count <- yelp_Mediterranean %>% group_by(user_id) %>% summarise(tot_rev = n())
table(review_Mediterranean_count$tot_rev)
```

```
## 
##     1     2     3     4     5     6     7     8     9    10    11    12 
## 22671  3163   917   404   181   115    63    51    30    10    16    13 
##    13    14    15    16    17    18    19    20    21    22    24    25 
##     8     3     5     3     7     2     2     1     1     4     1     2 
##    28    44 
##     1     1
```

```
count(review_Mediterranean_count)
```

```
## Source: local data table [1 x 1]
## 
##       n
##   (int)
## 1 27675
```

```
mean(review_Mediterranean_count$tot_rev)
```

```
## [1] 1.345872
```

SPANISH

```
yelp$is_Spanish <- grepl("Spanish", yelp$categories)  | grepl("Tapas", yelp$categories)
yelp_Spanish <- yelp[yelp$is_Spanish == T]
review_Spanish_count <- yelp_Spanish %>% group_by(user_id) %>% summarise(tot_rev = n())
table(review_Spanish_count$tot_rev)
```

```
## 
##     1     2     3     4     5     6     7     8     9    10    11    12 
## 15005  1629   382   173    74    29    26    18     8    11     7     4 
##    13    14    15    16    17    19    21    22    32 
##     1     1     1     2     1     2     1     1     1
```

```
count(review_Spanish_count)
```

```
## Source: local data table [1 x 1]
## 
##       n
##   (int)
## 1 17377
```

```
mean(review_Spanish_count$tot_rev)
```

```
## [1] 1.236232
```

JAPANESE

```r
yelp$is_Japanese <- grepl("Japanese", yelp$categories)  | grepl("Sushi", yelp$categories)
yelp_Japanese <- yelp[yelp$is_Japanese == T]
review_Japanese_count <- yelp_Japanese %>% group_by(user_id) %>% summarise(tot_rev = n())
table(review_Japanese_count$tot_rev)
```

```
## 
##     1     2     3     4     5     6     7     8     9    10    11    12 
## 48490  8957  3154  1420   838   533   324   256   159   117    96    77 
##    13    14    15    16    17    18    19    20    21    22    23    24 
##    56    53    42    40    31    22    24    22    12    10    13    10 
##    25    26    27    28    29    30    31    32    33    34    35    36 
##     5     4     6     7     3     3     3     5     1     1     3     2 
##    37    38    39    40    41    42    43    44    45    46    47    52 
##     1     6     2     2     1     1     2     1     1     1     1     1 
##    53    56    57    58    59    62    67    87 
##     2     1     1     1     2     2     2     1
```

```r
count(review_Japanese_count)
```

```
## Source: local data table [1 x 1]
## 
##       n
##   (int)
## 1 64831
```

```r
mean(review_Japanese_count$tot_rev)
```

```
## [1] 1.638537
```

Table of results of differeny Cusine.

Cuisine	Total Reviews	# >1 Review	% > 1 Review	Max Reviews
Indian	13763	2442	18	34
Chinese	46729	9908	21	54
Mexican	79355	21168	26	147
Italian	71694	16980	23	96
Greek	14990	2370	15	20
French	22983	3848	16	48
Thai	28532	5430	19	79
Medit	27675	5004	18	44
Spanish	17377	2372	13	32
Japanese	64831	16341	25	87

5. Apply new Weight and see the effect

Combine num_reviews information with original data frame of indian restaurant reviews We can see that all the cuisines had at least 10% of reviewers giving multiple reviews.

Let modify the rating using these wights and seeing what impact they have. Let's try first on Idian restaurants. We have # of reviews for each user in "review_Indian_count" Lets going this back to yelp_Indian data frame containing all individual ratings, we have a new table which has rating the user gave as well as the # of Indian restaurants they have reviewed.

```
cob_in <- inner_join(yelp_Indian, review_Indian_count) # join by user_id
```

```
## Joining by: "user_id"
```

Generate "weighted_stars" for later calculation

```
cob_in$Weighted_Star <- cob_in$stars * cob_in$tot_rev
```

Use "summarise" to generate a new rating for each restaurant

```
cal1 <- cob_in %>%  group_by(city, business_name, Avg_stars) %>%
                      summarise(count = n(),
                      new = sum(Weighted_Star) / sum(tot_rev))

cal2 <-  cob_in %>%  group_by(city, business_name, Avg_stars) %>%
                        summarise(sumOfStars = sum(stars))

new_rating_Indian <- inner_join(cal1,cal2)
```

```
## Joining by: c("city", "business_name", "Avg_stars")
```

```
new_rating_Indian$old <- new_rating_Indian$sumOfStars / new_rating_Indian$count
new_rating_Indian$diff <- new_rating_Indian$new - new_rating_Indian$old
```

Print summary data of the effect this new rating has

```
summary(new_rating_Indian$diff)
```

```
##     Min.  1st Qu.   Median     Mean  3rd Qu.     Max.
## -1.36400 -0.25500 -0.06966 -0.11070  0.04940  0.85000
```

We see that new weights can move the rating down by as many as 1.37 stars or up as high as 0.85 stars.

Limit to those with at least 5 ratings and redo summary

```
nri5 <- subset(new_rating_Indian, count > 5)
summary(nri5$newStar)
```

```
## Length  Class   Mode
##      0   NULL   NULL
```

We can see that the impact increases of unto 1.37 starts and decrease of as much as 0.8

6. Look at new and old ratings.

Checking the rating for few restaurants.

```
new_rating_Indian <- as.data.frame(new_rating_Indian)
head(new_rating_Indian[, c("business_name", "old", "new")],10)
```

```
##             business_name      old      new
## 1            India Garden 4.286713 4.010490
## 2              Le Tandoor 3.285714 2.333333
## 3       Restaurant Mysore 4.000000 4.000000
## 4          Shaan Tandoori 4.000000 4.214286
## 5                  Fusion 3.300000 2.954545
## 6              Cafe Delhi 4.014706 3.572650
## 7          Indian Village 4.323529 4.268293
## 8             Ambar India 3.548387 3.480769
## 9  Basmati Indian Cuisine 3.000000 3.000000
## 10    Bombay Indian Grill 3.586207 3.517442
```

We can see when there was a tie, new rating will help user.

6. Future analysis:

Creating an "immigrant" rating. Lets take an example of Indian restaurants, lot of immigrant indian workers working there temporarily for various tech companies. On the theory that those workers would actively seek out restaurants that remains them closing of "home cooking" and also that they tend to seek out places offering the most value, one thing people might do is check the rating given by those with clearly Indian names to see what they think. The proposal would be to check the user name in Yelp to guess at who might be an "immigrant" and create different rating for particular ethnic cuisine given specifically by those users. This method admittedly has some clear deficiencies - it will ignore any "immigrants" who do not use their real names and it will also mark as "immigrants" those who simply like an Indian name and choose to use it for Yep ID. The theory is that there might be enough information that cuts through the noise of those deficiencies to be able to provide useful information.