

개인 과제 레포트

제출일자 : 2021.01.12.(화)

이름 : 허지혜

1. 주제

운동 동작 분류 AI 알고리즘

2. 주제선정동기

스마트 시대에 맞게 스마트 헬스케어 산업에 적용 가능한 데이터 분석 방법을 찾아내고 운동 동작 인식 모형을 만들고 싶어 선정하게 되었습니다.

3. 사용데이터

3축 가속도계와 3축 자이로스코프를 활용해 측정된 센서 데이터

1) train_features.csv (1875000, 8)

id 별 600 time 간 동작 데이터

id 3125개 x 600 time =1875000 데이터

	id	time	acc_x	acc_y	acc_z	gy_x	gy_y	gy_z
0	0	0	1.206087	-0.179371	-0.148447	-0.591608	-30.549010	-31.676112
1	0	1	1.287696	-0.198974	-0.182444	0.303100	-39.139103	-24.927216
2	0	2	1.304609	-0.195114	-0.253382	-3.617278	-44.122565	-25.019629
3	0	3	1.293095	-0.230366	-0.215210	2.712986	-53.597843	-27.454013
4	0	4	1.300887	-0.187757	-0.222523	4.286707	-57.906561	-27.961234

2) train_labels.csv (3125, 3)

id 별 동작과 동작 label(61개)

	id	label	label_desc
0	0	37	Shoulder Press (dumbbell)
1	1	26	Non-Exercise
2	2	3	Biceps Curl (band)
3	3	26	Non-Exercise
4	4	26	Non-Exercise

3) test_features.csv (469200, 8)
 id 별 600 time간 동작 데이터
 id 782개 x 600 time =469200 데이터

	id	time	acc_x	acc_y	acc_z	gy_x	gy_y	gy_z
0	3125	0	-0.628100	-0.160155	0.151487	49.665357	88.435961	13.597668
1	3125	1	-0.462548	0.012462	-0.053726	56.953059	96.185341	16.278458
2	3125	2	-0.363481	-0.091789	-0.130004	29.557396	93.836453	13.329043
3	3125	3	-0.351750	-0.239870	-0.193053	23.686172	88.608721	13.449771
4	3125	4	-0.312934	-0.123762	-0.318621	20.410071	85.327707	13.884912

4) sample_submission.csv (782, 62)
 id별 동작을 예측해 작성하는 csv

4. 진행방식

1) 데이터 형태 파악

train_feature.csv data를 train 변수에 저장해서 사용

- 한 개의 id 값에 600 Time이 들어가 있기 때문에 위를 표로 표현
- 예시로 id가 0, 64, 298은 모두 같은 label_desc를 가지고 있음.

```
train_labels[train_labels['id']==0]
```

	id	label	label_desc
0	0	37	Shoulder Press (dumbbell)

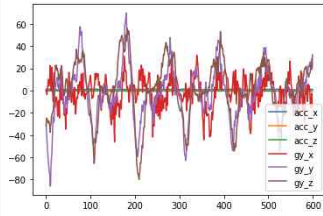
```
# id 별 동작들의 흐름 살펴보기
```

```
# id = 0 이면 (Slider Press(dumbell))
```

```
ex = train[train['id']==0]
```

```
ex.iloc[:,2:].plot()
```

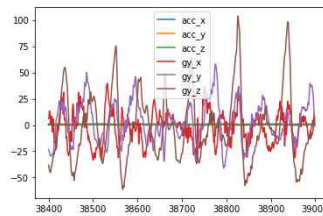
<matplotlib.axes._subplots.AxesSubplot at 0x203652172c8>



```
ex = train[train['id']==64]
```

```
ex.iloc[:,2:].plot()
```

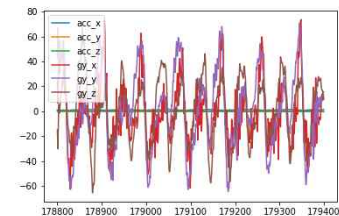
<matplotlib.axes._subplots.AxesSubplot at 0x20365c47b08>



```
ex = train[train['id']==298]
```

```
ex.iloc[:,2:].plot()
```

<matplotlib.axes._subplots.AxesSubplot at 0x20365b20208>



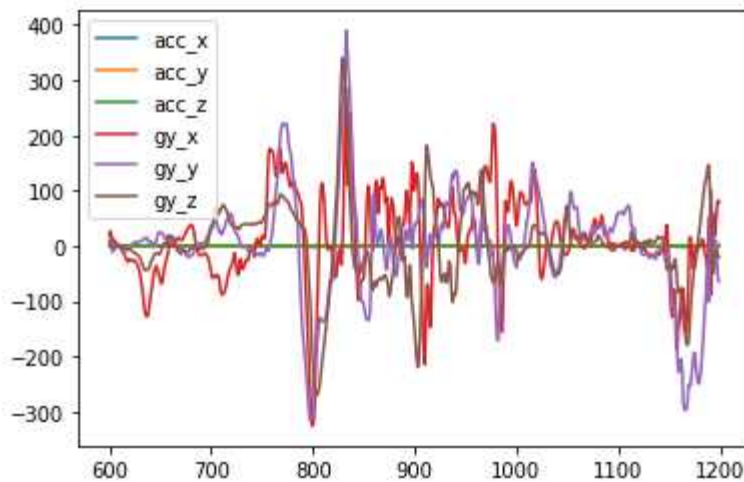
마찬가지로 Non - Exercise 에 관한 600초 그래프이다.

```
#운동이 아니라면,, 대쉬,,
```

```
ex = train[train['id']==1]
```

```
ex.iloc[:,2:].plot()
```

<matplotlib.axes._subplots.AxesSubplot at 0x20365ccbb08>



train에 혹시나 결측값이 있을까봐 살펴보았다.

```
train.isnull().sum() # 클린하노
```

```
id      0
time    0
acc_x    0
acc_y    0
acc_z    0
gy_x     0
gy_y     0
gy_z     0
dtype: int64
```

train_labels.csv를 train_labels 변수에 넣어주고 값을 살펴보았다.

```
train_labels['label_desc'].unique
```

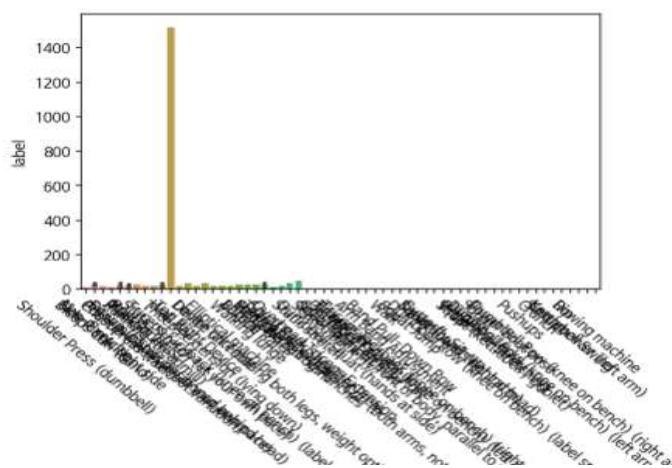
```
<bound method Series.unique of 0          Shoulder Press (dumbbell)
1          Non-Exercise
2          Biceps Curl (band)
3          Non-Exercise
4          Non-Exercise
...
3120        Non-Exercise
3121        Non-Exercise
3122  Dynamic Stretch (at your own pace)
3123        Non-Exercise
3124        Bicep Curl
Name: label_desc, Length: 3125, dtype: object>
```

가독성이 떨어지길래 표로 만들어보았다.

```
In [80]: import matplotlib.pyplot as plt
import seaborn as sns
plt.rcParams['font.family'] = 'NanumGothic'
import matplotlib as mpl
import matplotlib.pyplot as plt
import matplotlib.font_manager as fm
%config InlineBackend.figure_format = 'retina'
import matplotlib.font_manager as fm
fontpath = '/usr/share/fonts/truetype/nanum/NanumBarunGothic.ttf'
font = fm.FontProperties(fname=fontpath, size=9)
plt.rc('font', family='NanumBarunGothic')
```

```
sns.barplot(x = train_labels['label_desc'], y = b.value_counts())
plt.xticks(rotation=-45)
```

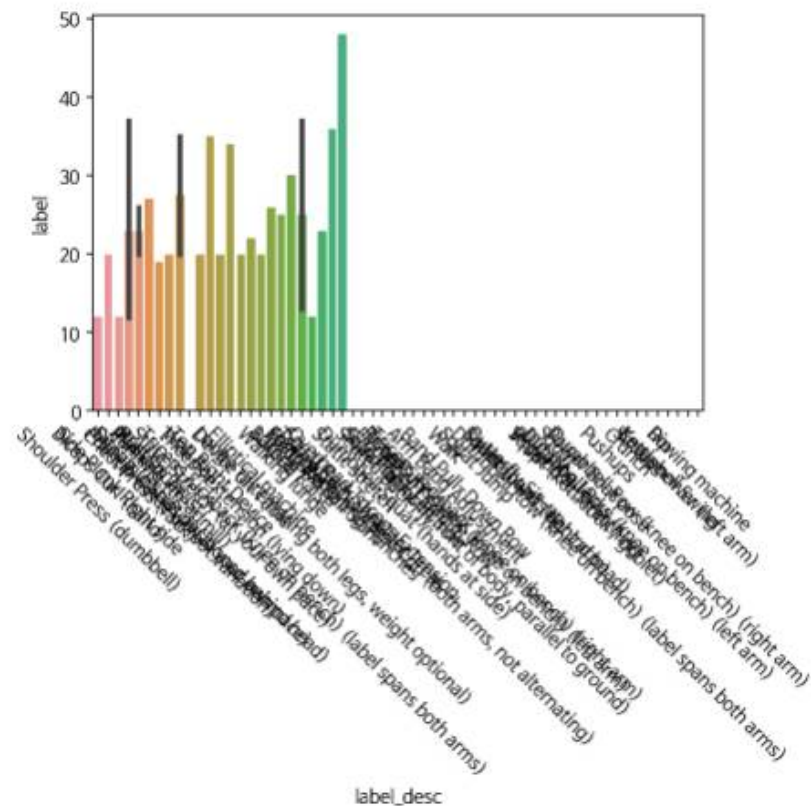
```
Out [80]: (array([ 0,  1,  2,  3,  4,  5,  6,  7,  8,  9, 10, 11, 12, 13, 14, 15, 16,
        17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33,
        34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50,
        51, 52, 53, 54, 55, 56, 57, 58, 59, 60]),
<a list of 61 Text xticklabel objects>)
```



는 더 떨어지는 것 같다. 그래서 가장 많은 값을 차지한 Non-Exercise를 빼고 나머지를 표로 나타내보았다.

```
In [96]: sns.barplot(x = x_1['label_desc'], y = b_1.value_counts())
plt.xticks(rotation=-45)
```

```
Out [96]: (array([ 0,  1,  2,  3,  4,  5,  6,  7,  8,  9, 10, 11, 12, 13, 14, 15, 16,
                  17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33,
                  34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50,
                  51, 52, 53, 54, 55, 56, 57, 58, 59]),
          <a list of 60 Text xticklabel objects>)
```



2) 학습 시킨 독립 변수, 종속 변수 지정

```
X=tf.reshape(np.array(train.iloc[:,2:]),[-1, 600, 6])  
X.shape  
TensorShape([3125, 600, 6])
```

```
import tensorflow  
from tensorflow.keras.utils import to_categorical  
y = to_categorical(train_labels['label'])
```

```
y.shape  
(3125, 61)
```

3) model 학습 시키기

- 분류 모델

4) 정확도 올리기

5. 활용방안