DMC@ISU: The 2015 Iowa State University Data Mining Cup Team

Creating Feature Matrix Version 0.3

Spring 2015, A Team as Strong as Steel

Last Day: May 19, 2015

I am using the following packages:

```
   library(magrittr)
   library(dplyr)
   library(tidyr)
   library(lubridate)
   library(ggplot2)
   library(rCharts)
   library(xtable)
   library(foreach)
   library(gtools)
   library(knitr)
   library(utils)
   source("~/dmc2015/ian/R/renm.R")

makeWideFeatureMatrix = function(w, uf, set) {
    ## Working with wide versions
    dropcols = c("userID", "couponsReceived", "couponID1", "couponID2", "couponID3",
        "couponsReceivedDate", "orderTimeDate", "batchID", "couponsExpire",
        "couponsSent")

    f1 = uf[, !(names(uf) %in% dropcols)] %>% left_join(w, by = "orderID") %>%
        arrange(orderID)
    f1 = f1[, !(grepl("ntimes_", names(f1)) | grepl("timesNotUsed_", names(f1)) |
        grepl("timesUsed_", names(f1)))]

    # check for names repeats
    data.frame(table(names(f1))) %>% arrange(Freq) %>% head

    # check for identical columns:
    nfeat = ncol(f1)
    f1 = f1[, !duplicated(t(f1))]

    # convert characters
    chars = names(f1)[sapply(1:ncol(f1), function(i) is.character(f1[, i]))]
    for (x in chars) f1[, x] = as.factor(f1[, x])

    factors = names(f1)[sapply(1:ncol(f1), function(i) is.factor(f1[, i]))]

    HTV = readRDS(paste0("~/dmc2015/data/featureMatrix/HTV", set, ".rds"))

    # train
    Xtrn = HTV$T %>% select(orderID, coupon1Used, coupon2Used, coupon3Used,
        basketValue) %>% left_join(f1, by = "orderID") %>% arrange(orderID)
    trn = list(X = Xtrn %>% select(-coupon1Used, -coupon2Used, -coupon3Used,
        -basketValue), y = Xtrn %>% select(orderID, coupon1Used, coupon2Used,
        coupon3Used, basketValue))
```

```
    # val
    Xval = HTV$V %>% select(orderID, coupon1Used, coupon2Used, coupon3Used,
        basketValue) %>% left_join(f1, by = "orderID") %>% arrange(orderID)
    val = list(X = Xval %>% select(-coupon1Used, -coupon2Used, -coupon3Used,
        -basketValue), y = Xval %>% select(orderID, coupon1Used, coupon2Used,
        coupon3Used, basketValue))

    # class
    Xcls = HTV$C %>% select(orderID) %>% left_join(f1, by = "orderID") %>% arrange(orderID)
    Xcls = HTV$C %>% select(orderID, coupon1Used, coupon2Used, coupon3Used,
        basketValue) %>% left_join(f1, by = "orderID") %>% arrange(orderID)
    cls = list(X = Xcls %>% select(-coupon1Used, -coupon2Used, -coupon3Used,
        -basketValue), y = Xcls %>% select(orderID, coupon1Used, coupon2Used,
        coupon3Used, basketValue))

    Fmat1 = list(train = trn, class = cls, validation = val)
    saveRDS(Fmat1, file = paste0("~/dmc2015/data/featureMatrix/featMat_based-on-HTV",
        set, "_WIDE_ver0.3.rds"))
}

makeLongFeatureMatrix = function(l, u, set) {
    ## Working with wide versions
    dropcols = c("userID", "couponsReceived", "couponID", "couponsReceivedDate",
        "orderTimeDate", "batchID", "couponsExpire", "couponsSent")

    f1 = u[, !(names(u) %in% dropcols)] %>% left_join(l, by = c("orderID", "couponCol")) %>%
        arrange(orderID, couponCol)
    f1 = f1[, !(grepl("ntimes_", names(f1)) | grepl("timesNotUsed_", names(f1)) |
        grepl("timesUsed_", names(f1)))]

    # check for names repeats
    data.frame(table(names(f1))) %>% arrange(-Freq) %>% head

    # check for identical columns:
    nfeat = ncol(f1)
    f1 = f1[, !duplicated(t(f1))]

    # convert characters
    chars = names(f1)[sapply(1:ncol(f1), function(i) is.character(f1[, i]))]
    for (x in chars) f1[, x] = as.factor(f1[, x])

    factors = names(f1)[sapply(1:ncol(f1), function(i) is.factor(f1[, i]))]

    HTV = readRDS(paste0("~/dmc2015/data/featureMatrix/HTV", set, ".rds"))
    names(HTV)
    dim(HTV$T)

    # train
    trn = HTV$T %>% select(orderID, coupon1Used, coupon2Used, coupon3Used, basketValue) %>%
        gather(colname, couponUsed, -orderID, -basketValue) %>% mutate(couponCol = gsub("coupon",
        "", gsub("Used", "", colname))) %>% select(orderID, couponCol, basketValue,
        couponUsed) %>% arrange(orderID, couponCol)
    trn$couponCol = as.numeric(trn$couponCol)
```

```
    Xtrn = trn %>% left_join(f1, by = c("orderID", "couponCol")) %>% arrange(orderID,
        couponCol)
    trn = list(X = Xtrn %>% select(-couponUsed, -basketValue), y = Xtrn %>%
        select(orderID, couponUsed, basketValue))

    # val
    val = HTV$V %>% select(orderID, coupon1Used, coupon2Used, coupon3Used, basketValue) %>%
        gather(colname, couponUsed, -orderID, -basketValue) %>% mutate(couponCol = gsub("coupon",
        "", gsub("Used", "", colname))) %>% select(orderID, couponCol, basketValue,
        couponUsed) %>% arrange(orderID, couponCol)
    val$couponCol = as.numeric(val$couponCol)

    Xval = val %>% left_join(f1, by = c("orderID", "couponCol")) %>% arrange(orderID,
        couponCol)
    val = list(X = Xval %>% select(-couponUsed, -basketValue), y = Xval %>%
        select(orderID, couponUsed, basketValue))

    # class
    cls = HTV$C %>% select(orderID, coupon1Used, coupon2Used, coupon3Used, basketValue) %>%
        gather(colname, couponUsed, -orderID, -basketValue) %>% mutate(couponCol = gsub("coupon",
        "", gsub("Used", "", colname))) %>% select(orderID, couponCol, basketValue,
        couponUsed) %>% arrange(orderID, couponCol)
    cls$couponCol = as.numeric(cls$couponCol)

    Xcls = cls %>% left_join(f1, by = c("orderID", "couponCol")) %>% arrange(orderID,
        couponCol)
    cls = list(X = Xcls %>% select(-couponUsed, -basketValue), y = Xcls %>%
        select(orderID, couponUsed, basketValue))

    Fmat1 = list(train = trn, class = cls, validation = val)
    saveRDS(Fmat1, file = paste0("~/dmc2015/data/featureMatrix/featMat_based-on-HTV",
        set, "_LONG_ver0.3.rds"))
}
```

0.1 Combine the features

```
## universal
ul = readRDS("../universal/combined/universalFeaturesCombined_long.rds")
uw = readRDS("../universal/combined/universalFeaturesCombined_wide.rds")

## set1
s1l = readRDS("../set1/combined/set1FeaturesCombined_long.rds")
s1w = readRDS("../set1/combined/set1FeaturesCombined_wide.rds")

# weichang's features
wc1 = readRDS("../set1/weichangsFeature.rds") %>% select(-userID)
names(wc1)[2] = "weichengProb"
s1l = s1l %>% left_join(wc1, by = "orderID")
s1w = s1w %>% left_join(wc1, by = "orderID")

# make the features
makeWideFeatureMatrix(s1w, uw, "set1")
```

```r
makeLongFeatureMatrix(s1l, ul, "set1")

## set2
s2l = readRDS("../set2/combined/set2FeaturesCombined_long.rds")
s2w = readRDS("../set2/combined/set2FeaturesCombined_wide.rds")

# weichang's features
wc2 = readRDS("../set2/weichangsFeature.rds") %>% select(-userID)
names(wc2)[2] = "weichengProb"
s2l = s2l %>% left_join(wc2, by = "orderID")
s2w = s2w %>% left_join(wc2, by = "orderID")

# make the features
makeWideFeatureMatrix(s2w, uw, "set2")
makeLongFeatureMatrix(s2l, ul, "set2")

## set3
s3l = readRDS("../set3/combined/set3FeaturesCombined_long.rds")
s3w = readRDS("../set3/combined/set3FeaturesCombined_wide.rds")

# weichang's features
wc3 = readRDS("../set3/weichangsFeature.rds") %>% select(-userID)
names(wc3)[2] = "weichengProb"
s3l = s3l %>% left_join(wc3, by = "orderID")
s3w = s3w %>% left_join(wc3, by = "orderID")

# make the features
makeWideFeatureMatrix(s3w, uw, "set3")
makeLongFeatureMatrix(s3l, ul, "set3")
```