Exploratory Data Analysis

# Media 6 Degrees 데이터

M6D에서는 광고 클릭과 관련하여 다음과 같은 세가지 문제에 관심이 있음

* 특징공학: 어떤 특성치를 어떻게 사용할 것인가?
* 사용자수준에서의 구매전환 예측: 언제 누가 클릭할 것인가?
* 입찰: 사용자에게 주어진 광고의 가치

어떤 사용자가 특정 광고를 클릭할 것인지(구매전환) 예측하기 위해 로지스틱 회귀

## 필요한 함수들 정의

# Author: Brian Dalessandro  
  
library(mgcv)

## Loading required package: nlme

## This is mgcv 1.8-12. For overview type 'help("mgcv-package")'.

# 일반화가법모형을 이용한 평활 함수  
plotrel <- function(x, y, b, title) {  
 # gam을 이용한 평활  
 g <- gam(as.formula("y ~ x"), family = "binomial",   
 data = set)  
 xs <- seq(min(x), max(x), length = 200)  
 p <- predict(g, newdata = data.frame(x = xs),  
 type = "response")  
 # 경험적 추정값 계산  
 if (length(unique(x)) > b) {  
 div <- floor(max(x) / b)  
 x\_b <- floor(x / div) \* div  
 c <- table(x\_b, y)  
 } else { c <- table(x, y) }  
   
 pact <- c[ , 2]/(c[ , 1]+c[, 2])  
 cnt <- c[ , 1]+c[ , 2]  
 xd <- as.integer(rownames(c))  
 plot(xs, p, type="l", main=title,  
 ylab = "P(Conversion | Ad, X)", xlab="X")  
 points(xd, pact, type="p", col="red")  
 rug(x+runif(length(x)))  
}  
  
  
library(plyr)  
# wMAE 도표와 추정 함수  
getmae <- function(p, y, b, title, doplot) {  
 # [0,1]로 정규화  
 max\_p <- max(p)  
 p\_norm <- p / max\_p  
 # b개의 빈으로 나누고 스케일 조정  
 bin <- max\_p \* floor(p\_norm \* b) / b  
 d <- data.frame(bin, p, y)  
 t <- table(bin)  
 summ <- ddply(d, .(bin), summarise, mean\_p = mean(p),  
 mean\_y = mean(y))  
 fin <- data.frame(bin = summ$bin, mean\_p = summ$mean\_p,  
 mean\_y = summ$mean\_y, t)  
 # wMAE 계산   
 num = 0  
 den = 0  
 for (i in c(1:nrow(fin))) {  
 num <- num + fin$Freq[i] \* abs(fin$mean\_p[i] -  
 fin$mean\_y[i])  
 den <- den + fin$Freq[i]  
 }  
 wmae <- num / den  
 if (doplot == 1) {  
 plot(summ$bin, summ$mean\_p, type = "p",  
 main = paste(title," MAE =", wmae),  
 col = "blue", ylab = "P(C | AD, X)",  
 xlab = "P(C | AD, X)")  
 points(summ$bin, summ$mean\_y, type = "p", col = "red")  
 rug(p)  
 }  
  
 return(wmae)  
}  
  
  
library(ROCR)

## Loading required package: gplots

##   
## Attaching package: 'gplots'

## The following object is masked from 'package:stats':  
##   
## lowess

get\_auc <- function(ind, y) {  
 pred <- prediction(ind, y)  
 perf <- performance(pred, 'auc', fpr.stop = 1)  
 auc <- as.numeric(substr(slot(perf, "y.values"), 1, 8),  
 double)  
 return(auc)  
}  
  
  
# 주어진 특징 세트에 대한 교차확인된 성능 측도 구하는 함수  
getxval <- function(vars, data, folds, mae\_bins) {  
 # 각 관측값을 묶음에 배정  
 data["fold"] <- floor(runif(nrow(data)) \* folds) + 1  
 auc <- c()  
 wmae <- c()  
 fold <- c()  
 # 공식 만들기  
 f = as.formula(paste("y", "~", paste(vars, collapse = "+")))  
 for (i in c(1:folds)) {  
 train <- data[(data$fold != i), ]  
 test <- data[(data$fold == i), ]  
 mod\_x <- glm(f, data=train, family = binomial(logit))  
 p <- predict(mod\_x, newdata = test, type = "response")  
 # wMAE  
 wmae <- c(wmae, getmae(p, test$y, mae\_bins, "dummy", 0))  
 fold <- c(fold, i)  
 auc <- c(auc, get\_auc(p, test$y))  
 }  
 return(data.frame(fold, wmae, auc))  
}

## 모형과 도표

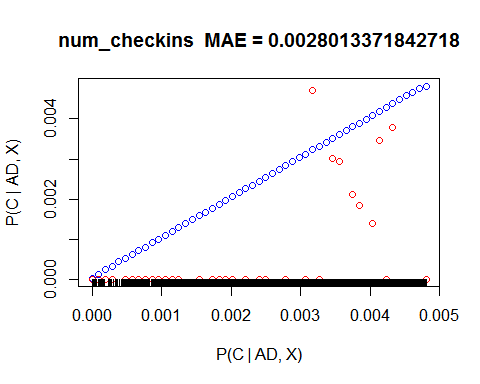
# 데이터 읽어서 변수를 살펴보고 데이터 분할하기  
setwd("D:/doingdatascience/dds\_datasets/")  
file <- "dds\_ch5\_binary-class-dataset.txt"  
set <- read.delim(file, header = TRUE, sep = "\t") #변수들이 tap으로 구분된파일..  
names(set)

## [1] "at\_buy\_boolean" "at\_freq\_buy"   
## [3] "at\_freq\_last24\_buy" "at\_freq\_last24\_sv"   
## [5] "at\_freq\_last24\_sv\_int\_buy" "at\_freq\_sv"   
## [7] "at\_freq\_sv\_int\_buy" "at\_interval\_buy"   
## [9] "at\_interval\_sv" "at\_interval\_sv\_int\_buy"   
## [11] "expected\_time\_buy" "expected\_time\_sv"   
## [13] "expected\_time\_sv\_int\_buy" "last\_buy"   
## [15] "last\_sv" "last\_sv\_int\_buy"   
## [17] "multiple\_buy" "multiple\_sv"   
## [19] "multiple\_sv\_int\_buy" "uniq\_content\_links"   
## [21] "num\_checkins" "y\_buy"

split <- .65  
set["rand"] <- runif(nrow(set)) #랜덤하게 분할. 65%, 35%의 비율로..  
train <- set[(set$rand <= split), ]  
test <- set[(set$rand > split), ]  
set$y <- set$y\_buy  
  
  
# 모든 변수에 대한 모형을 만들고 적합결과를 살펴보자   
vlist <- c("at\_buy\_boolean", "at\_freq\_buy",  
 "at\_freq\_last24\_buy","at\_freq\_last24\_sv",   
 "at\_freq\_sv", "expected\_time\_buy",   
 "expected\_time\_sv", "last\_buy", "last\_sv",   
 "num\_checkins")  
f = as.formula(paste("y\_buy", "~" , paste(vlist,   
 collapse = "+")))  
fit <- glm(f, data = train, family = binomial(logit))  
summary(fit)

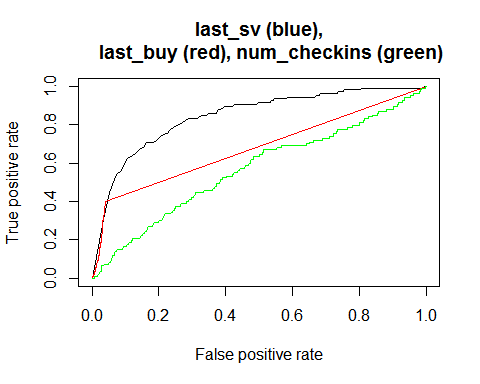
##   
## Call:  
## glm(formula = f, family = binomial(logit), data = train)  
##   
## Deviance Residuals:   
## Min 1Q Median 3Q Max   
## -1.6053 -0.0969 -0.0673 -0.0372 4.2371   
##   
## Coefficients:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) -5.0055559 0.1472369 -33.997 < 2e-16 \*\*\*  
## at\_buy\_boolean 2.2302650 0.3467684 6.432 1.26e-10 \*\*\*  
## at\_freq\_buy -0.2286804 0.1952229 -1.171 0.241446   
## at\_freq\_last24\_buy -2.5321895 1.1416236 -2.218 0.026551 \*   
## at\_freq\_last24\_sv 0.1372438 0.0309830 4.430 9.44e-06 \*\*\*  
## at\_freq\_sv 0.0367187 0.0102862 3.570 0.000357 \*\*\*  
## expected\_time\_buy -0.0039451 0.0071290 -0.553 0.580001   
## expected\_time\_sv -0.0044922 0.0049768 -0.903 0.366730   
## last\_buy -0.0024046 0.0035704 -0.673 0.500643   
## last\_sv -0.0221388 0.0037354 -5.927 3.09e-09 \*\*\*  
## num\_checkins -0.0001178 0.0001062 -1.109 0.267471   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## (Dispersion parameter for binomial family taken to be 1)  
##   
## Null deviance: 1893.8 on 35291 degrees of freedom  
## Residual deviance: 1623.9 on 35281 degrees of freedom  
## AIC: 1645.9  
##   
## Number of Fisher Scoring iterations: 10

# 각 변수에 대한 성능 측도 구하기   
vlist <- c("at\_buy\_boolean", "at\_freq\_buy",  
 "at\_freq\_last24\_buy","at\_freq\_last24\_sv",   
 "at\_freq\_sv", "expected\_time\_buy",   
 "expected\_time\_sv", "last\_buy", "last\_sv",   
 "num\_checkins")  
# 성능/평가 측도를 저장하기 위한 벡터 생성  
auc\_mu <- c()  
auc\_sig <- c()  
mae\_mu <- c()  
mae\_sig <- c()  
  
for (i in c(1:length(vlist))) {  
 a <- getxval(c(vlist[i]), set, 10, 100)  
 auc\_mu <- c(auc\_mu, mean(a$auc))  
 auc\_sig <- c(auc\_sig, sd(a$auc))  
 mae\_mu <- c(mae\_mu, mean(a$wmae))  
 mae\_sig <- c(mae\_sig, sd(a$wmae))  
}  
  
univar <- data.frame(vlist, auc\_mu, auc\_sig, mae\_mu, mae\_sig)  
  
# 한 변수에 대한 MAE 도표 그리기  
# 평가를 위해 holdout 이용  
  
  
fit <- glm(y\_buy ~ num\_checkins, data = train,   
 family = binomial(logit))  
y <- test$y\_buy  
p <- predict(fit, newdata = test, type = "response")  
getmae(p,y,50,"num\_checkins",1)



## [1] 0.002801337

# Greedy 전진 선택  
rvars <- c("last\_sv", "at\_freq\_sv", "at\_freq\_buy",  
 "at\_buy\_boolean", "last\_buy", "at\_freq\_last24\_sv",  
 "expected\_time\_sv", "num\_checkins",  
 "expected\_time\_buy", "at\_freq\_last24\_buy")  
# 빈 벡터 생성  
auc\_mu <- c()  
auc\_sig <- c()  
mae\_mu <- c()  
mae\_sig <- c()  
  
for (i in c(1:length(rvars))) {  
 vars <- rvars[1:i]  
 vars  
 a <- getxval(vars, set, 10, 100)  
 auc\_mu <- c(auc\_mu, mean(a$auc))  
 auc\_sig <- c(auc\_sig, sd(a$auc))  
 mae\_mu <- c(mae\_mu, mean(a$wmae))  
 mae\_sig <- c(mae\_sig, sd(a$wmae))  
}  
kvar<-data.frame(auc\_mu, auc\_sig, mae\_mu, mae\_sig)  
  
# 3가지 모형에 대한 AUC 도표  
y <- test$y\_buy  
fit <- glm(y\_buy~last\_sv, data=train,  
 family = binomial(logit))  
p1 <- predict(fit, newdata=test, type="response")  
  
fit <- glm(y\_buy~last\_buy, data=train,  
 family = binomial(logit))  
p2 <- predict(fit, newdata=test, type="response")  
  
fit <- glm(y\_buy~num\_checkins, data=train,   
 family = binomial(logit))  
p3 <- predict(fit, newdata=test,type="response")  
  
pred <- prediction(p1,y)  
perf1 <- performance(pred,'tpr','fpr')  
  
pred <- prediction(p2,y)  
perf2 <- performance(pred,'tpr','fpr')  
  
pred <- prediction(p3,y)  
perf3 <- performance(pred,'tpr','fpr')  
  
plot(perf1, color="blue", main="last\_sv (blue),  
 last\_buy (red), num\_checkins (green)")  
plot(perf2, col="red", add=TRUE)  
plot(perf3, col="green", add=TRUE)



## 연습문제

예제의 데이터에 대하여 선형 로지스틱 회귀분석을 실시하시오.