SMU

AI+X 선도인재양성 프로그램

데이터분석과 인사이트 도출

스마트폰 센서 데이터 기반 인간 행동 인식 분류

Outline

프로젝트 개요

Human Activity Recognition
Using Smartphone Data Set

- 목표

새로운 도메인의 데이터를 탐색하고
좋은 성능을 낼 수 있게 전처리한 후
최적의 머신러닝/딥러닝 모델을 완성한다.

Domain Knowledge

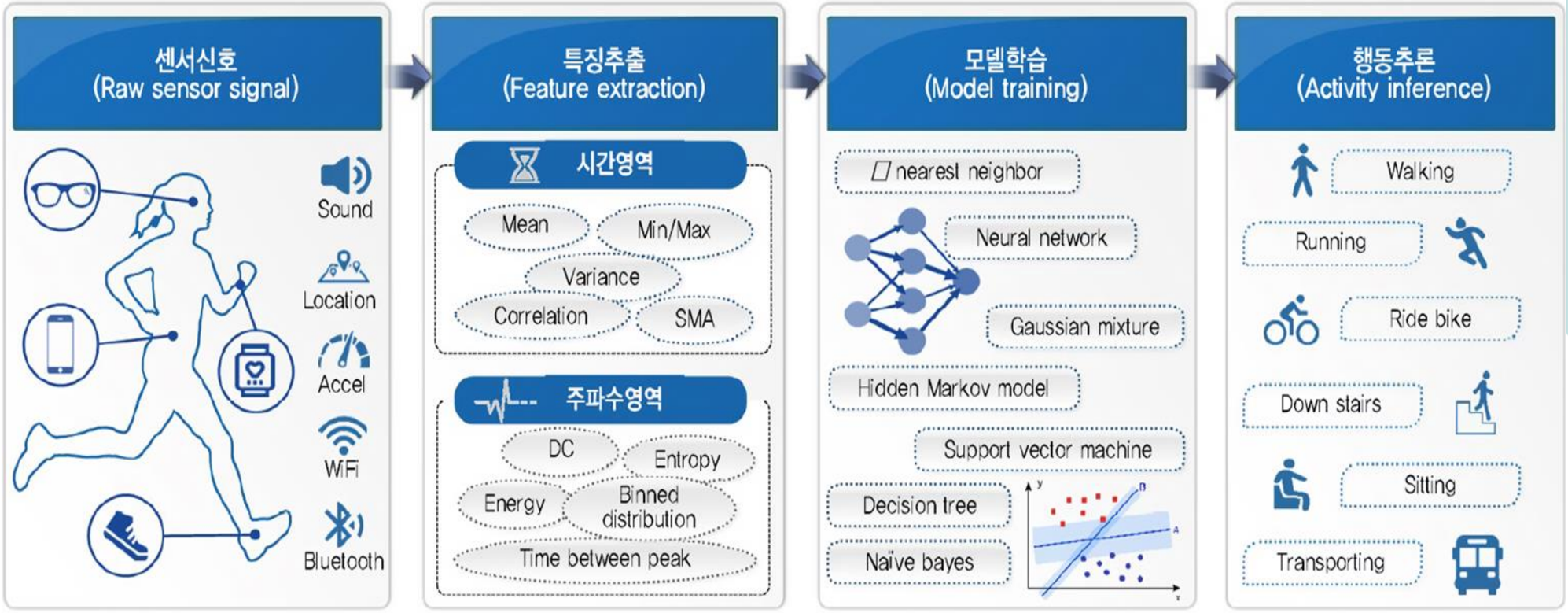
인간 행동 인식

인간 행동 인식

(HAR: Human Activity Recognition)

다양한 **센서를 활용**하여 사람의 모션에 관련된

정보를 수집하고 해석하여 행동을 인식 하는 기술



Domain Knowledge

인간 행동 인식

Domain Knowledge

인간 행동 인식

센서 신호

Raw sensor signal

특징 추출

Feature extraction

모델 학습

Model training

행동 추론

Activity inference

Dataset

데이터셋 소개

UCI – Human Activity Recognition

UCI Machine Learning Repository
Center for Machine Learning and Intelligent Systems

Check out the beta version of the new UCI Machine Learning Repository we are currently testing! Contact us if you have any issues, questions, or concerns. Click here to try out the new site.

Human Activity Recognition Using Smartphones Data Set

Download: Data Folder, Data Set Description

Abstract: Human Activity Recognition database built from the recordings of 30 subjects performing activities of daily living (ADL) while carrying a waist-mounted smartphone with embedded inertial sensors.

Data Set Characteristics:	Multivariate, Time-Series	**Number of Instances:**	10299	**Area:**	Computer
Attribute Characteristics:	N/A	**Number of Attributes:**	561	**Date Donated**	2012-12-10
Associated Tasks:	Classification, Clustering	**Missing Values?**	N/A	**Number of Web Hits:**	1301361

Source:

Jorge L. Reyes-Ortiz(1,2), Davide Anguita(1), Alessandro Ghio(1), Luca Oneto(1) and Xavier Parra(2)
1 - Smartlab - Non-Linear Complex Systems Laboratory
DITEN - Università degli Studi di Genova, Genoa (I-16145), Italy.
2 - CETpD - Technical Research Centre for Dependency Care and Autonomous Living
Universitat Politècnica de Catalunya (BarcelonaTech). Vilanova i la Geltrú (08800), Spain
activityrecognition '@' smartlab.ws

Dataset

데이터셋 소개

data.csv

(row **5881**, column **563**)

	tBodyAcc-mean()-X	tBodyAcc-mean()-Y	tBodyAcc-mean()-Z	tBodyAcc-std()-X	...	Subject	Activity		
1	0.288585	-0.02029	-0.13291	-0.99528		1			
2	0.278419	-0.01641	-0.12352	-0.99825		1			
3	0.279653	-0.01947	-0.11346	-0.99538		1			
4	0.279174	-0.0262	-0.12328	-0.99609		1			
5	0.276629	-0.01657	-0.11536	-0.99814		1			
6	0.277199	-0.0101	-0.10514	-0.99733		1			
7	0.279454	-0.01964	-0.11002	-0.99692		1			
...									

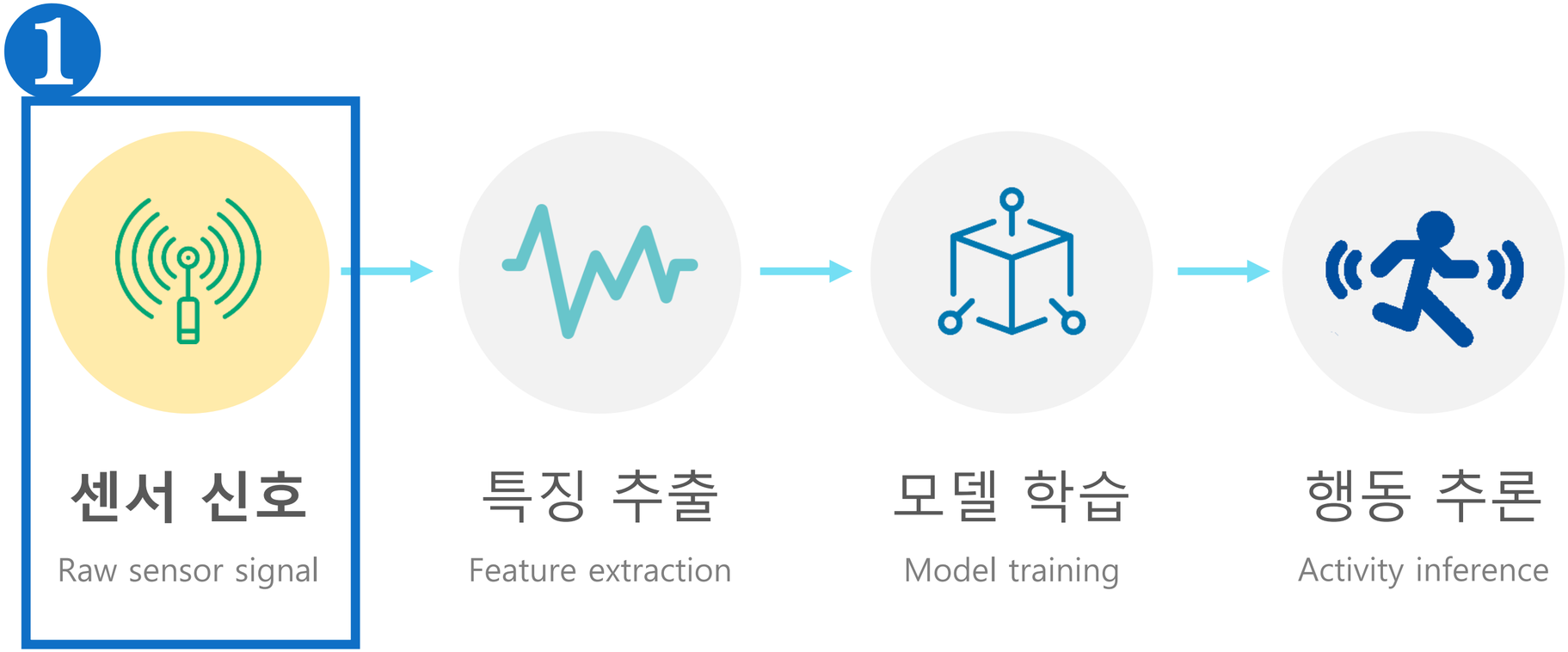
X_feature
562

tBodyAcc-mean()-X
tBodyAcc-mean()-Y
tBodyAcc-mean()-Z
tBodyAcc-std()-X
tBodyAcc-std()-Y
...

Y_label 1

Activity

데이터 소개 – 데이터 수집 방식



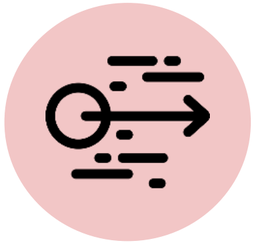
- ✓ **30 volunteers**
- ✓ **Wearing Samsung Galaxy S2**
- ✓ **Performing 6 posture activities**

Dataset

데이터셋 소개

Accelerometer

가속도 센서



일직선으로 움직이는 물체의 선형

가속도를 측정하는 센서

"일정 시간 동안 X, Y, Z 축으로 얼마나 빠르게 움직였는가?"

Gyroscope

자이로스코프 센서

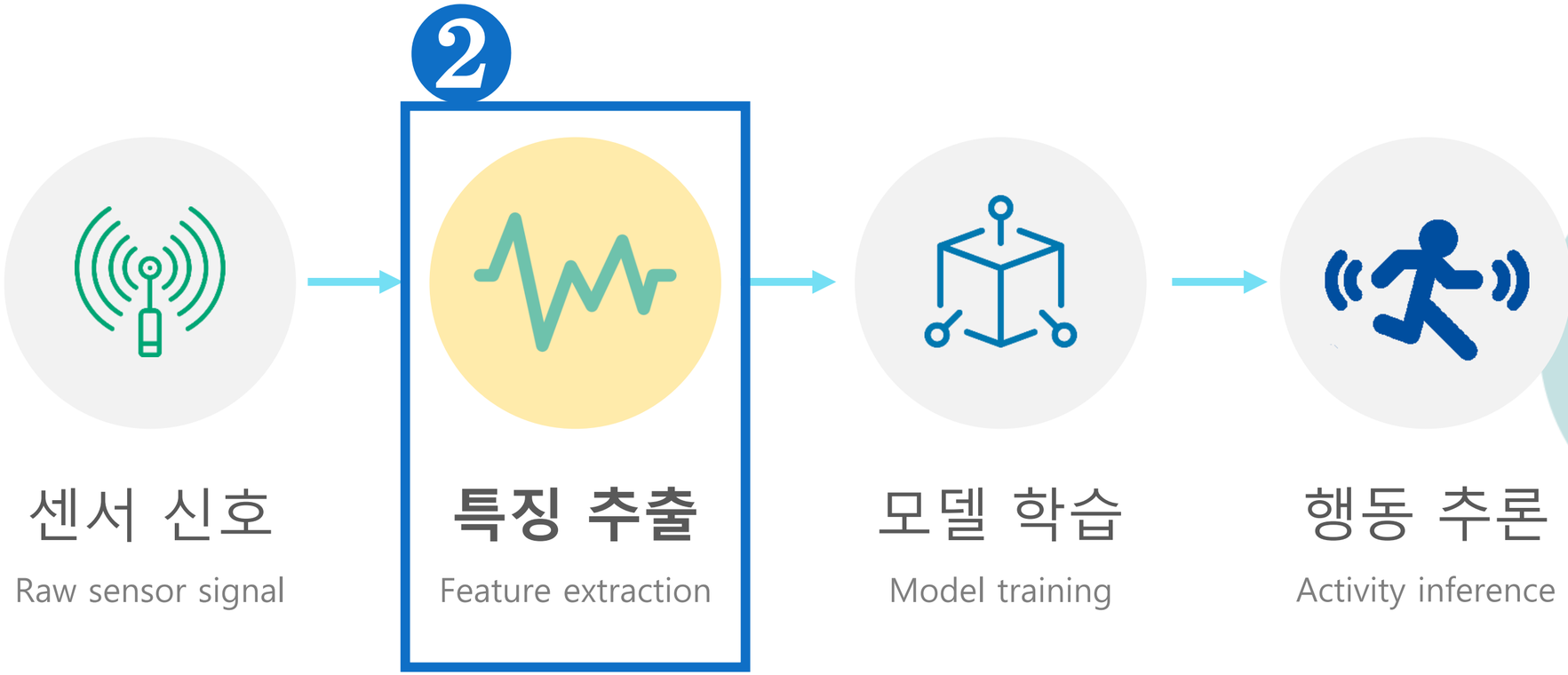
회전하는 물체의

각속도를 측정하는 센서

"일정 시간 동안 X, Y, Z 축으로 각도가 얼마나 변했는가?"

Dataset

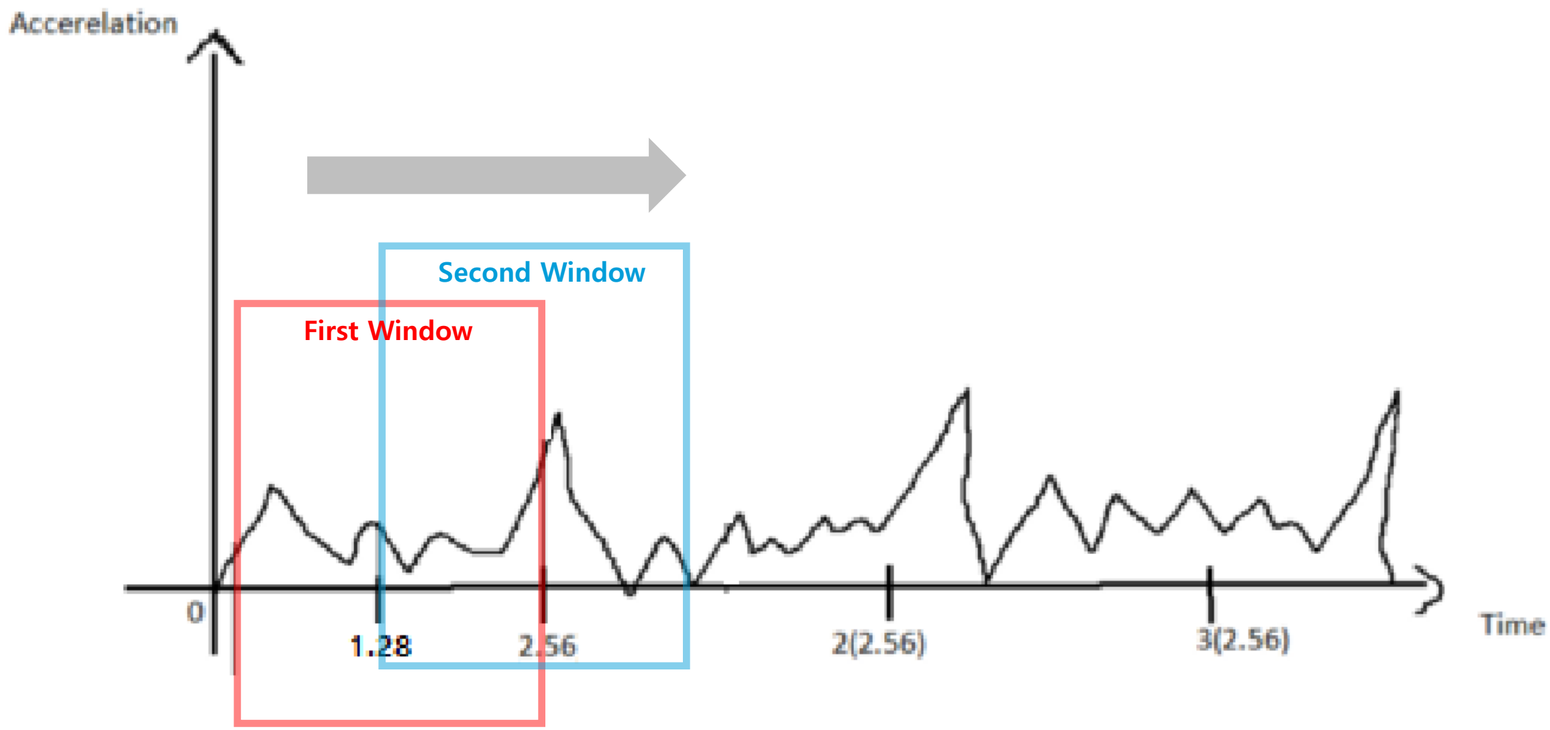
데이터셋 소개



Dataset

데이터셋 소개

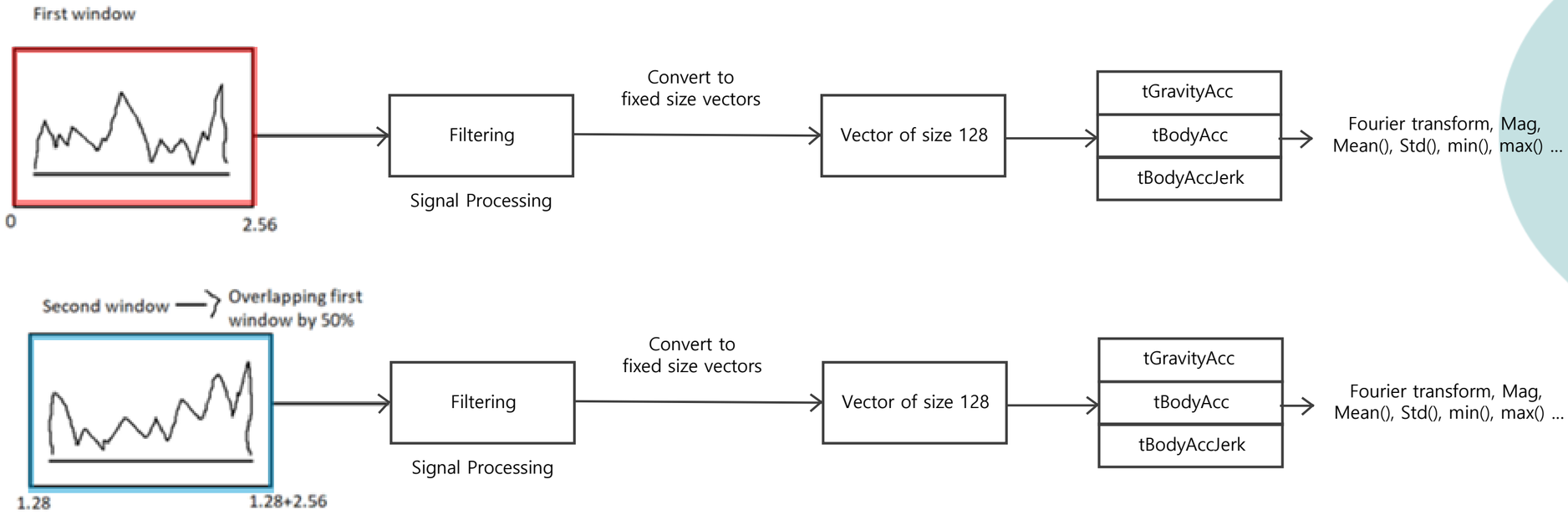
① 수집 : 2.56초 범위(window), 1.28초 간격으로 이동하며 데이터 샘플링(하나의 행)



Dataset

데이터셋 소개

② 집계 : 하나의 window에서 수집된 데이터를 신호별로 분리 후 집계 (mean, std, min, max..)



Dataset

데이터셋 소개

✓ **수집과 집계로 만들어진 데이터 세트의 일부분 입니다.**

✓ **Feature 이름들이 복잡하게 보입니다.**

- 모든 feature를 다 이해해야 하는 것은 아닙니다
- 그러나 구조를 이해하면 많은 feature들을 파악할 수 있습니다.

tBodyAcc-mean()-X	tBodyAcc-mean()-Y	tBodyAcc-mean()-Z	tBodyAcc-std()-X	tBodyAcc-std()-Y	tBodyAcc-std()-Z	tBodyAcc-mad()-X	tBodyAcc-mad()-Y	tBodyAcc-mad()-Z	tBodyAcc-max()-X	tBodyAcc-max()-Y	tBodyAcc-max()-Z	tBodyAcc-min()-X	tBodyAcc-min()-Y	tBodyAcc-min()-Z	tBodyAcc-sma()	tBodyAcc-energy()-X	tBodyAcc-energy()-Y
0.288508	-0.0092	-0.10336	-0.98899	-0.9628	-0.96742	-0.989	-0.9626	-0.96565	-0.92975	-0.5546	-0.79254	0.84868	0.681264	0.83028	-0.97219	-0.99987	-0.99948
0.265757	-0.01658	-0.09816	-0.98955	-0.99464	-0.98744	-0.99019	-0.99387	-0.98756	-0.93734	-0.57395	-0.81314	0.839654	0.694376	0.845888	-0.98894	-0.99987	-0.99997
0.278709	-0.01451	-0.10872	-0.99772	-0.98109	-0.99401	-0.99793	-0.98219	-0.99502	-0.94258	-0.56645	-0.82291	0.851978	0.681985	0.845253	-0.99424	-0.99999	-0.99985
0.289795	-0.03554	-0.15035	-0.23173	-0.00641	-0.33812	-0.27356	0.014245	-0.34792	0.008288	-0.13654	-0.41072	0.210761	0.067789	0.345664	-0.12894	-0.70359	-0.80839
0.394807	0.034098	0.091229	0.088489	-0.10664	-0.3885	-0.01047	-0.10968	-0.34637	0.584131	-0.11117	-0.36858	0.089686	0.135817	0.597771	-0.00491	-0.40427	-0.84052
0.330708	0.007561	-0.06137	-0.21576	0.101075	0.072949	-0.26986	0.06006	0.101298	-0.01926	0.187013	-0.12988	0.201169	0.029101	0.086293	0.006264	-0.69068	-0.7648
0.121465	-0.0319	-0.0052	-0.1522	-0.1131	-0.23942	-0.2024	-0.1647	-0.2471	0.114668	-0.0935	-0.06113	0.060086	0.026294	0.444026	-0.10349	-0.63456	-0.84706
0.272026	-0.00133	-0.12549	-0.99207	-0.91298	-0.97245	-0.99475	-0.94314	-0.97643	-0.92545	-0.40781	-0.80814	0.846043	0.69361	0.815227	-0.97381	-0.99993	-0.99778
0.284338	0.021956	-0.00693	-0.98015	-0.83839	-0.78236	-0.98368	-0.8162	-0.74392	-0.91401	-0.43033	-0.67713	0.83196	0.659073	0.811987	-0.85519	-0.99971	-0.99167

Dataset

데이터셋 소개

✓ **Sensor는 두 가지입니다.**

가속도 센서
Accelerometer

tBody**Acc** - **mean**() - **X**

자이로스코프 센서
Gyroscope

fBody**Gyro** - **std**() - **Y**

Dataset

데이터셋 소개

✓ **주요한 집계 함수는 다음과 같습니다.**

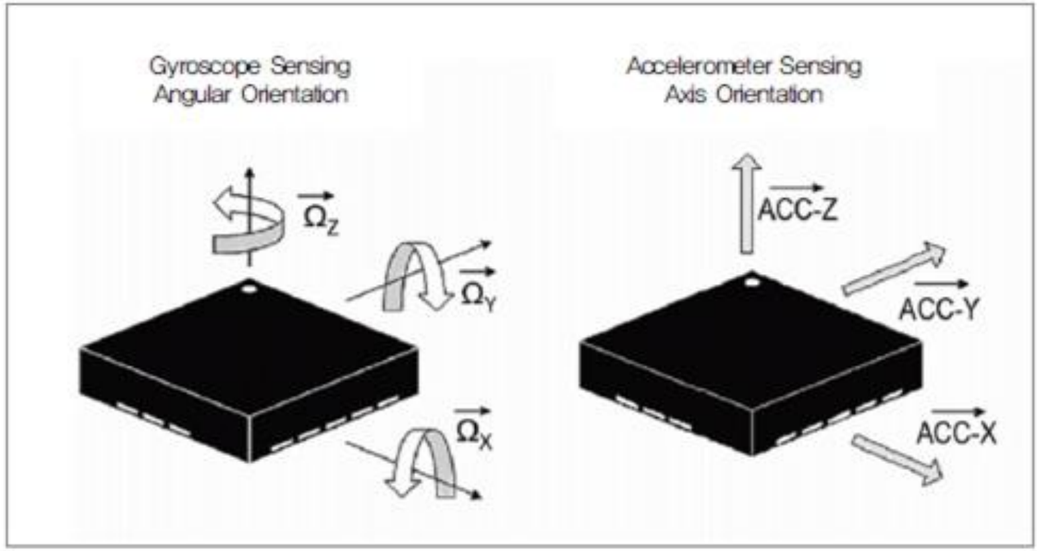
Aggregation	의미
mean()	평균; Mean value
std()	표준편차; Standard deviation
mad()	평균절대편차; Absolute deviation of the median
max()	최대값; Largest value in the array
min()	최소값; Smallest value in the array
sma()	시계열의 산술평균; Signal magnitude area
energy()	에너지 측정값 ; Energy measure;
iqr()	3사분위수 - 1사분위수; Interquartile range
entropy()	신호의 엔트로피; Signal entropy
arCoeff()	자기회귀계수; Autoregression coefficients
correlation()	상관계수; Correlation between two signals
maxInds	최대인덱스; Largest value in Indexs
meanFreq()	평균 빈도; Mean frequency
skewness()	왜도(비대칭도); skewness
kurtosis()	첨도(평균에 몰려 있는 정도); kurtosis

Dataset

✓ 데이터셋 소개

Axis는 x,y,z 축에 대해서 센서마다 차이가 있습니다.

- Gyroscope : 각 축(x, y, z) 방향으로 회전을 의미
- Accelerometer : 각 축 (x, y, z) 방향으로 이동을 의미



Dataset

데이터셋 소개

feature name	설명
tBodyAcc - mean() - X	X축 방향 가속도의 평균값
tBodyGyro - std() – X	X축 방향 각속도 의 표준편차
tBodyAccJerk - iqr() - Y	Y축 방향 가속도변화비율의 사분범위
tBodyGyroJerk - arCoeff() – Y,4	Y축 방향 각속도변화속도의 자기회귀계수
tGravityAcc - max() - Z	Z축 방향 중력가속도의 최댓값

프로젝트 진행

데이터 분석 - 1학기 진행 내용 warp-up

주어진 데이터의 구성 및 분포를 직접 확인 해 보고
분류 문제 해결에 유의미한 feature 및 label 변수를
탐색적으로 분석한다.

- 기본 정보 확인하기
- 변수별 중요도 확인하기
- 그룹별 중요도 확인하기
- 일부 변수의 실제값 분포 확인하기

프로젝트 진행

데이터 분석 – 1학기 진행 내용 warp-up

수 많은 feature들을 모두 살펴보는 것은 많은 시간과 노력이 필요합니다.
우리는 **선택**과 **집중**이 필요합니다
더불어 데이터를 다루는 능력도 트레이닝도 해보려고 합니다!

1) 주어진 데이터 (data, feature) 탐색 합니다

. 단변량 분석– 기초 통계량, 그래프 등 , 변수 10개 이상 진행 해보세요.

. 이변량 분석– 이변량 분석 및 그래프

2) 기본 모델을 생성한 후 **변수 중요도**를 구합니다. (옵션)

(random forest 알고리즘 사용하여 제공해 드립니다.)

3) 중요한 featur와 중요하지 않은 feature 상위 N개를 선정하고, 이들을 대상으로 EDA 수행합니다.

4) 각 feature 그룹별 중요도도 파악해보며 EDA를 수행

Feature 이름에는 계층구조를 담고 있습니다. 그렇다 보니 feature들을 적절하게 그룹으로 묶을 수 있습니다.

참고로, feature 그룹의 중요도는 개별 feature 중요도의 합으로 계산 할 수 있습니다.

프로젝트 진행

데이터 분석 – data.csv 소개

data.csv

	tBodyAcc-mean()-X	tBodyAcc-mean()-Y	tBodyAcc-mean()-Z	tBodyAcc-std()-X	tBodyAcc-std()-Y	tBodyAcc-std()-Z	tBodyAcc-mad()-X	tBodyAcc-mad()-Y	tBodyAcc-mad()-Z	tBodyAcc-max()-X	...	angle(Z,gravityMean)	Activity
0	0.288508	-0.009196	-0.103362	-0.988986	-0.962797	-0.967422	-0.989000	-0.962596	-0.965650	-0.929747	...	0.165163	STANDING
1	0.265757	-0.016576	-0.098163	-0.989551	-0.994636	-0.987435	-0.990189	-0.993870	-0.987558	-0.937337	...	-0.147944	LAYING
2	0.278709	-0.014511	-0.108717	-0.997720	-0.981088	-0.994008	-0.997934	-0.982187	-0.995017	-0.942584	...	-0.032755	STANDING
3	0.289795	-0.035536	-0.150354	-0.231727	-0.006412	-0.338117	-0.273557	0.014245	-0.347916	0.008268	...	0.111388	WALKING
4	0.394807	0.034098	0.091229	0.088489	-0.106636	-0.388502	-0.010469	-0.109680	-0.346372	0.584131	...	0.137758	WALKING_DOWNSTAIRS

프로젝트 진행

데이터 분석 – feature.csv 소개

feature.csv

	sensor	agg	axis	feature_name
0	tBodyAcc	mean()	X	tBodyAcc-mean()-X
1	tBodyAcc	mean()	Y	tBodyAcc-mean()-Y
2	tBodyAcc	mean()	Z	tBodyAcc-mean()-Z
3	tBodyAcc	std()	X	tBodyAcc-std()-X
4	tBodyAcc	std()	Y	tBodyAcc-std()-Y

프로젝트 진행

데이터 분석 – fi_analysis.csv **생성방법**

random forest 의 feature Importance 사용

```
]: #생성
model = RandomForestClassifier()

#학습
model.fit(x_train, y_train)
pred = model.predict(x_val)

#평가
print('accuracy :',accuracy_score(y_val, pred))
print('='*60)
print(confusion_matrix(y_val, pred))
print('='*60)
print(classification_report(y_val, pred))
```

97.4% 정확도

```
accuracy : 0.9745042492917847
============================================================
[[346   0   0   0   0   0]
 [  0 297  16   0   0   1]
 [  0  14 317   0   0   0]
 [  0   0   0 287   4   0]
 [  0   0   0   3 240   1]
 [  0   0   0   1   5 233]]
============================================================
                    precision    recall  f1-score   support

           LAYING       1.00      1.00      1.00       346
          SITTING       0.95      0.95      0.95       314
         STANDING       0.95      0.96      0.95       331
          WALKING       0.99      0.99      0.99       291
WALKING_DOWNSTAIRS       0.96      0.98      0.97       244
  WALKING_UPSTAIRS       0.99      0.97      0.98       239

          accuracy                           0.97      1765
         macro avg       0.97      0.97      0.97      1765
      weighted avg       0.97      0.97      0.97      1765
```

97% 정도의 정확도를 갖는 model에 기여하는 변수의 중요도로 Target 변수에 영향을 주는 중요도 자료로 활용

프로젝트 진행

데이터 분석 – fi_analysis.csv 소개

fi_analysis.csv

	sensor	agg	axis	feature_name	fi_all	fi_dynamic	fi_standing	fi_sitting	fi_laying	fi_walking	fi_walking_up	fi_walking_down
0	tBodyAcc	mean()	X	tBodyAcc-mean()-X	0.000211	0.000108	0.000603	0.000222	0.000152	0.000194	0.000170	0.000233
1	tBodyAcc	mean()	Y	tBodyAcc-mean()-Y	0.000460	0.000049	0.001376	0.000663	0.000078	0.000156	0.000566	0.000312
2	tBodyAcc	mean()	Z	tBodyAcc-mean()-Z	0.000180	0.000020	0.000190	0.000329	0.000153	0.000097	0.000244	0.000189
3	tBodyAcc	std()	X	tBodyAcc-std()-X	0.005055	0.000007	0.007170	0.001380	0.001158	0.011469	0.004446	0.016283
4	tBodyAcc	std()	Y	tBodyAcc-std()-Y	0.000360	0.000000	0.000765	0.000243	0.000251	0.000106	0.000565	0.001756

데이터 분석 – 개인 미션 ①

중요도 기반 feature 분석 (상, 하위 N개 분석)

[18]:
```
# 중요도 상위 top 5
r0.head(5)
```

[18]:

	feature_name	feature_importance
0	tGravityAcc-max()-X	0.032798
1	angle(Y,gravityMean)	0.030676
2	tGravityAcc-energy()-X	0.030037
3	tGravityAcc-min()-X	0.025349
4	angle(X,gravityMean)	0.024910

[19]:
```
# 중요도 하위 top 5
r0.tail(5)
```

[19]:

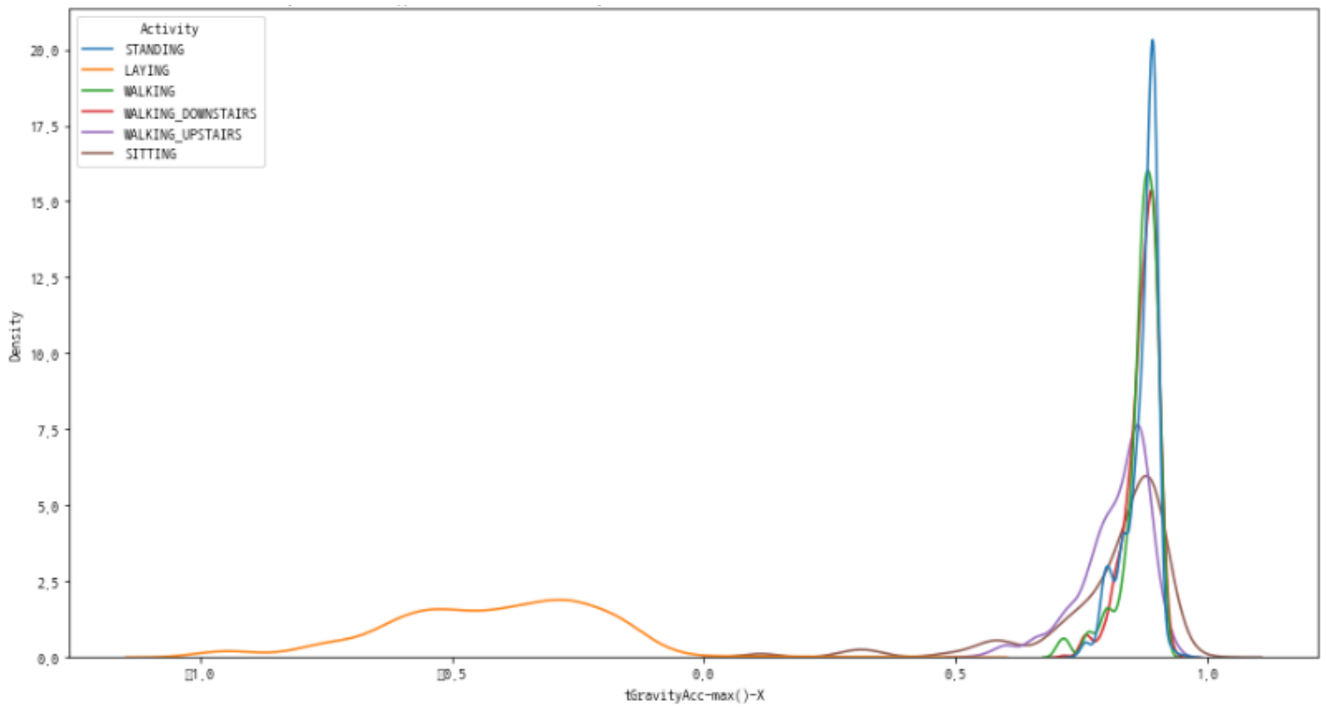
	feature_name	feature_importance
556	fBodyAcc-bandsEnergy()-49,64.2	0.000093
557	fBodyAccJerk-bandsEnergy()-49,64	0.000092
558	fBodyAcc-bandsEnergy()-25,32.1	0.000087
559	fBodyBodyGyroJerkMag-iqr()	0.000079
560	fBodyBodyAccJerkMag-entropy()	0.000078

데이터 분석 – 개인 미션 ①

중요도 상위 변수 분석

2) 상위 5개 변수에 대한 분석

[20]:
```
# 1위 : tGravityAcc-max()-X
var = 'tGravityAcc-max()-X'
plt.figure(figsize = (15,8))
sns.kdeplot(x=var, data = data, hue =target, common_norm = False)
```

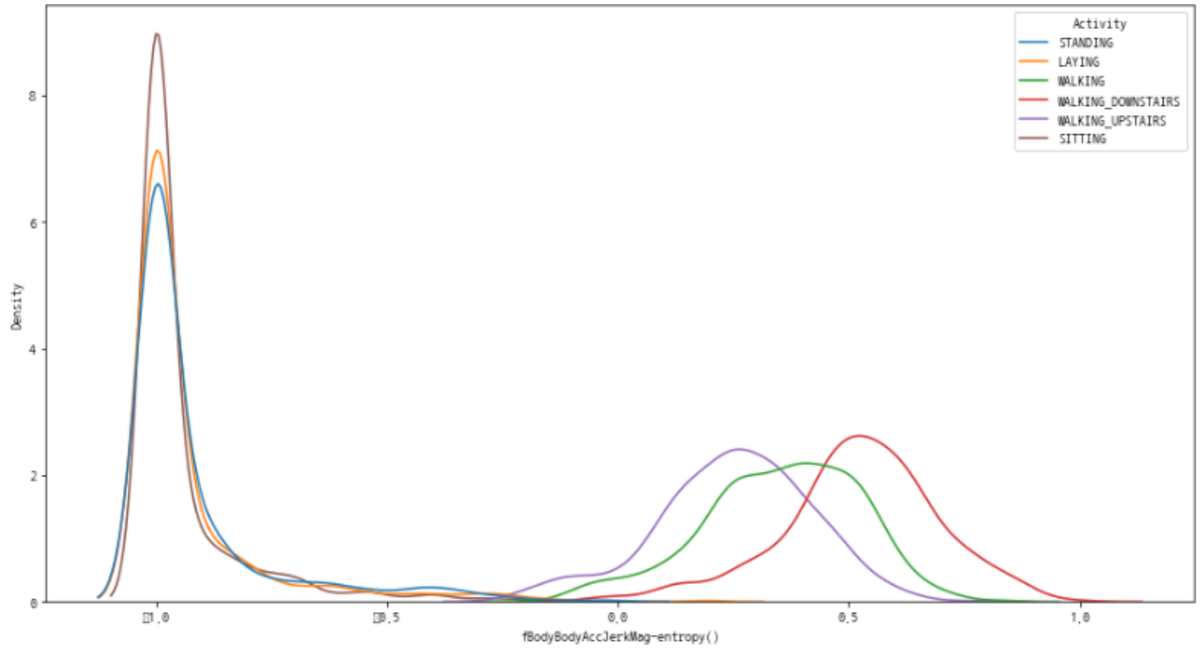


데이터 분석 – 개인 미션 ①

중요도 하위 변수 분석

3) 하위 5개 변수에 대한 분석

[25]:
```
# 1위 : fBodyBodyAccJerkMag-entropy()
var = 'fBodyBodyAccJerkMag-entropy()'
plt.figure(figsize = (15,8))
sns.kdeplot(x=var, data = data, hue =target, common_norm = False)
```

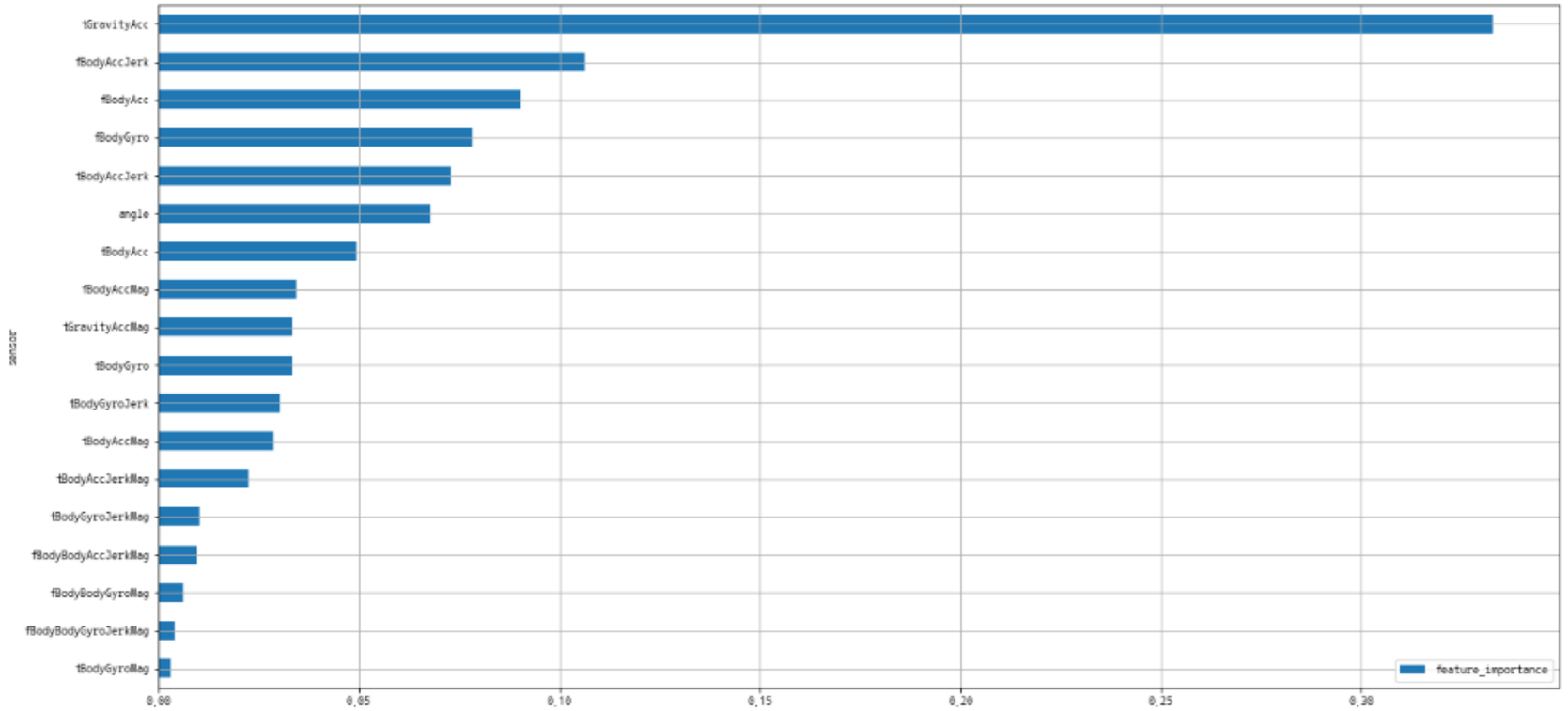


데이터 분석 – 개인 미션 ①

sensor 별 중요도

2) sensor 별 중요도

[32]:
```
result.groupby('sensor')[['feature_importance']].sum().sort_values('feature_importance').plot.barh(figsize=(20,10))
plt.grid()
plt.show()
```

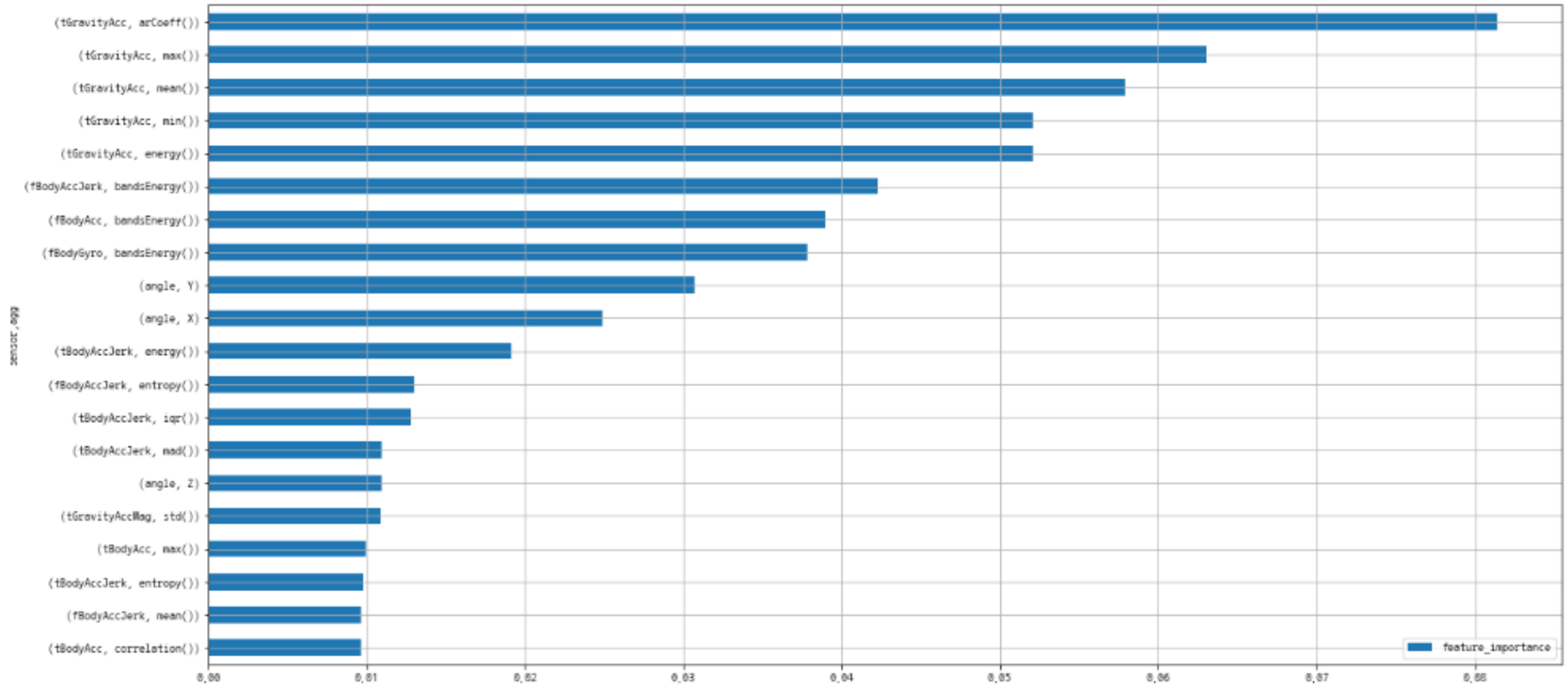


데이터 분석 – 개인 미션 ①

Sensor + agg 별 중요도

3) sensor + agg 별 중요도

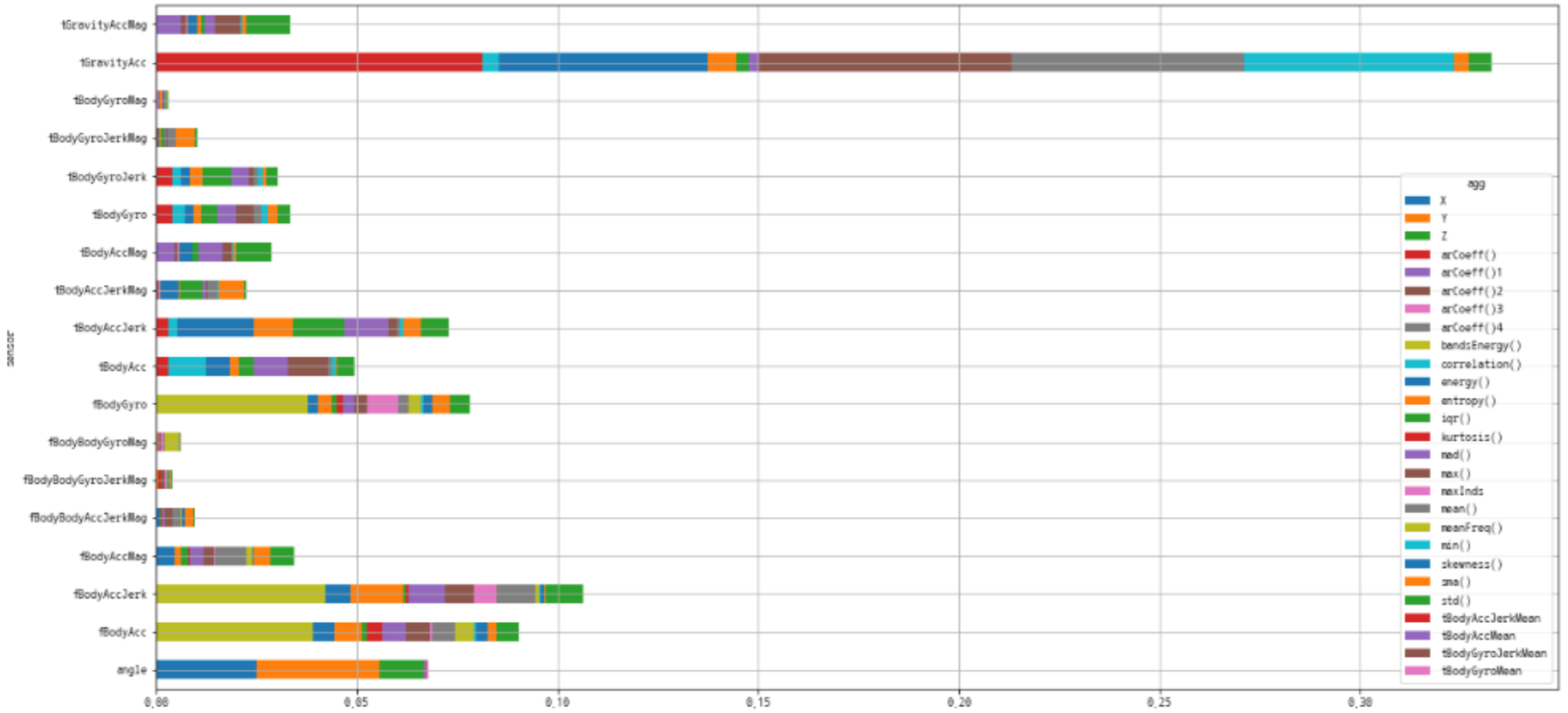
[33]:
```
# 상위 20개만 조회
temp = result.groupby(['sensor','agg'])[['feature_importance']].sum().sort_values('feature_importance')
temp.tail(20).plot.barh(figsize=(20,10))
plt.grid()
plt.show()
```



데이터 분석 – 개인 미션 ①

Sensor + agg 별 중요도

[34]:
```
# Sensor 별, agg로 나눠서 분석하기
result.groupby(['sensor','agg'])['feature_importance'].sum().unstack().plot(kind='barh', stacked=True, figsize = (20,10))
plt.grid()
plt.show()
```



데이터 분석 – 개인 미션 ②

다음의 case에 맞게 feature 및 feature 그룹 중요도를 기반으로 탐색적 데이터 분석을 수행합니다.

1) Target을 정적/동적 행동으로 구분

6개의 행동은 2개의 그룹(정적행동, 동적행동)으로 나뉩니다. 어떤 feature(혹은 feature 그룹)이 2개 class 그룹(정적행동, 동적행동)를 구분하는데 중요한지를 찾아보고 탐색해봅시다.

2) Target을 개별 행동 여부로 구분

6가지의 행동을 분류하는 분석도 중요하지만, 개별 행동에만 특별히 영향을 받는 feature들도 있습니다.

예를 들어, 계단을 오르는 행동(Walking_upstairs)과 관련이 큰 feature가 있을 것입니다.

[계단을 오르는 행동]인지 아닌지로 구분하는 targe을 추가하여 EDA를 수행해 봅시다.

데이터 분석 주요 설명

- ✓ Label인 Activity는 6가지 class.
- ✓ 실제 스마트폰을 이용하여 측정하는 사람들이 입력한 Activity 입니다.
- ✓ 이 값을 가지고 다양하게 label을 추가해서 분석해야 합니다.

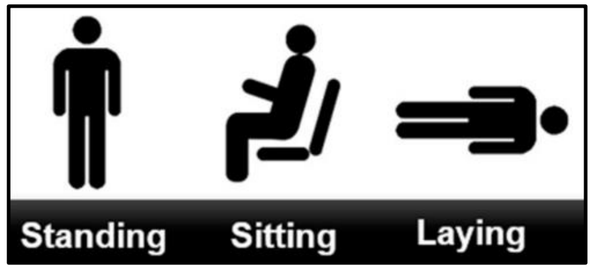
Activity
STANDING
SITTING
LAYING
WALKING
WALKING_UPSTAIRS
WALKING_DOWNSTAIRS

데이터 분석 주요 설명

✓ 두가지 Class(정적, 동적)로 나눠서 분석하고 모델링할 수 있습니다.

- 정적 행동 : STANDING, SITTING, LAYING
- 동적 행동 : WALKING, WALKING-UPSTAIRS, WALKING-DOWNSTAIRS

Activity	is_dynamic
STANDING	0
SITTING	0
LAYING	0
WALKING	1
WALKING_UPSTAIRS	1
WALKING_DOWNSTAIRS	1



데이터 분석 주요 설명

✓ 각 행동별로 분석 및 모델링하기 위한 target을 추가할 수 있습니다.

- Standing 인지 아닌지 구분하는데 중요한 feature는? ☒ is_Standing
- Walking 인지 아닌지 분류하기 위한 최적의 모델은? ☒ is_Walking

Activity	is_dynamic	is_Standing	is_Sitting	…	is_Down
STANDING	0	1	0		0
SITTING	0	0	1		0
LAYING	0	0	0		0
WALKING	1	0	0		0
WALKING_UPSTAIRS	1	0	0		0
WALKING_DOWNSTAIRS	1	0	0		

참고 설명

✓ joblib 의 두 함수

- dump : 저장

```
joblib.dump(data, 'data_df.pkl')
```

- Load : 불러오기

```
data2 = joblib.load('data_df.pkl')
```

✓ 저장할 수 있는 대상

- 데이터 자료형 : 리스트, 딕셔너리, 데이터프레임, 넘파이어레이…
- 피팅한 전처리 함수들 : SimpleImputer, KNNImputer, Scaler…
- 학습한 모델
  - .fit 으로 학습해 놓은 모델을 파일로 저장할 수 있습니다.

Outline

문제정의

이번 미니 프로젝트의 의도는?

스마트폰 행동 인식 데이터를 활용 해 동작을 분류하는 최적의 모델을 만들어 봅시다.

프로젝트 진행

과제 핵심 사항

Challenge

너무 많은 Feature들 (561개)

EDA 수행의 어려움

모든 feature들에 대해서
다 그래프를 그려야 할까?

모델의 복잡도 증가

모든 feature가 모델링에 필요할까?

다중분류

6개 Class

상호 관련이 있는 Class들

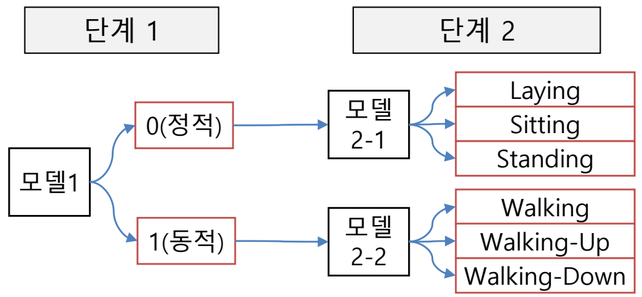
- 정적 : Laying, Sitting, Standing
- 동적 : Walking, Walking-Up, Walking-Down

과제 수행 전략

선택과 집중

- ✓ 트리 기반 모델로 부터 변수 중요도 추출
- ✓ 상위 N개 변수에 대해 탐색 및 모델링

단계별 모델링



프로젝트 진행

진행 순서

ML/DL 모델링 'A~Z' 경험

이번 과제입니다!

도메인 이해

데이터 수집

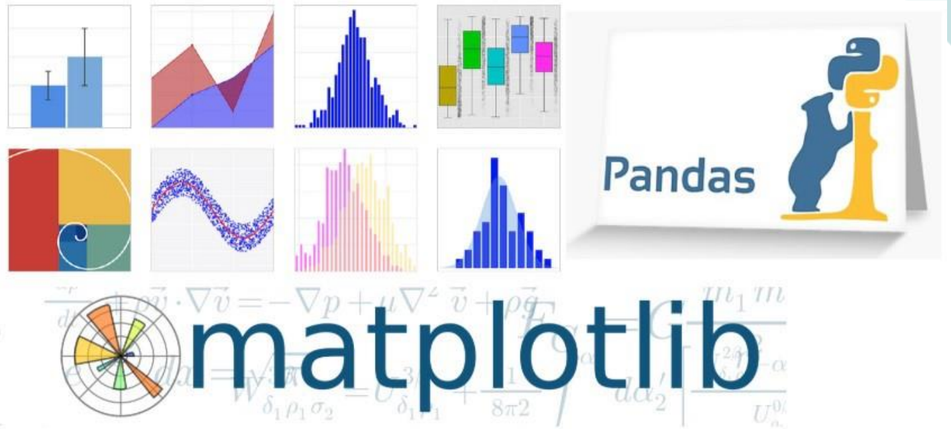
데이터 전처리

데이터 분석

모델링

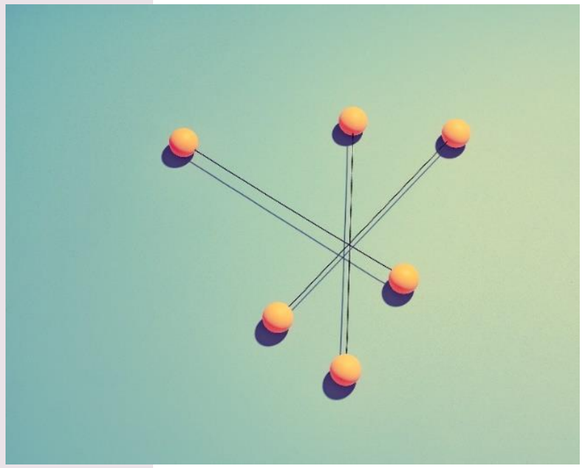
프로젝트 진행

참고 기술



프로젝트 진행

과제 수행



기본 모델링

✓ 미션

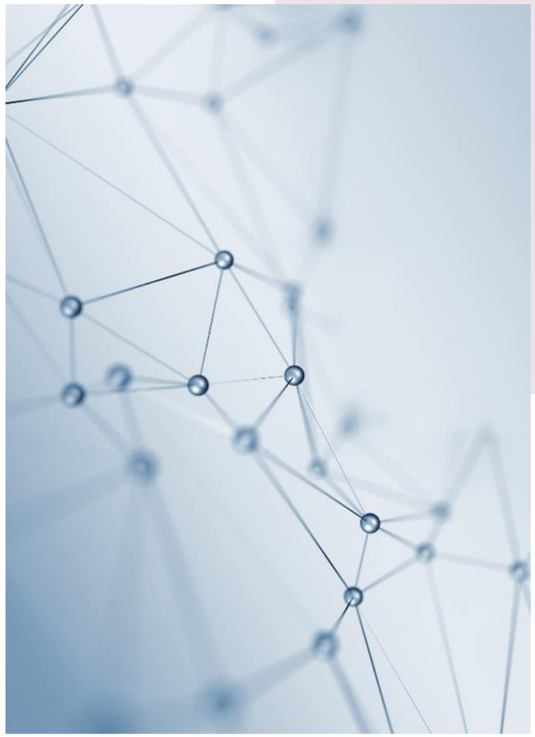
- **데이터 전처리**
  - 가변수화, 데이터 분할, NaN 확인 및 조치, 스케일링 등 필요한 전처리 수행
- **다양한 알고리즘으로 분류 모델 생성**
  - **최소 4개 이상의 알고리즘**을 적용하여 모델링 수행
- **성능 비교**
  - 각 모델의 성능을 저장하는 별도 데이터 프레임을 만들고 비교

✓ 옵션

- 전체 변수로 모델링, 중요도 상위 N개 변수를 선택하여 모델링을 수행하고 성능 비교

프로젝트 진행

과제 수행



단계별 모델링

✓ **미션**

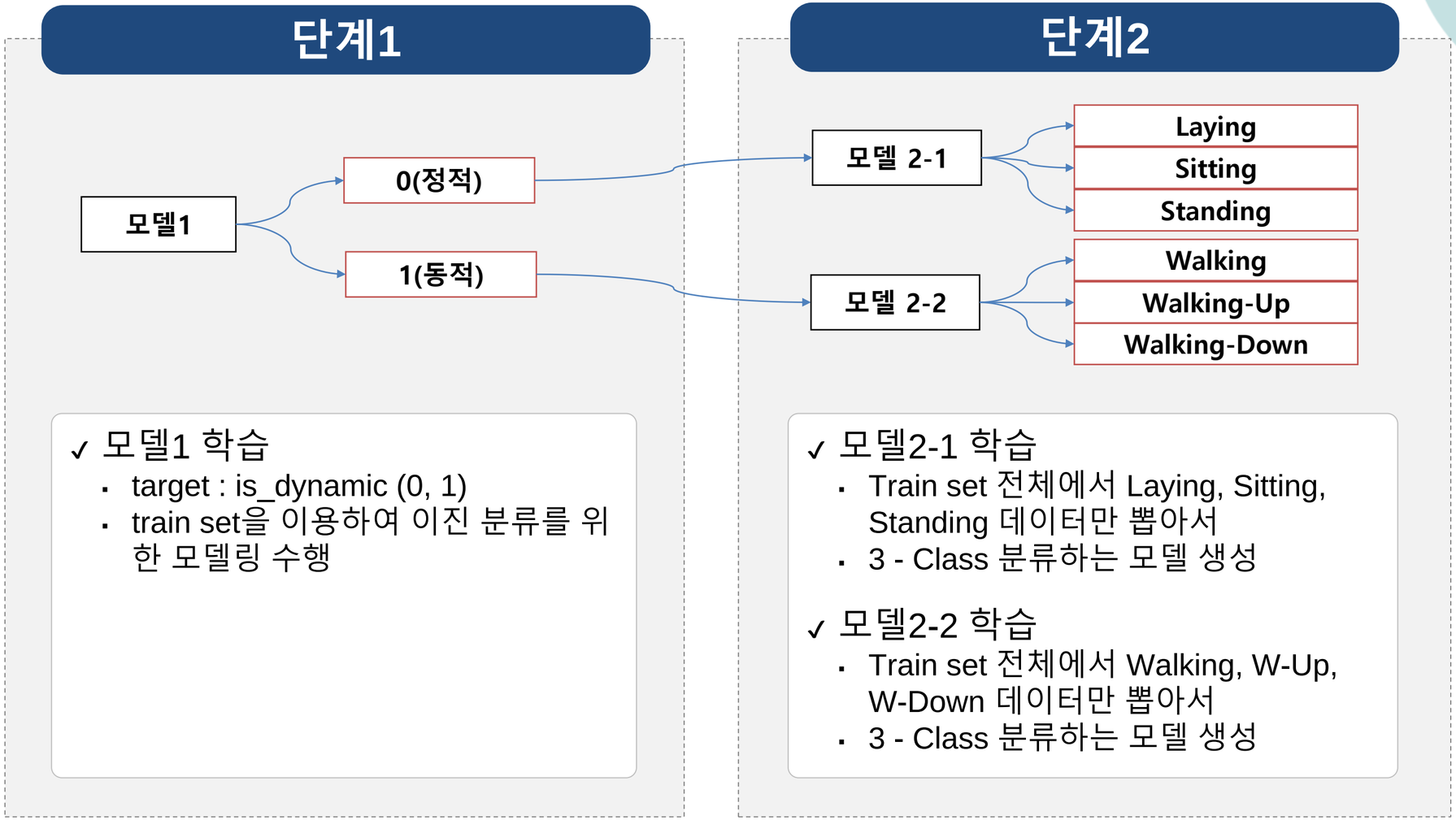
- **단계1 : 정적(0), 동적(1) 행동 분류 모델 생성**
- **단계2 : 세부 동작에 대한 분류모델 생성**
  - 단계1 모델에서 0으로 예측 - 정적 행동 3가지 분류 모델링
  - 단계1 모델에서 1으로 예측 - 동적 행동 3가지 분류 모델링

- **모델 통합**
  - 두 단계 모델을 통합하고, 새로운 데이터에 대해서 최종 예측결과와 성능평가가 나오도록 함수로 만들기

- **성능 비교**
  - 기본 모델링의 성능과 비교
  - 모든 모델링은 [다양한 알고리즘 + 성능 튜닝]을 수행해야 합니다.

프로젝트 진행

과제 수행



✓ 모델1 학습
- target : is_dynamic (0, 1)
- train set을 이용하여 이진 분류를 위한 모델링 수행

✓ 모델2-1 학습
- Train set 전체에서 Laying, Sitting, Standing 데이터만 뽑아서
- 3 - Class 분류하는 모델 생성

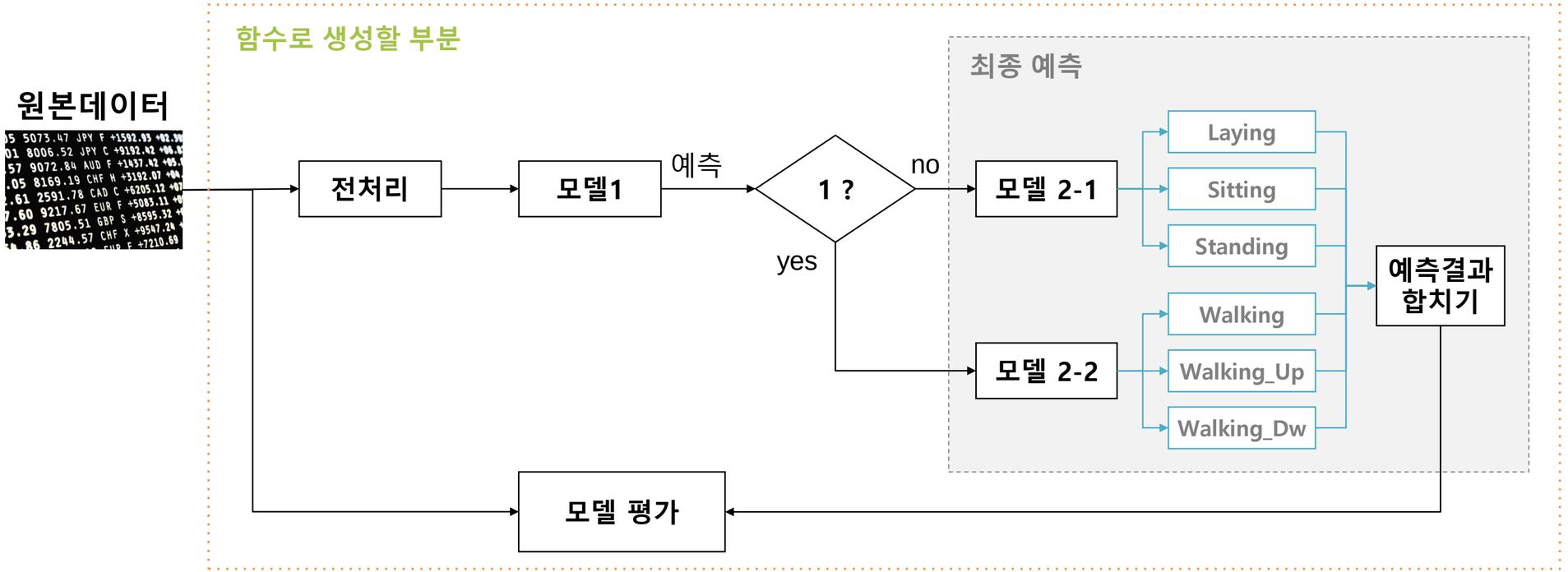
✓ 모델2-2 학습
- Train set 전체에서 Walking, W-Up, W-Down 데이터만 뽑아서
- 3 - Class 분류하는 모델 생성

프로젝트 진행

과제 수행

✓ **아래 절차를 참조하여, 각 단계별 코드를 준비하고 하나의 함수로 실행되도록 개발합니다.**

- 입력 : 원본 데이터(제공되는 test set)
- 출력 : 예측 결과, 성능평가 결과 등.



<< 실습 코드 & 데이터

hhttps://shorturl.at/ukoE5

Thank you !

감사합니다.