(2주차) 데이터 확인

2023.05.01 ~ 05.07

- [x] 5.2 웹페이지 레이아웃 초안
- [x] 5.4. Flask - Spring Boot 데이터 전송 코드 변경
- [x] 5.4. 데이터 확인, 차량별 GPS 데이터 시각화

💬 한상곤교수님 데이터 관련 팁!

- 예측 결과를 반영할 칼럼 만든다고 생각해야함 (원활, 서행, 정체)

(단위: km/h)

원활	서행	정체	정보없음
25 이상	15 ~ 25	15 미만	-

* 서울시 교통정보 참조

- 데이터는 분석과정에서 줄여가면서 CSV 파일 많이 만들기
- 최종 데이터베이스에 넣을 버전 생각하기!

💬 웹페이지 출력할 것들

- 도로 교통 상황
  - 카카오 지도 API 사용
  - 리액트에서 스크립트로 바로 추가하기
- 일자별 속도 (달력 형식)
  - 년 - 월 - 일 - 시간 - 속도데이터 DB에 입력
- 시간대별 주요 노선유형별 통행속도 예측, 주간 버스 통행속도 예보
  - 데이터 분석 후 예측 모델링에 따른 결과

Untitled Edited 16 days ago

데이터 확인 1차 (5/2)

- 차량 4대분 2022년 1달 간 1초간격 속도데이터, …
  - 자동차 운행기록 및 장치에 관한 관리지침
  - 파일명 : [별표 3]
  - 내용 : [별표 2]
  - 양식 : txt파일
- 하나로 어떻게 합치지…?
  - => 5/2 한상곤교수님한테 도움 요청 해놓음..

Flask - Spring Boot 데이터 전송 코드 변경

- 기존
  - Flask : return {{}, {}}
  - SpringBoot에서 String으로 받음
  - DB에 객체로 저장하려고 하니 parsing 노가다…?
- 변경
  - Flask : return [{}, {}]
    - entity column 개수, 이름 안맞으면 데이터 안들어가서 깡통seq 만들어서 넣어줌!
  - SpringBoot에서 `ObjectMapper` 사용하여 Json으로 받음

```
ObjectMapper objMapper = new ObjectMapper();
List<Test2> testList = null;

try{
  ...

  BufferedReader br = new BufferedReader(new InputStreamReader(urlConnection.getInputStream(), Standard

  testList = objMapper.readValue(br.readLine(), new TypeReference<>() {});
  test2Repo.saveAll(testList);

   ...
}
```

    - Database에 저장

데이터 확인 2차 (5/4)

- 차량번호별(총 4대), 날짜시간별(2022년12월 한달간, 초단위) 분리되어 있는 txt파일을 csv로 변환
  - rows 약 500만건……
  - S, C 데이터 파일 중 속도데이터만 사용 할 예정

```
from glob import glob
files = glob("./data/**/**/*")

```

```
for file in files:
  ...
  #파일 읽기
  temp_df = pd.read_fwf(file, encoding ="cp949", skiprows=1, header=None)
  ...
  #csv 내보내기
  df.to_csv("./data/raw.csv", index=False)
```

- 66자리 숫자 칼럼별 분리 (칼럼정보 : 자동차 운행기록 및 장치에 관한 관리지침, 별표2)

항 목		자릿수	표기방법	표기시기
주행거리 (km)	일일주행거리	4	00시부터 24시까지 주행한 거리 (범위: 0000~9999)	실시간
	누적주행거리	7	최초등록일로부터 누적한 거리 (범위: 0000000~9999999)	"
정보발생 일시		14	YYMMDDhhmmssss (연/월/일/시/분/0.01초)	"
차량속도(km/h)		3	범위: 000~255	"
분당 엔진회전수(RPM)		4	범위: 0000~9999	"
브레이크 신호		1	범위: 0(off) 또는 1(on)	"
차량위치 (GPS X, Y 좌표)	X	9	10진수로 표기 (예: 127.123456*1000000⇒127123456)	"
	Y	9		
위성항법 장치(GPS) 방위각		3	범위: 0~360 (0~360°에서 1°를 1로 표현)	"
가속도 (m/sec2)	ΔVx	6	범위: -100.0~+100.0	"
	ΔVy	6		

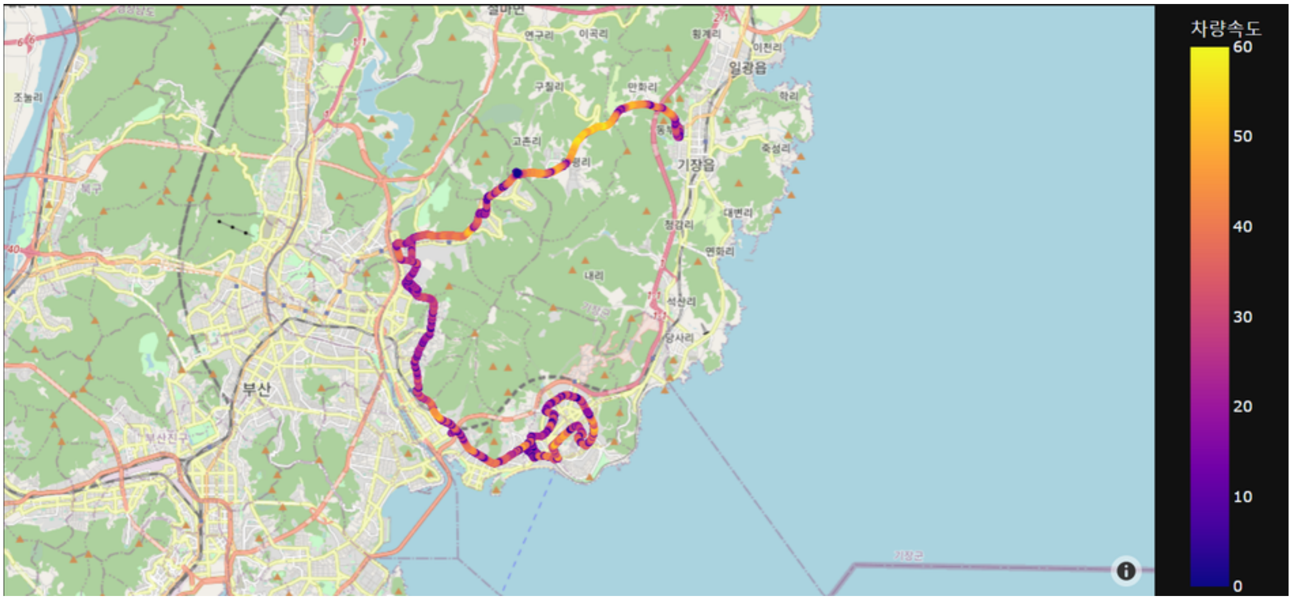
- 데이터 분석 방향 잡기 (시계열) GPS_X, GPS_Y : 시간대별
  - 먼저 버스1대, 하루치에 대해서 시간대별 GPS_X, GPS_Y 좌표 찍어서 시각화
  - 운행방향, 회차여부, 운행시간범주 정하기
  - 버스 4대로 확대 시각화
  - 한달치 분석
- 데이터 차량별로 시각화
  - import plotly.express as px → px.scatter.mapbox()
  - 1854, 1860, 1893, 1894 모두 같은 노선

```
df = pd.read_csv('./dataset/slice_data_1894.csv')

gps_x = np.array(df['GPS_X']) / 1000000
gps_y = np.array(df['GPS_Y']) / 1000000
gps_x = gps_x.tolist()
gps_y = gps_y.tolist()
df['gps_x'] = gps_x
df['gps_y'] = gps_y

fig = px.scatter_mapbox(df, lat='gps_y', lon='gps_x', size_max=10,
                        zoom=11, center=dict(lat=df["gps_y"].mean(), lon=df["gps_x"].mean()),
                        mapbox_style="open-street-map", template="plotly_dark", color='차량속도')
fig.update_layout(title_x=0.5, height=600)
```

- 차량속도에 따라 색 다르게 하여 시각화


차량속도
60
50
40
30
20
10
0
부산
기장읍
일광읍
철마면
연구리
이곡리
구칠리
고촌리
만화리
횡계리
이천리
학리
죽성리
대변리
청강리
연화리
내리
석산리
당사리
조놀리
북구
부산진구