2022 UOS
빅데이터
알고리즘
경진대회

TEAM : PORTION

TEAM : Portion

- Portion-jack : 김정우 : 시립대 물리학과, 통계학과(복수전공)
- BlackSkirts : 송치호 : 광운대 전자바이오물리학과
- 닉넴닉 : 안동현 : 계명대 의과대학

목차

데이터 관련하여

모델링과 관련하여

결론

데이터와 관련하여

- 외부 데이터를 추가적으로 사용할 수 있다.

  하지만 2022년도 예측을 위한 외부 데이터는 사용할 수 없다.

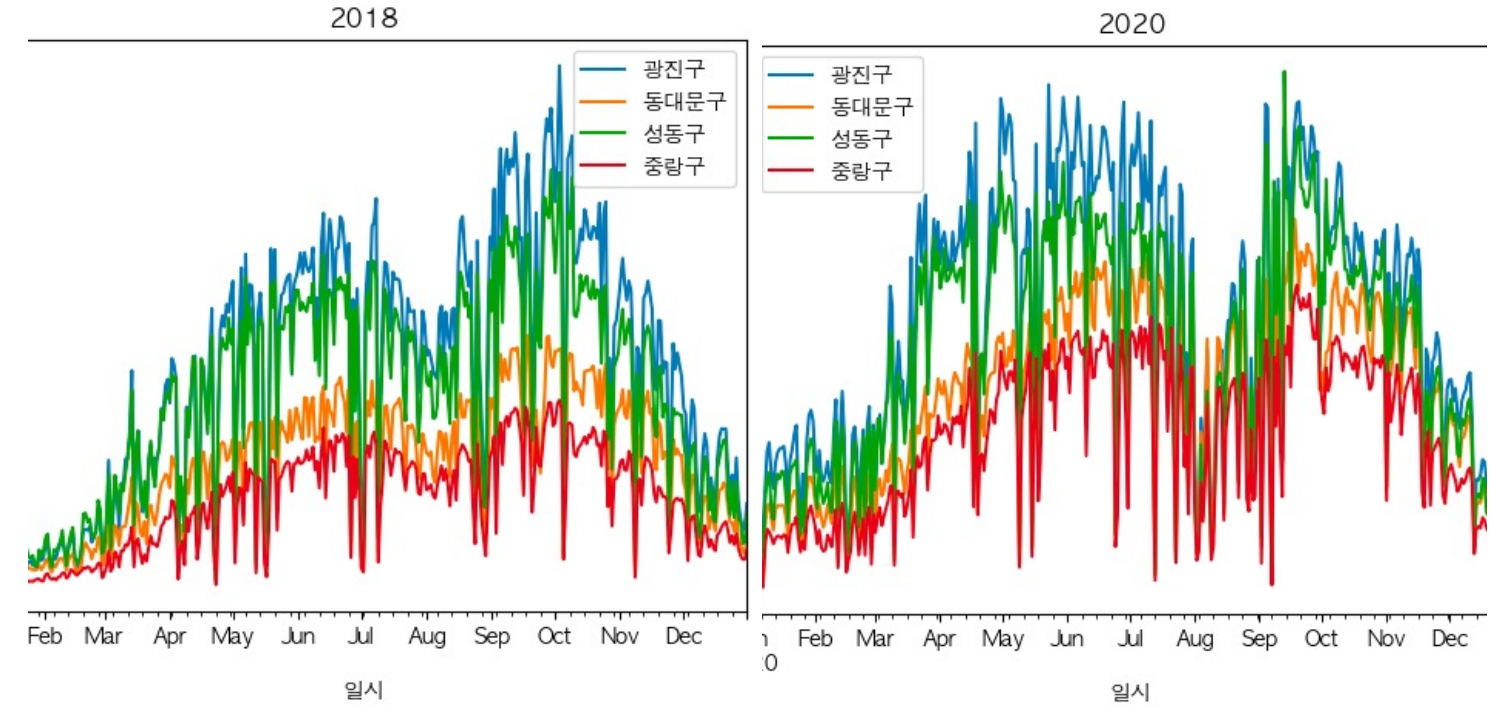
- 기상데이터를 통한 비온날의 데이터를 제거하여 smoothing?

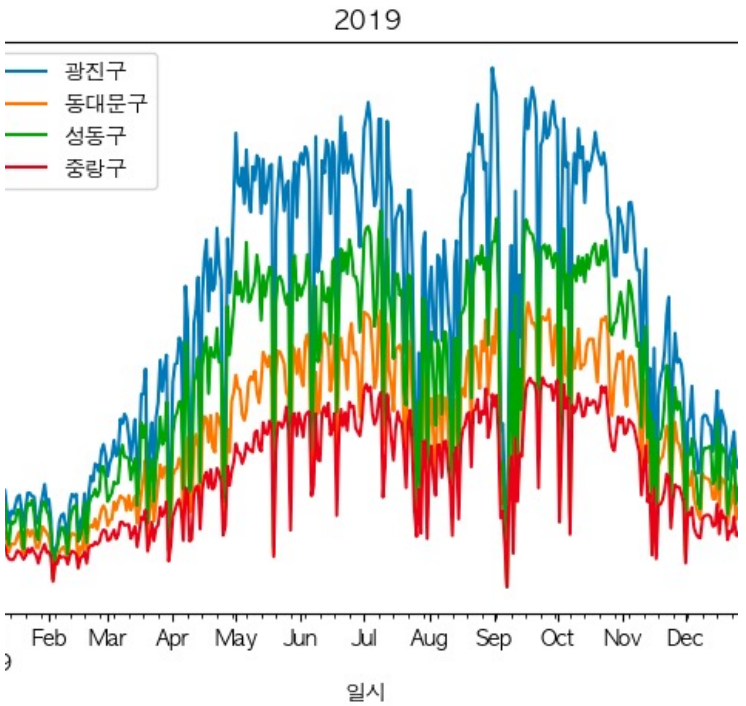
  굳이 외부데이터를 사용하지 않고 수치적으로 처리하자.

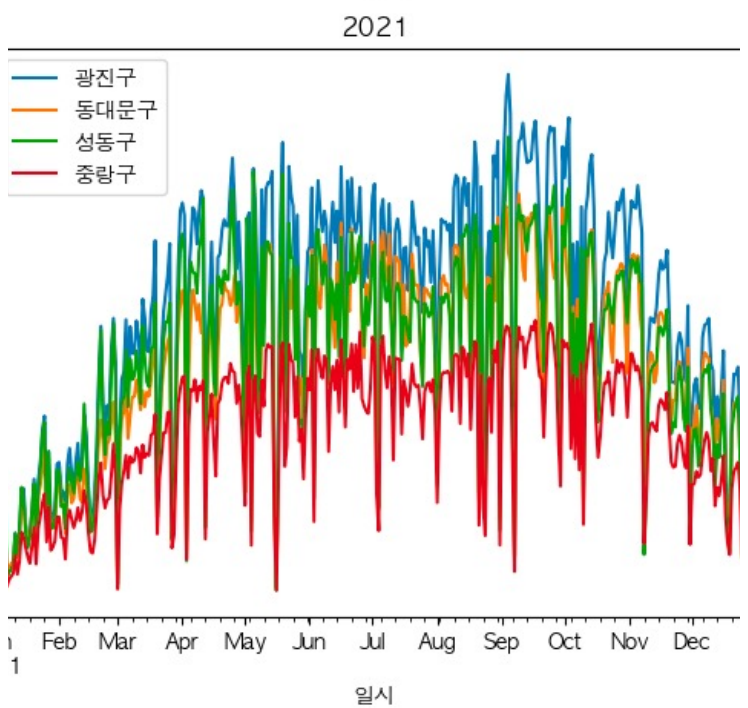
데이터와 관련하여
- 패턴찾기_v1

=>모델이 비를 예측하기 시작한다면

정확도가 상당히 떨어질것이다.

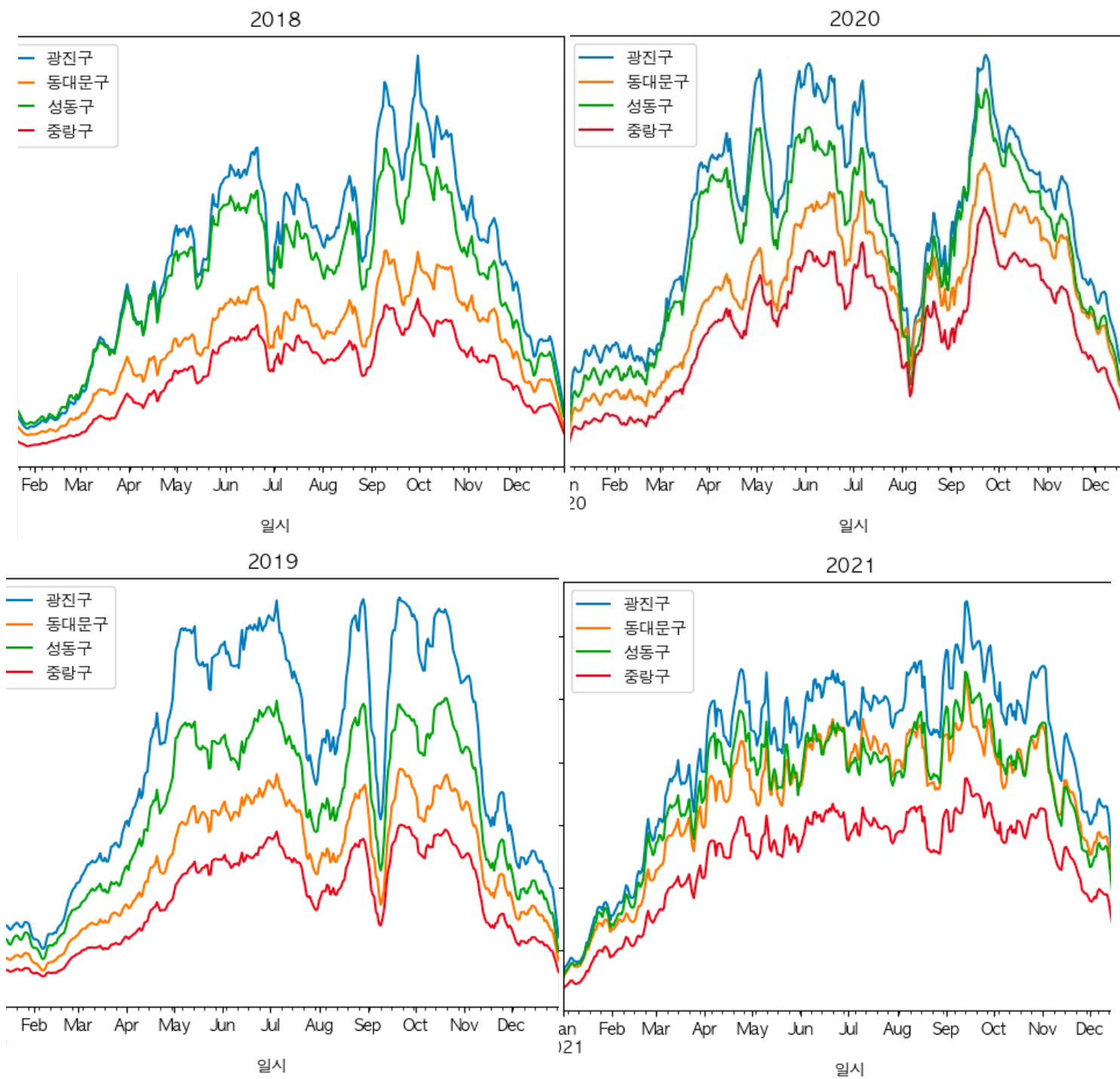






데이터와 관련하여
- 패턴찾기_v2

=> 장마를 예측할 것인가?



모델링

- 시계열 예측?

  2018 ~ 2021**의 데이터로 각 년도의 날짜별로 시계열 예측을 하는 방식**

  => 4**개의** data**로 하나의** target**을 예측하는 것**

  => **유의미한 결과를 만들 수 없을것으로 생각된다**.

=> **시간 데이터에 따른** groupby**를 통한 패턴 찾기**

모델링_1
날짜에서 필요한 정보 도출

- 각 날짜가 일년중 몇 주차인지 추출
- 각 날짜가 주말인지 평일인지 추출

  추가적으로 년도별 공휴일 주말로 선택

- 주차별 (평일,주말) 에 따른 따릉이 예측량의 평균값을 구하자

모델링_2
주차별 (주말/평일) 예측량의 최종 평균 구하기

년도마다 따릉이 대여량의 scale이 꽤 차이가 나기 때문에 일년간의 대여량 추세를 보기 위해선 scaling이 필요하다.

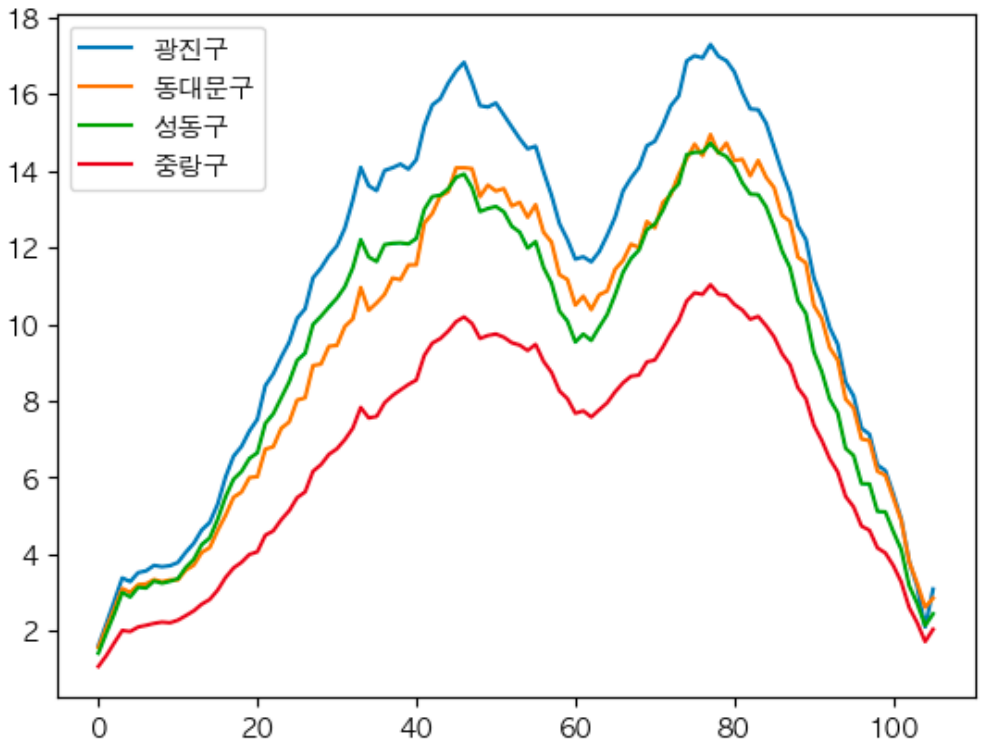
```
# 각 연도별 minmaxscaling 진행
## => 모든 연도별 주간 평균을 값을 baseline으로 2022년을 예측하기 위해
from sklearn.preprocessing import MinMaxScaler
mms_2018 = MinMaxScaler()
mms_2019 = MinMaxScaler()
mms_2020 = MinMaxScaler()
mms_2021 = MinMaxScaler()

# 각 연도별 minmaxscaling 진행
year_df_list=[df_2018,df_2019,df_2020,df_2021]
scaler_list=[mms_2018,mms_2019,mms_2020,mms_2021]
for df,mms in zip(year_df_list,scaler_list):
    df.iloc[:,2:] = mms.fit_transform(df.iloc[:,2:])
```

모델링_3
주차별 (주말/평일) 대여량에 대한 연도별 평균값

```
# 연도별 대여량을 스케일링 한후 주별,(평일,주말별) 대여량 평균구하기
week_mean=pd.concat(year_df_list).\
    groupby(['week','weekend']).mean().reset_index()
```

모델링_4
이후 이를 년도별 스케일러를 활용한 inverse_transform

- 그리고 이 데이터를 바탕으로 년도별 차이를 구하여 2022 따릉이 대여량을 주차별 (주말/평일) 로 예측한다.



모델링_4
추가설명자료

```
# 이제 minmaxscaling된 자료를 바탕으로
# 각 년도에 맞게 inverse_transform 진행 => 2018년도 스케일의 (주간평균), 2019년도 스케일의 (주간평균),2020년도 스케일의 (주간평균),2021년도 스케일의 (주간평균)
df_2018_weekmean=pd.DataFrame(mms_2018.inverse_transform(week_mean.iloc[:,2:]),columns=week_mean.iloc[:,2:].columns)
df_2018_weekmean['week'] = week_mean['week']
df_2018_weekmean['weekend'] = week_mean['weekend']
df_2019_weekmean=pd.DataFrame(mms_2019.inverse_transform(week_mean.iloc[:,2:]),columns=week_mean.iloc[:,2:].columns)
df_2019_weekmean['week'] = week_mean['week']
df_2019_weekmean['weekend'] = week_mean['weekend']
df_2020_weekmean=pd.DataFrame(mms_2020.inverse_transform(week_mean.iloc[:,2:]),columns=week_mean.iloc[:,2:].columns)
df_2020_weekmean['week'] = week_mean['week']
df_2020_weekmean['weekend'] = week_mean['weekend']
df_2021_weekmean=pd.DataFrame(mms_2021.inverse_transform(week_mean.iloc[:,2:]),columns=week_mean.iloc[:,2:].columns)
df_2021_weekmean['week'] = week_mean['week']
df_2021_weekmean['weekend'] = week_mean['weekend']
```

Q&A