Recruit Restaurant Visitor Forecasting

Yoan Bidart

2/24/2018

Introduction

For the Kaggle Competition "Recruit Restaurant Visitor Forecasting", the aim is to predict the number of visitors for various restaurants in Japan. We have multiple datasets so some features engineering is needed as well as some machine learning skills.

Preprocessing

```
library(dplyr)
require(lubridate)
library(caret)
library(ModelMetrics)
library(ggplot2)
library(gridExtra)
```

We choose to use the air datasets, and merge together to match all the informations we need on the restaurants.

```
airReserve <- read.csv(file="air_reserve.csv")
airInfo <- read.csv(file="air_store_info.csv")
airVisit <- read.csv(file="air_visit_data.csv")
date <- read.csv(file="date_info.csv")
str(airReserve)
```

```
## 'data.frame':    92378 obs. of  4 variables:
##  $ air_store_id    : Factor w/ 314 levels "air_00a91d42b08b08d9",..: 168 270 270 168 271 271 271 72 1
##  $ visit_datetime  : Factor w/ 4975 levels "2016-01-01 19:00:00",..: 1 1 1 2 2 3 3 4 4 5 ...
##  $ reserve_datetime: Factor w/ 7513 levels "2016-01-01 01:00:00",..: 6 8 8 6 1 6 5 17 9 20 ...
##  $ reserve_visitors: int  1 3 6 2 5 2 4 2 2 2 ...
```

```
str(airInfo)
```

```
## 'data.frame':    829 obs. of  5 variables:
##  $ air_store_id  : Factor w/ 829 levels "air_00a91d42b08b08d9",..: 43 391 826 528 413 500 789 342 32
##  $ air_genre_name: Factor w/ 14 levels "Asian","Bar/Cocktail",..: 7 7 7 7 7 7 7 7 7 7 ...
##  $ air_area_name : Factor w/ 103 levels "Fukuoka-ken Fukuoka-shi Daimyō",..: 28 28 28 28 75 75 75 75
##  $ latitude      : num  34.7 34.7 34.7 34.7 35.7 ...
##  $ longitude     : num  135 135 135 135 140 ...
```

```
str(airVisit)
```

```
## 'data.frame':    252108 obs. of  3 variables:
##  $ air_store_id: Factor w/ 829 levels "air_00a91d42b08b08d9",..: 604 604 604 604 604 604 604 604 604
##  $ visit_date  : Factor w/ 478 levels "2016-01-01","2016-01-02",..: 13 14 15 16 18 19 20 21 22 23 ..
##  $ visitors    : int  25 32 29 22 6 9 31 21 18 26 ...
```

```
str(date)
```

```
## 'data.frame':    517 obs. of  3 variables:
##  $ calendar_date: Factor w/ 517 levels "2016-01-01","2016-01-02",..: 1 2 3 4 5 6 7 8 9 10 ...
```

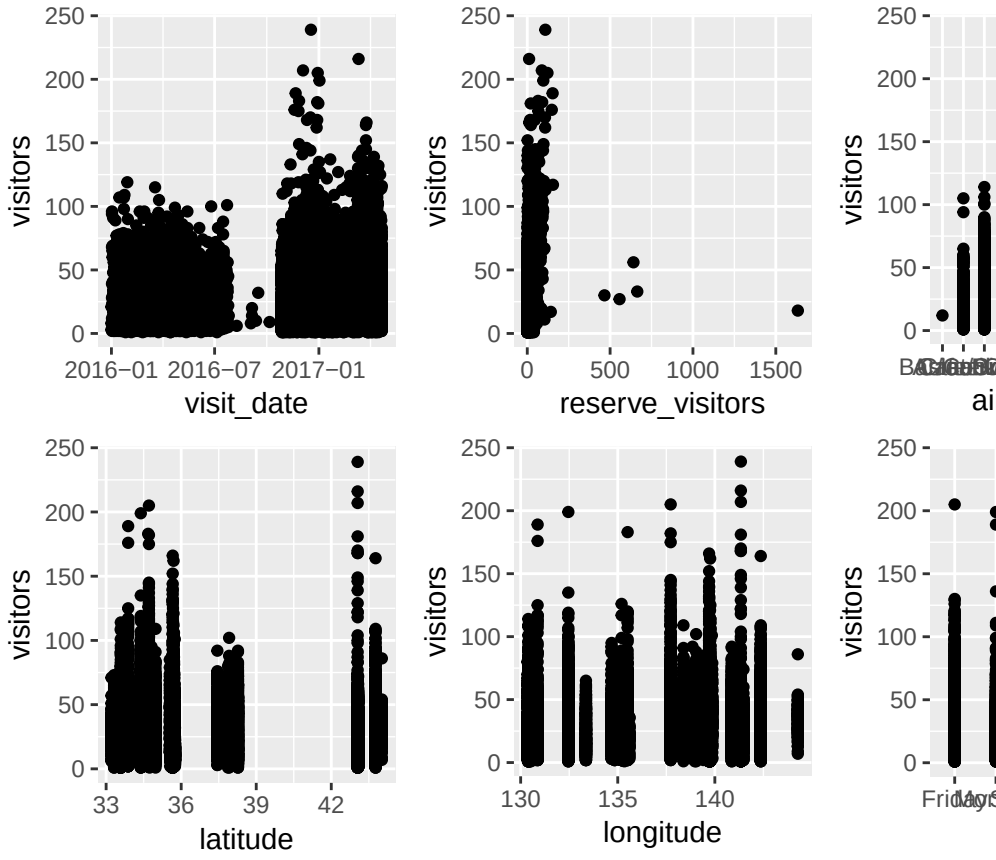
```
##  $ day_of_week  : Factor w/ 7 levels "Friday","Monday",..: 1 3 4 2 6 7 5 1 3 4 ...
##  $ holiday_flg  : int  1 1 1 0 0 0 0 0 0 0 ...
```

Feature Engineering

We will create some date features and merge the datasets.

```
#Date features engineering
airReserve$visit_datetime <- ymd_hms(airReserve$visit_datetime)
airReserve$visit_date <- date(airReserve$visit_datetime)
#Summarise visitors by date
airReserve <- select(airReserve, -(reserve_datetime))
airReserve <- aggregate(airReserve$reserve_visitors,
                        by=list(airReserve$air_store_id, airReserve$visit_date),
                        FUN=sum)
names(airReserve) <- c("air_store_id", "visit_date", "reserve_visitors")
#create weekday variable
airReserve$wday <- weekdays(airReserve$visit_date)
#Merge datasets
df <- merge(airReserve, airInfo)
df <- merge(df, airVisit, all.x=TRUE)
#Remove Nas
airTraining <- df[!is.na(df$visitors),]
#air area name is correlated with longitude and latitude so we will remove it
airTraining <- select(airTraining, -air_area_name)
airTraining <- arrange(airTraining, by=visit_date)
str(airTraining)
```

```
## 'data.frame':    28064 obs. of  8 variables:
##  $ air_store_id    : Factor w/ 314 levels "air_00a91d42b08b08d9",..: 168 270 271 49 72 134 153 168 27
##  $ visit_date      : Date, format: "2016-01-01" "2016-01-01" ...
##  $ reserve_visitors: int  3 9 5 6 4 11 67 4 34 2 ...
##  $ wday            : chr  "Friday" "Friday" "Friday" "Saturday" ...
##  $ air_genre_name  : Factor w/ 14 levels "Asian","Bar/Cocktail",..: 9 5 5 11 8 7 8 9 5 9 ...
##  $ latitude        : num  35.7 34.7 43.8 34.4 38.3 ...
##  $ longitude       : num  140 135 142 132 141 ...
##  $ visitors        : int  3 21 8 28 39 96 68 12 94 2 ...
```



Let's plot our features against visitors!

Machine Learning

Create train and test sets.

```
set.seed(12345)
inTrain <- createDataPartition(airTraining$visitors, p=0.7, list=FALSE)
airTrain <- airTraining[inTrain, ]
airTest <- airTraining[-inTrain, ]
```

Fit the model.

```
airTrainA <- select(airTrain, -c(air_store_id, wday))
controls <- trainControl(method="repeatedcv", number=10, repeats=3)
airfit <- train(visitors~., data=airTrainA, method="rf", metric="RMSE",
                trainControl=controls)
```

Prediction and evaluation.

```
airTestA <- select(airTest, -c(air_store_id, visitors, wday))
airPred <- predict(airfit, airTestA)
accu <- rmsle(airTest$visitors,airPred)
```

Our Root Mean Square Logarithmic Error is 9.2891611.