Team 2 Group Project

**Names:**

Liz Stell​

Yu Sheng Lu​

Michael Chilton Zuber​

Thomas Szot

The Business Opening Dilemma Solved Using YELP Business and Review Data​

Business Questions:

* Where in Portland Oregon is the best spot to open our new business?​
* What kind of business should we open?​
* Based on reviews, what is something unique that we can do?​
* What are some important pieces of information we can gather from the reviews? From this information, how can we apply it to our new business?​

Our Data:

* Business Data:
  + We used this data to create a map that helped us decide about where to open our new business.​
  + Some of the fields included a unique business ID, business name and address, total number of reviews, and average rating for a business.​
  + We found a place in Portland with a high density of high YELP ratings​.
* Review Data
  + Yelp Business, Yelp Review, and Positive & Negative word list data were used in the word cloud section.
  + We cleaned up the data and set the city equal to Portland, Oregon. Then we used the merge function to merge with the yelp review by Business ID. This helped us get the businesses that are in the Portland, Oregon.
  + We then used the clean dataset to create a document term matrix for word cloud.
  + The word cloud is generated based on the most word used in the review.
  + Then we imported the positive and negative word list, then we match with our document term matrix that we created.
  + Finally, we formed a frequency list for a better view on the result.

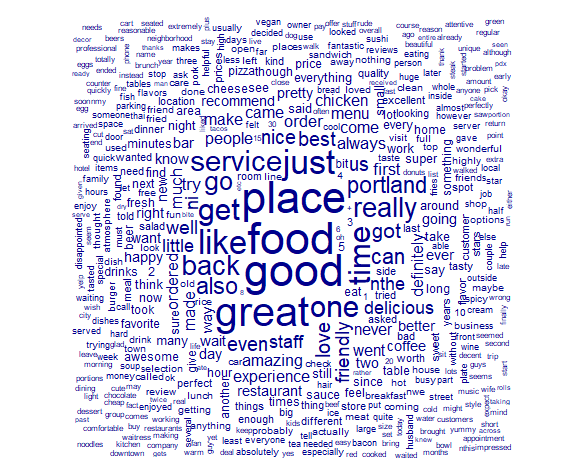
Our Business Solution Based on Data:

* Where are we starting our business based on the map?​
  + Based on the data from YELP, we have seen that the best reviews happen in the Alberta Park area of Portland OR. ​
  + We can see this because the light blue shading is visibly the best in that area on this map.​
  + Based on the data, this seems to be the best area to open a business in. Good reviews mean other good business and heavy foot traffic to attract customers for a new business.

Graphical user interface, chart

Description automatically generated

* What are we opening based on the feedback that was most received?
  + Based on the word “FOOD” we have decided to open a restaurant.
  + Based on the words “SERVICE”, “EXPERIENCE”, “FRIENDLY” we know that we must hire the best staff for our patrons. The better the staff, the better the reviews.
  + Based on the words “CHICKEN”, “CHEESE”, “EGG”, “BURGER”, “COFFEE” we are going to open a breakfast/brunch restaurant.



* From the YELP reviews what should we make sure we do for good reviews and what should we make sure we don’t do?
  + Based on what good words were used the most, it looks like we should stick to opening a restaurant. Based on the words “DELICIOUS”. It was used almost 84k times, clearly good food translates to good reviews.
  + Based on what bad words were used the most, good service will keep our good reviews up. We can see that because of the word “RUDE”. We must hire the best staff to keep up positive reviews.

Table

Description automatically generatedTable

Description automatically generated

Map Code:

library(readr)

library(dplyr)

# read cleaned csv and delete outlier row

yelpBusiness <- read\_csv("yelpBusiness.csv")

yelpOregon <- yelpBusiness[yelpBusiness$state == "OR",]

yelpOregon<- yelpOregon[-c(3315), ]

library(ggplot2)

library(maps)

library(openintro)

library(ggmap)

library(readxl)

# create a state map and filter just Oregon

us <- map\_data('state')

usOregon <- us[us$region == "oregon",]

yelpOregon$state <- gsub('OR', 'oregon', yelpOregon$state)

colnames(usOregon) <- c("long", "lat", "group", "order", "state", "subregion")

# merge csv and map dataset

dfOR <- dfUS <- merge(usOregon, yelpOregon, by = 'state')

# create base Oregon state map

map <- ggplot(dfOR) + geom\_polygon(color="black", fill="white", aes(x=long.x, y=lat.x)) + expand\_limits(x=dfOR$long.y, y=dfOR$lat.y) + coord\_map()

map

# add map with businesses as points

map1 <- map + geom\_point(aes(x=long.y, y=lat.y, size=review\_count, color=stars))

map1

# create county map and filter just Multnomah

county <- map\_data('county')

PortCounty <- county[county$subregion == "multnomah",]

colnames(PortCounty) <- c("long", "lat", "group", "order", "state", "subregion")

# merge csv and map dataset

dfPort <- dfUS <- merge(PortCounty, yelpOregon, by = 'state')

# create base county map

map2 <- ggplot(dfPort) + geom\_polygon(color="black", fill="white", aes(x=long.x, y=lat.x)) + expand\_limits(x=dfPort$long.y, y=dfPort$lat.y) + coord\_map()

# add map with businesses as points

map3 <- map + geom\_point(aes(x=long.y, y=lat.y, size=review\_count, color=stars))

map3

Word Cloud Code:

library(readr)

yelp\_academic\_dataset\_business <- read.csv("yelp\_academic\_dataset\_business.json", header=FALSE)

#import the business data set

library(dplyr)

#start cleaning the data set

yelpBusiness <- select(yelpBusiness, -c(12:41))

yelpBusiness <- rename(yelpBusiness, c(business\_id=X1, name=X2, address=X3, city=X4, state=X5, postal\_code=X6, lat=X7, long=X8, stars=X9, review\_count=X10, is\_open=X11))

yelpBusiness$business\_id <- gsub('"', '', yelpBusiness$business\_id)

yelpBusiness$business\_id <- gsub('{business\_id:', '', yelpBusiness$business\_id, fixed = TRUE)

yelpBusiness$name <- gsub('name:', '', yelpBusiness$name)

yelpBusiness$address <- gsub('address:', '', yelpBusiness$address)

yelpBusiness$city <- gsub('city:', '', yelpBusiness$city)

yelpBusiness$state <- gsub('state:', '', yelpBusiness$state)

yelpBusiness$postal\_code <- gsub('postal\_code:', '', yelpBusiness$postal\_code)

yelpBusiness$lat <- gsub('latitude:', '', yelpBusiness$lat)

yelpBusiness$long <- gsub('longitude:', '', yelpBusiness$long)

yelpBusiness$stars <- gsub('stars:', '', yelpBusiness$stars)

yelpBusiness$review\_count <- gsub('review\_count:', '', yelpBusiness$review\_count)

yelpBusiness$is\_open <- gsub('is\_open:', '', yelpBusiness$is\_open)

yelpBusiness$lat <- as.numeric(yelpBusiness$lat)

yelpBusiness$long <- as.numeric(yelpBusiness$long)

yelpBusiness$review\_count <- as.numeric(yelpBusiness$review\_count)

yelpBusiness$stars <- as.factor(yelpBusiness$stars)

yelpBusiness$state <- tolower(yelpBusiness$state)

yelpBusiness$city <- tolower(yelpBusiness$city)

yelpOregon <- yelpBusiness[yelpBusiness$state == "OR",]

yelpOregon$state <- gsub('OR', 'oregon', yelpOregon$state)

colSums(is.na(yelpBusiness))

#import the review data set

yelp\_academic\_dataset\_review <- read.csv("yelp\_academic\_dataset\_review.json", header=FALSE)

yelpReview <- yelp\_academic\_dataset\_review

yelpReview <- yelpReview[, -c(1:2, 4:7, 9)]

yelpReview <- rename(yelpReview, c( business\_id=V3, text=V8))

yelpReview$business\_id <- gsub('business\_id:', '', yelpReview$business\_id)

yelpReview$text <- gsub('text:', '', yelpReview$text)

#merge the review data set with business data set

yelpText <- merge(yelpReview, yelpOregon)

#filter For Portland

yelpText <- yelpText[yelpText$city == "Portland",]

library(quanteda)

#changing the reviews into document term matrix

dfCor <- corpus(yelpText$text)

dftok <- tokens(dfCor)

dfDFM <- dfm(dftok, remove\_punct=TRUE, remove=c(stopwords("english"), "n", "$"), removeNumbers = TRUE, tolower = TRUE)

dfDFM

library(quanteda.textplots)

#do word cloud on review data

textplot\_wordcloud(dfDFM, min\_termfeq = 10)

#import the positive and negative word list

posWords <- scan("https://intro-datascience.s3.us-east-2.amazonaws.com/positive-words.txt", character(0), sep = "\n")

negWords <- scan("https://intro-datascience.s3.us-east-2.amazonaws.com/negative-words.txt", character(0), sep = "\n")

posWords <- posWords[-c(1:34)]

negWords <- negWords[-c(1:34)]

#match the word list with the review from yelp data set

posDfm <- dfm\_match(dfDFM, posWords)

negDfm <- dfm\_match(dfDFM, negWords)

posDfm

negDfm

textplot\_wordcloud(posDfm, min\_termfeq = 2)

textplot\_wordcloud(negDfm, min\_termfeq = 2)

#sum up the frequency for the text used

dfFreq <- textstat\_frequency(dfDFM)

posFreq <- textstat\_frequency(posDfm)

negFreq <- textstat\_frequency(negDfm)

#frequency list for word cloud positive and negative

dfFreq[1:10,]

posFreq[1:10,]

negFreq[1:10,]