PROJECT DATA ANALYSIS

Luis Garcia Anna
Udelman Leon Samuel
Yanling Xiao

12/12/2018

Introduction

In this task, we are challenged to use reservation and visitation data to predict the total number of visitors to a restaurant for future dates.This information will help restaurants be much more efficient and allow them to focus on creating an enjoyable dining experience for their customers.

The data we are provided is a time-series forecasting problem centred on restaurantvisitors. The data comes from: AirREGI/Restaurant Board (air): similar to Square, a reservation control and cash register system. We are required to use the reservations, visits, and other information from these sites to forecast future restaurant visitor totals on a given date. The training data covers the dates from January 2017 until March 2017. The test set covers the first three weeks of April 2017. There are days in the test set where the restaurant were closed and had no visitors. These are ignored in scoring. The training set omits days where the restaurants were closed

```
# Loading required packages
library(dplyr)
library(chron)
library(ggplot2)
library(ggmap)
library(rpart)
library(rpart.plot)
library(randomForest)
library(caTools)
library(class)
library(caret)
library(e1071)
library(rfUtilities)
library(MLmetrics)
library(miscTools)
```

We have 4 dataset given by sql file with different variables:

air_reserve

This table contains reservations made in the air system. Note that the reserve_datetime indicates the time when the reservation was created, whereas the visit_datetime is the time in the future where the visit will occur.

- ID - the restaurant's id in the air system
- visit_datetime - the time of the reservation
- reserve_datetime - the time the reservation was made
- reserve_visitors - the number of visitors for that reservation

restaurant_info

This table contains information about selected air restaurants. Column names and contents are self-explanatory.

- ID
- air_genre_name
- air_area_name
- latitude
- longitude

visit

This table contains historical visit data for the air restaurants.

- ID
- visit_date - the date
- visitors - the number of visitors to the restaurant on the date

date_info

This table gives basic information about the calendar dates in the dataset.

- calendar_date
- day_of_week
- holiday_flg - is the day a holiday in Japan

MySQLWorkbench

We have studied the 4 tables in MySQLWorkbench an generate a finall table, joining the other 4, containing all the data. Now we are going to work with these data and build a prediction model.

Loading and cleaning data

```
#Read the data:
restData <- read.csv(file = "FinalData_complet.csv",
                     header = TRUE,
                     sep = ",",
                     quote = "",
                     stringsAsFactors = FALSE,
                     na.strings = "NULL")
#Check the variables:
str(restData)
```

```
## 'data.frame':    84201 obs. of  13 variables:
##  $ ID              : chr  "\"restaurant_ 1\"" "\"restaurant_ 1\"" "\"restaurant_ 1\"" "\"restaurant_
##  $ visit_date      : chr  "2017-01-02" "2017-01-03" "2017-01-04" "2017-01-06" ...
##  $ visitors        : int  10 38 31 22 22 22 45 17 32 32 ...
##  $ day_of_week     : chr  "Monday" "Tuesday" "Wednesday" "Friday" ...
##  $ holiday_flg     : int  1 1 0 0 0 0 0 0 1 1 ...
##  $ air_genre_name  : chr  "Italian/French" "Italian/French" "Italian/French" "Italian/French" ...
##  $ air_area_name   : chr  "\"Hyōgo-ken Kōbe-shi Kumoidōri\"" "\"Hyōgo-ken Kōbe-shi Kumoidōri\"" "\"Hy
```

```
##  $ latitude        : num  34.7 34.7 34.7 34.7 34.7 ...
##  $ longitude       : num  135 135 135 135 135 ...
##  $ reserve_visitors: int  NA NA NA 2 11 2 3 NA 14 2 ...
##  $ reserve_date    : chr  NA NA NA "2017-01-02" ...
##  $ visit_time      : chr  NA NA NA "18:00:00" ...
##  $ reserve_time    : chr  NA NA NA "23:00:00" ...
```

From the result, we see there are 84201 rows and 13 columns in the dataset. Variables as time and hours, that before were togheter in the same column, now are separated in two. The other keep the same format as the original tables.

Let's change some variable categories:

```
# #Change the category of the variables:
# restData[, c(3,5)] <- sapply(restData[, c(3,5)], as.numeric)
# restData[, c(10)] <- sapply(restData[, c(10)], as.integer)
# restData[, c(1)] <- sapply(restData[, c(1)], as.factor)
# restData[, c(2)] <- sapply(restData[, c(2)], as.factor)
```

```
#Adjust datetime format:
master <- restData %>%
  mutate(visit_date = as.POSIXct(visit_date,
                                 format="%Y-%m-%d", tz="")) %>%
  mutate(reserve_date = as.POSIXct(reserve_date,
                                 format="%Y-%m-%d", tz=""))
#Check the changes:
head(master)
```

```
##                ID visit_date visitors day_of_week holiday_flg
## 1 "restaurant_ 1" 2017-01-02       10      Monday           1
## 2 "restaurant_ 1" 2017-01-03       38     Tuesday           1
## 3 "restaurant_ 1" 2017-01-04       31   Wednesday           0
## 4 "restaurant_ 1" 2017-01-06       22      Friday           0
## 5 "restaurant_ 1" 2017-01-06       22      Friday           0
## 6 "restaurant_ 1" 2017-01-06       22      Friday           0
##   air_genre_name                  air_area_name latitude longitude
## 1 Italian/French "Hyōgo-ken Kōbe-shi Kumoidōri" 34.69512  135.1979
## 2 Italian/French "Hyōgo-ken Kōbe-shi Kumoidōri" 34.69512  135.1979
## 3 Italian/French "Hyōgo-ken Kōbe-shi Kumoidōri" 34.69512  135.1979
## 4 Italian/French "Hyōgo-ken Kōbe-shi Kumoidōri" 34.69512  135.1979
## 5 Italian/French "Hyōgo-ken Kōbe-shi Kumoidōri" 34.69512  135.1979
## 6 Italian/French "Hyōgo-ken Kōbe-shi Kumoidōri" 34.69512  135.1979
##   reserve_visitors reserve_date visit_time reserve_time
## 1               NA         <NA>       <NA>         <NA>
## 2               NA         <NA>       <NA>         <NA>
## 3               NA         <NA>       <NA>         <NA>
## 4                2   2017-01-02   18:00:00     23:00:00
## 5               11   2017-01-04   19:00:00     21:00:00
## 6                2   2017-01-05   19:00:00     21:00:00
```

Sanity checks

Now we are going to check all reservations as been done at the time of visiting or prior to that.

```
reservLag <- as.numeric(master$visit_date)-
  as.numeric(master$reserve_date)
reservLag <- reservLag[complete.cases(reservLag)]
if(all(reservLag >= 0)){
    print("Reservations' times are correct")
  } else {
    warning("You cannot travel in time!
            Some reservations are for past days")
  }
```

```
## [1] "Reservations' times are correct"
```

From the result, we can see all the reservation data we have are correct.

Exploratory analysis

```
# Set graph counter
t = 1
```

Now we are going to classify the type of restaurants in our dataset. We will get the sum of the reservation for each restaurant each day, and the mean of the reservation

```
#Number of restaurants
n_distinct(master$ID)
```

```
## [1] 825
```

```
#Generating compact data
visitorsCompact <- master %>% select(-visit_time,-reserve_date,-reserve_time,
                                     -reserve_visitors)%>%.[!duplicated(.),]
reserveCompact <- master %>%
  mutate(reserve_lag = visit_date - reserve_date) %>%
  group_by(ID, visit_date) %>%
  summarise_at(vars(reserve_lag, reserve_visitors), funs(mean, sum)) %>%
  select(-reserve_lag_sum, -reserve_visitors_mean) %>%
  as.data.frame()

masterCompact <- inner_join(visitorsCompact, reserveCompact,
                            by = c("ID", "visit_date"))
head(masterCompact)
```

```
##                ID visit_date visitors day_of_week holiday_flg
## 1 "restaurant_ 1" 2017-01-02       10      Monday           1
## 2 "restaurant_ 1" 2017-01-03       38     Tuesday           1
## 3 "restaurant_ 1" 2017-01-04       31   Wednesday           0
## 4 "restaurant_ 1" 2017-01-06       22      Friday           0
## 5 "restaurant_ 1" 2017-01-07       45    Saturday           0
## 6 "restaurant_ 1" 2017-01-08       17      Sunday           0
##   air_genre_name                  air_area_name latitude longitude
## 1 Italian/French "Hyōgo-ken Kōbe-shi Kumoidōri" 34.69512  135.1979
## 2 Italian/French "Hyōgo-ken Kōbe-shi Kumoidōri" 34.69512  135.1979
## 3 Italian/French "Hyōgo-ken Kōbe-shi Kumoidōri" 34.69512  135.1979
## 4 Italian/French "Hyōgo-ken Kōbe-shi Kumoidōri" 34.69512  135.1979
## 5 Italian/French "Hyōgo-ken Kōbe-shi Kumoidōri" 34.69512  135.1979
```

```
## 6 Italian/French "Hyōgo-ken Kōbe-shi Kumoidōri" 34.69512  135.1979
##   reserve_lag_mean reserve_visitors_sum
## 1          NA secs                   NA
## 2          NA secs                   NA
## 3          NA secs                   NA
## 4      201600 secs                   15
## 5      172800 secs                    3
## 6          NA secs                   NA
```

```
summary(masterCompact)
```

```
##       ID              visit_date                     visitors
##  Length:61803       Min.   :2017-01-01 00:00:00   Min.   :  1.00
##  Class :character   1st Qu.:2017-01-26 00:00:00   1st Qu.:  9.00
##  Mode  :character   Median :2017-02-16 00:00:00   Median : 17.00
##                     Mean   :2017-02-16 04:09:13   Mean   : 20.77
##                     3rd Qu.:2017-03-10 00:00:00   3rd Qu.: 29.00
##                     Max.   :2017-03-31 00:00:00   Max.   :877.00
##
##  day_of_week         holiday_flg      air_genre_name
##  Length:61803       Min.   :0.00000   Length:61803
##  Class :character   1st Qu.:0.00000   Class :character
##  Mode  :character   Median :0.00000   Mode  :character
##                     Mean   :0.03929
##                     3rd Qu.:0.00000
##                     Max.   :1.00000
##
##  air_area_name         latitude       longitude     reserve_lag_mean
##  Length:61803       Min.   :33.21   Min.   :130.2   Length:61803
##  Class :character   1st Qu.:34.69   1st Qu.:135.3   Class :difftime
##  Mode  :character   Median :35.66   Median :139.7   Mode  :numeric
##                     Mean   :35.62   Mean   :137.4
##                     3rd Qu.:35.69   3rd Qu.:139.8
##                     Max.   :44.02   Max.   :144.3
##
##  reserve_visitors_sum
##  Min.   :  1.00
##  1st Qu.:  4.00
##  Median :  9.00
##  Mean   : 12.76
##  3rd Qu.: 17.00
##  Max.   :214.00
##  NA's   :50805
```

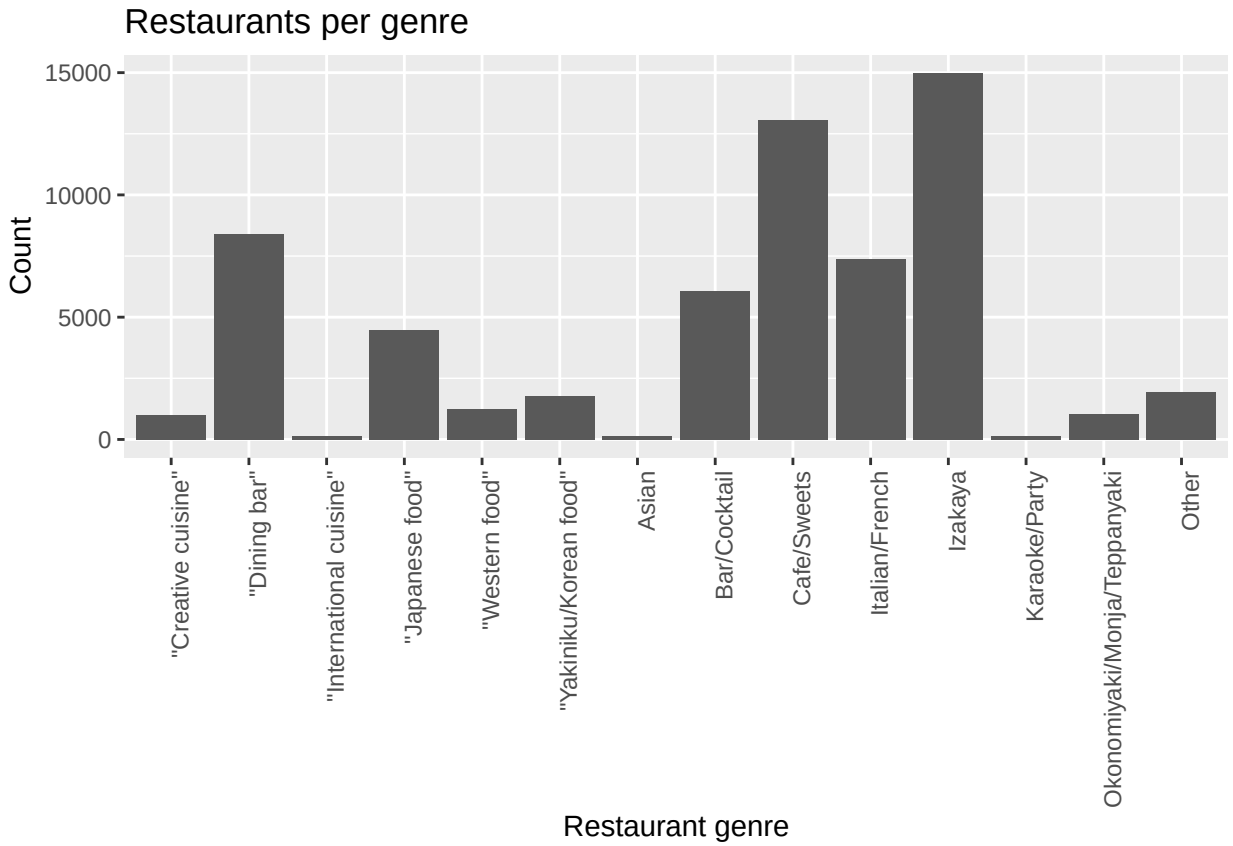
```
masterCompact[,"reserve_lag_mean"] <- sapply(masterCompact[,"reserve_lag_mean"], as.numeric)
summary(masterCompact)
```

```
##       ID              visit_date                     visitors
##  Length:61803       Min.   :2017-01-01 00:00:00   Min.   :  1.00
##  Class :character   1st Qu.:2017-01-26 00:00:00   1st Qu.:  9.00
##  Mode  :character   Median :2017-02-16 00:00:00   Median : 17.00
##                     Mean   :2017-02-16 04:09:13   Mean   : 20.77
##                     3rd Qu.:2017-03-10 00:00:00   3rd Qu.: 29.00
##                     Max.   :2017-03-31 00:00:00   Max.   :877.00
##
```

```
##  day_of_week         holiday_flg      air_genre_name
##  Length:61803       Min.   :0.00000   Length:61803
##  Class :character   1st Qu.:0.00000   Class :character
##  Mode  :character   Median :0.00000   Mode  :character
##                     Mean   :0.03929
##                     3rd Qu.:0.00000
##                     Max.   :1.00000
##
##  air_area_name         latitude       longitude     reserve_lag_mean
##  Length:61803       Min.   :33.21   Min.   :130.2   Min.   :      0
##  Class :character   1st Qu.:34.69   1st Qu.:135.3   1st Qu.:  86400
##  Mode  :character   Median :35.66   Median :139.7   Median : 259200
##                     Mean   :35.62   Mean   :137.4   Mean   : 395864
##                     3rd Qu.:35.69   3rd Qu.:139.8   3rd Qu.: 518400
##                     Max.   :44.02   Max.   :144.3   Max.   :6825600
##                                                     NA's   :50805
##  reserve_visitors_sum
##  Min.   :  1.00
##  1st Qu.:  4.00
##  Median :  9.00
##  Mean   : 12.76
##  3rd Qu.: 17.00
##  Max.   :214.00
##  NA's   :50805
```

Plot restaurants by genre

```
g <- ggplot(masterCompact, aes(air_genre_name)) +
  geom_bar() +
  labs(x = "Restaurant genre", y = "Count",
       title = "Restaurants per genre") +
  theme(axis.text.x = element_text(angle = 90, hjust = 1))
t <- t + 1
g
```



From the result, we can see "Dinning bar", "Cafe/Sweets" and "Izakaya" are the top 3 restaurants.

```
  # Define map borders
minLon <- min(masterCompact$longitude)*1.03
maxLon <- max(masterCompact$longitude)*1.03
minLat <- min(masterCompact$latitude)*1.03
maxLat <- max(masterCompact$latitude)*1.03

# Download map
japan <- c(left = minLon, bottom = minLat, right = maxLon, top = maxLat)
map <- get_stamenmap(japan, zoom = 5, maptype = "toner-lite")
```

```
## Map from URL : http://tile.stamen.com/toner-lite/5/27/11.png
## Map from URL : http://tile.stamen.com/toner-lite/5/28/11.png
## Map from URL : http://tile.stamen.com/toner-lite/5/29/11.png
## Map from URL : http://tile.stamen.com/toner-lite/5/27/12.png
## Map from URL : http://tile.stamen.com/toner-lite/5/28/12.png
## Map from URL : http://tile.stamen.com/toner-lite/5/29/12.png
```

```
  # Table of avg visitors per restaurant
avgVisitors <- masterCompact %>% select(ID, longitude, latitude, visitors) %>%
  group_by(ID) %>% summarise_all(mean) %>% arrange(visitors) %>%
  as.data.frame()

  # Plot avg visitors on the map
g <- ggmap(map) +
```

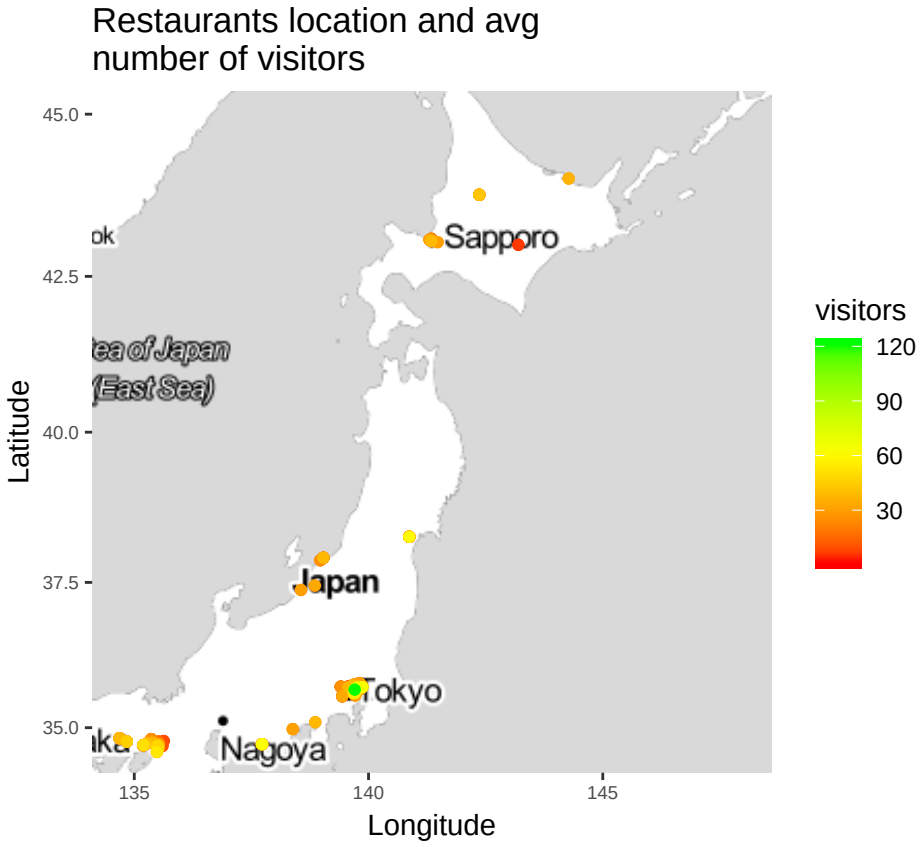
```
  geom_point(data = avgVisitors,
             aes(x = longitude, y = latitude, color = visitors)) +

  labs(x = "Longitude", y = "Latitude",
       title = "Restaurants location and avg\nnumber of visitors") +

  theme(axis.text.x = element_text(size = 7),
        axis.text.y = element_text(size = 7)) +

  scale_colour_gradientn(
    colours=c('red','yellow','green'),
    oob = scales::squish)
t <- t+1
g
```

```
## Warning: Removed 159 rows containing missing values (geom_point).
```



Here we can see the restaurant are spread across Japan. More restaurants are located in Tokyo and has also have more visitors.

```
#Barplot avg visitors
g <- ggplot(avgVisitors, aes(x = reorder(ID,-visitors),
                             y = visitors)) +
  geom_bar(stat = "identity") +
  labs(y = "Avg. number of visitors",
       title = "Average visitors per restaurant") +
  theme(axis.title.x=element_blank(),
        axis.text.x=element_blank(),
        axis.ticks.x=element_blank())
```

Figure 1: Average visitor of each resturant

```
t <- t + 1
g
```

```
# Boxplot visitors per restaurant
set.seed(100)
split <- sample.split(masterCompact, SplitRatio = 0.001)

plotData <- subset(masterCompact, split == TRUE)
g <- ggplot(plotData, aes(x = reorder(ID,-visitors), y = visitors)) +
  geom_boxplot() +
  labs(y = "Number of visitors",
       title = "Comparison visitors per restaurant") +
  theme(axis.title.x=element_blank(),
        axis.text.x=element_blank(),
        axis.ticks.x=element_blank())
t <- t + 1
g
```

```
# Number of visitors in time. Include holidays.
dateInfo <- masterCompact %>% select(visit_date, day_of_week, holiday_flg) %>%
  .[!duplicated(.),]
dateInfo[,3] <- as.factor(dateInfo[,3])

avgVisitorsDay <- masterCompact %>%
```

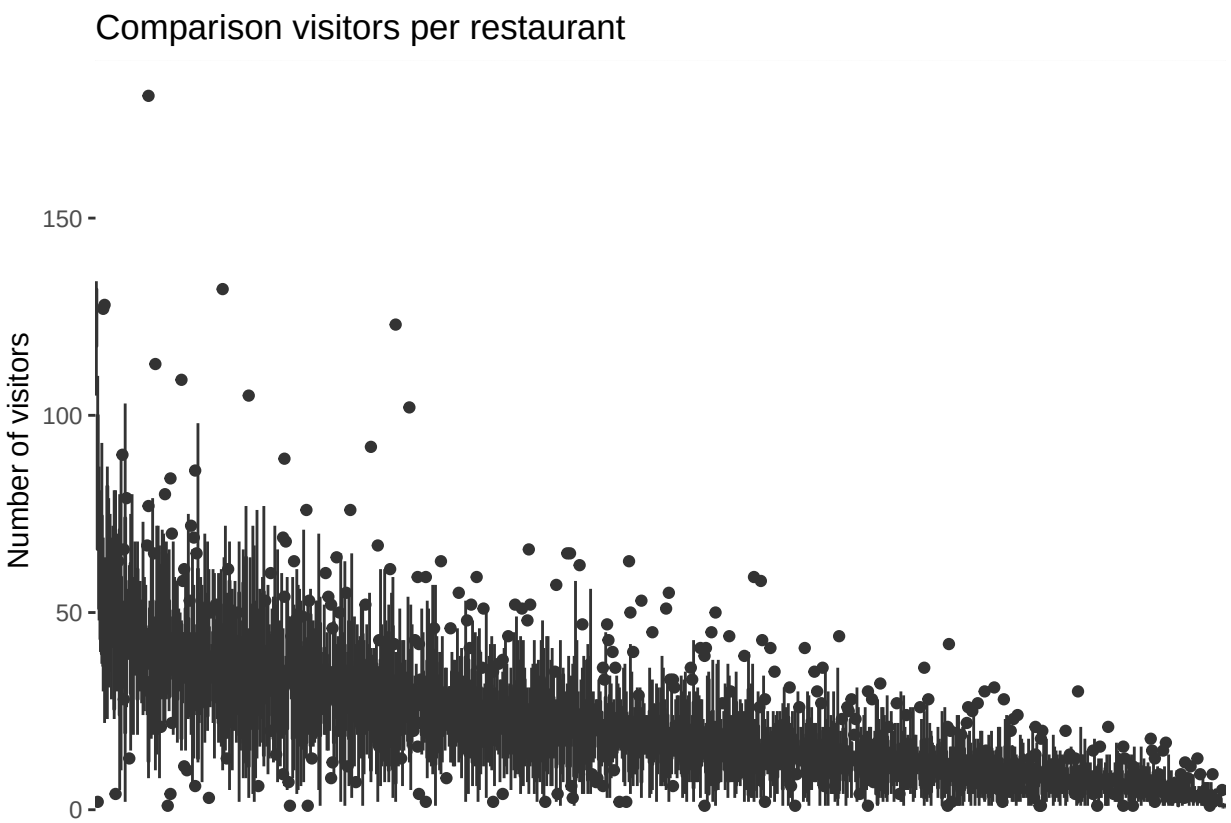


Figure 2: Comparison visitors per restaurant

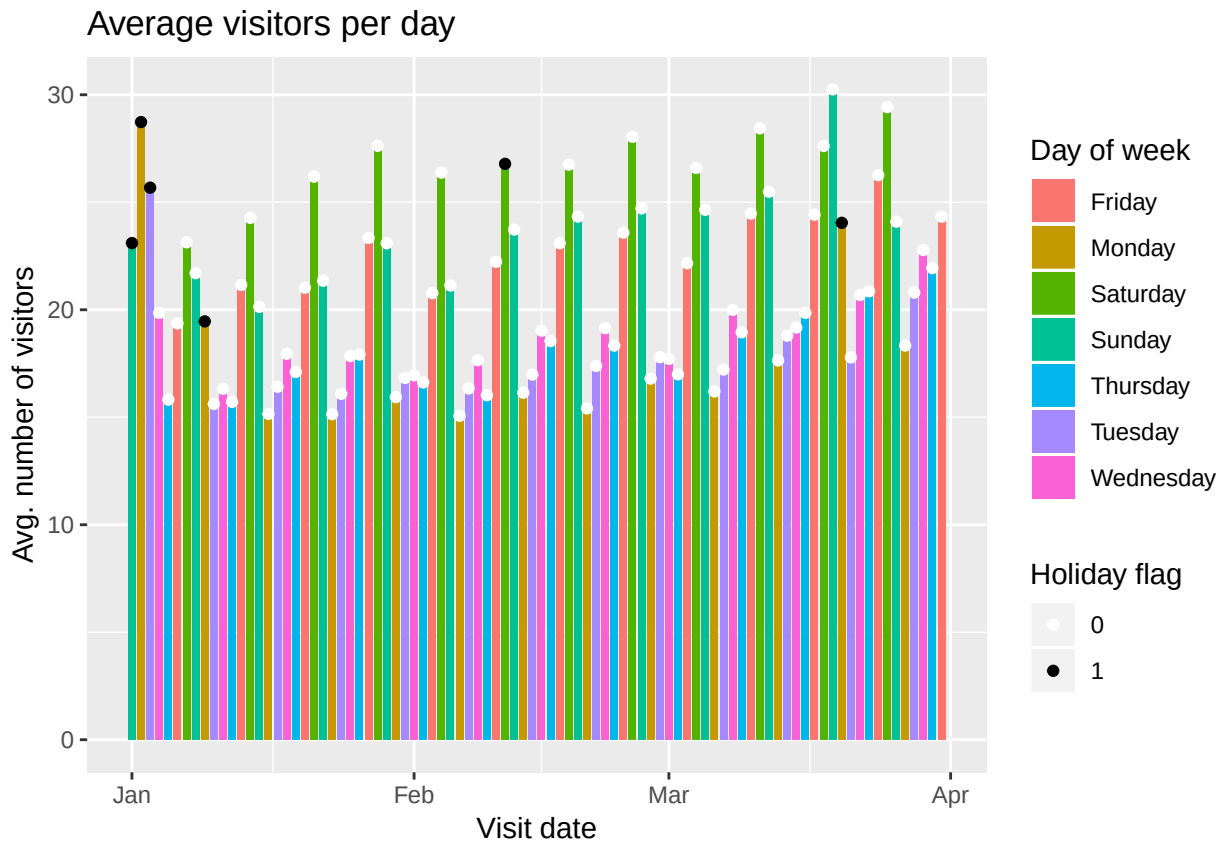


Figure 3: Average visitors per day

```
  select(visit_date, visitors) %>%
  group_by(visit_date) %>%
  summarise_all(mean) %>%
  arrange(visit_date) %>%
  left_join(.,dateInfo, by = "visit_date") %>%
  as.data.frame()

g <- ggplot(avgVisitorsDay, aes(x = visit_date, y = visitors)) +
  geom_col(aes(fill = day_of_week)) +
  scale_fill_discrete(name = "Day of week") +
  geom_point(aes(color = holiday_flg)) +
  scale_color_manual(name = "Holiday flag",
                     values = c("white","black")) +
  labs(x = "Visit date",
       y = "Avg. number of visitors",
       title = "Average visitors per day")

g
```

```
# Compare number of actual visitors vs sum of reservations for each day vs how long ago were they made
resVisRatio <- masterCompact %>%
  mutate(res_vis_ratio = reserve_visitors_sum / visitors)

summary(resVisRatio)
```

```
##       ID              visit_date                     visitors
##  Length:61803       Min.   :2017-01-01 00:00:00   Min.   :  1.00
##  Class :character   1st Qu.:2017-01-26 00:00:00   1st Qu.:  9.00
##  Mode  :character   Median :2017-02-16 00:00:00   Median : 17.00
##                     Mean   :2017-02-16 04:09:13   Mean   : 20.77
##                     3rd Qu.:2017-03-10 00:00:00   3rd Qu.: 29.00
##                     Max.   :2017-03-31 00:00:00   Max.   :877.00
##
##  day_of_week         holiday_flg      air_genre_name
##  Length:61803       Min.   :0.00000   Length:61803
##  Class :character   1st Qu.:0.00000   Class :character
##  Mode  :character   Median :0.00000   Mode  :character
##                     Mean   :0.03929
##                     3rd Qu.:0.00000
##                     Max.   :1.00000
##
##  air_area_name         latitude       longitude     reserve_lag_mean
##  Length:61803       Min.   :33.21   Min.   :130.2   Min.   :      0
##  Class :character   1st Qu.:34.69   1st Qu.:135.3   1st Qu.:  86400
##  Mode  :character   Median :35.66   Median :139.7   Median : 259200
##                     Mean   :35.62   Mean   :137.4   Mean   : 395864
##                     3rd Qu.:35.69   3rd Qu.:139.8   3rd Qu.: 518400
##                     Max.   :44.02   Max.   :144.3   Max.   :6825600
##                                                     NA's   :50805
##  reserve_visitors_sum res_vis_ratio
##  Min.   :  1.00       Min.   : 0.01
##  1st Qu.:  4.00       1st Qu.: 0.26
##  Median :  9.00       Median : 0.49
##  Mean   : 12.76       Mean   : 0.56
##  3rd Qu.: 17.00       3rd Qu.: 0.75
##  Max.   :214.00       Max.   :21.00
##  NA's   :50805        NA's   :50805
```

```
g<- ggplot(data = resVisRatio, aes(resVisRatio$res_vis_ratio)) +
  geom_histogram(breaks = seq(0,1.5,0.05)) +
  labs(x = "Reserves to visit ratio",
    title="Histogram for reserve/visitors ratio")
t<-t+1
g
```

```
## Warning: Removed 50805 rows containing non-finite values (stat_bin).
```

Encoding Categorical Data and creating new variables

After this first part, explorating the data, we are going to modify and generate new variables that helps us to do a better model for the visits.

First, we are encoding categorical data. We convert the day_of_week(Monday to Sunday) to number(1 to7).

For restaurant genre name, we are also conver air_genre_name to number(1 to 14).

```
encodedData <- master %>%
  mutate(day_number = ifelse(day_of_week == "Wednesday" , 3,
```

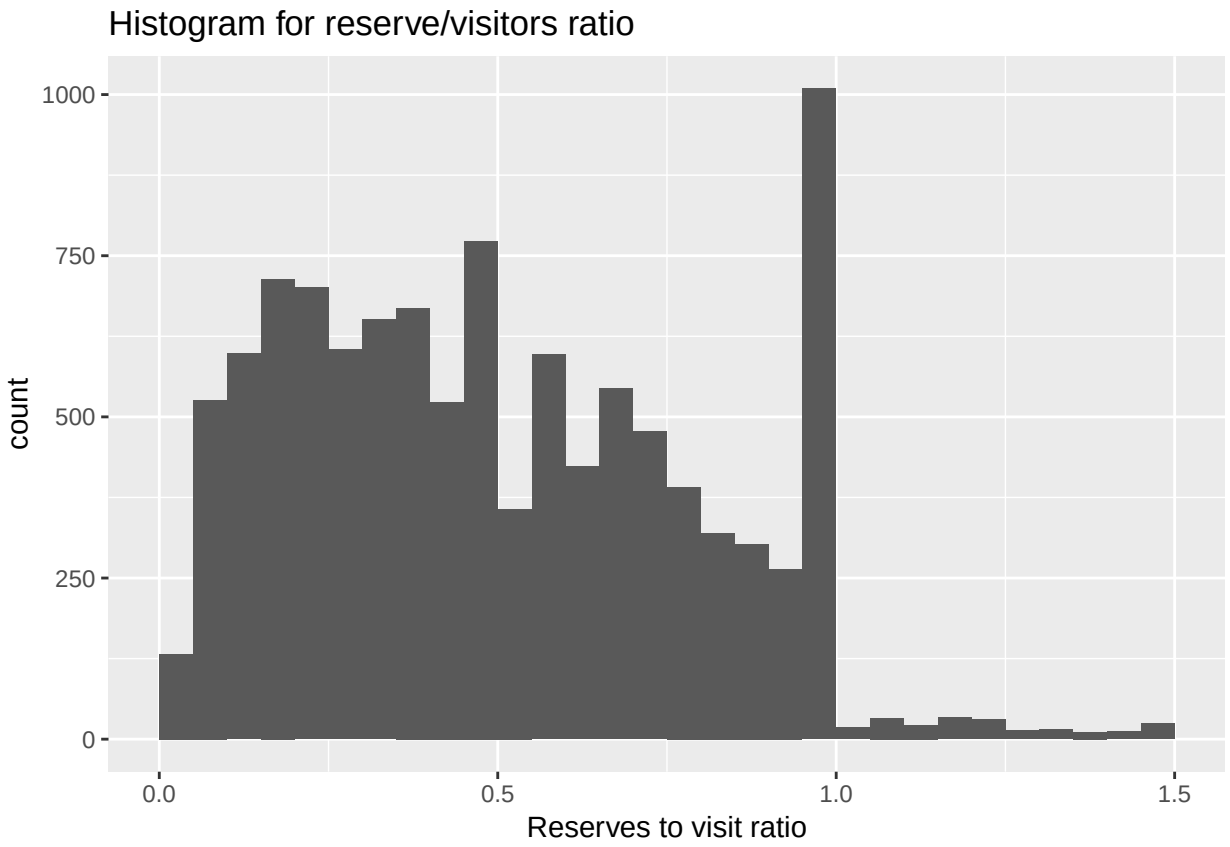


Figure 4: Histogram for reserve/visitors ratio

```
                    ifelse(day_of_week == "Thursday" , 4,
                    ifelse(day_of_week == "Friday" , 5,
                    ifelse(day_of_week == "Saturday" , 6,
                    ifelse(day_of_week == "Sunday" , 7,
                    ifelse(day_of_week == "Monday" , 1,
                    ifelse(day_of_week == "Tuesday" , 2, 0)))))))
head(encodedData)
```

```
##                ID visit_date visitors day_of_week holiday_flg
## 1 "restaurant_ 1" 2017-01-02       10      Monday           1
## 2 "restaurant_ 1" 2017-01-03       38     Tuesday           1
## 3 "restaurant_ 1" 2017-01-04       31   Wednesday           0
## 4 "restaurant_ 1" 2017-01-06       22      Friday           0
## 5 "restaurant_ 1" 2017-01-06       22      Friday           0
## 6 "restaurant_ 1" 2017-01-06       22      Friday           0
##   air_genre_name                  air_area_name latitude longitude
## 1 Italian/French "Hyōgo-ken Kōbe-shi Kumoidōri" 34.69512  135.1979
## 2 Italian/French "Hyōgo-ken Kōbe-shi Kumoidōri" 34.69512  135.1979
## 3 Italian/French "Hyōgo-ken Kōbe-shi Kumoidōri" 34.69512  135.1979
## 4 Italian/French "Hyōgo-ken Kōbe-shi Kumoidōri" 34.69512  135.1979
## 5 Italian/French "Hyōgo-ken Kōbe-shi Kumoidōri" 34.69512  135.1979
## 6 Italian/French "Hyōgo-ken Kōbe-shi Kumoidōri" 34.69512  135.1979
##   reserve_visitors reserve_date visit_time reserve_time day_number
## 1               NA         <NA>       <NA>         <NA>          1
## 2               NA         <NA>       <NA>         <NA>          2
## 3               NA         <NA>       <NA>         <NA>          3
## 4                2   2017-01-02   18:00:00     23:00:00          5
## 5               11   2017-01-04   19:00:00     21:00:00          5
## 6                2   2017-01-05   19:00:00     21:00:00          5
```

```
encodedData2 <- master %>%
  mutate(genre = ifelse(air_genre_name == "Cafe/Sweets", 1,
                 ifelse(air_genre_name =="\"Dining bar\"", 2,
                 ifelse(air_genre_name == "Izakaya" , 3,
                 ifelse(air_genre_name == "Bar/Cocktail" , 4,
                 ifelse(air_genre_name == "\"Western food\""  , 5,
                 ifelse(air_genre_name == "Other" , 6,
                 ifelse(air_genre_name == "Karaoke/Party" , 7,
                 ifelse(air_genre_name == "Italian/French"  , 8,
                 ifelse(air_genre_name==
                          "\"Yakiniku/Korean food\"",9,
                 ifelse(air_genre_name==
                          "Okonomiyaki/Monja/Teppanyaki", 10,
                 ifelse(air_genre_name == "\"Creative cuisine\"", 11,
                 ifelse(air_genre_name == "\"Japanese food\"", 12,
                 ifelse(air_genre_name ==
                          "\"International cuisine\"", 13,
                 ifelse(air_genre_name == "Asian", 14,0)
                 ))))))))))))))

encodedData$genre <- encodedData2$genre

summary(factor(encodedData$genre,
               seq(1,14)) == encodedData$genre)
```

```
##    Mode    TRUE
## logical   84201
```

```
encodedData$genre <- factor(encodedData$genre, seq(1,14))

head(encodedData)
```

```
##                ID visit_date visitors day_of_week holiday_flg
## 1 "restaurant_ 1" 2017-01-02       10      Monday           1
## 2 "restaurant_ 1" 2017-01-03       38     Tuesday           1
## 3 "restaurant_ 1" 2017-01-04       31   Wednesday           0
## 4 "restaurant_ 1" 2017-01-06       22      Friday           0
## 5 "restaurant_ 1" 2017-01-06       22      Friday           0
## 6 "restaurant_ 1" 2017-01-06       22      Friday           0
##   air_genre_name                  air_area_name latitude longitude
## 1 Italian/French "Hyōgo-ken Kōbe-shi Kumoidōri" 34.69512  135.1979
## 2 Italian/French "Hyōgo-ken Kōbe-shi Kumoidōri" 34.69512  135.1979
## 3 Italian/French "Hyōgo-ken Kōbe-shi Kumoidōri" 34.69512  135.1979
## 4 Italian/French "Hyōgo-ken Kōbe-shi Kumoidōri" 34.69512  135.1979
## 5 Italian/French "Hyōgo-ken Kōbe-shi Kumoidōri" 34.69512  135.1979
## 6 Italian/French "Hyōgo-ken Kōbe-shi Kumoidōri" 34.69512  135.1979
##   reserve_visitors reserve_date visit_time reserve_time day_number genre
## 1               NA         <NA>       <NA>         <NA>          1     8
## 2               NA         <NA>       <NA>         <NA>          2     8
## 3               NA         <NA>       <NA>         <NA>          3     8
## 4                2   2017-01-02   18:00:00     23:00:00          5     8
## 5               11   2017-01-04   19:00:00     21:00:00          5     8
## 6                2   2017-01-05   19:00:00     21:00:00          5     8
```

```
str(encodedData)
```

```
## 'data.frame':    84201 obs. of  15 variables:
##  $ ID              : chr  "\"restaurant_ 1\"" "\"restaurant_ 1\"" "\"restaurant_ 1\"" "\"restaurant_
##  $ visit_date      : POSIXct, format: "2017-01-02" "2017-01-03" ...
##  $ visitors        : int  10 38 31 22 22 22 45 17 32 32 ...
##  $ day_of_week     : chr  "Monday" "Tuesday" "Wednesday" "Friday" ...
##  $ holiday_flg     : int  1 1 0 0 0 0 0 0 1 1 ...
##  $ air_genre_name  : chr  "Italian/French" "Italian/French" "Italian/French" "Italian/French" ...
##  $ air_area_name   : chr  "\"Hyōgo-ken Kōbe-shi Kumoidōri\"" "\"Hyōgo-ken Kōbe-shi Kumoidōri\"" "\"Hy
##  $ latitude        : num  34.7 34.7 34.7 34.7 34.7 ...
##  $ longitude       : num  135 135 135 135 135 ...
##  $ reserve_visitors: int  NA NA NA 2 11 2 3 NA 14 2 ...
##  $ reserve_date    : POSIXct, format: NA NA ...
##  $ visit_time      : chr  NA NA NA "18:00:00" ...
##  $ reserve_time    : chr  NA NA NA "23:00:00" ...
##  $ day_number      : num  1 2 3 5 5 5 6 7 1 1 ...
##  $ genre           : Factor w/ 14 levels "1","2","3","4",..: 8 8 8 8 8 8 8 8 8 8 ...
```

Next, we are including information on median, mean, standard deviation and maximum number of visitors per restaurant.

```
medianVisit <- encodedData %>%
  group_by(ID) %>%
  summarise_at(vars(visitors),
               funs(median, max, mean, sd)) %>%
  as.data.frame()
```

Then we are including mean reserves/visit ratio per restaurant.

```
modelData <- left_join(encodedData,medianVisit, by = "ID")
head(modelData)
```

```
##                ID visit_date visitors day_of_week holiday_flg
## 1 "restaurant_ 1" 2017-01-02       10      Monday           1
## 2 "restaurant_ 1" 2017-01-03       38     Tuesday           1
## 3 "restaurant_ 1" 2017-01-04       31   Wednesday           0
## 4 "restaurant_ 1" 2017-01-06       22      Friday           0
## 5 "restaurant_ 1" 2017-01-06       22      Friday           0
## 6 "restaurant_ 1" 2017-01-06       22      Friday           0
##   air_genre_name                 air_area_name latitude longitude
## 1 Italian/French "Hyōgo-ken Kōbe-shi Kumoidōri" 34.69512  135.1979
## 2 Italian/French "Hyōgo-ken Kōbe-shi Kumoidōri" 34.69512  135.1979
## 3 Italian/French "Hyōgo-ken Kōbe-shi Kumoidōri" 34.69512  135.1979
## 4 Italian/French "Hyōgo-ken Kōbe-shi Kumoidōri" 34.69512  135.1979
## 5 Italian/French "Hyōgo-ken Kōbe-shi Kumoidōri" 34.69512  135.1979
## 6 Italian/French "Hyōgo-ken Kōbe-shi Kumoidōri" 34.69512  135.1979
##   reserve_visitors reserve_date visit_time reserve_time day_number genre
## 1               NA         <NA>       <NA>         <NA>          1     8
## 2               NA         <NA>       <NA>         <NA>          2     8
## 3               NA         <NA>       <NA>         <NA>          3     8
## 4                2   2017-01-02   18:00:00     23:00:00          5     8
## 5               11   2017-01-04   19:00:00     21:00:00          5     8
## 6                2   2017-01-05   19:00:00     21:00:00          5     8
##   median max     mean       sd
## 1   36.5  68 34.91667 13.76344
## 2   36.5  68 34.91667 13.76344
## 3   36.5  68 34.91667 13.76344
## 4   36.5  68 34.91667 13.76344
## 5   36.5  68 34.91667 13.76344
## 6   36.5  68 34.91667 13.76344
```

```
summary(modelData)
```

```
##       ID              visit_date                     visitors
##  Length:84201       Min.   :2017-01-01 00:00:00   Min.   :  1.00
##  Class :character   1st Qu.:2017-01-27 00:00:00   1st Qu.: 11.00
##  Mode  :character   Median :2017-02-18 00:00:00   Median : 21.00
##                     Mean   :2017-02-17 06:05:39   Mean   : 24.48
##                     3rd Qu.:2017-03-11 00:00:00   3rd Qu.: 34.00
##                     Max.   :2017-03-31 00:00:00   Max.   :877.00
##
##  day_of_week         holiday_flg      air_genre_name
##  Length:84201       Min.   :0.00000   Length:84201
##  Class :character   1st Qu.:0.00000   Class :character
##  Mode  :character   Median :0.00000   Mode  :character
##                     Mean   :0.04031
##                     3rd Qu.:0.00000
##                     Max.   :1.00000
##
##  air_area_name         latitude       longitude     reserve_visitors
##  Length:84201       Min.   :33.21   Min.   :130.2   Min.   :  1.0
##  Class :character   1st Qu.:34.69   1st Qu.:135.2   1st Qu.:  2.0
```

```
##  Mode  :character   Median :35.66   Median :139.7   Median :  3.0
##                     Mean   :35.86   Mean   :137.4   Mean   :  4.2
##                     3rd Qu.:35.69   3rd Qu.:139.8   3rd Qu.:  4.0
##                     Max.   :44.02   Max.   :144.3   Max.   :100.0
##                                                     NA's   :50805
##   reserve_date                 visit_time        reserve_time
##  Min.   :2017-01-01 00:00:00   Length:84201       Length:84201
##  1st Qu.:2017-01-25 00:00:00   Class :character   Class :character
##  Median :2017-02-16 00:00:00   Mode  :character   Mode  :character
##  Mean   :2017-02-15 00:11:19
##  3rd Qu.:2017-03-08 00:00:00
##  Max.   :2017-03-31 00:00:00
##  NA's   :50805
##    day_number        genre           median           max
##  Min.   :1.000   3      :23026   Min.   :  1.00   Min.   :  2.00
##  1st Qu.:3.000   1      :13938   1st Qu.: 13.00   1st Qu.: 34.00
##  Median :4.000   8      :12791   Median : 22.00   Median : 52.00
##  Mean   :4.164   2      :10224   Mean   : 23.51   Mean   : 56.31
##  3rd Qu.:6.000   12     : 6649   3rd Qu.: 33.00   3rd Qu.: 70.00
##  Max.   :7.000   4      : 6184   Max.   :116.00   Max.   :877.00
##                  (Other):11389
##       mean              sd
##  Min.   :  1.044   Min.   :  0.000
##  1st Qu.: 14.171   1st Qu.:  6.788
##  Median : 23.389   Median : 10.284
##  Mean   : 24.479   Mean   : 11.255
##  3rd Qu.: 33.771   3rd Qu.: 14.328
##  Max.   :119.372   Max.   :101.371
##
```

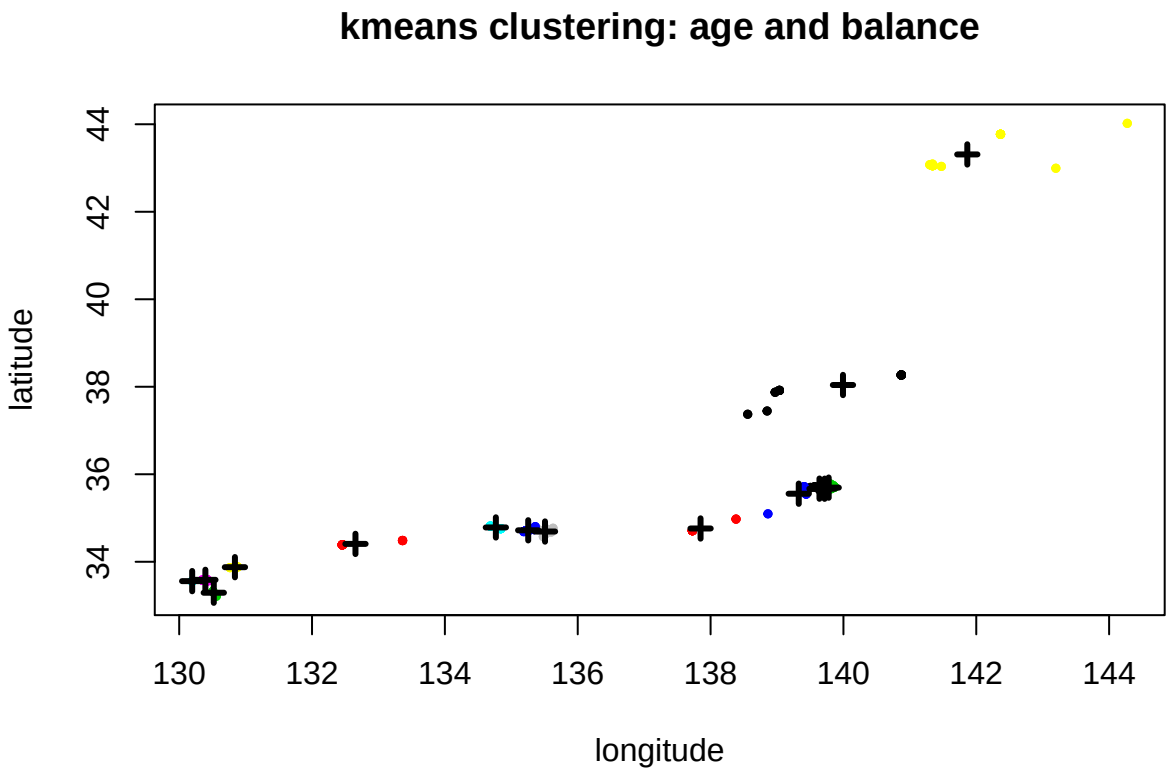
Having too many different factors for a variable in the model, may cause problems while trying to solve the trees. To fix that, we are going to creat a cluster with the variables "longitude" and "latitude" trying to reflect the data of the diferent locations included in the variable air_area_name.

```
#Creat new data set restaurants with just the longitude and latitude by ID
restaurants <-modelData %>% group_by(ID) %>%
  select(longitude, latitude) %>%
  summarise(longitude = first(longitude),
            latitude =first (latitude))
```

```
## Adding missing grouping variables: `ID`
```

```
#Generate the cluster with 15 centers (qualitative number of groups found in the exploration analysis)
longitude<- restaurants$longitude
latitude <- restaurants$latitude
kmeansClustering <- data.frame (longitude, latitude)
kmeansObj <- kmeans(kmeansClustering, centers = 15)

#plot the cluster:
colours <- kmeansObj$cluster
plot(kmeansClustering ,col=colours,pch=19,cex=0.5,
     main = "kmeans clustering: age and balance")
points(kmeansObj$centers,pch=3,cex=1,lwd=3)
```




```
#Include the new variable cluster_area in the data set restaurants:
restaurants ["cluster_area"]<-kmeansObj$cluster

#Join restaurant data set with the general data set modelData:
modelData<-merge(modelData, restaurants[,c(1, 4)], by="ID" )

plotData <- as.data.frame(cbind(kmeansClustering,
                                as.character(kmeansObj$cluster)))

g <- ggmap(map) +
  geom_point(data = plotData,
             aes(x = longitude, y = latitude,
                 color = as.character(kmeansObj$cluster))) +

  labs(x = "Longitude", y = "Latitude",
       title = "Restaurants by cluster") +

  theme(axis.text.x = element_text(size = 7),
        axis.text.y = element_text(size = 7),
        legend.position = "none")
```

Random Forest

Random forest is the method choose to creat the model. After trying linear model and tree, we have found the random forest has more accuracy and performs better trying to predict the visitors for a concrete restaurant and day.

First, Correct the type of the variables.

```
#adjust the type of the variables:
modelData$cluster_area<-as.factor(modelData$cluster_area)
modelData$air_genre_name<-as.factor(modelData$air_genre_name)
modelData$air_area_name<-as.factor(modelData$air_area_name)
```

Let's creat two subsets for training and testing the model:

```
#creat Train & Test sets:
set.seed(100)
split <- sample.split(modelData, SplitRatio = 0.75)

dataTrain <- subset(modelData, split == TRUE)
dataTest <- subset (modelData, split == FALSE)

summary(is.na(dataTrain))
```

```
##      ID          visit_date       visitors       day_of_week
##  Mode :logical   Mode :logical   Mode :logical   Mode :logical
##  FALSE:63151     FALSE:63151     FALSE:63151     FALSE:63151
##
##  holiday_flg     air_genre_name  air_area_name    latitude
##  Mode :logical   Mode :logical   Mode :logical   Mode :logical
##  FALSE:63151     FALSE:63151     FALSE:63151     FALSE:63151
##
##  longitude       reserve_visitors reserve_date    visit_time
##  Mode :logical   Mode :logical    Mode :logical   Mode :logical
##  FALSE:63151     FALSE:25076      FALSE:25076     FALSE:25076
##                  TRUE :38075      TRUE :38075     TRUE :38075
##  reserve_time    day_number        genre           median
##  Mode :logical   Mode :logical   Mode :logical   Mode :logical
##  FALSE:25076     FALSE:63151     FALSE:63151     FALSE:63151
##  TRUE :38075
##     max             mean             sd          cluster_area
##  Mode :logical   Mode :logical   Mode :logical   Mode :logical
##  FALSE:63151     FALSE:63151     FALSE:63151     FALSE:63151
##
```

```
str(dataTrain)
```

```
## 'data.frame':    63151 obs. of  20 variables:
##  $ ID              : chr  "\"restaurant_ 1\"" "\"restaurant_ 1\"" "\"restaurant_ 1\"" "\"restaurant_
##  $ visit_date      : POSIXct, format: "2017-01-02" "2017-01-03" ...
##  $ visitors        : int  10 38 31 22 22 22 17 32 32 32 ...
##  $ day_of_week     : chr  "Monday" "Tuesday" "Wednesday" "Friday" ...
##  $ holiday_flg     : int  1 1 0 0 0 0 0 1 1 1 ...
##  $ air_genre_name  : Factor w/ 14 levels "\"Creative cuisine\"",..: 10 10 10 10 10 10 10 10 10 10 ..
##  $ air_area_name   : Factor w/ 103 levels "\"Fukuoka-ken Fukuoka-shi Daimyō\"",..: 28 28 28 28 28 28
##  $ latitude        : num  34.7 34.7 34.7 34.7 34.7 ...
##  $ longitude       : num  135 135 135 135 135 ...
##  $ reserve_visitors: int  NA NA NA 2 11 2 NA 14 2 3 ...
##  $ reserve_date    : POSIXct, format: NA NA ...
##  $ visit_time      : chr  NA NA NA "18:00:00" ...
##  $ reserve_time    : chr  NA NA NA "23:00:00" ...
##  $ day_number      : num  1 2 3 5 5 5 7 1 1 1 ...
##  $ genre           : Factor w/ 14 levels "1","2","3","4",..: 8 8 8 8 8 8 8 8 8 8 ...
```

```
##  $ median          : num  36.5 36.5 36.5 36.5 36.5 36.5 36.5 36.5 36.5 36.5 ...
##  $ max             : num  68 68 68 68 68 68 68 68 68 68 ...
##  $ mean            : num  34.9 34.9 34.9 34.9 34.9 ...
##  $ sd              : num  13.8 13.8 13.8 13.8 13.8 ...
##  $ cluster_area    : Factor w/ 15 levels "1","2","3","4",..: 12 12 12 12 12 12 12 12 12 12 ...
```

There are some varibles not useful for the model. Let's study them one by one. - ID. We can not use it as a variable in the model. Factors can not have to many diferent values in a variable - Visit day and day of the week. We already convert this variable in number of the week. We have not take into account the month in the model. - Air genre name. Is substituted by the new numerical variable genre - Air area name. Is substituted by the cluster area, because of the impossibility to use so many factors - Latitud and Longitude. The information of this variables is also contained in the cluster area, using them do not imporve our model, so we remove it. - Reservation data. As we have only few restaurants with any information about reservations (20% aprox.), and also, only few restaurants with reservations in April for making the prediction, we decided not to use this variables in our model. - Holiday flag. Also remove even though seems meaninful variable.

Considering these, we are selecting the variables that we consider more important for our model:

```
#Select the subset with the best variables choosen for the mmodel:
dataTrainSub<- dataTrain %>%
  select (visitors, day_number, median, max,
          mean, cluster_area, genre)


#Creat a Random forest model with the variables selected
Tree <- randomForest(visitors ~., data= dataTrainSub, ntree=100, importance=TRUE)

#Check the importance of the variables
Tree$importance
```

```
##                %IncMSE IncNodePurity
## day_number   107.89416     2058316.4
## median       115.73777     3708253.0
## max           62.24723     2366638.3
## mean         205.67237     5515410.9
## cluster_area  36.98451      491791.4
## genre         33.35108      525744.4
```

Let's check which accuracy we have in the same data train, to get an idea about the precision of the model and to be aware of overfitting:

```
#make the prediction:
predictionTrain<-predict(Tree, newdata = dataTrainSub)
predictionTrain <- as.numeric(predictionTrain)
dataTrainSub<-cbind(dataTrainSub,predictionTrain)

#Check r2:
r2Train <- rSquared(dataTrainSub$visitors,
                    resid = dataTrainSub$visitors-predictionTrain)
r2Train
```

```
##           [,1]
## [1,] 0.7029254
```

Now, let's use the model in the data set:

```
#Creat the subset of the data test:
dataTestSub <- dataTest %>% select (visitors, day_number, median, max, mean, cluster_area, genre)
```

```r
#make the predicition
predictionTest<-predict(Tree, newdata = dataTestSub)
predictionTest <- as.numeric(predictionTest)
dataTestSub<-cbind(dataTestSub,predictionTest)

#Check r2:
r2Test <- rSquared(dataTestSub$visitors, resid = dataTestSub$visitors-predictionTest)
r2Test
```

```
##           [,1]
## [1,] 0.6116594
```

From the result, we can see we've got a optimistic prediction.

Prediction

As final data set to generate the model we will use both test and train data togheter. This will our model will be feed with more data and we hope that improve the accuracy of the model while predicting.

```
#Final data set for generating the model
finalData <- modelData %>%
  select(visitors, day_number, median,
         max, mean, cluster_area, genre)
```

Now, let's work with the data to predict.

Generating the new data file

To make the predictions for the month of April, we are provided by a csv file with the name of the restaurants and the date (day/month/year). To be able to apply our predictions, we have joined with this data the other data relative to the other variables used in the model. - day_number - transforming the date into a single number from 1 to 7 - median of visitors - the historical data we get from each restaurant - max of visitors- - mean of visitors - cluster area of the restaurant - genre of the restaurant.

```
#let's read the data to predict
predictData <- read.csv(file = "PredictData.csv",
                        header = TRUE,
                        sep = ",",
                        stringsAsFactors = FALSE)
#Check the data:
str(predictData)
```

```
## 'data.frame':    15770 obs. of  13 variables:
##  $ X             : int  1 2 3 4 5 6 7 8 9 10 ...
##  $ ID            : chr  "\"restaurant_113\"" "\"restaurant_113\"" "\"restaurant_113\"" "\"restaurant
##  $ Date          : chr  " 2017-04-01" " 2017-04-02" " 2017-04-03" " 2017-04-04" ...
##  $ X.visitors.   : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ latitude      : num  35.7 35.7 35.7 35.7 35.7 ...
##  $ longitude     : num  140 140 140 140 140 ...
##  $ median        : num  17 17 17 17 17 17 17 17 17 17 ...
##  $ max           : int  54 54 54 54 54 54 54 54 54 54 ...
##  $ mean          : num  18 18 18 18 18 ...
##  $ cluster_area  : int  15 15 15 15 15 15 15 15 15 15 ...
##  $ air_genre_name: chr  "Dining bar" "Dining bar" "Dining bar" "Dining bar" ...
##  $ day_of_week   : chr  "Saturday" "Sunday" "Monday" "Tuesday" ...
##  $ day_number    : int  6 7 1 2 3 4 5 6 1 2 ...
```

```
str(finalData)
```

```
## 'data.frame':    84201 obs. of  7 variables:
##  $ visitors    : int  10 38 31 22 22 22 45 17 32 32 ...
##  $ day_number  : num  1 2 3 5 5 5 6 7 1 1 ...
##  $ median      : num  36.5 36.5 36.5 36.5 36.5 36.5 36.5 36.5 36.5 36.5 ...
##  $ max         : num  68 68 68 68 68 68 68 68 68 68 ...
##  $ mean        : num  34.9 34.9 34.9 34.9 34.9 ...
##  $ cluster_area: Factor w/ 15 levels "1","2","3","4",..: 12 12 12 12 12 12 12 12 12 12 ...
##  $ genre       : Factor w/ 14 levels "1","2","3","4",..: 8 8 8 8 8 8 8 8 8 8 ...
```

```
#check missing data:
summary(is.na(predictData))
```

```
##      X               ID              Date         X.visitors.
##  Mode :logical   Mode :logical   Mode :logical   Mode :logical
##  FALSE:15770     FALSE:15770     FALSE:15770     FALSE:15770
##
##   latitude       longitude         median           max
##  Mode :logical   Mode :logical   Mode :logical   Mode :logical
##  FALSE:15754     FALSE:15754     FALSE:15754     FALSE:15754
##  TRUE :16        TRUE :16        TRUE :16        TRUE :16
##     mean         cluster_area    air_genre_name  day_of_week
##  Mode :logical   Mode :logical   Mode :logical   Mode :logical
##  FALSE:15754     FALSE:15754     FALSE:15754     FALSE:15770
##  TRUE :16        TRUE :16        TRUE :16
##  day_number
##  Mode :logical
##  FALSE:15770
##
```

```r
#two restaurants does not have any reserve or visits in the training data.
#Let's impute the numbers with 0
predictData[is.na(predictData)] <- 0

#Fix the name of the column visitors:
colnames(predictData)[colnames(predictData)=="X.visitors."] <- "visitors"

#eliminate variables not usefull for the model:
predictData$X <- NULL
predictData$Date <- NULL
predictData$day_of_week <- NULL
predictData$ID <- NULL
predictData$longitude <-NULL
predictData$latitude <-NULL

#add  genre as a factor, with the same order as done for the prediction
predictData <- predictData %>%
  mutate(genre = ifelse(air_genre_name == "Cafe/Sweets", 1,
                ifelse(air_genre_name =="Dining bar", 2,
                ifelse(air_genre_name == "Izakaya", 3,
                ifelse(air_genre_name == "Bar/Cocktail", 4,
                ifelse(air_genre_name == "Western food", 5,
                ifelse(air_genre_name == "Other", 6,
                ifelse(air_genre_name == "Karaoke/Party" , 7,
                ifelse(air_genre_name == "Italian/French", 8,
                ifelse(air_genre_name == "Yakiniku/Korean food", 9,
                ifelse(air_genre_name == "Okonomiyaki/Monja/Teppanyaki", 10,
                ifelse(air_genre_name ==  "Creative cuisine", 11,
                ifelse(air_genre_name ==  "Japanese food", 12,
                ifelse(air_genre_name ==  "International cuisine"  , 13,
                ifelse(air_genre_name ==  "Asian"  , 14,0))))))))))))))

summary(factor(predictData$genre, seq(1,14)) == predictData$genre)
```

```
##    Mode    TRUE    NA's
## logical   15754      16
```

```
predictData$genre <- factor(predictData$genre, seq(1,14))

unique(predictData$genre)
```

```
##  [1] 2    3    1    4    8    10   6    13   12   9    7    5    11   14
## [15] <NA>
## Levels: 1 2 3 4 5 6 7 8 9 10 11 12 13 14
```

```
# the categorical variable with the genre names can be now removed:
predictData$air_genre_name <-NULL

#change the type of the variables to adapt it to the model
predictData$visitors <- as.numeric(predictData$visitors)
predictData$day_number <- as.numeric(predictData$day_number)
predictData[predictData$cluster_area==0, 5] <- 2
predictData$cluster_area <- as.factor(predictData$cluster_area)

str(predictData)
```

```
## 'data.frame':    15770 obs. of  7 variables:
##  $ visitors    : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ median      : num  17 17 17 17 17 17 17 17 17 17 ...
##  $ max         : num  54 54 54 54 54 54 54 54 54 54 ...
##  $ mean        : num  18 18 18 18 18 ...
##  $ cluster_area: Factor w/ 15 levels "1","2","3","4",..: 15 15 15 15 15 15 15 15 15 15 ...
##  $ day_number  : num  6 7 1 2 3 4 5 6 1 2 ...
##  $ genre       : Factor w/ 14 levels "1","2","3","4",..: 2 2 2 2 2 2 2 2 2 2 ...
```

```
str(finalData)
```

```
## 'data.frame':    84201 obs. of  7 variables:
##  $ visitors    : int  10 38 31 22 22 22 45 17 32 32 ...
##  $ day_number  : num  1 2 3 5 5 5 6 7 1 1 ...
##  $ median      : num  36.5 36.5 36.5 36.5 36.5 36.5 36.5 36.5 36.5 36.5 ...
##  $ max         : num  68 68 68 68 68 68 68 68 68 68 ...
##  $ mean        : num  34.9 34.9 34.9 34.9 34.9 ...
##  $ cluster_area: Factor w/ 15 levels "1","2","3","4",..: 12 12 12 12 12 12 12 12 12 12 ...
##  $ genre       : Factor w/ 14 levels "1","2","3","4",..: 8 8 8 8 8 8 8 8 8 8 ...
```

```
finalData$cluster_area <- as.integer(finalData$cluster_area)
finalData$genre <- as.integer(finalData$genre)
predictData$cluster_area <- as.integer(predictData$cluster_area)
predictData$genre  <- as.integer(predictData$genre)

#concatenate the two tables two have the same factors.
sumData <-rbind(finalData, predictData)

sumData$cluster_area <- as.factor(sumData$cluster_area)
sumData$genre <- as.factor(sumData$genre)

#separate the two data frames:
nrow(finalData)
```

```
## [1] 84201
```

```
finalData <- sumData[1:nrow(finalData),]
predictData <- sumData[(nrow(finalData)+1):nrow(sumData),]
```

Generate the model and make the prediction

```
#creat a Random forest model with the variables selected
finalModel <- randomForest(visitors ~., data=finalData,
                           ntree=100, importance=TRUE)

#Check the importance of the variables
finalModel$importance
```

```
##                %IncMSE IncNodePurity
## day_number   108.78728     2676045.7
## median       128.64478     5334812.4
## max           57.91476     2507352.7
## mean         213.92753     7543425.9
## cluster_area  37.74297      631208.3
## genre         32.66714      606018.4
```

```
#make the prediction in the same data train:
tPrediction<-predict(finalModel, newdata = finalData)
tPrediction <- as.numeric(tPrediction)
finalData<-cbind(finalData,tPrediction)

r2tfinal <- rSquared(finalData$visitors, resid = finalData$visitors-tPrediction)
r2tfinal
```

```
##           [,1]
## [1,] 0.6840432
```

```
#Use the model selected to make the final predictions:
finalPrediction <- predict(finalModel, newdata = predictData)
predictData ["prediction"] <- finalPrediction
```

Now we are adjust our prediction data in the required format. ##Submission

```
#Read again the data:
predictData2 <- read.csv(file = "PredictData.csv",
                         header = TRUE,
                         sep = ",",
                         stringsAsFactors = FALSE)



predictData2["X.visitors."]<- predictData$prediction

#generate the table submission:
submission <- predictData2 %>% mutate (ID= paste(substr(ID,2,nchar(ID)-1),
                                                 substr(Date,2,nchar(Date)), sep= "_")) %>%
  select(ID,X.visitors.)
names(submission)<-c("ID","visitors")

#round the number of visitors
submission$visitors <- round(submission$visitors)
```

```r
#extract the csv file

write.csv(submission, file="submission.csv",
          quote = FALSE,
          row.names= FALSE)
```

Conclusion

1. The aim of this project was to predict the number of visitors per restaurant given information about date, location, genre and reservations.
2. Some of the variables were manipulated in order to use them within the prediction models.
3. New variables were also created from the available data, which prove to be an effective strategy.
4. A random forest was proposed and refined in order to minimize the error on prediction.
5. Final model had a resultant Rsquare = 61% and included the following variables: day_number, median of visitors, max of visitors, mean of visitors, cluster area of the restaurant, genre of the restaurant.

Further thoughts

1. It has been a challenging and engaging project.
2. Despite reaching reasonably good results, further improvements can be done to future models:

a. Use holiday flag as a variable.

b. Create a new binary feature for the days before a holiday flag, as usually people go to restaurants the night before.

c. Fill the lack of information of the restaurants without an historic, imputing the data instead of using 0.

d. Add information regarding weather and prime time events, refined by location.

e. Try other models like neural network.

Reference

[1] R Core Team (2016). R: A language and environment for statistical computing. R Foundation for Statistical Computing, Vienna, Austria. URL https://www.R-project.org/.

[2]Hadley Wickham, Romain François, Lionel Henry and Kirill Müller (2018). dplyr: A Grammar of Data Manipulation. R package version 0.7.6. URL https://CRAN.R-project.org/package=dplyr

[3]David James and Kurt Hornik (2018). chron: Chronological Objects which Can Handle Dates and Times. R package version 2.3-53. URL https://CRAN.R-project.org/package=chron

[4]D. Kahle and H. Wickham. ggmap: Spatial Visualization with ggplot2. The R Journal, 5(1), 144-161. URL http://journal.r-project.org/archive/2013-1/kahle-wickham.pdf

[5]A. Liaw and M. Wiener (2002). Classification and Regression by randomForest. R News 2(3), 18–22. URL https://CRAN.R-project.org/doc/Rnews/

[6]Jarek Tuszynski (2018). caTools: Tools: moving window statistics, GIF, Base64, ROC AUC, etc.. R package version 1.17.1.1. URL https://CRAN.R-project.org/package=caTools

[7]Venables, W. N. & Ripley, B. D. (2002) Modern Applied Statistics with S. Fourth Edition. Springer, New York. ISBN 0-387-95457-0 URL http://www.stats.ox.ac.uk/pub/MASS4

[8]Max Kuhn. Contributions from Jed Wing, Steve Weston, Andre Williams, Chris Keefer, Allan Engelhardt, Tony Cooper, Zachary Mayer, Brenton Kenkel, the R Core Team, Michael Benesty, Reynald Lescarbeau, Andrew Ziem, Luca Scrucca, Yuan Tang, Can Candan and Tyler Hunt. (2018). caret: Classification and Regression Training. R package version 6.0-81. URL https://CRAN.R-project.org/package=caret

[9]David Meyer, Evgenia Dimitriadou, Kurt Hornik, Andreas Weingessel and Friedrich Leisch (2018). e1071: Misc Functions of the Department of Statistics, Probability Theory Group (Formerly: E1071), TU Wien. R package version 1.7-0.URL https://CRAN.R-project.org/package=e1071

[10]citation for class balance Evans & Cushman 2009 citation for model selection and significance test Murphy et al., 2010 citation for other methods Evans et al., 2011. Evans JS, Murphy MA (2018).rfUtilities. R package version 2.1-3, URL https://cran.r-project.org/package=rfUtilities.

[11]Yachen Yan (2016). MLmetrics: Machine Learning Evaluation Metrics. R package version 1.1.1. URL https://CRAN.R-project.org/package=MLmetrics