Data Mining for BI Supervised Learning Project

DATA MINING FOR BI

Supervised Learning Project

Loading libraries

```
library(RCurl)
library(e1071)
#install.packages("caret")
library(caret)
```

```
## Loading required package: lattice
```

```
## Loading required package: ggplot2
```

```
# install.packages("doSNOW")
library(doSNOW)
```

```
## Loading required package: foreach
```

```
## Loading required package: iterators
```

```
## Loading required package: snow
```

```
library(ipred)
# install.packages("xgboost")
library(xgboost)
#library(devtools)
#devtools::install_github('topepo/caret/pkg/caret')
```

Importing training set features using 'getURL' method and excluding the 4th col (i.e. week start date)

```
trfeat <- getURL("https://s3.amazonaws.com/drivendata/data/44/public/dengue_features_train.csv")
trfeat <-read.csv(text = trfeat)
tr <- trfeat[, -c(4)]
```

Importing training set labels: i.e., number of dengue cases by the same week, year, and city as in the training set features dataset,

```
trlabel <- getURL("https://s3.amazonaws.com/drivendata/data/44/public/dengue_labels_train.csv")
trlabel <- read.csv(text = trlabel)
```

Merging the training set features and lablels by city, year, and week of year

```
trfinal <- merge(tr, trlabel, by=c("city", "year", "weekofyear"))
train <- trfinal
```

```
str(trfinal)
```

```
## 'data.frame':    1456 obs. of  24 variables:
##  $ city                                 : Factor w/ 2 levels "iq","sj": 1 1 1 1 1 1 1 1 1 1 ...
##  $ year                                 : int  2000 2000 2000 2000 2000 2000 2000 2000 2000 2000 ...
##  $ weekofyear                           : int  26 27 28 29 30 31 32 33 34 35 ...
##  $ ndvi_ne                              : num  0.193 0.217 0.177 0.228 0.329 ...
##  $ ndvi_nw                              : num  0.132 0.276 0.173 0.145 0.322 ...
##  $ ndvi_se                              : num  0.341 0.289 0.204 0.254 0.254 ...
##  $ ndvi_sw                              : num  0.247 0.242 0.128 0.2 0.361 ...
##  $ precipitation_amt_mm                 : num  25.4 60.6 55.5 5.6 62.8 ...
##  $ reanalysis_air_temp_k                : num  297 297 296 295 296 ...
##  $ reanalysis_avg_temp_k                : num  298 298 297 296 298 ...
##  $ reanalysis_dew_point_temp_k          : num  295 295 296 293 294 ...
##  $ reanalysis_max_air_temp_k            : num  307 307 304 304 307 ...
##  $ reanalysis_min_air_temp_k            : num  293 291 293 289 292 ...
##  $ reanalysis_precip_amt_kg_per_m2      : num  43.2 46 64.8 24 31.8 ...
##  $ reanalysis_relative_humidity_percent : num  92.4 93.6 95.8 87.2 88.2 ...
##  $ reanalysis_sat_precip_amt_mm         : num  25.4 60.6 55.5 5.6 62.8 ...
##  $ reanalysis_specific_humidity_g_per_kg: num  16.7 16.9 17.1 14.4 15.4 ...
##  $ reanalysis_tdtr_k                    : num  8.93 10.31 7.39 9.11 9.5 ...
##  $ station_avg_temp_c                   : num  26.4 26.9 26.8 25.8 26.6 ...
##  $ station_diur_temp_rng_c              : num  10.8 11.6 11.5 10.5 11.5 ...
##  $ station_max_temp_c                   : num  32.5 34 33 31.5 33.3 32 34 33 34 34 ...
##  $ station_min_temp_c                   : num  20.7 20.8 20.7 14.7 19.1 17 19.9 20.5 19 20 ...
##  $ station_precip_mm                    : num  3 55.6 38.1 30 4 11.5 72.9 50.1 89.2 78 ...
##  $ total_cases                          : int  0 0 0 0 0 0 0 0 0 0 ...
```

grouping all features and dummy coding of features

```
train$city <- as.factor(train$city)
features <- c("city","year","weekofyear","ndvi_ne","ndvi_nw","ndvi_se",
"ndvi_sw", "precipitation_amt_mm", "reanalysis_air_temp_k", "reanalysis_avg_temp_k","reanalysis_dew_poi
"reanalysis_specific_humidity_g_per_kg", "reanalysis_tdtr_k",                  "station_avg_temp_c", "s
train <- train[, features]
dummy.vars <- dummyVars(~ ., data = train[, -c(1:3,24)])
train.dummy <- predict(dummy.vars, train[, -c(1:3,24)])
```

Imputation of missing values using bag imputation method

```
pre.process <- preProcess(train.dummy, method = "bagImpute")
imputed.data <- predict(pre.process, train.dummy)
```

```
## Warning in cprob[tindx] + pred: longer object length is not a multiple of
## shorter object length

## Warning in cprob[tindx] + pred: longer object length is not a multiple of
## shorter object length

## Warning in cprob[tindx] + pred: longer object length is not a multiple of
## shorter object length

## Warning in cprob[tindx] + pred: longer object length is not a multiple of
```

```
## shorter object length

## Warning in cprob[tindx] + pred: longer object length is not a multiple of
## shorter object length

## Warning in cprob[tindx] + pred: longer object length is not a multiple of
## shorter object length

## Warning in cprob[tindx] + pred: longer object length is not a multiple of
## shorter object length

## Warning in cprob[tindx] + pred: longer object length is not a multiple of
## shorter object length

## Warning in cprob[tindx] + pred: longer object length is not a multiple of
## shorter object length

## Warning in cprob[tindx] + pred: longer object length is not a multiple of
## shorter object length
```

```
train$ndvi_ne <- imputed.data[,1]
train$ndvi_nw <- imputed.data[,2]
train$ndvi_se <- imputed.data[,3]
train$ndvi_sw <- imputed.data[,4]
train$precipitation_amt_mm <- imputed.data[,5]
train$reanalysis_air_temp_k <- imputed.data[, 6]
train$reanalysis_avg_temp_k <- imputed.data[,7]
train$reanalysis_dew_point_temp_k <- imputed.data[,8]
train$reanalysis_max_air_temp_k <- imputed.data[,9]
train$reanalysis_min_air_temp_k <- imputed.data[,10]
train$reanalysis_precip_amt_kg_per_m2 <- imputed.data[,11]
train$reanalysis_relative_humidity_percent <- imputed.data[,12]
train$reanalysis_sat_precip_amt_mm <- imputed.data[,13]
train$reanalysis_specific_humidity_g_per_kg <- imputed.data[,14]
train$reanalysis_tdtr_k <- imputed.data[,15]
train$station_avg_temp_c <- imputed.data[,16]
train$station_diur_temp_rng_c <- imputed.data[,17]
train$station_max_temp_c <- imputed.data[,18]
train$station_min_temp_c <- imputed.data[,19]
train$station_precip_mm <- imputed.data[,20]
```

Checking any missing values in imputed data

```
anyNA(train)
```

```
## [1] FALSE
```

Splitting the training set 80:20, and checking the dimensions of split sets

```
set.seed(54321)
indexes <- createDataPartition(train$total_cases, times = 1, p = .8, list = FALSE)
deng.train <- train[indexes,]
deng.test <- train[-indexes,]
dim(deng.test)
```

```
## [1] 288  24
```

```
dim(deng.train)
```

```
## [1] 1168   24
```

Defining the training control using repeated cross validation

```
train.control <- trainControl(method = "repeatedcv",
                              number = 5,
                              repeats = 20,
                              search = "grid")
```

Defining the tuning grid

```
tune.grid <- expand.grid(eta = c(0.05, 0.06, 0.1),
                         nrounds = c(30, 60, 70),
                         max_depth = 3:8,
                         min_child_weight = c(2.0, 2.25, 2.5),
                         colsample_bytree = c(0.3, 0.4, 0.5),
                         gamma = 0,
                         subsample = 1)
```

Registering clusters on RAM to improve computer performance while running the ML algorithm

```
cl <- makeCluster(3, type = "SOCK")
registerDoSNOW(cl)
```

Subsetting the traiing data for selective predictors in the subset, based on regression tree analysis performed separately for determing the significant contributers

```
train2 = train[, c(1, 3, 4, 5, 6, 7, 11, 13, 17, 18, 24)]
```

Obtaining the regressor from training dataset using the tuning grid and train controls and selective predictors from decision tree analysis

```
caret.cv <- train(total_cases ~ .,
                  data = train2,
                  method = "xgbTree",
                  tuneGrid = tune.grid,
                  trControl = train.control)
```

Stopping the clusters started earlier to free RAM

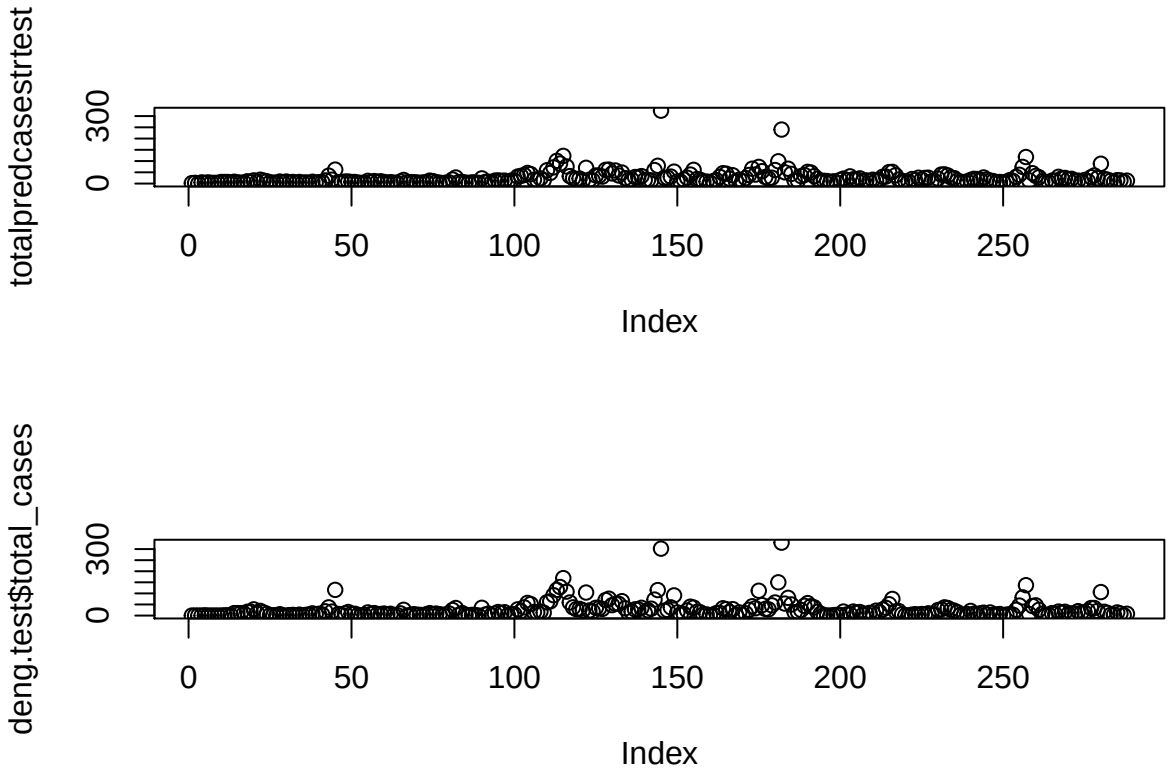
```
stopCluster(cl)
```

Predicting 'total_cases' (rounded to nearest 0 place of decimal) on split test-set using the regressor from split train-set

```
preds <- predict(caret.cv, deng.test)
totalpredcasestrtest <- round(preds)
```

Plotting predicted and acutal total_cases

```
par(mfrow=c(2,1))
plot(totalpredcasestrtest)
plot(deng.test$total_cases)
```



Checking mean absolute error (MAE)

```
actual = deng.test$total_cases
predicted = totalpredcasestrtest
mae <- function(error)
{
  mean(abs(error))
}
error <- (actual-predicted)
mae(error)
```

```
## [1] 7.295139
```

Importing the test set from the host site

```
testset <- getURL("https://s3.amazonaws.com/drivendata/data/44/public/dengue_features_test.csv")
testset <- read.csv(text=testset)
ts <- testset[, -c(4)]
names(ts)
```

```
##  [1] "city"
##  [2] "year"
##  [3] "weekofyear"
##  [4] "ndvi_ne"
##  [5] "ndvi_nw"
##  [6] "ndvi_se"
##  [7] "ndvi_sw"
##  [8] "precipitation_amt_mm"
##  [9] "reanalysis_air_temp_k"
## [10] "reanalysis_avg_temp_k"
## [11] "reanalysis_dew_point_temp_k"
## [12] "reanalysis_max_air_temp_k"
## [13] "reanalysis_min_air_temp_k"
## [14] "reanalysis_precip_amt_kg_per_m2"
## [15] "reanalysis_relative_humidity_percent"
## [16] "reanalysis_sat_precip_amt_mm"
## [17] "reanalysis_specific_humidity_g_per_kg"
## [18] "reanalysis_tdtr_k"
## [19] "station_avg_temp_c"
## [20] "station_diur_temp_rng_c"
## [21] "station_max_temp_c"
## [22] "station_min_temp_c"
## [23] "station_precip_mm"
```

```
ts$total_cases <- NA
```

Imputing missing values for the test set

```
ts <- ts[, features]
ts$city <- as.factor(ts$city)
ts$weekofyear <- as.numeric(ts$weekofyear)

tsdummy.vars <- dummyVars(~ ., data = ts[, -c(1:3,24)])
ts.dummy <- predict(tsdummy.vars, ts[, -c(1:3,24)])

tspre.process <- preProcess(ts.dummy, method = "bagImpute")
tsimputed.data <- predict(tspre.process, ts.dummy)
```

```
## Warning in cprob[tindx] + pred: longer object length is not a multiple of
## shorter object length

## Warning in cprob[tindx] + pred: longer object length is not a multiple of
## shorter object length

## Warning in cprob[tindx] + pred: longer object length is not a multiple of
## shorter object length

## Warning in cprob[tindx] + pred: longer object length is not a multiple of
## shorter object length

## Warning in cprob[tindx] + pred: longer object length is not a multiple of
## shorter object length

## Warning in cprob[tindx] + pred: longer object length is not a multiple of
```

```
## shorter object length

## Warning in cprob[tindx] + pred: longer object length is not a multiple of
## shorter object length

## Warning in cprob[tindx] + pred: longer object length is not a multiple of
## shorter object length

## Warning in cprob[tindx] + pred: longer object length is not a multiple of
## shorter object length

## Warning in cprob[tindx] + pred: longer object length is not a multiple of
## shorter object length
```

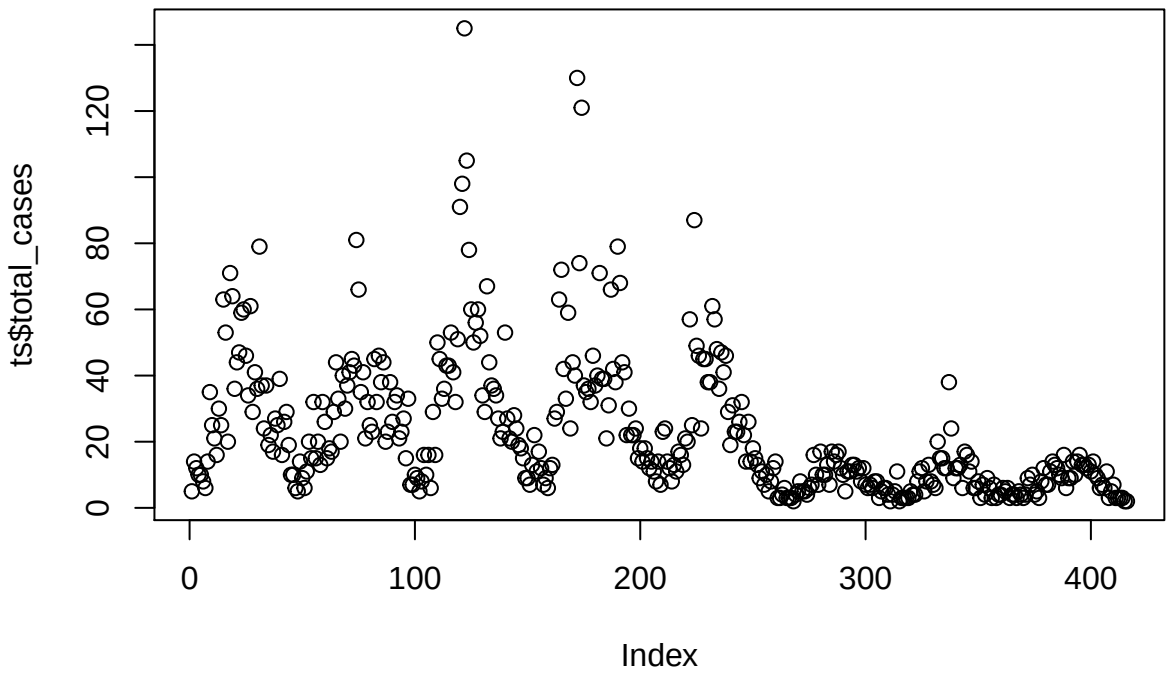
```
ts$ndvi_ne <- tsimputed.data[,1]
ts$ndvi_nw <- tsimputed.data[,2]
ts$ndvi_se <- tsimputed.data[,3]
ts$ndvi_sw <- tsimputed.data[,4]
ts$precipitation_amt_mm <- tsimputed.data[,5]
ts$reanalysis_air_temp_k <- tsimputed.data[, 6]
ts$reanalysis_avg_temp_k <- tsimputed.data[,7]
ts$reanalysis_dew_point_temp_k <- tsimputed.data[,8]
ts$reanalysis_max_air_temp_k <- tsimputed.data[,9]
ts$reanalysis_min_air_temp_k <- tsimputed.data[,10]
ts$reanalysis_precip_amt_kg_per_m2 <- tsimputed.data[,11]
ts$reanalysis_relative_humidity_percent <- tsimputed.data[,12]
ts$reanalysis_sat_precip_amt_mm <- tsimputed.data[,13]
ts$reanalysis_specific_humidity_g_per_kg <- tsimputed.data[,14]
ts$reanalysis_tdtr_k <- tsimputed.data[,15]
ts$station_avg_temp_c <- tsimputed.data[,16]
ts$station_diur_temp_rng_c <- tsimputed.data[,17]
ts$station_max_temp_c <- tsimputed.data[,18]
ts$station_min_temp_c <- tsimputed.data[,19]
ts$station_precip_mm <- tsimputed.data[,20]
```

Predicting the 'total_cases' for the test set using the regressor from the training set

```
ts$total_cases <- round(predict(caret.cv, ts))
```

Plotting the time-series for the total_cases in the test set

```
par(mfrow=c(1,1))
plot(ts$total_cases)
```



Downloading the submission form

```
Submitformat <- getURL("https://s3.amazonaws.com/drivendata/data/44/public/submission_format.csv")
submitformat2 <- read.csv(text=Submitformat)
```

Entering the predicted 'total_cases' from the test-set into the submission form

```
submitformat2$total_cases<- ts$total_cases
```

Exporting the submission form to local drive for uploading to the competition site

```
write.csv(submitformat2, "D://STUDY//MSIS//DM//submit031920xgb_send.csv", row.names = FALSE)
```