**휴대폰 센서 데이터 추가 변수 추출**

**20171468 서민지**

**1.통계 특징과 관련된 변수 추출**

* 기존에는 **roationRate**와 **userAcceleraion** 2개의 변수만 사용해서 통계 특징을 추출하였습니다.

1. “**gravity**” 와 “**attitude**” 관련 변수들에 대한 통계 특징도 추출하였습니다.

**\*** (“gravity” 변수는 mag 함수를 적용하여 통계 특징을 추출하였고, “attitude” 변수는 기존 값에서 통계 특징을 추출하였습니다.)

* 기존에는 mag함수를 사용해서 x, y, z 크기를 구한 후 통계 특징을 추출하였습니다. 각각의 값을 그대로 사용해보는 것도 의미가 있을것이라는 생각이 들었습니다.

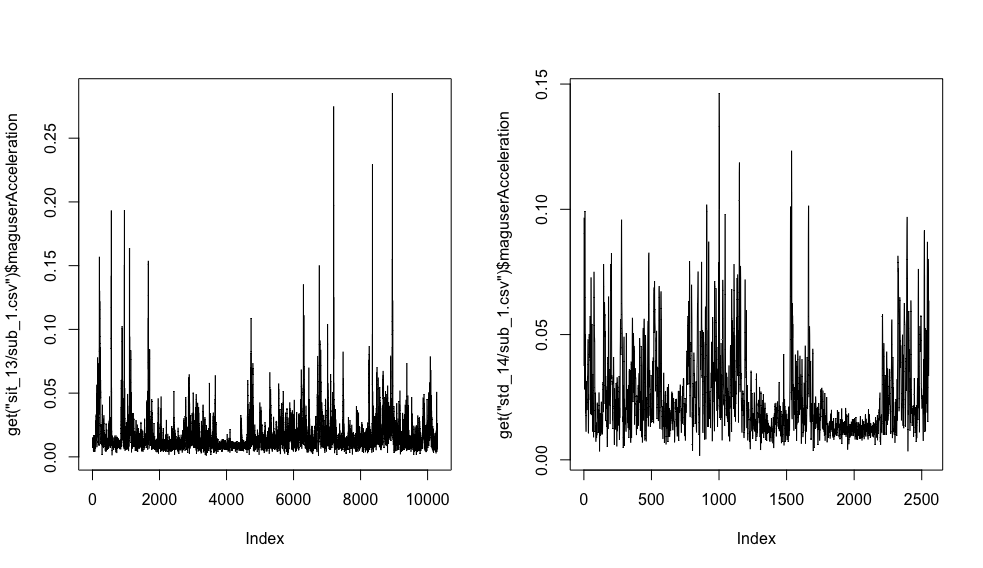
1. Mag함수를 적용하지 않은 x, y, z 원래의 값에서 통계 특징을 추출하였습니다.
2. (mean, min, max, sd, skewness, rms, rss, IQR, kurtosis) + **range** 통계 특징을 추가 하였습다.

**\*** (range는 최대값과 최솟값의 차이로 dispersion 관련 통계 특징입니다.)

|  |  |
| --- | --- |
| **변수 명** | **변수 설명** |
| 변수명\_range | 각 변수에서 range 통계 특징 추출 |
| attitude.roll\_fn1  ~  attitude.roll\_fn9 | attitude.roll 변수에 (mean, min, max, sd, skewness, rms, rss, IQR, kurtosis) 통계 특징 추출 |
| attitude.pitch\_fn1  ~  attitude.pitch\_fn9 | attitude.pitch 변수에 (mean, min, max, sd, skewness, rms, rss, IQR, kurtosis) 통계 특징 추출 |
| attitude.yaw\_fn1  ~  attitude.yaw\_fn9 | attitude.yaw 변수에 (mean, min, max, sd, skewness, rms, rss, IQR, kurtosis) 통계 특징 추출 |
| maggravity\_fn1  ~  maggravity\_fn9 | Gravity 변수에 mag 함수를 취하여 (mean, min, max ,sd ,skewness, rms, rss, IQR, kurtosis) 통계 특징 추출 |
| 변수.x\_fn1 ~ fn9, 변수.y\_fn1~fn9,  변수.z\_fn1~fn9 | mag 함수를 적용하지 않은 상태에서 통계 특징 추출 |

**2. 피크와 관련된 변수 추출**

1. 기존에는 피크 크기에서 (min, max, min, std) 통계 특징을 사용하였습니다. 여기에 **range** 통계 특징을 추가하였습니다.
2. 기존에는 magrotationRate 변수의 피크만 가지고 변수를 도출하였습니다. 이 부분을 똑같이 **maguserAcceleration** 에도 적용을 하였습니다.

****

maguserAcceleration의 그래프를 그려본 후 **피크 기준값을 1**로 사용하기로 결정 하였습니다.

ex<- get(fls[1])

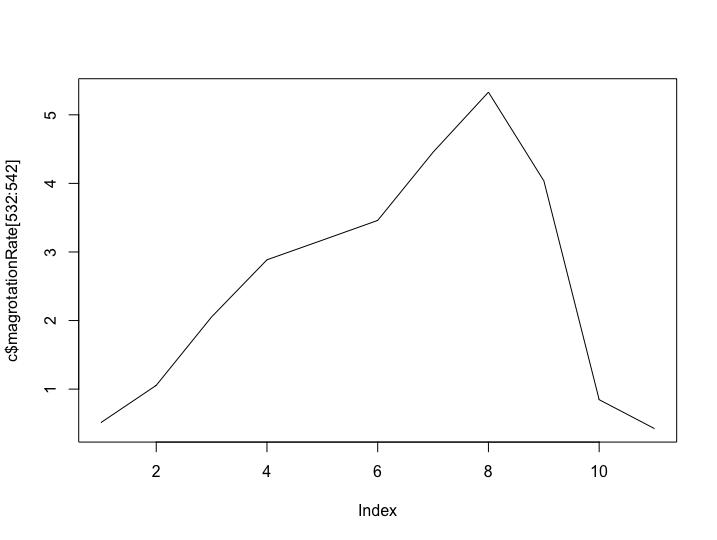
p\_ro<- findpeaks(ex$magrotationRate, threshold=4)

텍스트, 테이블이(가) 표시된 사진

자동 생성된 설명

다음과 같이 예제로 피크를 추출하고, 피크, 피크 발생 전, 피크 발생 후에 관한 그래프를 그려보았습니다.

plot(ex$magrotationRate[532:542], type="l")



1. 피크가 **발생하기 전과 후**의 값도 의미가 있을 것이라고 생각하여 피크가 발생하기 바로 이전 값과 이후 값을 구하여 (min, max, min, std) 통계 특징을 구한 변수를 추가하였습니다.
2. **순간적인 변화**를알기 위해 피크가 발생하는 순간과 그 다음 순간의 차이를 구하여 (min, max, min, std) 통계 특징을 구한 변수를 추가하였습니다.

|  |  |
| --- | --- |
| **변수 명** | **변수 설명** |
| p\_ac\_n, p\_ac\_interval, p\_ac\_interval\_std  p\_ac\_mean, p\_ac\_max, p\_ac\_min, p\_ac\_std | magrotationRate로 도출했던 피크 특징들을 maguserAcceleration에 똑같이 적용 |
| p\_ro\_be\_mean, p\_ro\_be\_max, p\_ro\_be\_min, p\_ro\_be\_std, p\_ro\_af\_mean, p\_ro\_af\_max, p\_ro\_ad\_min, p\_ro\_af\_std  p\_ac\_be\_mean, p\_ac\_be\_max, p\_ac\_be\_min, p\_ac\_be\_std, p\_ac\_af\_mean, p\_ac\_af\_max, p\_ac\_af\_min, p\_ac\_af\_std | magrotationRate과 maguserAcceleration 변수에서 피크값을 구한 후 피크 바로 이전 값과 이후 값을 구해 (min, max, min, std) 통계 특징을 추출 |
| p\_ro\_range, p\_ac\_range | 피크 크기를 구하여 range 통계 특징 추출 |
| p\_ro\_moment\_mean, p\_ro\_moment\_max, p\_ro\_moment\_min, p\_ro\_moment\_std,  p\_ac\_moment\_mean, p\_ac\_moment\_max, p\_ac\_moment\_min, p\_ac\_moment\_std, | magrotationRate과 maguserAcceleration 변수에서 피크값과 바로 이후 값의 차이를 구한 후 (min, max, min, std) 통계 특징을 추출 |

**3.고속 푸리 변환 적용 후 통계 특징 변수 추출**

**fft** 함수로 푸리에 변환을 적용해 데이터를 변환하였습니다. 변환 후 실수 부분만 가져와서 통계 특징을 추출하였습니다.

퓨리 변환 적용

테이블이(가) 표시된 사진

자동 생성된 설명

실수 부분 추출

테이블이(가) 표시된 사진

자동 생성된 설명

|  |  |
| --- | --- |
| **변수 명** | **변수 설명** |
| 변수명\_fft1  ~  변수명\_fft9 | 먼저, 퓨리의 변한 함수인 fft()를 사용하여 데이터를 변환시킨 후 실수 부분만 선택하여 데이터를 만듦  그리고 각 변수에 (mean, min, max, sd, skewness, rms, rss, IQR, kurtosis, **range**) 통계 특징을 추출함 |

**4.변화 분석과 관련된 변수 추출**

* 기존에는 cpt.mean, cpt.var, cpt.meanvar에서 변화 분석을 진행할 때 method 파라미터를 기본값인 AMOC(single changepoint)를 사용하였습니다.

1. method 방식 중 **PELT(multiple changepoints)** 방법이 좀 더 특징을 잘 추출하는 것으로 나타나 method를 변경하여 변화 분석을 진행하였습니다.

텍스트이(가) 표시된 사진

자동 생성된 설명테이블이(가) 표시된 사진

자동 생성된 설명

RWeka에 RF로 실험을 진행해 봤을때 AMOC(왼쪽)를 사용했을때 보다 PELT(오른쪽)를 사용했을때 더 높은 정확도를 가졌습니다.

* 기존에는 변화가 몇 번 일어났는지에 관한 변수만 사용하였습니다.

1. 변화가 일어났던 위치의 값을 가져와서 (mean, max, min, std) 통계 특징을 추출한 변수를 추가하였습니다.

|  |  |
| --- | --- |
| **변수 명** | **변수 설명** |
| cp1\_mean, cp1\_max, cp1\_min, cp1\_std,  cp2\_mean, cp2\_max, cp2\_min, cp2\_std,  cp3\_mean, cp3\_max, cp3\_min, cp3\_std,  cp4\_mean, cp4\_max, cp4\_min, cp4\_std,  cp5\_mean, cp5\_max, cp5\_min, cp5\_std,  cp6\_mean, cp6\_max, cp6\_min, cp6\_std | \*method 방식을 PELT를 사용하여 분석을 진행하였습니다.  변화가 일어났던 위치의 값을 가져와서 (mean, max, min, std) 통계 특징을 추출함 |

**5.모델 결과**

**5-1 변수 추출 특징 별 모델 비교**

xgboost 모델을 사용해서 학습을 진행하였습니다.

**[통계 특징 변수 모델]**

**테이블이(가) 표시된 사진

자동 생성된 설명**

- 통계 특징 변수만 사용했을 때 정확도는 **약 97%**가 나왔습니다. 아주 높은 정확도가 나왔습니다. 다른 변수들에 비해 중요한 변수들이 많이 담겨있는 것 같습니다.

**[피크 특징 변수 모델]**

테이블이(가) 표시된 사진

자동 생성된 설명

피크 특징 변수만 사용했을 때 정확도는 **약 78%**가 나왔습니다. 통계 특징보다는 낮은 결과지만 그래도 꽤 높은 정확도가 나왔다고 생각합니다.

**[푸리에 변환 변수 모델]**

테이블이(가) 표시된 사진

자동 생성된 설명

푸리에 변환을 적용한 변수를 사용했을 때 정확도는 **약 95%**가 나왔습니다. 높은 정확도가 나왔습니다.

**[변화 분석 변수 모델]**

테이블이(가) 표시된 사진

자동 생성된 설명

변화 분석 변수를 사용했을 때 정확도는 **약 78%**가 나왔습니다. 수업시간에 실습을 하며 40%밖에 나오지 못해 좋은 예측 변수가 될 수 없다고 생각하였지만, method방식을 바꾸고, 통계 특징을 추출하니 꽤 높은 정확도가 나왔습니다.

**5-2 최종 모델 선택 기준**

1. **모든 변수들을 통합한 모델**
2. **가장 정확도가 높았던 두 특징(통계 특징 변수, 푸리에 변환 변수) 변수를 사용한 모델**
3. **각 변수의 특징에서 중요 변수를 상위 30개씩 추출하여 합한 모델**

저는 다음과 같이 3개의 모델로 모델링을 진행하여 정확도가 가장 높은 모델을 최종으로 선택하고 나머지 2개의 모델을 후보로 선택하였습니다.

**5-3 최종 모델 선택**

**[모든 변수들을 통합한 모델]**

**테이블이(가) 표시된 사진

자동 생성된 설명**

- 모든 변수들을 통합한 모델은 정확도가 약 **98%**가 나왔습니다. 앞에서 가장 높았던 통계 특징만 추출한 모델보다 약 0.01% 증가하였습니다.

**[가장 정확도가 높았던 두 특징(통계 특징 변수, 푸리에 변환 변수) 변수를 사용한 모델]**

**테이블이(가) 표시된 사진

자동 생성된 설명**

- 위에 모델과 동일한 결과가 나왔습니다. 이것으로 보아 역시 통계 특징에 관한 변수가 중요한 변수들인 것 같습니다.

**[각 변수의 특징에서 중요 변수를 상위 30개씩 추출하여 합한 모델]**

변수별 중요도를 파악하기 위해서는 xgboost로 학습을 시킨 후, xgb.importance 사용해서 확인할 수 있습니다.

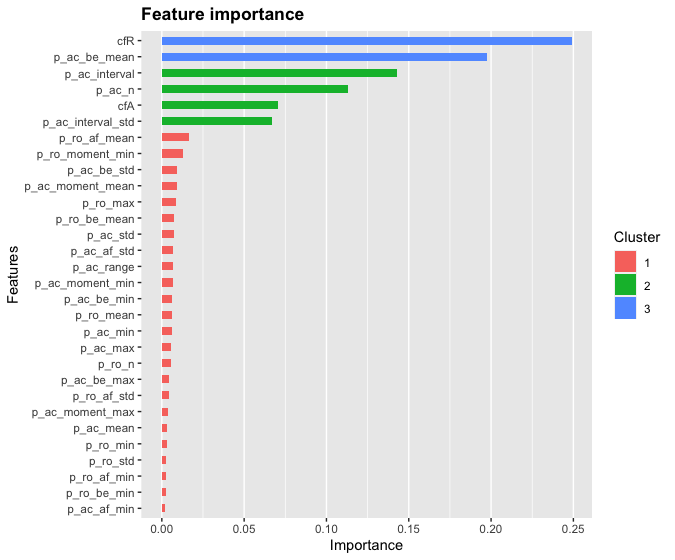
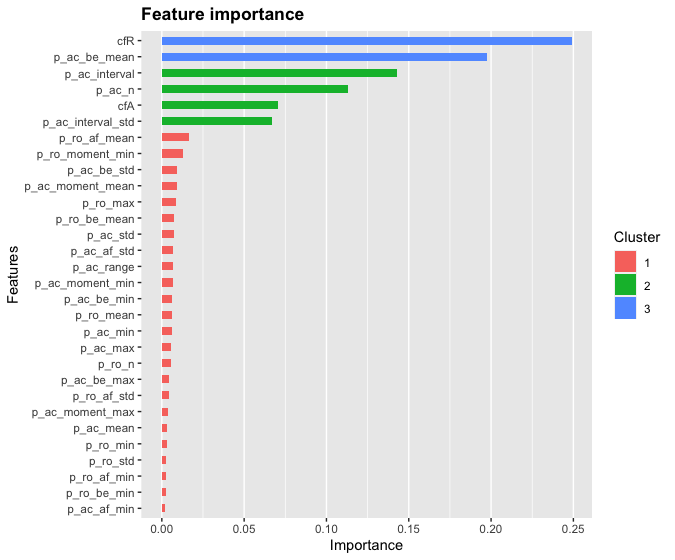
테이블이(가) 표시된 사진

자동 생성된 설명

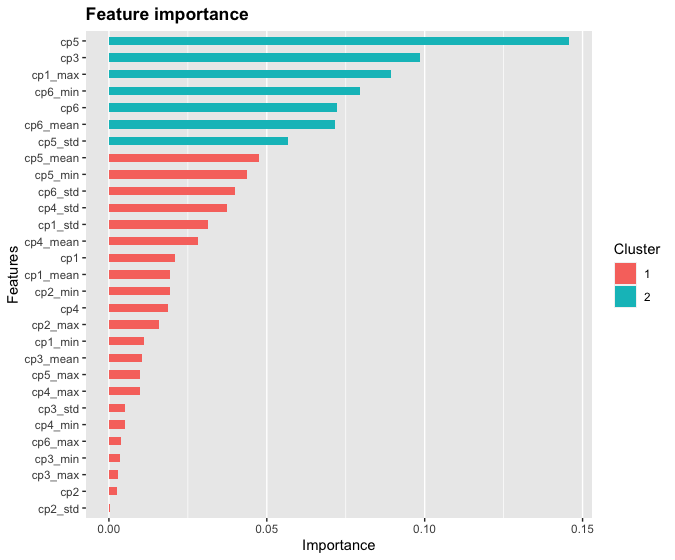
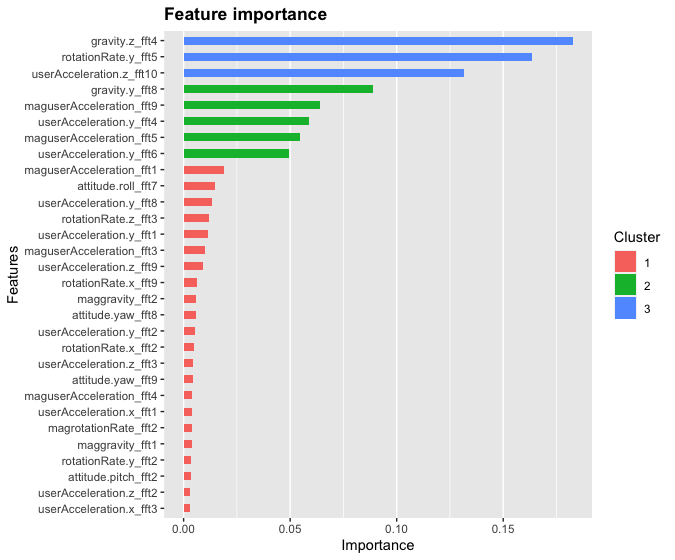
- 정확도는 약 98%가 나왔습니다. 위의 두 모델보다는 아주 약간 떨어지는 결과가 나왔습니다.

각 변수별 중요도는 xgb.ggplot.importance 함수를 사용해서 다음과 같이 시각화도 해볼 수 있습니다.

**[통계 특징 중요 변수] [피크 특징 중요 변수]**



**[푸리에 변환 특징 중요 변수] [변화 분석 특징 중요 변수]**



위 3가지 모델의 결과를 비교해본 결과 최종 모델로 **“모든 변수를 통합한 모델”** 을 선택하였습니다. 통계적 모델과 푸리에 변환 모델을 합한 모델도 정확도가 높았지만, 여러 특징을 담은 모델을 사용하고 싶어 이 모델을 선택하였습니다.

[코드 부록]

## library load

library(dplyr)

library(stringr)

library(ggplot2)

library(tidyverse)

library(RWeka)

library(fBasics)

library(pracma)

library(signal)

library(seewave)

library(e1071)

library(caret)

library(xgboost)

library(changepoint)

# 경로 설정

setwd("/Users/seominji/Desktop/Unstruct\_DA/A\_DeviceMotion\_data")

d<-getwd()

fls <-dir(d,recursive = TRUE)

# 객체 생성

for(f in fls){

a<- file.path(str\_c(d,"/",f))

temp<- read.csv(a)

assign(f,temp)

}

#### 통계치 관련 변수 ####

# 전체 데이터 생성(피크변수와 merge를 안전하게 하기 위해 d(파일 이름)을 추가)

HAR\_total<-data.frame()

i<-0

for(f in fls){

temp<-get(f)

print(f)

i<-i+1

print(i)

HAR\_total <-rbind(HAR\_total,

temp%>%mutate(exp\_no=unlist(regmatches(f,gregexpr("[[:digit:]]+", f)[1]))[1],

id=unlist(regmatches(f,gregexpr("[[:digit:]]+", f)[1]))[2],

activity=unlist(str\_split(f,"\\\_"))[1],d=f))

}

# mag 함수 정의

mag<- function(df, column){

df[,str\_c("mag", column)]<- with(df, sqrt(get(str\_c(column, ".x"))^2+get(str\_c(column,".y"))^2+get(str\_c(column, ".z"))^2))

return(df)

}

# skewness 함수

skewness <- function(x){

(sum((x-mean(x))^3)/length(x))/((sum((x-mean(x))^2)/length(x)))^(3/2)

}

# rss 함수 정의

rss<-function(x) rms(x)\*(length(x))\*0.5

# range 함수 정의

range\_ <- function(x){

(diff(range(x)))

}

# mag 적용(gravity 추가)

HAR\_total<- mag(HAR\_total, "userAcceleration")

HAR\_total<- mag(HAR\_total, "rotationRate")

HAR\_total<-mag(HAR\_total, "gravity")

## 변수 추출 ##

# 통계 특징 구하기("maggravity","attitude.roll","attitude.pitch","attitude.yaw" 추가)

HAR\_summary\_extend<- HAR\_total %>% group\_by(id,exp\_no,activity,d) %>%

summarise\_at(.vars=c("gravity.x","gravity.y","gravity.z",

"userAcceleration.x","userAcceleration.y","userAcceleration.z",

"rotationRate.x","rotationRate.y","rotationRate.z",

"maguserAcceleration","magrotationRate",

"maggravity","attitude.roll","attitude.pitch","attitude.yaw"),

.funs=c(mean, min, max ,sd ,skewness, rms, rss, IQR, e1071::kurtosis, range\_))

# 널값 확인

colSums(is.na(HAR\_summary\_extend))

## 통계 특징 변수만 학습 ##

# x,y 구분

x = HAR\_summary\_extend %>% ungroup %>% select(-d, -exp\_no, -id, -activity) %>% data.matrix

y = HAR\_summary\_extend$activity

# xgboost 모델 학습(10-fold)

set.seed(1004)

Static\_model = xgb.cv(data = x,label = as.integer(as.factor(y))-1, num\_class = levels(as.factor(y)) %>% length,

nfold = 10, nrounds = 500, early\_stopping\_rounds = 8, booster = 'gbtree',

objective = 'multi:softprob', eval\_metric = 'mlogloss',

verbose = F, prediction = T)

# 예측 데이터 생성(max.col 사용해서 가장 큰값 추출)

pred\_df = Static\_model$pred %>% as.data.frame %>%

mutate(pred = levels(as.factor(y))[max.col(.)] %>% as.factor, actual = as.factor(y))

pred\_df %>% select(pred,actual) %>% table

# confusionMatrix 생성

Static\_conf<- caret::confusionMatrix(pred\_df$pred, pred\_df$actual)

Static\_conf

# 중요 변수 추출

Static\_model = xgboost(data = x,label = as.integer(as.factor(y))-1, num\_class = levels(as.factor(y)) %>% length,

nfold = 10, nrounds = 500, early\_stopping\_rounds = 8, booster = 'gbtree',

objective = 'multi:softprob', eval\_metric = 'mlogloss',

verbose = F)

Static\_imp<- xgb.importance(model = Static\_model)

#### ####

#### 피크 관련 변수 ####

Peak\_rslt<-data.frame()

for(d in fls){

f<-get(d)

f<-mag(f,"rotationRate")

f<-mag(f,"userAcceleration")

assign(d,f)}# d는 객체명 f는 객체가 들어간다

## 변수 추출 ##

for(d in fls){

f<-get(d)

p\_ro<-findpeaks(f$magrotationRate,threshold = 4)

p\_ac<-findpeaks(f$maguserAcceleration,threshold = 1)

Peak\_rslt<-rbind(Peak\_rslt, data.frame(d,

p\_ro\_n=ifelse(!is.null(p\_ro),dim(p\_ro)[1],0),

p\_ro\_interval=ifelse(!is.null(p\_ro),ifelse(dim(p\_ro)[1]>2,mean(diff(p\_ro[,2])),0),0),

p\_ro\_interval\_std=ifelse(!is.null(p\_ro),ifelse(dim(p\_ro)[1]>2,std(diff(p\_ro[,2])),0),0),

p\_ro\_mean=ifelse(!is.null(p\_ro),mean(p\_ro[,1]),0),

p\_ro\_max=ifelse(!is.null(p\_ro),max(p\_ro[,1]),0),

p\_ro\_min=ifelse(!is.null(p\_ro),min(p\_ro[,1]),0),

p\_ro\_range=ifelse(!is.null(p\_ro),diff(range(p\_ro[,1])),0),

p\_ro\_std=ifelse(!is.null(p\_ro),std(p\_ro[,1]),0),

p\_ro\_be\_mean=ifelse(!is.null(p\_ro),mean(f$magrotationRate[p\_ro[,2]-1]),0),

p\_ro\_be\_max=ifelse(!is.null(p\_ro),max(f$magrotationRate[p\_ro[,2]-1]),0),

p\_ro\_be\_min=ifelse(!is.null(p\_ro),min(f$magrotationRate[p\_ro[,2]-1]),0),

p\_ro\_be\_std=ifelse(!is.null(p\_ro),std(f$magrotationRate[p\_ro[,2]-1]),0),

p\_ro\_af\_mean=ifelse(!is.null(p\_ro),mean(f$magrotationRate[p\_ro[,2]+1]),0),

p\_ro\_af\_max=ifelse(!is.null(p\_ro),max(f$magrotationRate[p\_ro[,2]+1]),0),

p\_ro\_af\_min=ifelse(!is.null(p\_ro),min(f$magrotationRate[p\_ro[,2]+1]),0),

p\_ro\_af\_std=ifelse(!is.null(p\_ro),std(f$magrotationRate[p\_ro[,2]+1]),0),

p\_ro\_moment\_mean=ifelse(!is.null(p\_ro),mean(p\_ro[,1] - f$magrotationRate[p\_ro[,2]+1]),0),

p\_ro\_moment\_max=ifelse(!is.null(p\_ro),max(p\_ro[,1] - f$magrotationRate[p\_ro[,2]+1]),0),

p\_ro\_moment\_min=ifelse(!is.null(p\_ro),min(p\_ro[,1] - f$magrotationRate[p\_ro[,2]+1]),0),

p\_ro\_moment\_std=ifelse(!is.null(p\_ro),std(p\_ro[,1] - f$magrotationRate[p\_ro[,2]+1]),0),

p\_ac\_n=ifelse(!is.null(p\_ac),dim(p\_ac)[1],0),

p\_ac\_interval=ifelse(!is.null(p\_ac),ifelse(dim(p\_ac)[1]>2,mean(diff(p\_ac[,2])),0),0),

p\_ac\_interval\_std=ifelse(!is.null(p\_ac),ifelse(dim(p\_ac)[1]>2,std(diff(p\_ac[,2])),0),0),

p\_ac\_mean=ifelse(!is.null(p\_ac),mean(p\_ac[,1]),0),

p\_ac\_max=ifelse(!is.null(p\_ac),max(p\_ac[,1]),0),

p\_ac\_min=ifelse(!is.null(p\_ac),min(p\_ac[,1]),0),

p\_ac\_range=ifelse(!is.null(p\_ac),diff(range(p\_ac[,1])),0),

p\_ac\_std=ifelse(!is.null(p\_ac),std(p\_ac[,1]),0),

p\_ac\_be\_mean=ifelse(!is.null(p\_ac),mean(f$maguserAcceleration[p\_ac[,2]-1]),0),

p\_ac\_be\_max=ifelse(!is.null(p\_ac),max(f$maguserAcceleration[p\_ac[,2]-1]),0),

p\_ac\_be\_min=ifelse(!is.null(p\_ac),min(f$maguserAcceleration[p\_ac[,2]-1]),0),

p\_ac\_be\_std=ifelse(!is.null(p\_ac),std(f$maguserAcceleration[p\_ac[,2]-1]),0),

p\_ac\_af\_mean=ifelse(!is.null(p\_ac),mean(f$maguserAcceleration[p\_ac[,2]+1]),0),

p\_ac\_af\_max=ifelse(!is.null(p\_ac),max(f$maguserAcceleration[p\_ac[,2]+1]),0),

p\_ac\_af\_min=ifelse(!is.null(p\_ac),min(f$maguserAcceleration[p\_ac[,2]+1]),0),

p\_ac\_af\_std=ifelse(!is.null(p\_ac),std(f$maguserAcceleration[p\_ac[,2]+1]),0),

p\_ac\_moment\_mean=ifelse(!is.null(p\_ac),mean(p\_ac[,1] - f$maguserAcceleration[p\_ac[,2]+1]),0),

p\_ac\_moment\_max=ifelse(!is.null(p\_ac),max(p\_ac[,1] - f$maguserAcceleration[p\_ac[,2]+1]),0),

p\_ac\_moment\_min=ifelse(!is.null(p\_ac),min(p\_ac[,1] - f$maguserAcceleration[p\_ac[,2]+1]),0),

p\_ac\_moment\_std=ifelse(!is.null(p\_ac),std(p\_ac[,1] - f$maguserAcceleration[p\_ac[,2]+1]),0)

))}

# 파고율 변수

temp<- data.frame()

for(d in fls){

f<-get(d)

f<-f %>% select(magrotationRate, maguserAcceleration)

cfR<- crest(f$magrotationRate, 50, plot=TRUE)

cfA<- crest(f$maguserAcceleration, 50, plot=TRUE)

temp<- rbind(temp, data.frame(d, cfR=cfR$C, cfA=cfA$C))

}

Peak\_final<- merge(Peak\_rslt, temp, by="d")

# exp, id, activity 추출

id\_f<-function(x){

exp\_no=unlist(regmatches(x,gregexpr("[[:digit:]]+", x)[1]))[1]

id=unlist(regmatches(x,gregexpr("[[:digit:]]+", x)[1]))[2]

activity=unlist(str\_split(x, "\\\_"))[1]

return(cbind(exp\_no, id, activity))

}

temp<-data.frame()

for(i in 1:nrow(Peak\_final)){

temp<-rbind(temp, id\_f(Peak\_final$d[i]))

}

Peak\_final2<-cbind(Peak\_final,temp)

## 피크 특징 변수만 학습 ##

# x,y 구분

x = Peak\_final2 %>% ungroup %>% select(-d, -exp\_no, -id, -activity) %>% data.matrix

y = Peak\_final2$activity

# xgboost 모델 학습(10-fold)

set.seed(1004)

Peak\_model = xgb.cv(data = x,label = as.integer(as.factor(y))-1, num\_class = levels(as.factor(y)) %>% length,

nfold = 10, nrounds = 500, early\_stopping\_rounds = 8, booster = 'gbtree',

objective = 'multi:softprob', eval\_metric = 'mlogloss',

verbose = F, prediction = T)

# 예측 데이터 생성(max.col 사용해서 가장 큰값 추출)

pred\_df = Peak\_model$pred %>% as.data.frame %>%

mutate(pred = levels(as.factor(y))[max.col(.)] %>% as.factor, actual = as.factor(y))

pred\_df %>% select(pred,actual) %>% table

# confusionMatrix 생성

Peak\_conf<- caret::confusionMatrix(pred\_df$pred, pred\_df$actual)

Peak\_conf

# 중요 변수 추출

Peak\_model = xgboost(data = x,label = as.integer(as.factor(y))-1, num\_class = levels(as.factor(y)) %>% length,

nfold = 10, nrounds = 500, early\_stopping\_rounds = 8, booster = 'gbtree',

objective = 'multi:softprob', eval\_metric = 'mlogloss',

verbose = F)

Peak\_imp<- xgb.importance(model = Peak\_model)

#### ####

####고속 푸리에 변환 (Fast Fourier transform) 적용 변수 ####

fft\_data <- data.frame()

for(f in fls){

temp<-get(f)

temp<- as.data.frame(lapply(temp, fft)) # fft 적용

temp<- as.data.frame(lapply(temp, Re)) # 앞에 실수부분만 가져옴

fft\_data <-rbind(fft\_data,

temp%>%mutate(exp\_no=unlist(regmatches(f,gregexpr("[[:digit:]]+", f)[1]))[1],

id=unlist(regmatches(f,gregexpr("[[:digit:]]+", f)[1]))[2],

activity=unlist(str\_split(f,"\\\_"))[1],d=f))

}

fft\_data<- mag(fft\_data, "userAcceleration")

fft\_data<- mag(fft\_data, "rotationRate")

fft\_data<-mag(fft\_data, "gravity")

colnames(fft\_data)

## 변수 추출 ##

fft\_data\_summary<- fft\_data %>% group\_by(id,exp\_no,activity,d) %>%

summarise\_at(.vars=c("gravity.x","gravity.y","gravity.z",

"userAcceleration.x","userAcceleration.y","userAcceleration.z",

"rotationRate.x","rotationRate.y","rotationRate.z",

"maguserAcceleration","magrotationRate",

"maggravity","attitude.roll","attitude.pitch","attitude.yaw"),

.funs=c(mean, min, max ,sd ,skewness, rms, rss, IQR, e1071::kurtosis, range\_))

# 변수명 변경

names(fft\_data\_summary)<- str\_replace\_all(colnames(fft\_data\_summary),'fn',"fft")

## FFT 변환 변수만 학습 ##

# x,y 구분

x = fft\_data\_summary %>% ungroup %>% select(-d, -exp\_no, -id, -activity) %>% data.matrix

y = fft\_data\_summary$activity

# xgboost 모델 학습(10-fold)

set.seed(1004)

fft\_model = xgb.cv(data = x,label = as.integer(as.factor(y))-1, num\_class = levels(as.factor(y)) %>% length,

nfold = 10, nrounds = 500, early\_stopping\_rounds = 8, booster = 'gbtree',

objective = 'multi:softprob', eval\_metric = 'mlogloss',

verbose = F, prediction = T)

# 예측 데이터 생성(max.col 사용해서 가장 큰값 추출)

pred\_df = fft\_model$pred %>% as.data.frame %>%

mutate(pred = levels(as.factor(y))[max.col(.)] %>% as.factor, actual = as.factor(y))

pred\_df %>% select(pred,actual) %>% table

# confusionMatrix 생성

Fourier\_conf<- caret::confusionMatrix(pred\_df$pred, pred\_df$actual)

Fourier\_conf

# 중요 변수 추출

Fourier\_model = xgboost(data = x,label = as.integer(as.factor(y))-1, num\_class = levels(as.factor(y)) %>% length,

nfold = 10, nrounds = 500, early\_stopping\_rounds = 8, booster = 'gbtree',

objective = 'multi:softprob', eval\_metric = 'mlogloss',

verbose = F)

Fourier\_imp<- xgb.importance(model = Fourier\_model)

#### ####

#### 변화 분석 변수 ####

## 변수 추출

ch\_pt<-data.frame()

for(d in fls){

f<-get(d)

f<-mag(f, "rotationRate")

f<-mag(f, "userAcceleration")

f<-mag(f, "gravity")# method 값을 PELT 방식으로 바꿈(기본값은 AMOC)

rslt<-sapply(f %>% select(magrotationRate, maguserAcceleration, maggravity), cpt.mean, method = "PELT")

rslt\_cpts1<-cpts(rslt$magrotationRate)

# 변화시점에 해당하는 값을 가져와 통계 특징을 추출

cp1\_mean<- ifelse(length(rslt\_cpts1) != 0, mean(f$magrotationRate[rslt\_cpts1]),0)

cp1\_max<- ifelse(length(rslt\_cpts1) != 0, max(f$magrotationRate[rslt\_cpts1]),0)

cp1\_min<- ifelse(length(rslt\_cpts1) != 0, min(f$magrotationRate[rslt\_cpts1]),0)

cp1\_std<- ifelse(length(rslt\_cpts1) != 0, std(f$magrotationRate[rslt\_cpts1]),0)

rslt\_cpts2<-cpts(rslt$maguserAcceleration)

cp2\_mean<- ifelse(length(rslt\_cpts2) != 0, mean(f$maguserAcceleration[rslt\_cpts2]),0)

cp2\_max<- ifelse(length(rslt\_cpts2) != 0, max(f$maguserAcceleration[rslt\_cpts2]),0)

cp2\_min<- ifelse(length(rslt\_cpts2) != 0, min(f$maguserAcceleration[rslt\_cpts2]),0)

cp2\_std<- ifelse(length(rslt\_cpts2) != 0, std(f$maguserAcceleration[rslt\_cpts2]),0)

rslt2<-sapply(f %>% select(magrotationRate, maguserAcceleration), cpt.var, method = "PELT")

rslt2\_cpts1<-cpts(rslt2$magrotationRate)

cp3\_mean<- ifelse(length(rslt2\_cpts1) != 0, mean(f$magrotationRate[rslt2\_cpts1]),0)

cp3\_max<- ifelse(length(rslt2\_cpts1) != 0, max(f$magrotationRate[rslt2\_cpts1]),0)

cp3\_min<- ifelse(length(rslt2\_cpts1) != 0, min(f$magrotationRate[rslt2\_cpts1]),0)

cp3\_std<- ifelse(length(rslt2\_cpts1) != 0, std(f$magrotationRate[rslt2\_cpts1]),0)

rslt2\_cpts2<-cpts(rslt2$maguserAcceleration)

cp4\_mean<- ifelse(length(rslt2\_cpts2) != 0, mean(f$maguserAcceleration[rslt2\_cpts2]),0)

cp4\_max<- ifelse(length(rslt2\_cpts2) != 0, max(f$maguserAcceleration[rslt2\_cpts2]),0)

cp4\_min<- ifelse(length(rslt2\_cpts2) != 0, min(f$maguserAcceleration[rslt2\_cpts2]),0)

cp4\_std<- ifelse(length(rslt2\_cpts2) != 0, std(f$maguserAcceleration[rslt2\_cpts2]),0)

rslt3<-sapply(f %>% select(magrotationRate, maguserAcceleration), cpt.meanvar, method = "PELT")

rslt3\_cpts1<-cpts(rslt3$magrotationRate)

cp5\_mean<- ifelse(length(rslt3\_cpts1) != 0, mean(f$magrotationRate[rslt3\_cpts1]),0)

cp5\_max<- ifelse(length(rslt3\_cpts1) != 0, max(f$magrotationRate[rslt3\_cpts1]),0)

cp5\_min<- ifelse(length(rslt3\_cpts1) != 0, min(f$magrotationRate[rslt3\_cpts1]),0)

cp5\_std<- ifelse(length(rslt3\_cpts1) != 0, std(f$magrotationRate[rslt3\_cpts1]),0)

rslt3\_cpts2<-cpts(rslt3$maguserAcceleration)

cp6\_mean<- ifelse(length(rslt3\_cpts2) != 0, mean(f$maguserAcceleration[rslt3\_cpts2]),0)

cp6\_max<- ifelse(length(rslt3\_cpts2) != 0, max(f$maguserAcceleration[rslt3\_cpts2]),0)

cp6\_min<- ifelse(length(rslt3\_cpts2) != 0, min(f$maguserAcceleration[rslt3\_cpts2]),0)

cp6\_std<- ifelse(length(rslt3\_cpts2) != 0, std(f$maguserAcceleration[rslt3\_cpts2]),0)

ch\_pt<-rbind(ch\_pt, data.frame(d, cp1=length(rslt\_cpts1),cp2=length(rslt\_cpts2),

cp3=length(rslt2\_cpts1), cp4=length(rslt2\_cpts2),

cp5=length(rslt3\_cpts1), cp6=length(rslt3\_cpts2),

cp1\_mean, cp1\_max, cp1\_min, cp1\_std,

cp2\_mean, cp2\_max, cp2\_min, cp2\_std,

cp3\_mean, cp3\_max, cp3\_min, cp3\_std,

cp4\_mean, cp4\_max, cp4\_min, cp4\_std,

cp5\_mean, cp5\_max, cp5\_min, cp5\_std,

cp6\_mean, cp6\_max, cp6\_min, cp6\_std))

}

for(d in fls){

f<-get(d)

f<-mag(f,"rotationRate")

f<-mag(f,"userAcceleration")

assign(d,f)}# d는 객체명 f는 객체야!!!

# exp, id, activity 추출

id\_f<-function(x){

exp\_no=unlist(regmatches(x,gregexpr("[[:digit:]]+", x)[1]))[1]

id=unlist(regmatches(x,gregexpr("[[:digit:]]+", x)[1]))[2]

activity=unlist(str\_split(x, "\\\_"))[1]

return(cbind(exp\_no, id, activity))

}

temp<-data.frame()

for(i in 1:nrow(ch\_pt)){

temp<-rbind(temp, id\_f(ch\_pt$d[i]))

}

ch\_pt2<-cbind(ch\_pt, temp)

# 널값 확인 후 0으로 변경

colSums(is.na(ch\_pt2))

ch\_pt2[is.na(ch\_pt2)] <- 0

## 변화 변수만 학습 ##

# x,y 구분

x = ch\_pt2 %>% ungroup %>% select(-d, -exp\_no, -id, -activity) %>% data.matrix

y = ch\_pt2$activity

# xgboost 모델 학습(10-fold)

set.seed(1004)

ch\_pt\_model = xgb.cv(data = x,label = as.integer(as.factor(y))-1, num\_class = levels(as.factor(y)) %>% length,

nfold = 10, nrounds = 500, early\_stopping\_rounds = 8, booster = 'gbtree',

objective = 'multi:softprob', eval\_metric = 'mlogloss',

verbose = F, prediction = T)

# 예측 데이터 생성(max.col 사용해서 가장 큰값 추출)

pred\_df = ch\_pt\_model$pred %>% as.data.frame %>%

mutate(pred = levels(as.factor(y))[max.col(.)] %>% as.factor, actual = as.factor(y))

pred\_df %>% select(pred,actual) %>% table

# confusionMatrix 생성

Chpoint\_conf<- caret::confusionMatrix(pred\_df$pred, pred\_df$actual)

Chpoint\_conf

# 중요 변수 추출

Chpoint\_model = xgboost(data = x,label = as.integer(as.factor(y))-1, num\_class = levels(as.factor(y)) %>% length,

nfold = 10, nrounds = 500, early\_stopping\_rounds = 8, booster = 'gbtree',

objective = 'multi:softprob', eval\_metric = 'mlogloss',

verbose = F)

Chpoint\_imp<- xgb.importance(model = Chpoint\_model)

### 모든 변수를 합하여 학습 진행 ####

Static<- HAR\_summary\_extend %>% ungroup() %>% select(-exp\_no,-id,-activity)

Fourier<- fft\_data\_summary %>% ungroup() %>% select(-exp\_no,-id,-activity)

Peak<-Peak\_final2 %>% ungroup() %>% select(-exp\_no,-id,-activity)

Chpoint<- ch\_pt2 %>% ungroup() %>% select(-exp\_no,-id)

final<- merge(Static, Fourier, by="d")

final<- merge(final, Peak, by="d")

final<- merge(final, Chpoint, by="d")

final2<- final %>% ungroup() %>% select(-d)

# 널값 확인 후 0으로 변경

colSums(is.na(final2))

final2[is.na(final2)] <- 0

# x,y 구분

x = final2 %>% ungroup %>% select(-activity) %>% data.matrix

y = final2$activity

# xgboost 모델 학습(10-fold)

set.seed(1004)

all\_model = xgb.cv(data = x,label = as.integer(as.factor(y))-1, num\_class = levels(as.factor(y)) %>% length,

nfold = 10, nrounds = 500, early\_stopping\_rounds = 8, booster = 'gbtree',

objective = 'multi:softprob', eval\_metric = 'mlogloss',

verbose = F, prediction = T)

# 예측 데이터 생성(max.col 사용해서 가장 큰값 추출)

pred\_df = all\_model$pred %>% as.data.frame %>%

mutate(pred = levels(as.factor(y))[max.col(.)] %>% as.factor, actual = as.factor(y))

pred\_df %>% select(pred,actual) %>% table

# confusionMatrix 생성

all\_conf<- caret::confusionMatrix(pred\_df$pred, pred\_df$actual)

all\_conf

# accuracy 확인

cat("통계 특징 accuracy는", round(Static\_conf$overall[[1]],2),"입니다.")

cat("피크 특징 accuracy는", round(Peak\_conf$overall[[1]],2),"입니다.")

cat("푸리에 변환 특징 accuracy는", round(Fourier\_conf$overall[[1]],2),"입니다.")

cat("변화 특징 accuracy는", round(Chpoint\_conf$overall[[1]],2),"입니다.")

cat("총 accuracy는", round(all\_conf$overall[[1]],2),"입니다.")

## 가장 높은 정확도를 나타낸 통계특징과 푸리에 변환 특징 변수들만 사용하여 예측 ##

Static<- HAR\_summary\_extend %>% ungroup() %>% select(-exp\_no,-id,-activity)

Fourier<- fft\_data\_summary %>% ungroup() %>% select(-exp\_no,-id)

Sta\_Fo<- merge(Static, Fourier, by = 'd')

Sta\_Fo<- Sta\_Fo %>% ungroup() %>% select(-d)

# 널값 확인 후 0으로 변경

colSums(is.na(Sta\_Fo))

Sta\_Fo[is.na(Sta\_Fo)] <- 0

# x,y 구분

x = Sta\_Fo %>% ungroup %>% select(-activity) %>% data.matrix

y = Sta\_Fo$activity

# xgboost 모델 학습(10-fold)

set.seed(1004)

Sta\_Fo\_model = xgb.cv(data = x,label = as.integer(as.factor(y))-1, num\_class = levels(as.factor(y)) %>% length,

nfold = 10, nrounds = 500, early\_stopping\_rounds = 8, booster = 'gbtree',

objective = 'multi:softprob', eval\_metric = 'mlogloss',

verbose = F, prediction = T)

# 예측 데이터 생성(max.col 사용해서 가장 큰값 추출)

pred\_df = Sta\_Fo\_model$pred %>% as.data.frame %>%

mutate(pred = levels(as.factor(y))[max.col(.)] %>% as.factor, actual = as.factor(y))

pred\_df %>% select(pred,actual) %>% table

# confusionMatrix 생성

Sta\_Fo\_conf<- caret::confusionMatrix(pred\_df$pred, pred\_df$actual)

Sta\_Fo\_conf

## 각 특징별로 상위 변수들만 추출하여 예측 ##

xgb.ggplot.importance(Static\_imp[1:30])

xgb.ggplot.importance(Peak\_imp[1:30])

xgb.ggplot.importance(Fourier\_imp[1:30])

xgb.ggplot.importance(Chpoint\_imp[1:29])

Staitc\_imp\_df<- cbind(d = HAR\_summary\_extend$d,HAR\_summary\_extend[,Static\_imp[1:30]$Feature])

Peak\_imp\_df<- cbind(d = Peak\_final2$d, Peak\_final2[,Peak\_imp[1:30]$Feature])

Fourier\_imp\_df<- cbind(d = fft\_data\_summary$d,fft\_data\_summary[,Fourier\_imp[1:30]$Feature])

Chpoint\_imp\_df<- cbind(d = ch\_pt2$d, activity = ch\_pt2$activity, ch\_pt2[,Chpoint\_imp[1:29]$Feature])

SP\_df<- merge(Staitc\_imp\_df, Peak\_imp\_df, by = 'd')

SPF\_df<- merge(SP\_df, Fourier\_imp\_df, by = 'd')

import\_df<- merge(SPF\_df, Chpoint\_imp\_df, by = 'd')

# x,y 구분

x = import\_df %>% ungroup %>% select(-d, -activity) %>% data.matrix

y = import\_df$activity

# xgboost 모델 학습(10-fold)

set.seed(1004)

import\_model = xgb.cv(data = x,label = as.integer(as.factor(y))-1, num\_class = levels(as.factor(y)) %>% length,

nfold = 10, nrounds = 500, early\_stopping\_rounds = 8, booster = 'gbtree',

objective = 'multi:softprob', eval\_metric = 'mlogloss',

verbose = F, prediction = T)

# 예측 데이터 생성(max.col 사용해서 가장 큰값 추출)

pred\_df = import\_model$pred %>% as.data.frame %>%

mutate(pred = levels(as.factor(y))[max.col(.)] %>% as.factor, actual = as.factor(y))

pred\_df %>% select(pred,actual) %>% table

# confusionMatrix 생성

import\_conf<- caret::confusionMatrix(pred\_df$pred, pred\_df$actual)

import\_conf