2020 오픈소스전문프로젝트 11조

박수창 교수님

제출일: 2020/06/09

2014040003 윤민호  
2014040017 현승호  
2014040019 하태연  
2018076051 김주은  
2018076052 옥보라

스마트 버스 시스템

2차 데이터 수집 및 분석 결과

1. 1차 데이터 수집
2. 수집한 데이터에 대한 분석 계획
3. 1차 수집 데이터에 대한 분석 결과
4. 피드백 후 새 데이터 분석
5. 파이썬 Keras를 통한 데이터 학습
6. 분석 데이터 이용 계획

**목차**

**1. 1차 데이터 수집**

데이터 수집은 저희조의 어플의 특수성을 고려하여 새로운 데이터를 모으기 위해서는 어플 제작 및 서비스 제공부터 수집이 가능하기에 기존 버스 시스템을 통해 모인 공공데이터를 이용하였습니다. 저희조가 사용 한 공공데이터는 서울시에서 제공하는 서울시의 모든 버스 정류장 및 버스들의 시간대별 승하차 인원에 대한 공공 데이터입니다.

- **공공데이터 파일명:** 2020년\_버스노선별\_정류장별\_시간대별\_승하차\_인원\_정보(04월)

**- 서울특별시 제공 서울시 버스노선별 정류장별 시간대별 승하차 인원 정보 공공데이터 링크:**

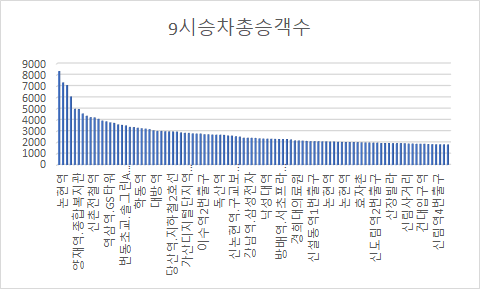
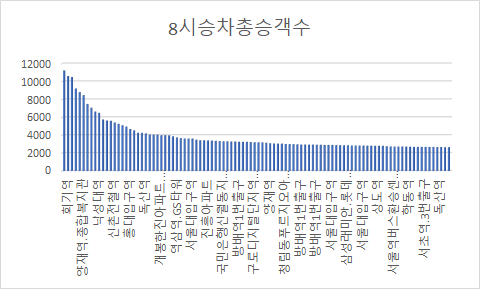
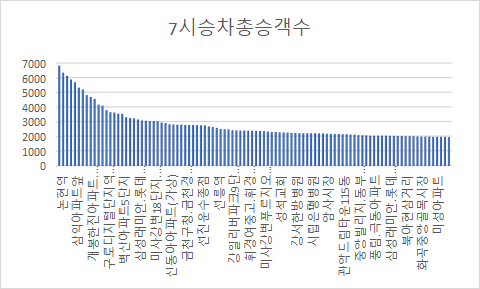
<https://data.seoul.go.kr/dataList/OA-12913/S/1/datasetView.do>

**2. 수집한 데이터에 대한 분석 계획**

데이터에 대한 분석은 크게 두가지로 나누어 집니다. 첫째는 서비스 이용자(대표적으로 버스 이용 시민)에게 맞춤 서비스를 제공하기 위한 데이터 분석이며 두번째는 저희가 만든 시스템을 사용할 버스회사에게 제공될 데이터 분석입니다.

**3. 1차 수집 데이터에 대한 분석 결과**

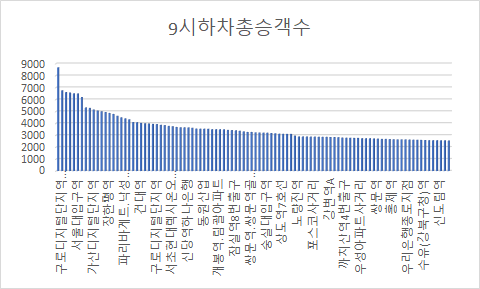
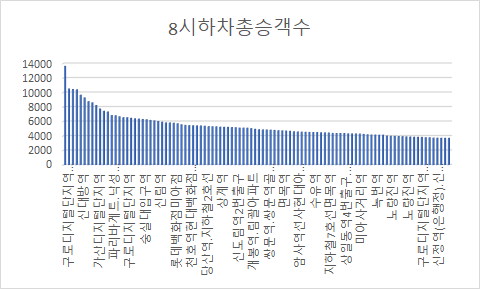
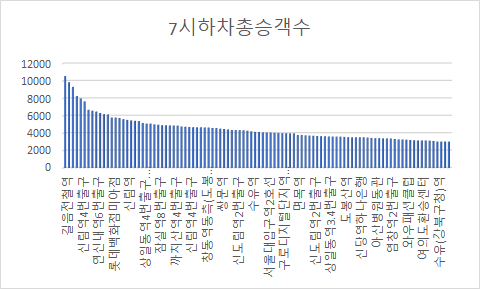
1. 오전 7시 – 오전 9시 (승차)

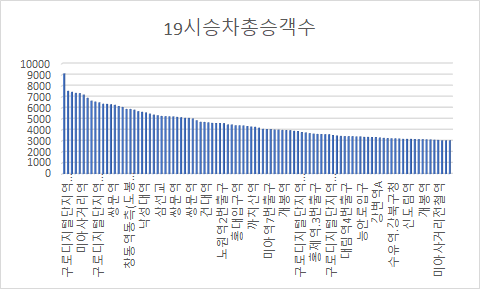
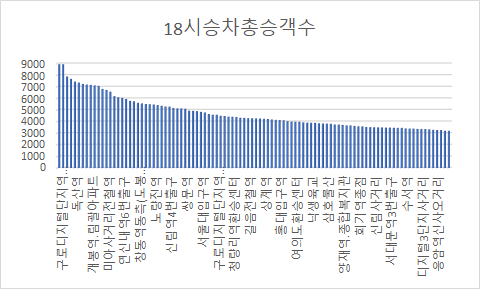


* 1. 직장과 학교로 가기 위해 탑승
     + - 이 때 높은 이용률을 기록하는 정류장들은 사람들의 거주지일 확률이 높음
  2. 활용 방안
     + - 인근 지역에 생활편의시설을 확충하여 주민들의 삶의 질을 향상시킬 수 있다.
       - 주변 사업체와의 제휴를 통해 서비스 이용자에게 부가서비스(할인) 제공

Ex) 마트, 헬스장 등

1. 오전 7시 – 오전 9시 (하차), 오후 6시 – 오후 7시 (승차)

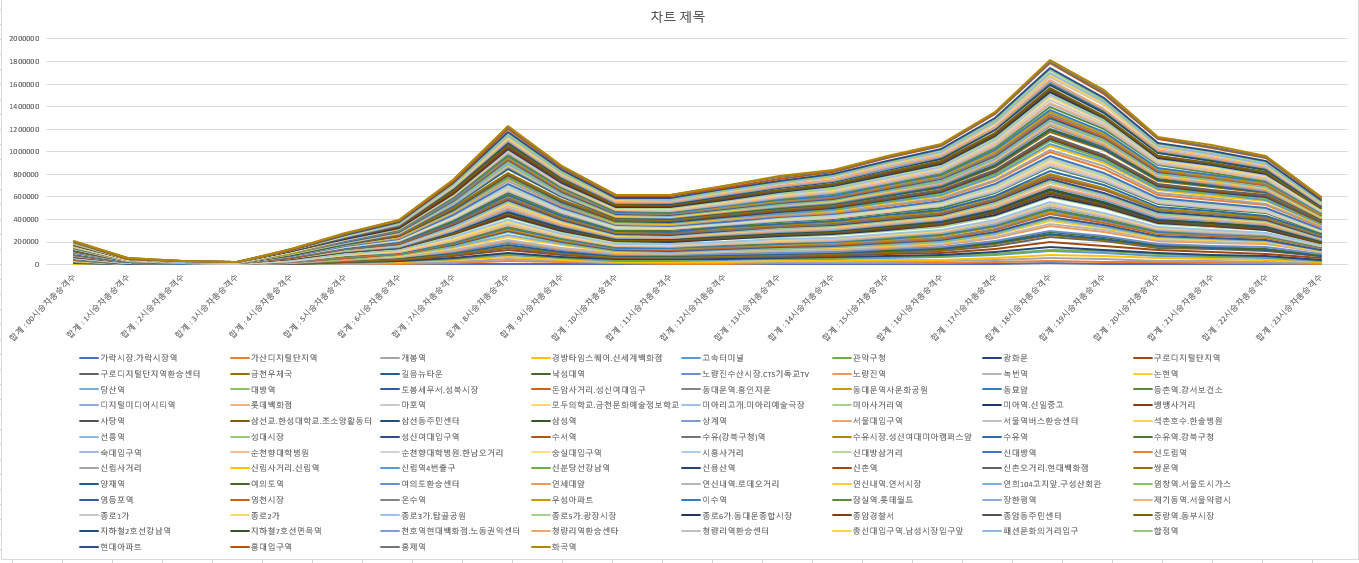




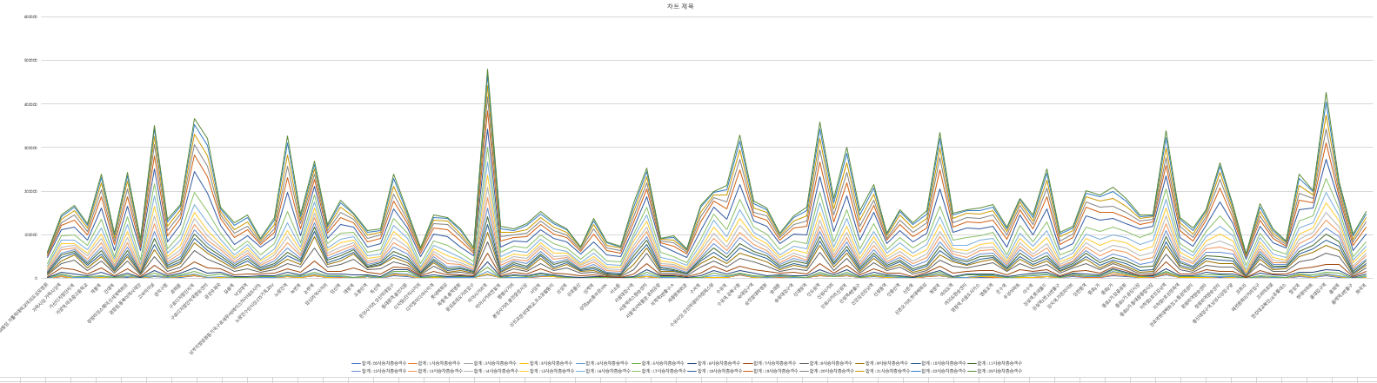
* 1. 하루 중 가장 높은 이용률을 보이는 시간대 (평균 9천 - 1만명)
     + - 유동인구가 많은 지역(사업체 밀집 / 번화가 / 학교)
  2. 활용 방안
     + - 경제적 가치가 높은 지역으로 향후 사업체 투자 유치나 상권 부흥에 활용
       - 주변 사업체와의 제휴를 통해 서비스 이용자에게 부가서비스(할인) 제공

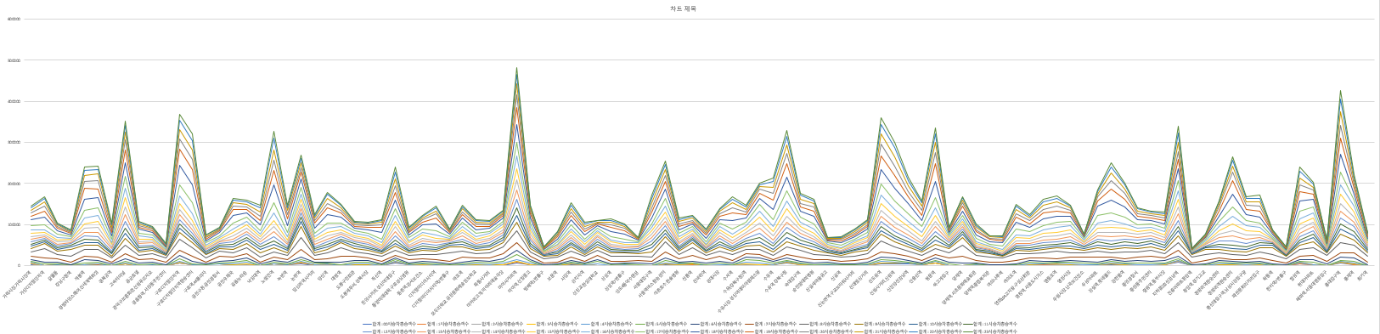
Ex) 식당, 의류판매점, 카페 등

1. 승하차율이 높은 지역
   * + - 노선을 증설하여 교통혼잡을 줄이는 효과
2. 승하차율이 저조한 지역
   * + - 거주 인구 또는 유동인구가 적은 지역으로 이용률에 비해 많은 노선이 배치된 경우 배차간격을 늘리고 노선 수를 줄여 비용을 절감하는 효과



1. 각 정류장들의 시간대 별 승객수를 분석
   1. 8시와 18시에 승객수가 가장 많음
   2. 출퇴근, 등하교 시간으로 유동 인구가 가장 많음





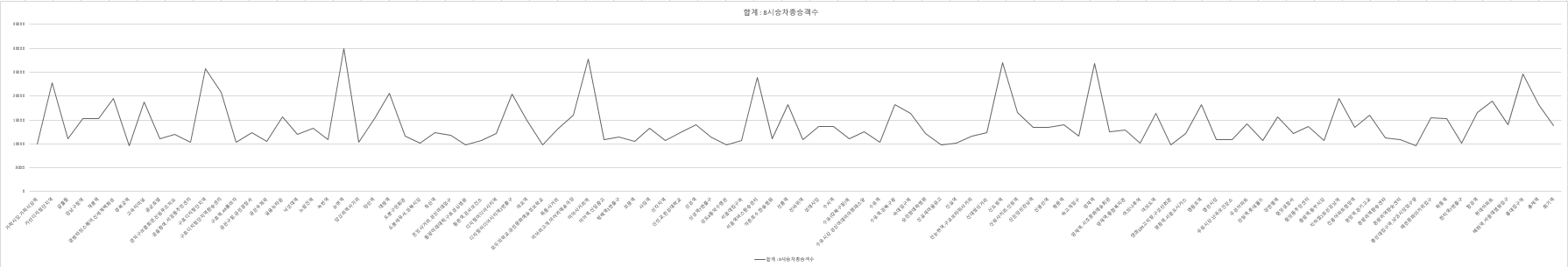
1. 8시와 18시의 정류장별 승객수를 다시 분석
2. 평균적으로 승객수가 많은 곳과 적은 곳을 선별
   1. 주로 \_\_\_\_역이 수치가 높음
      * + 지하철역 주변으로 유동인구가 많고 상가가 밀집해 있음

=> 주변 상가들의 광고창(맛집 추천 서비스) 운영 및 할인 쿠폰 제공 등의 서비스를 지원할 수 있음

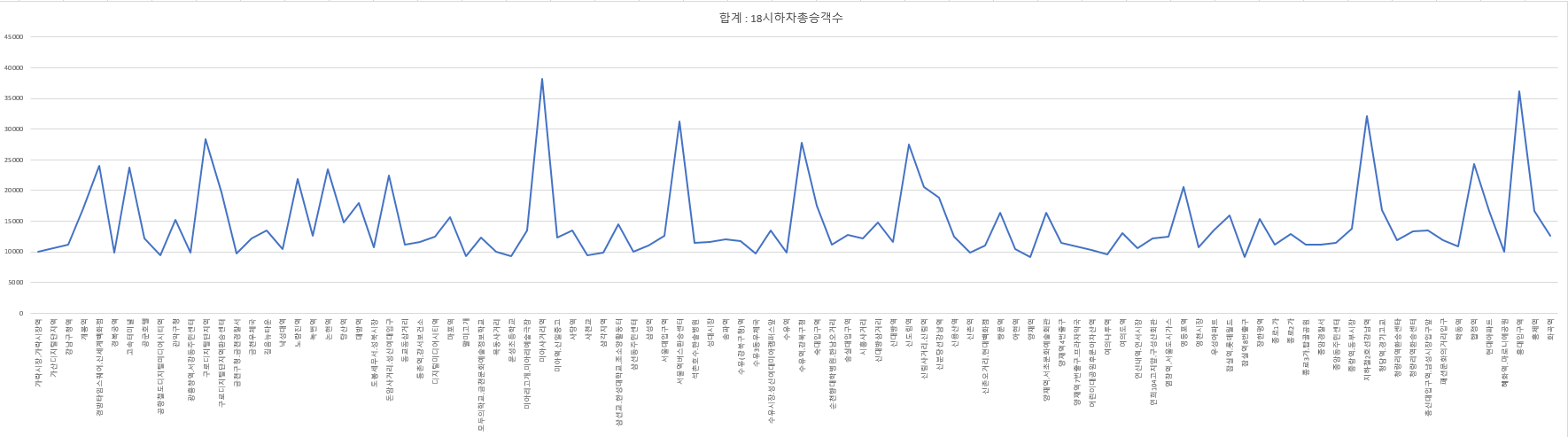
* + - * 환승이 편리함

=> 환승역까지의 길찾기를 지원할 수 있음

* 1. 추후에 노선 변경, 버스 수 조정에 필요한 데이터 사용 가능



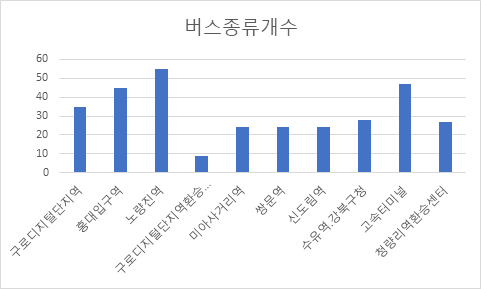
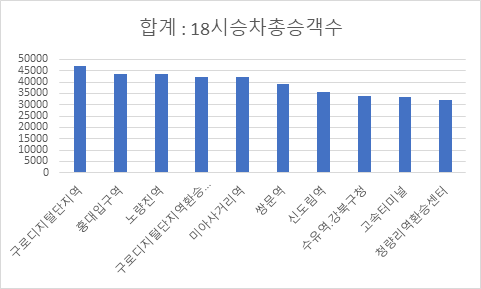
-8시 승차인원이 상위100인 버스정류장-



-18시 하차인원이 상위100인 버스정류장-

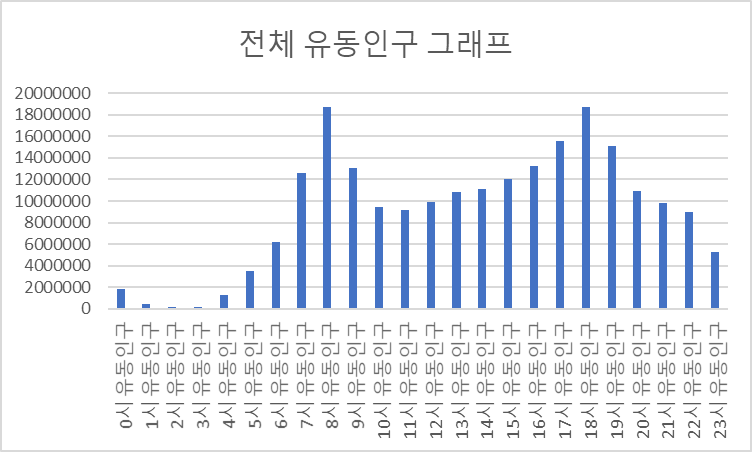
위의 그래프를 기준으로 분석한 결과

1. 8시 승차그래프의 값과 18시 하차 그래프의 값이 비슷한 경우 해당 정류장은 주거지역 근처로 분석 할 수 있다.
2. 또한 8시의 승차그래프의 인원보다 18시 하차그래프의 인원이 더 높은 버스 정류장은 상권이 발달한 지역으로 분석할 수 있다.
   1. 위의 분석한 결과들을 토대로 주거지역과 상권이 발달한 지역을 산정할 수 있는데 이 데이터들을 가지고 주거지역 근처의 편의시설들과 상권이 발달한 지역 근처의 음식점들과 제휴를 맺어 버스를 자주 이용하는 승객들에 한해 각 제휴 업체들에 대한 할인 쿠폰 등의 서비스를 제공할 수 있다.
3. 시내 버스 운행에 있어서 사람이 많이 탑승하는 시간에 버스가 상대적으로 적게오면 삶의 질이 떨어지므로 가장 유동인구가 많은 오후 6시의 정류장 승차 인구와 각 정류장으로 오는 버스를 조사함.



다음 분석은 위의 그래프를 토대로 분석한 내용이다.

1. 18시 승차
   1. 구로디지털단지역 47219명 승차 버스종류 35개
   2. 홍대입구역 43587명 승차 버스종류 45개
   3. 노량진역 43450명 승차 버스 종류 55개
   4. **구로디지털단지역환승센터 42425명 승차 버스종류 9개**
   5. 미아사거리역 42388명 승차 버스종류 24개
   6. 쌍문역 39237명 승차 버스종류 24개
   7. 신도림역 35436명 승차 버스종류 24개
   8. 수유역.강북구청 33671명 승차 버스종류 28개
   9. 고속터미널 333232명 승차 버스종류 47개
   10. 청량리역환승센터 32112명 승차 버스종류 27개
2. 유동인구가 가장 많은 오후 6시에 대체적으로 27개~55개의 종류의 버스가 오는데 구로디지털단지역환승센터만 오직 버스 종류가 9개고 승차인구는 42425명으로 많다. 실조사 후에 운행 경로 수정이 필요한지 확인해야 좋을 것 같다.



1. 위의 그래프를 보면 하루 유동인구 중 8시와 18시가 제일 많다. 그 중에서도 앞뒤 시간과 차이가 많이 나는 8시를 기준으로 분석을 우선 진행하고자 한다.
2. 8시 기준 유동인구 제일 많은 5개의 정거장의 유동인구수와 배치 버스 수
   1. 구로디지털단지역 85434명 35개
   2. 신도림역 66803명 51개
   3. 노량진역 65731명 55개
   4. 구로디지털단지역환승센터 65485명 9개
   5. 미아사거리역 59913명 52개
3. 구로디지털단지환승센터 정류장의 경우 유동인구는 많으나 그에 비해 버스의 수가 부족하므로 해당하는 노선의 해당 시간대 버스 배차 간격을 좁게 하거나 다양한 버스의 배치가 필요하다 분석.
4. 또한 구로디지털단지역 정류장의 경우도 다른 정류장에 비해 월등히 높은 유동인구 수치를 보이나 상대적으로 버스의 수가 부족하므로 해당 노선의 해당 시간대 버스의 배차 간격을 좁게 하거나 추가적인 버스 노선의 배치가 필요하다 분석.

**4. 피드백 후 새 데이터 분석**

피드백 후 주요적으로 사용할 데이터를 버스 승하차 데이터가 아닌 스마트폰의 가속도 센서를 이용하여 수집한 데이터로 변경하였다. 이는 사용자가 버스를 예약 하였을 때 제시간에 버스 정류장까지 도착 할 수 있는가에 대한 분석을 하기 위함이다. 사용자의 상태를 크게 3가지 (걷기, 뛰기, 정지)로 나누었으며 이에 대한 데이터를 다양한 방식으로 수집하여 분석을 진행하였다.

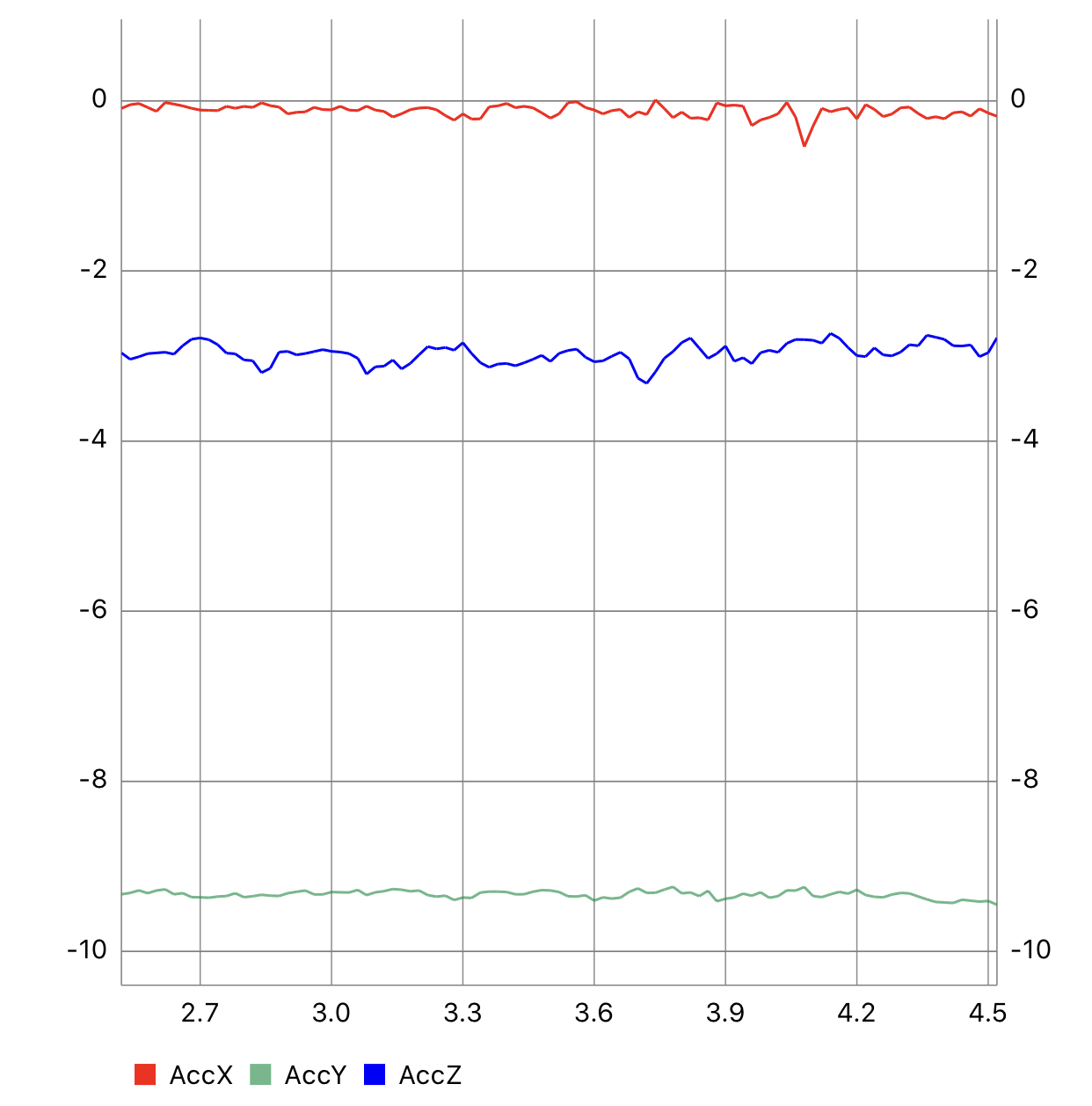
**- 센서 데이터 수집 어플을 이용한 분석**

|  |  |
| --- | --- |
| 걷다가 뛴 상태 | 뛰다가 걸은 상태 |
|  |  |
| 걷다가 멈춘 상태 | 뛰다가 멈춘 상태 |
|  |  |

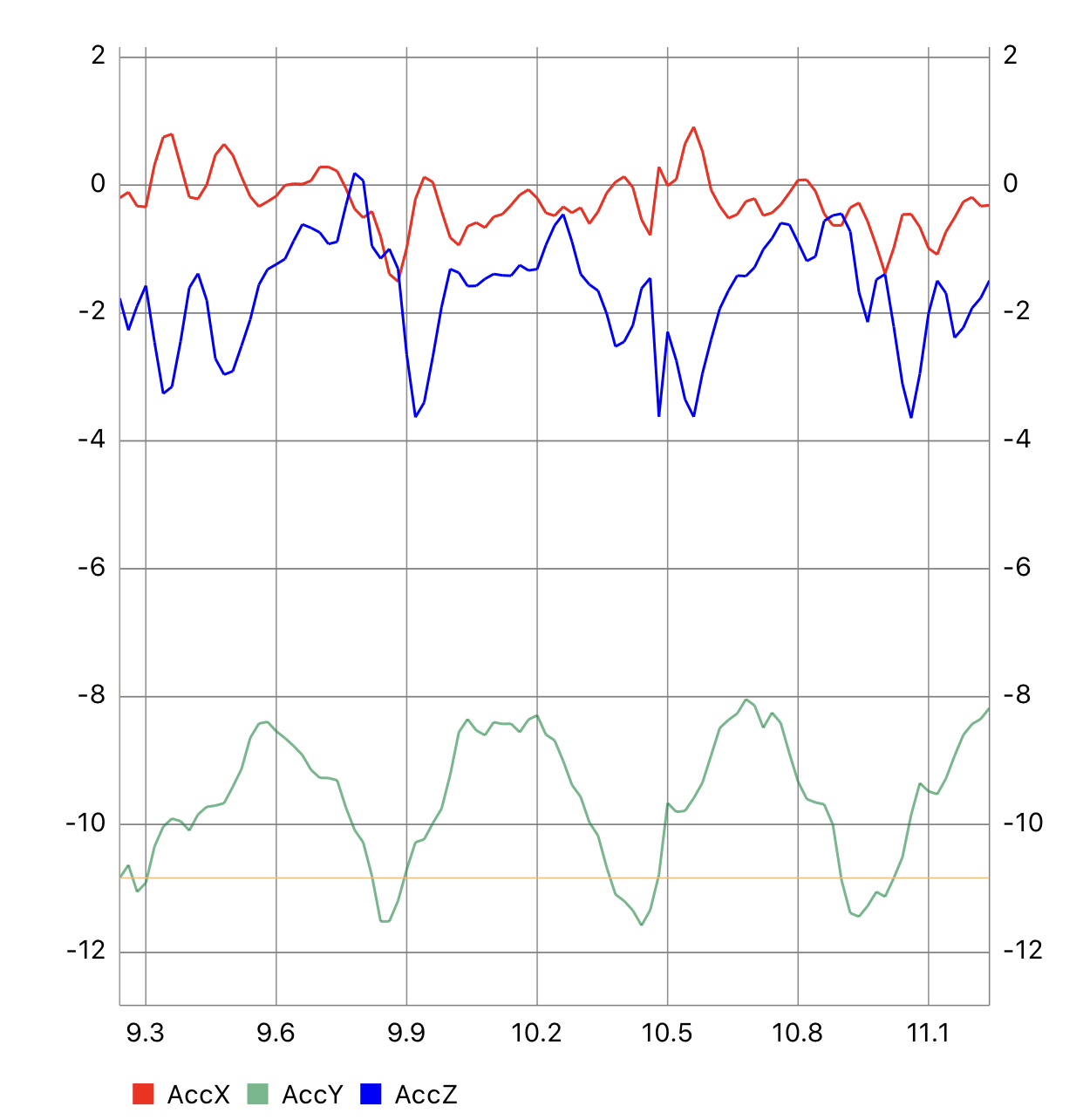
위의 네 가지 상태는 휴대폰 어플에 있는 가속도 센서를 통해 직접 실험한 사진들이다. 위의 사진들을 보면 뛴 상태, 걸은 상태, 멈춘 상태 세가지의 상태를 알아볼 수 있는데 가장 폭이 큰 경우가 뛴 상태고 폭이 중간 정도인 것이 걸은 상태 1자에 가까운 폭을 가진 상태가 멈춘 상태로 분석할 수 있다. 이와 같은 가속도의 데이터를 인공지능에게 학습시킴으로써 인공지능은 휴대폰을 들고 있는 사용자의 상태를 뛰는 상태, 걷는 상태, 멈춘 상태의 총 3가지 상태를 판단 할 수 있게 된다.

**- 가속도 센서의 증감 유형에 따른 분석**

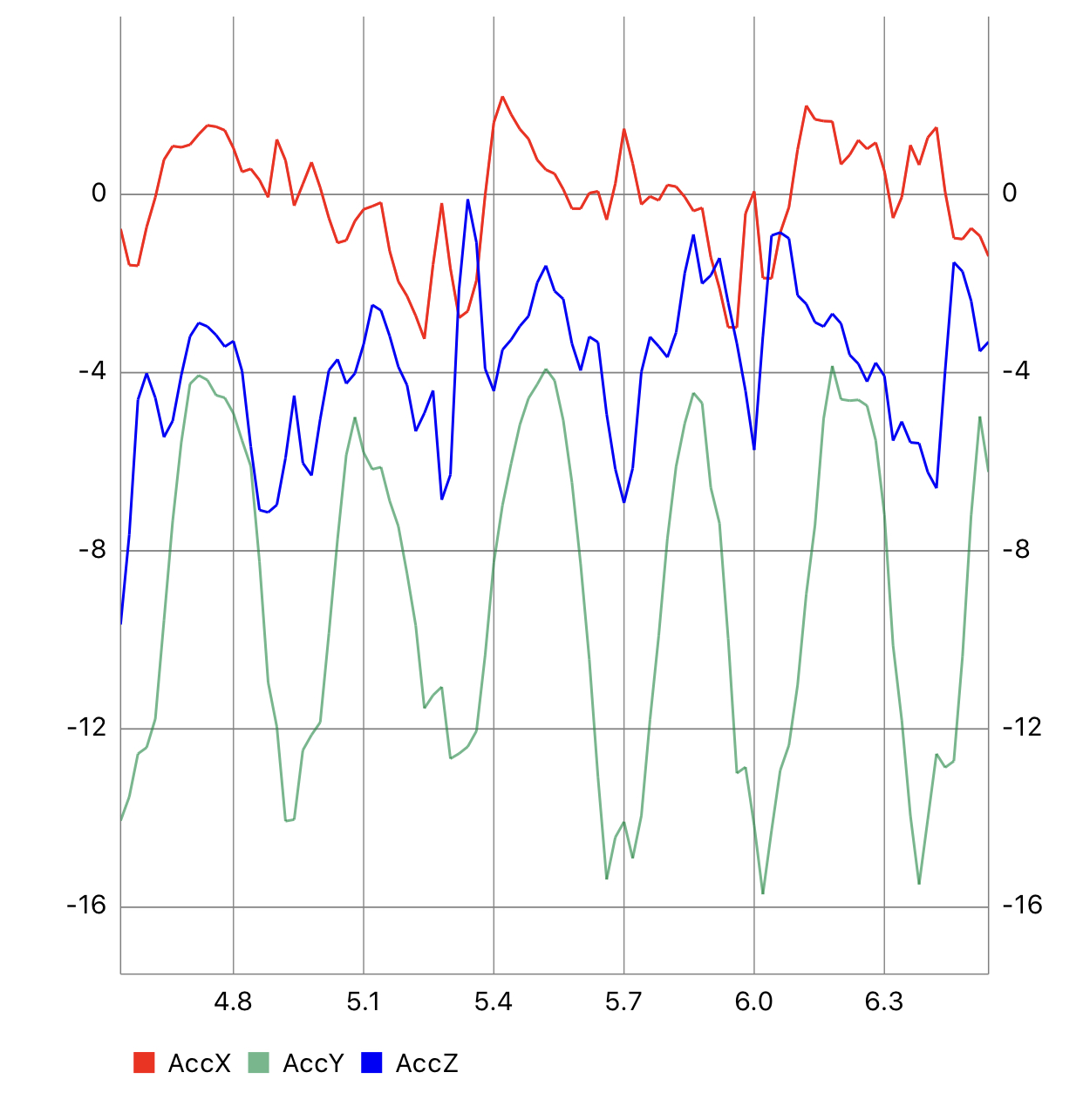
**멈춰있을 때**



**걸을 때**



**뛸 때**



정지와 걷기, 달리기에 따라 점점 가속도의 x, y, z 성분 값의 변동 폭이 커지는 양상이다.

세가지 상태의 그래프 개형을 분석했을 때 AccX와 AccY는 움직임이 격할수록 평균값이 기존보다 증가하는 경향을 보이고 AccZ는 같은 상황에서 평균값이 기존보다 감소하는 경향을 보인다.

가속도 센서는 자이로 센서가 측정한 변위로부터 가속도 성분들을 계산하는 식으로 동작하기 때문에 결과적으로 스마트폰의 가속도 센서만 활용해도 충분히 사용자의 현재 상태를 파악할 수 있을것으로 보인다. 따라서 가속도 성분의 증감 유형에 따라 사용자의 이동 상태를 판단할 수 있다.

**5. 파이썬의 Keras를 통한 데이터의 학습**

캐글을 통해 걷고 뛰는 것에 대한 가속도센서 x, y ,z 와 자이로 센서의 x, y, z의 값의 데이터를 수집했다. 이 데이터에서 걷는 것은 “0”, 뛰는 것은 “1”의 값을 가진다. 이 데이터들을 파이썬의 Keras 패키지를 통해 신경망을 구현해 학습을 진행하였다.

밑의 소스는 Sequential 모델을 사용해 레이어를 구성하고 학습을 진행한 소스이다.

-소스-

import numpy as np

import pandas as pd

import os

import numpy as np

import pandas as pd

from keras.layers import Dense, Input, BatchNormalization, Dropout

from keras.models import Sequential, Model

data = pd.read\_csv('./input/dataset.csv')

x\_train = data[['wrist', 'acceleration\_x', 'acceleration\_y', 'acceleration\_z', 'gyro\_x', 'gyro\_y', 'gyro\_z']]

y\_train = data[['activity']]

x\_train = np.array(x\_train)

y\_train = np.array(y\_train)

model = Sequential()

model.add(Dense(250, activation = 'relu', input\_shape = (7,)))

model.add(BatchNormalization())

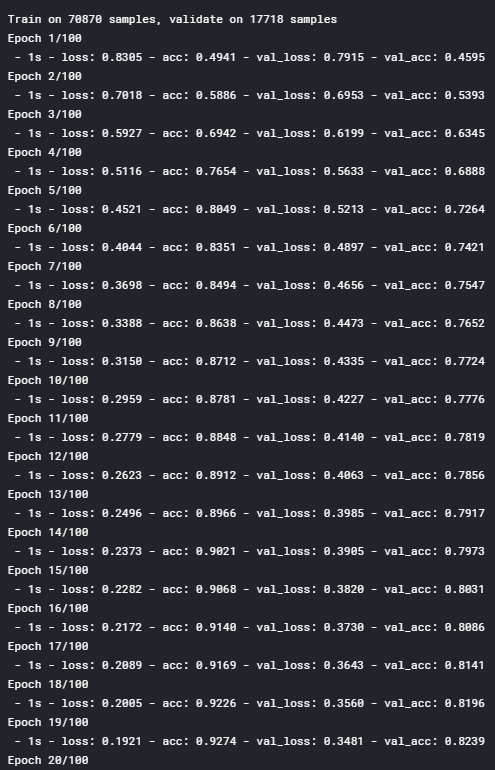
model.add(Dropout(rate = 0.25))

model.add(Dense(1, activation = 'sigmoid'))

model.compile(loss = 'binary\_crossentropy', optimizer = 'adam', metrics = ['accuracy'])

model.fit(x\_train, y\_train, batch\_size = 40000, epochs = 100, validation\_split = 0.2, verbose = 2)

마지막 fit 메소드를 통해 model에 데이터셋을 통해 학습을 진행했다. 아래 사진을 학습한 사진이다.



위의 사진은 학습하는 것을 나타낸 사진의 일부로 총 70870개의 데이터를 학습하고 17718개의 샘플을 검증한 것을 의미한다.

loss값은 손실 값으로 결과값과의 차이를 의미하며 0에 가까울수록 좋은 모델이다.

acc는 정확도로 1에 가까울수록 좋은 모델이다.

그 후 test case인 x\_test를 생성한 후 precdict 메소드를 통해 출력된 y값을 각 클래스에 대한 판별함수로 가정하고 predict\_classes 메소드로 분류를 진행한 소스이다.

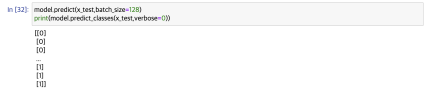
data1 = pd.read\_csv('./input/dataset1.csv')

x\_test = data1[['wrist', 'acceleration\_x', 'acceleration\_y', 'acceleration\_z', 'gyro\_x', 'gyro\_y', 'gyro\_z']]

x\_test = np.array(x\_test)

model.predict(x\_test,batch\_size=128)

print(model.predict\_classes(x\_test,verbose=0))



결과는 다음과 같이 나왔고 예측한 결과 테스트 데이터 셋의 처음에는 걷는 값인 0을 예측했고 후반에는 뛰는 값인 1을 예측했다.

**6. 분석 데이터 이용 계획**

위와 같은 데이터들을 인공지능은 학습함으로 사용자의 상태에 대한 정확한 판별을 하고 이를 사용자가 도착지까지 걸리는 시간을 계산할 때에 실시간으로 계산에 반영시켜 사용자가 얼마나 걷고 얼마나 뛰어야 하는지, 다른곳을 들럿다 가도 시간이 충분히 여유로운지 등 도착 가능성에 대한 판별을 가능토록 할 것이다.