**Introduction**

The purpose of this project is to perform data analysis, to predict the review scores for individual pizza restaurants, to find out the relationship between restaurant locations and price ranges and to forecast the price levels of the restaurants. To figure out the answers to these questions can help pizza restaurants improve the customers’ satisfaction in order to have more business.

I used R to perform data analysis and visualization to explore and identify the characters in the three datasets, and uncover insights to understand how the location affects the price and customers reviews through the following steps:

1. Load Required Packages
2. Clean Up and Prepare Data for Analysis
3. Exploratory Data Analysis
4. Data Visualization
5. Summary of Findings

In the analysis, the main algorithms that I used are nonlinear multivariant regression for review score prediction and random forecast classification for price level prediction. The result of regression is the predicted value of review scores so that the restaurant can improve the service according to the critical predictors. The result of classification is the predicted group of price level the restaurant may belong to. It can help the restaurant re-adjust their prices.

**1. Packages Required**

**library(readr)***# used to load the datasets*

**library("corrplot")***# used to visualize correlation matrix*

**library('MASS')***# used for stepwise regression to remove variables*

**library('caret')***# used for cross validation methods*

**library(dismo)***# used to calculate the AUC values*

**library('sqldf')***# used to adjust the dataset via SQL*

**library(caTools)***# used to split the train-test dataset*

**library(e1071)***# used for SVM algorithm*

**2. Data Preparation - Source Data and Code Book**

The source data can be found at GitHub [https://github.com/rfordatascience /tidytuesday/tree/master/data/2019/2019-10-01](https://github.com/rfordatascience%20/tidytuesday/tree/master/data/2019/2019-10-01) in the form of csv files.

Jared's dataset is from top NY pizza restaurants, with a 6 point likers scale survey on ratings. The Barstool sports dataset has critic and public. And the Barstool Staff's rating as well as pricing, location, and geo-location. There are 22 pizza places that overlap between the two datasets. More information about geo-location of pizza places can be found from pizza\_datafiniti. This includes 10000 pizza places, their price ranges and geo-locations.

There are 463 observations and 22 variables in Barstool, 10k observations and 10 variables in datafiniti, 375 observations and 9 variables in Jared’s. Some of latitude and longitude of restaurants noted as NA or blanks are in Barstool datasets.

The following is the code used to evaluate the variables in the source data. From the information, we can learn that these datasets were collected on 2019-10-01.

library(tidyverse)

library(jsonlite)

# Get barstool data off github

pizza\_raw <- read\_csv("https://raw.githubusercontent.com/tylerjrichards/Barstool\_Pizza/master/pizza\_data.csv")

pizza\_cooked <- pizza\_raw %>%

select(name, address1, city, zip, country, latitude, longitude, priceLevel,

providerRating, providerReviewCount,

reviewStats.all.averageScore:reviewStats.dave.totalScore) %>%

janitor::clean\_names()

# Get jared data off his website (json)

url <- "https://jaredlander.com/data/PizzaPollData.php"

jared\_pizza <- fromJSON(readLines(url), flatten = TRUE) %>%

as\_tibble() %>%

janitor::clean\_names()

write\_csv(jared\_pizza, here::here("2019", "2019-10-01", "pizza\_jared.csv"))

write\_csv(pizza\_cooked, here::here("2019", "2019-10-01", "pizza\_barstool.csv"))

The detail descriptions for each attribute in every dataset are as below:

pizza\_jared

| **variable** | **class** | **description** |
| --- | --- | --- |
| polla\_qid | integer | Quiz ID |
| answer | character | Answer (likert scale) |
| votes | integer | Number of votes for that question/answer combo |
| pollq\_id | integer | Poll Question ID |
| question | character | Question |
| place | character | Pizza Place |
| time | integer | Time of quiz |
| total\_votes | integer | Total number of votes for that pizza place |
| percent | double | Vote percent of total for that pizza place |

pizza\_barstool

| **variable** | **class** | **description** |
| --- | --- | --- |
| name | character | Pizza place name |
| address1 | character | Pizza place address |
| city | character | City |
| zip | double | Zip |
| country | character | Country |
| latitude | double | Latitude |
| longitude | double | Longitude |
| price\_level | double | Price rating (fewer $ = cheaper, more $$$ = expensive) |
| provider\_rating | double | Provider review score |
| provider\_review\_count | double | Provider review count |
| review\_stats\_all\_average\_score | double | Average Score |
| review\_stats\_all\_count | double | Count of all reviews |
| review\_stats\_all\_total\_score | double | Review total score |
| review\_stats\_community\_average\_score | double | Community average score |
| review\_stats\_community\_count | double | community review count |
| review\_stats\_community\_total\_score | double | community review total score |
| review\_stats\_critic\_average\_score | double | Critic average score |
| review\_stats\_critic\_count | double | Critic review count |
| review\_stats\_critic\_total\_score | double | Critic total score |
| review\_stats\_dave\_average\_score | double | Dave (Barstool) average score |
| review\_stats\_dave\_count | double | Dave review count |
| review\_stats\_dave\_total\_score | double | Dave total score |

pizza\_datafiniti

| **variable** | **class** | **description** |
| --- | --- | --- |
| name | character | Pizza place |
| address | character | Address |
| city | character | City |
| country | character | Country |
| province | character | State |
| latitude | double | Latitude |
| longitude | double | Longitude |
| categories | character | Restaurant category |
| price\_range\_min | double | Price range min |
| price\_range\_max | double | Price range max |

For Barstool data table, to clean the data, I first inspected the source data and Learned that there are missing values of some observations for the “position” fields (longitude and latitude). So for each of the invalid positions, I looked them up online and filled in the blanks in R. In addition, some of the average or total scores are 0 or missing. But it does not mean the restaurant is the worst. It is just because no customers visited or reviewed. Hence, I keep all these values unchanged.

For pizza\_datafiniti, I found that there are huge number of rows that are exactly the same. So to remove the redundant data lines is necessary. I used unique() function to drop those rows which duplicate the same information.

For Jared dataset, it deploys the percentage of people who vote among ‘Excellent’, ‘Good’, ‘Average’, ‘Poor’ and ‘Never Again’ for each restaurant. Since all the questions are ‘How was xxx?’ (xxx represent the specific restaurant), I deleted the redundant columns like ‘polla\_qid’, ‘pllq\_id’ and ‘question’ from the list.

As such, we used the following code to tidy up our data:

#######load datasets and remove redundant columns and rows#########

library(readr)

pizza\_barstool <- read\_csv("C:/Users/Henry/Desktop/pizza/pizza\_barstool.csv")

pizza1 <- subset(pizza\_barstool, select = -c(2,5)) #(10,14:22)

pizza\_datafiniti <- read\_csv("C:/Users/Henry/Desktop/pizza/pizza\_datafiniti.csv")

pizza2 <- unique(pizza\_datafiniti[ ,1:10])

pizza2 <- subset(pizza2, select = -c(2,4) )

pizza\_jared <- read\_csv("C:/Users/Henry/Desktop/pizza/pizza\_jared.csv")

pizza3 <- subset(pizza\_jared, select = -c(1,4,5) )

#find and fill in missing values

sapply(pizza1, function(x) sum(is.na(x)))

which(is.na(pizza1$latitude))

pizza1[6,4]<-40.68060

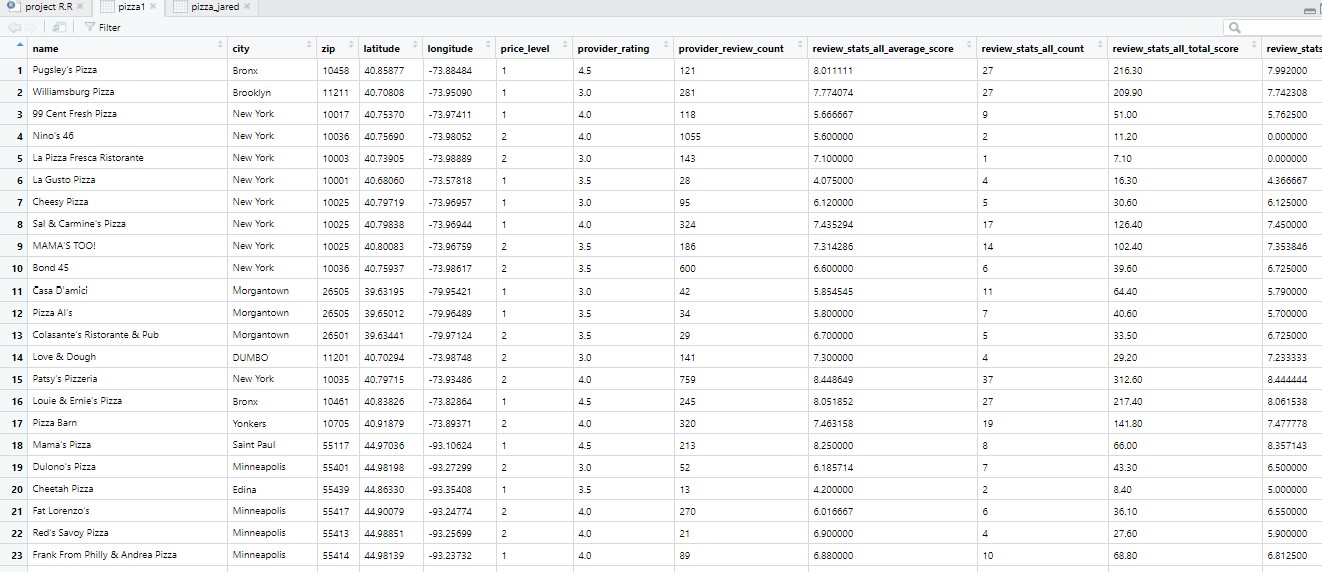
pizza1[6,5]<--73.57818

pizza1[266,4] <- 40.72277

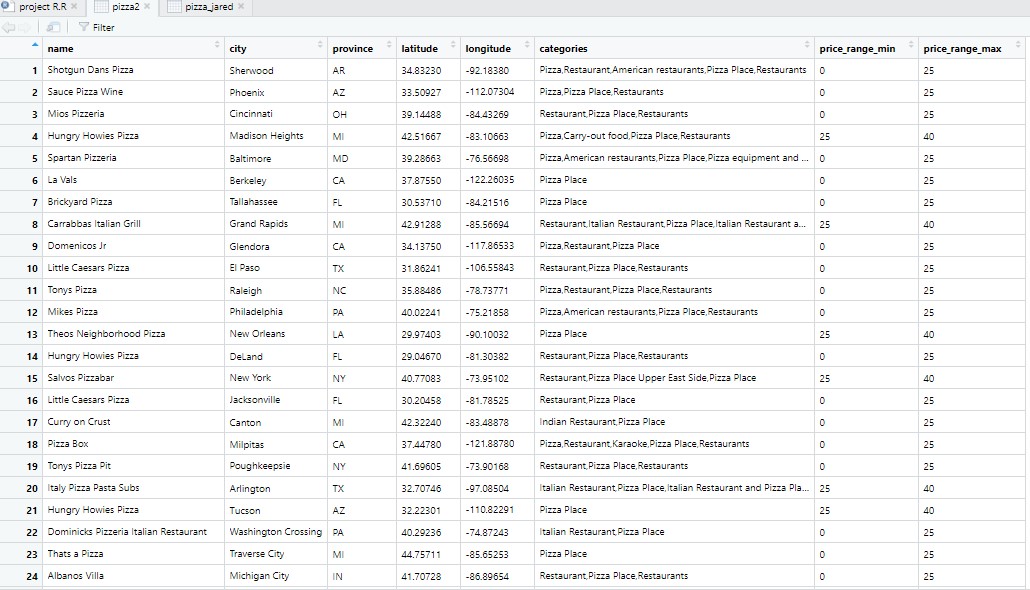
pizza1[266,5] <- -73.99620

Below is a brief snapshot of the cleaned data set:

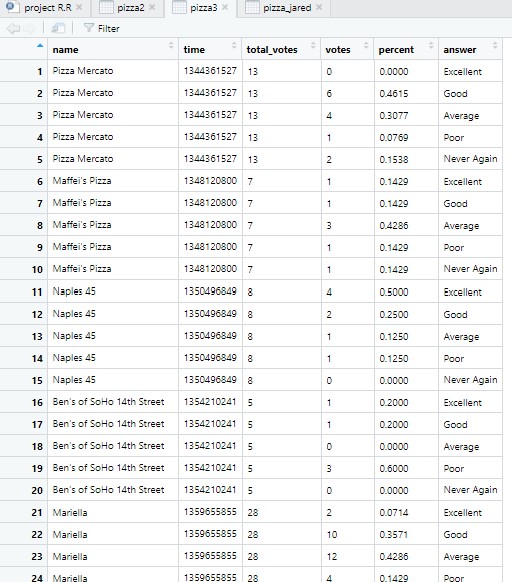
Pizza1



Pizza2



Pizza3



Based on the cleaned datasets, we performed preliminary analysis and obtained the summary statistics for each variable using the following code. Note, I chose not to display the entire output herein; instead I will summarize the results of the preliminary analysis below for each dataset.

*# Get summary statistics for each variable*

**summary**(pizza1)

*# Display structure of cleaned data set*

**str**(pizza1)

*# Average duration of each session*

**mean**(mean(pizza1$review\_stats\_all\_average\_score))

In details, for pizza1(barstool), the average review score of all restaurants shown as review\_stats\_all\_average\_score is 6.876 which means majority of restaurants are above average or people are likely to score higher according to a 0-10 scaling range. In addition, except for the name and city, all the other attributes are numbers. For pizza2(datafiniti), the average price range is between 4.67 to 27.8, slightly higher than normal level 0 to 25. As for pizza3(jared), the average of total votes is 14.16. That means, about 14 customers would like to evaluate the service for each restaurant online for others’ reference.

**3 and 4 - Exploratory Data Analysis and Visualization**

First I analyzed pizza1(barstool) by further adjusting the table. Some replicated information such as zip code and latitude/longitude can be removed since they have similar functions which might be highly correlative. Then I split the dataset into train/test groups by 0.55/0.45 at this step to avoid further test leakage. Actually at first the split rate was 0.75/0.25 but the trained fitting model is not good enough due to overfitting so I tuned the parameters with less training input. Additionally, there are still too many independent variables, so correlation matrix is employed for further feature removal.

#remove first two columns and lon/lat just keep zip code(correlative)

pizza1\_no2 <- subset(pizza1, select = -c(1,2,4,5,19)) #19 is ? in correlation matrix

#split train and test here to avoid test leakage

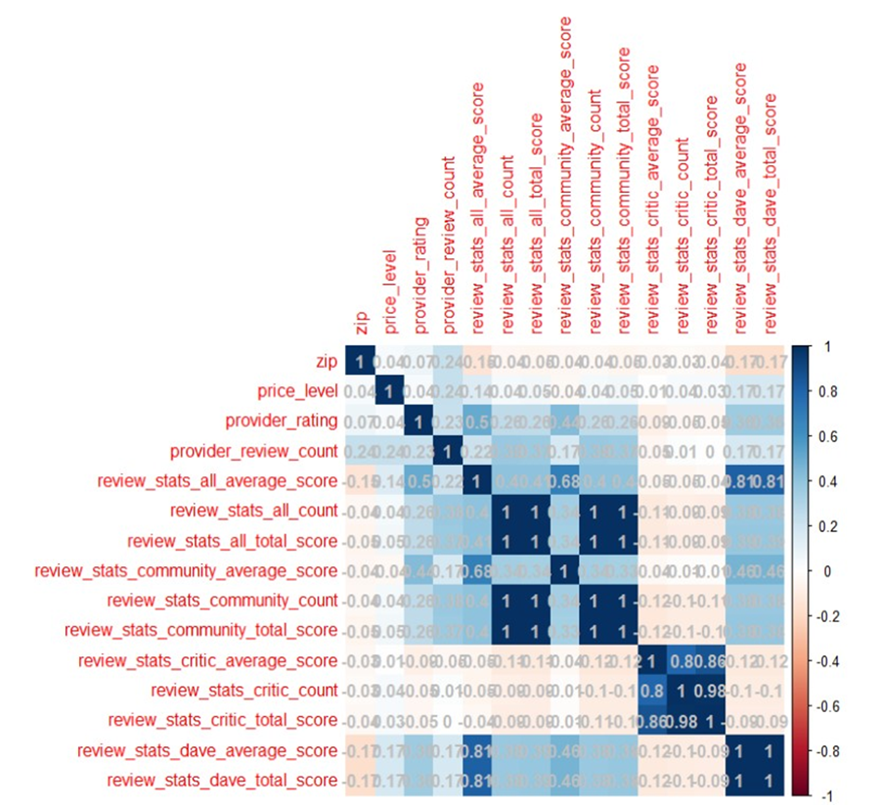
per=0.55

train<-sample(nrow(pizza1\_no2),per\*nrow(pizza1\_no2))

pizza1\_no2.train<-pizza1\_no2[train,]

pizza1\_no2.test<-pizza1\_no2[-train,]

In correlation matrix, the darker color refers to the higher correlation between any two attributes. That means one of them could be dropped and the other one could represent the dropped feature instead during the analysis. After that, an approach named stepwise regression can also be used to remove more attributes. It is an idea that when a new feature is removed or added, how far the dependent variable y is changed. We just need to keep those critical features that impact y the most. After the table is re-organized, there are only zip, price\_level, provider\_rating and review\_stats\_all\_total\_average as X which affect Y review\_stats\_all\_average\_score.



#filter out X for pizza1 using correlation matrix

library("corrplot")

b<-cor(pizza1\_no2.train)

corrplot(b,method="color",addCoef.col="grey")

#remove high correlative attributes

pizza1\_no2.train <- subset(pizza1\_no2.train, select = -c(6,8:10,12:15) )

pizza1\_no2.train <- pizza1\_no2.train[c(1:4,6,7,5)]

#remove more attributes with stepwise regression

library('MASS')

fit <- lm(review\_stats\_all\_average\_score~.,data=pizza1\_no2.train) #y~all others x

stepAIC(fit,direction='backward')

pizza1\_ftrain <- pizza1\_no2.train[c(1:3,5,7)]

The model could be fit after all the attributes are fixed. As it is a regression problem that we need to predict the scores based on all the input Xs. To consider using linear or nonlinear multivariant regression model is necessary. Moreover, after splitting the dataset, we just have 254 observations which are too few to fit an accurate model. So I used k fold cross validation method to increase the samples.

There are three options that I need to choose from to predict the trend and values. The first option is the normal linear regression model and its final accuracy of AUC is 71% (Area Under Curve represents the areas under ROC receiver Operating Characteristic Curve. The higher the better.). The second option is linear model with k fold method but the AUC accuracy is as low as 67% which is even worse. So this method needs to be abandoned. The third approach is nonlinear multivariant regression which is the best. Its AUC accuracy is about 76%. This model shows that the multiple independent variables might have nonlinear relationships with y, rather than a linear relationship. So I adopt option 3 as the optimal solution for the prediction. Furthermore, as I mentioned, although option 3 is the best, there were too many observations at first that led to overfitting. So I tuned the hyperparameter the training size from 75% to 55% to have more better model.

#########fit the model:

#option1 normal linear fitting

model1 <- lm(review\_stats\_all\_average\_score~.,data=pizza1\_ftrain)

#option2: using k fold to increase the samples to train the model

library('caret')

model2 <- train(

review\_stats\_all\_average\_score~.,

data=pizza1\_ftrain,

method = "lm",

trControl = trainControl(

method = "cv",

number = 10,

verboseIter = TRUE

)

)

#option3: nonlinear multivariant regression

model3 <- nls(review\_stats\_all\_average\_score~exp(a\*price\_level+b/provider\_rating+c\*review\_stats\_all\_total\_score+d\*zip),

data=pizza1\_ftrain, start=c(a=0,b=0,c=0,d=0))

#evaluate the model

par(mfrow=c(2,2))

plot(model3)

#test the datasets

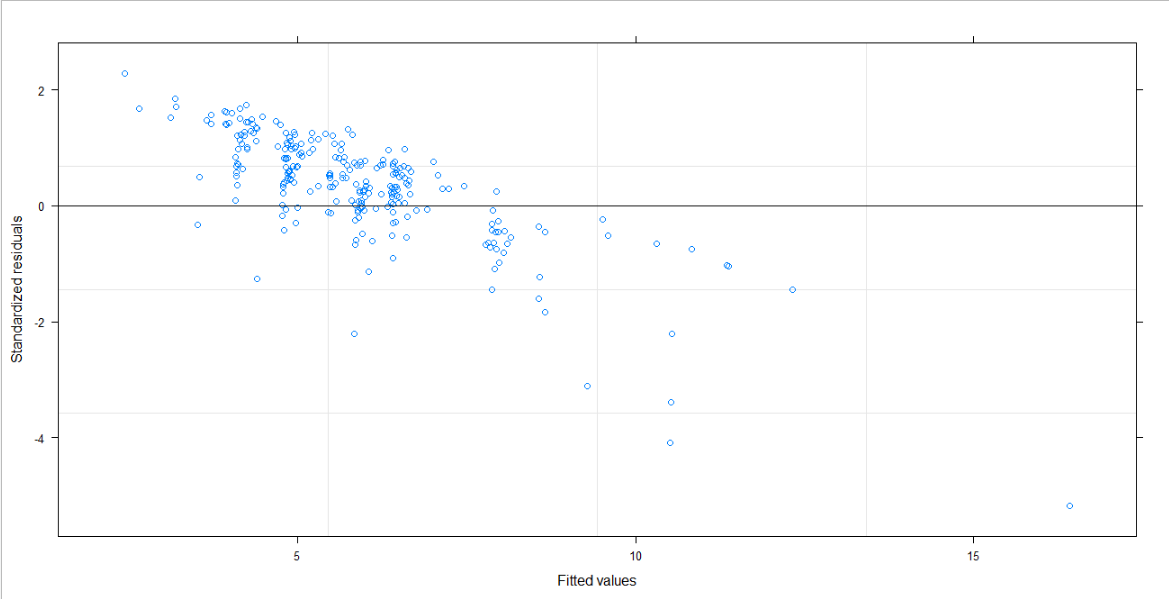
pizza1\_ftest <- subset(pizza1\_no2.test, select = c(1:3,7))

pizza1\_no2.test$prediction <- predict(model3,pizza1\_ftest)

#evaluate the test prediction result

library(dismo)

evaluate(pizza1\_no2.test$review\_stats\_all\_average\_score, pizza1\_no2.test$prediction)



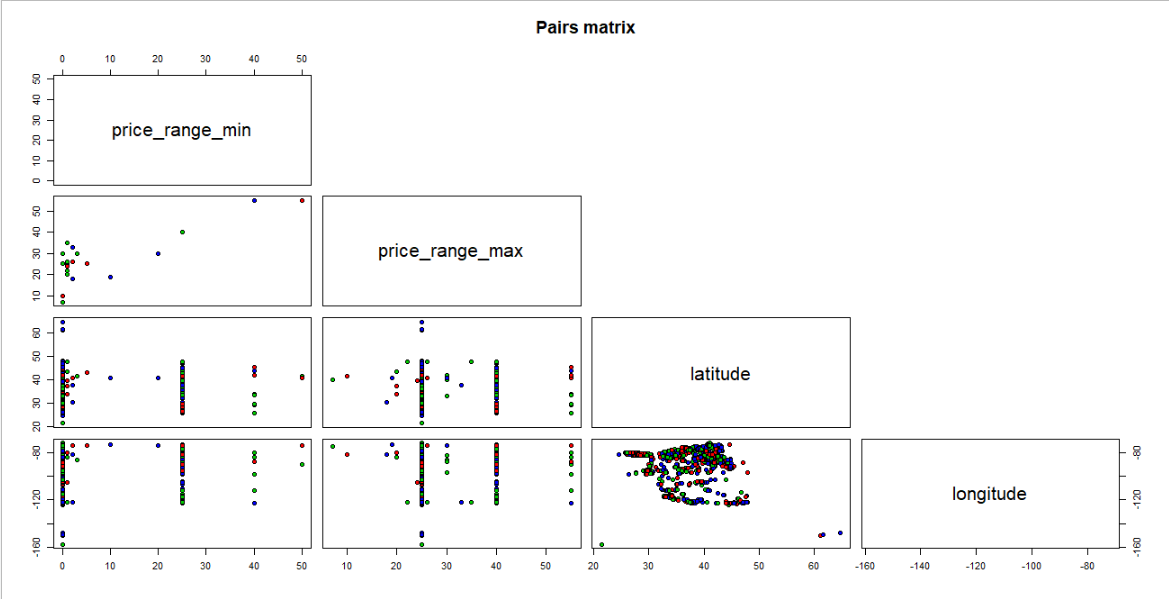
Second I used pizza2(datafiniti) to find out if there is any relationship between restaurant position and its price range. If yes, position is one of the predictors for the price range. I first proposed null Hypothesis H0: No relationship between geography variables and Price Range variables and alternate Hypothesis HA: There exists a relationship between geography variables and Price Range variables. Hence based on the assumption and research objective, I filtered out the only concerned variables which are latitude, longitude and PriceRange (Max and Min).

Geo\_data <- subset(pizza2, select = c(7,8,4,5) )

#Relationship between every pair of variables

pairs(Geo\_data, labels = colnames(Geo\_data), main = "Pairs matrix", pch = 21,

bg = c("red", "green3", "blue"), upper.panel = NULL)



From the glance of Pairs Matrix for the attributes, we can learn that there are simple linear correlations between the geography variables (latitude, longitude) and price ranges(max, min). I will go further analysis to see the prediction. After splitting the dataset into training and test sets by 50/50, I selected all predictors of training set with degree one for predicting Maxprice and found out that the p value is smaller than 2.2e-16 which is far less than alpha 0.05. That is, the null hypothesis is wrong and alternative hypothesis should be adopted. There is a relationship between geography variables and Price Range variables. Moreover, a fitted model is trained and tested with test set. The results are train RMSE = 0.828965 and test RMSE = 0.7605214 which are too much different.

# Select all predictors with degree one for predicting Maxprice

RMSE <- function(pred, actual, narm = False){

sqrt(mean((pred - actual) ^ 2, na.rm = narm))}

full\_model <- lm(price\_range\_max ~ ., data = train\_data)

summary(full\_model)

# test RMSE

findRMSE <- function(model){

cat("Train RMSE = ", RMSE(train\_data$price\_range\_max, model$fitted.values, narm = TRUE), "\n")

cat("Test RMSE = ", RMSE(test\_data$price\_range\_max, predict(model, test\_data), narm = TRUE))}

findRMSE(full\_model)

Finally, for pizza3(jared), I merged this dataset with the other two, because it has too few effective attributes to determine the final reviewing answers at first. Furthermore, different records for the same restaurants are shown for many times only because of the different question times. So I used SQL to group the same rows which add up the total numbers of votes. Then column name ‘place’ needs to be changed to ‘name’ as primary key to cross join with the other two datasets.

#rearrange the columns to show y at last

pizza3 <- pizza3[,c(3,4,5,2,6,1)]

colnames(pizza3)[colnames(pizza3)=="place"] <- "name"

#combine same rows using SQL

library('sqldf')

pizza3\_sql <- sqldf('SELECT name,sum(total\_votes) as t\_v,sum(votes) as v,answer FROM pizza3 GROUP BY name,answer')

The joined dataset has minor problem with column ‘answer’ which is y. There should be 5 categories in this column which are ‘Excellent’, ‘Good’, ‘Average’, ‘Poor’ and ‘Never Again’. But in some rows, there are ‘Fair’ instead of ‘Average’. So I changed it to ‘Average’ and added the numbers up for the same items. To simplify the analysis, I calculated one weighted average value for ‘answer’ instead of using 5 different values for each restaurant. Specifically, the 5 categories respectively represent the scores from 4 to 0. Then 5 scores times their respective numbers of voters and each total is divided by each voter number. The result is the weighted average score.

#optimize the dataset

pizza3\_sql <- pizza3\_sql[-c(228),] #has 'Fair'

# convert level to weighted scores

pizza3\_sql$score <- c(rep(0,280))

pizza3\_sql$score[which(pizza3\_sql$answer=="Excellent")] <- '4'

pizza3\_sql$score[which(pizza3\_sql$answer=="Good")] <- '3'

pizza3\_sql$score[which(pizza3\_sql$answer=="Average")] <- '2'

pizza3\_sql$score[which(pizza3\_sql$answer=="Poor")] <- '1'

pizza3\_sql$score[which(pizza3\_sql$answer=="Never Again")] <- '0'

pizza3\_sql$score <- as.numeric(pizza3\_sql$score)

pizza3\_sql$ws <- pizza3\_sql$v\*pizza3\_sql$score

pizza3\_sql$fin<-c(rep(0,280)) #fill all with 0

sequence <- seq(1, nrow(pizza3\_sql), by=5)

for(i in sequence) {

pizza3\_sql$fin[i] <- (pizza3\_sql$ws[i]+pizza3\_sql$ws[i+1]+pizza3\_sql$ws[i+2]+pizza3\_sql$ws[i+3]+pizza3\_sql$ws[i+4])/pizza3\_sql$t\_v[i]

}

After splitting the train/test and scaling the dataset, I used two classification methods to test the accuracy of classifier. Due to their better performance, I used Support Vector Machine and Parallel Random Forecast to train and test the model. As result, SVM has the accuracy of 85.7% but PRF is 89.5% when the models are evaluated with test set. This is the model that we want to make the accurate classification.

#svm without k fold

library(e1071)

classifier = svm(formula =price\_level ~ .,

data = training\_set[,2:8],

type = 'C-classification',

kernel = 'radial')

y\_pred = predict(classifier, newdata = test\_set[,2:8])

cm = table(test\_set[,8], y\_pred)

cm

y\_pred

0 1 2

1 0 6 0

2 0 1 0

#Parallel Random Forest with repeated cv is best acc0.895

library(caret)

# define training control

train\_control <- trainControl(method="repeatedcv", number=10, repeats=3)

# train the model

model <- train(price\_level~., data=training\_set, trControl=train\_control, method="parRF")

# summarize results

print(model)

Parallel Random Forest

20 samples

7 predictor

3 classes: '0', '1', '2'

No pre-processing

Resampling: Cross-Validated (10 fold, repeated 3 times)

Summary of sample sizes: 18, 18, 19, 18, 19, 18, ...

Resampling results across tuning parameters:

mtry Accuracy Kappa

2 0.8950617 0

12 0.8580247 0

23 0.8580247 0

**5. Summary**

According to our analysis, we uncovered some insights about the rating for pizza restaurant with three different datasets. Based on the work above, it appears that the geography of restaurant, price level and provider’s rating can decide its final average review score. Meanwhile, as our common sense, the location of a restaurant can determine its price range. And the pricing strategy can also be determined by the reviewers’ evaluating scores. Hence the owner of restaurant can employ all these conclusions to improve their service.

In the research, I used nonlinear multivariant regression model to find out the relationship among average review scores and all the other independent variables. I then used the fitted model to make the prediction for the review scores. After joining the datasets, I employed Parallel Random Forest with k fold cross validation to make the classification of price level for the restaurant. So the owner can have correct pricing strategies based on that.

The only limitation of the analysis is that the pool of sample records is too small to have an accurate result. There are only 300 to 400 effective records in each dataset, after I removed the duplicate information. What is more, after merging the tables to have a complete dataset, there are only 50 effective records. That is, majority of the records in each table is not relative to each other. There is not extensive information for most the restaurant. So collecting more raw data for the same restaurants is necessary to make further research.