# 패키지 3주차

25기 박서영

# 라이브러리

## 우선 필요한 라이브러리를 불러온다.

[코드]

```
library(tidyverse)
library(plyr)
library(magrittr)
library(data.table)
library(tictoc)
library(foreach)
library(parallel)
library(doSNOW)
library(randomForest)
library(MLmetrics)
```

# CH2_1,2

## 코드

```
#문제1
tic()
Sys.sleep(3)
toc()


#문제2
data1 <- NULL
system.time(data1 <-
foreach(i=1:10, .combine=cbind) %do% {
  rnorm(5000*1000)
  print(i)}
  )
rm(i)
```

## 결과

```
#문제1

3.1 sec slapsed


#문제2
[1] 1 [1] 2 [1] 3 [1] 4 [1] 5 [1] 6 [1] 7 [1] 8
[1] 9 [1] 10
사용자 시스템 elapsed
 7.68    0.31    8.08
```

# CH2_3,4

## 코드

```
#문제3
data2 <- NULL
worker=makeCluster(10)
registerDoSNOW(worker)
system.time(data2 <-
foreach(i=1:10, .combine=cbind)%dopar%
rnorm(5000*1000))
stopCluster(worker)

#문제4
```

dopar가 시간이 적게 나왔다. 그 이유는, dopar를 쓰면 병렬처리에 용이하기에 효율적인 계산이 가능하기 때문이다.

## 결과

```
#문제3
사용자 시스템 elapsed
 2.13 1.97 7.73
```

# CH2_5,6

## 문제5

```
dir <- ("C:/Users/LG/Desktop/3주차 패키지/데이터 2")
file_list <- list.files(dir)
data_train <- data.frame()
worker2=makeCluster(99)
registerDoSNOW(worker2)
data_train <-
foreach(i = 1:99, .combine=rbind) %dopar% {
 read.csv(paste(dir,file_list[i],sep ="/"),
header=TRUE,stringsAsFactors=FALSE)
}
stopCluster(worker)
```

## 문제6

```
data_test <- fread("C:/Users/LG/Desktop/3주차 패키지/데이터 1.csv", stringsAsFactors = F)
data_train %<>% mutate_if(is.character, as.factor)
data_test %<>% mutate_if(is.character, as.factor)

data_test <- rbind(data_train[1, ], data_test)
data_test <- data_test[-1, ]
data_train <- rbind(data_test[1, ], data_train)
data_train <- data_train[-1, ]
data_train$click %<>% as.character %>% as.factor
data_test$click %<>% as.character %>% as.factor
```

# CH2_7,8

## 코드

```
#문제7
set.seed(1)
system.time(rf1 <-
randomForest(click~.,data=data_train))
#문제8
worker3=makeCluster(3)
registerDoSNOW(worker3)
system.time(rf2<-foreach(i =
rep(500%/%3,3), .combine=randomForest::combi
ne,.multicombine =TRUE,.packages
='randomForest') %dopar% {
  set.seed(1)
  randomForest(click~.,data=data_train,ntree=i)
})
stopCluster(worker3)
```

## 결과

```
#문제7
사용자 시스템 elapsed
 30.99 0.61 31.78

#문제8
사용자 시스템 elapsed
 0.61 0.37 18.70
```

# CH2_9

## 코드

```
prediction1 <- predict(rf1, newdata=data_test, type="prob")
LogLoss(prediction1[,2], (as.numeric(data_test$click)-1))

prediction2 <- predict(rf2, newdata=data_test, type="prob")
LogLoss(prediction2[,2], (as.numeric(data_test$click)-1))
```

## 결과

```
[1] 0.2646974
[1] 0.275836
```

# CH3_1,2

## 코드

```
#문제1
data_train$click %<>%  as.character %>%
as.numeric

data_test$click %<>%  as.character %>%
as.numeric

#문제2
devtools::install_url('https://github.com/catboost
/catboost/releases/download/v0.21/catboost-R-
Windows-0.21.tgz', INSTALL_opts = c("--no-
multiarch"))
```

## 결과

#문제1 해설

Click 변수가 factor(명목형 변수)에서 숫자형태로 바뀐다.

# CH3_3

## 코드

```
#문제3
library(catboost)

cat_features <-
which(sapply(data_train[,colnames(data_train)[-
1]], is.factor))

pool_train <- catboost.load_pool(data =
data_train[,-1], label= data_train[,1], cat_features
= cat_features )

pool_test <- catboost.load_pool(data =
data_test[,-1], label= data_test[,1],  cat_features
= cat_features)
```

## 해설

Catboost.load.pool을 이용할 때, data에는 x변수들을, label에는 targe인 click 변수를, 그리고 cat_features를 통해서 명목형 변수를 추려냈다.

# CH3_4

## 코드

```
params <- list(random_seed = 1, loss_function = "Logloss", logging_level = "Verbose", iterations= 200, learning_rate = 0.1 , task_type = "CPU")
model_catboost <- catboost.train(pool_train,NULL, params)

prediction3 <- catboost.predict(model_catboost, pool_test,prediction_type = "Probability")
LogLoss(prediction3, data_test$click)
```

## 결과

```
0: learn: 0.5857718 total: 277ms remaining: 55.1s
1: learn: 0.5157096 total: 426ms remaining: 42.1s
2: learn: 0.4633053 total: 472ms remaining: 31s
3: learn: 0.4146753 total: 543ms remaining: 26.6s
4: learn: 0.3845470 total: 689ms remaining: 26.9s
5: learn: 0.3611479 total: 823ms remaining: 26.6s (~199까지 출력됨)

0.2515626 (LogLoss값)
```

# CH3_5

## 코드

```
catBoost_cv<- catboost.cv(pool_train, params,
fold_count=3, type="Classical")

best_iteration <-
which(catBoost_cv$test.Logloss.mean ==
min(catBoost_cv$test.Logloss.mean)) -1
params2 <- list(random_seed = 1, loss_function
= "Logloss", logging_level = "Verbose",
iterations= 70, learning_rate = 0.1 , task_type =
"CPU")
model_catboost2 <-
catboost.train(pool_train,NULL, params2)

prediction4 <- catboost.predict(model_catboost2,
pool_test,prediction_type = "Probability")
LogLoss(prediction4, data_test$click)
```

## 결과

```
0: learn: 0.5946245 test: 0.5950192 best:
0.5950192 (0) total: 464ms remaining: 1m
32s
1: learn: 0.5230186 test: 0.5236458 best:
0.5236458 (1) total: 911ms remaining: 1m
30s (~199까지 출력)

0: learn: 0.5998765 total: 90.7ms remaining:
6.26s
1: learn: 0.5279523 total: 141ms remaining:
4.78s (~63까지 출력 / best iteration 값 :64)



0.2514341 (LogLoss값) -> 값이 줄었음!!
```