서울시, 공공데이터 기반 서울시 생활인구 예측

F조
김아연, 김진석, 김유민, 김만서, 이상준

1. 데이터 전처리 :

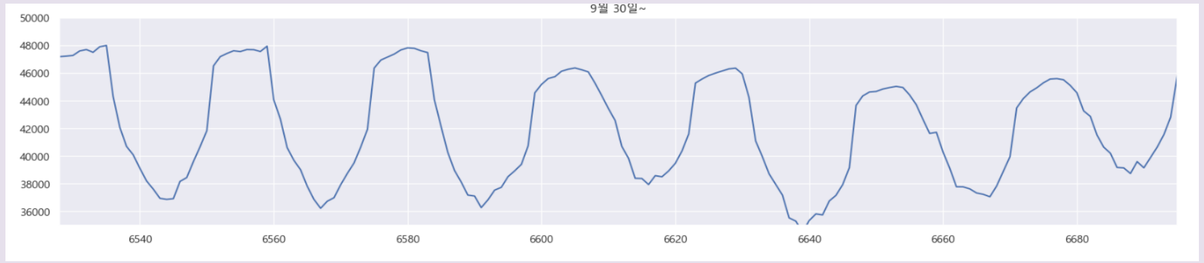
1) 데이터를 계속 들여다 보면서 패턴을 찾으려 노력함.

2) 머신러닝, 딥러닝으로는 절대 좋은 예측을 할 수 없다는 결론을 내리게됨. (설날의 변수 때문)

2. 데이터 그리기 :

1) 윤희남 교수님께서 계속 그려보라고 하셔서 데이터를 계속 그림

*주의사항 : xlim, ylim을 꼭 지정해 놓고 그려야함.



2) 데이터가 큰 범위 안에서 달라지는 것이 아닌 비슷한 패턴이 이어짐을 발견.

3. 핵심 아이디어:

1) 주어진 데이터와 비슷한 데이터가 있다면 결국 1, 2월도 따라가지 않을까?

2) 코치님들께서 데이터가 다르다고 했지만 결국 패턴은 같지 않을까?

3) 행정동 426개동의 패턴을 본다면 어떤 결과가 나올까?

4) Xlim, Ylim을 정해놓고 그려본다면 비슷한 패턴이 나오는 행정동이 있지않을까?

4. 구현 :

1) 426개 동 X 7일(144) = 61,344개 값을 그리면서 패턴을 추적

2) 몇 개의 후보군들의 패턴이 보이기 시작함 약 20개동으로 좁혀짐.

3) 20개 동을 계속(2주 .. N주) 그려가면서 비슷한 패턴을 찾아냄.

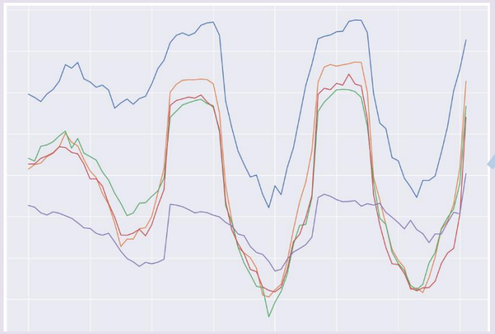
4) 패턴이 비슷한 행정동을 찾아냄. (그러나 코치님께서 데이터를 왜곡 시켰기에 결코 이 과정이 쉽지 않았음)

1. 데이터 전처리 :

13 (음 9.15)	14	15 ✓	16 ✓	17 ✓	18 ✓	19 ✓
20 ✓	21 ✓	22 ✓	23 ✓	24 상강 ✓	25 ✓	26 ✓
27 ✓	28 (음 10.1) ○	29	30	31		

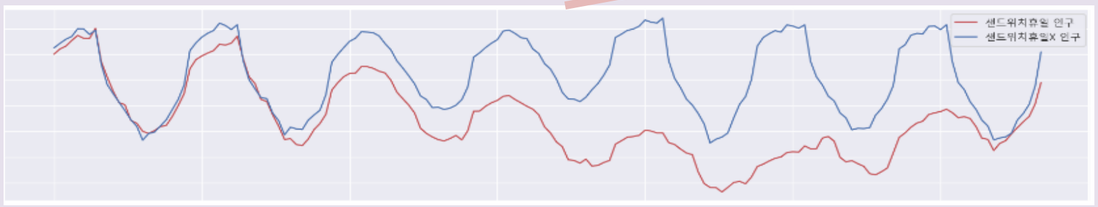
2019 10 15 ~ 2019 10 27일 데이터가 없다
-> 차라리 28일도 빼서 1주일 주기를 맞추자
코로나 상황인 19,20,21년 데이터를 학습

2. 데이터 그리기 :



각 연도별 1월 1일~
1월 3일 그래프

2019년 설 연휴 총
6일동안의 그래프



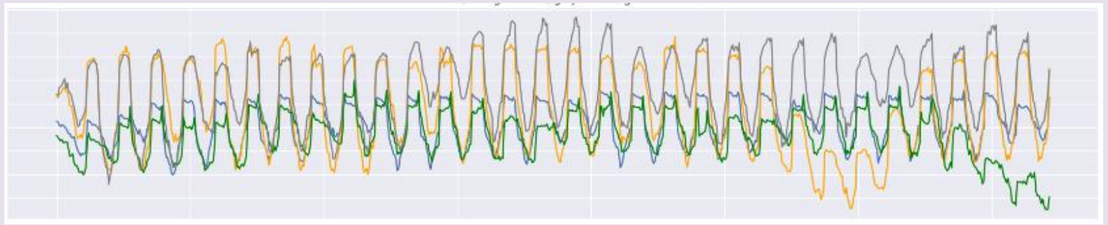
3. 핵심 아이디어:

선형 회귀 모델을 돌리고 나온 값은 후처리가 중요한 것이 아닐까?

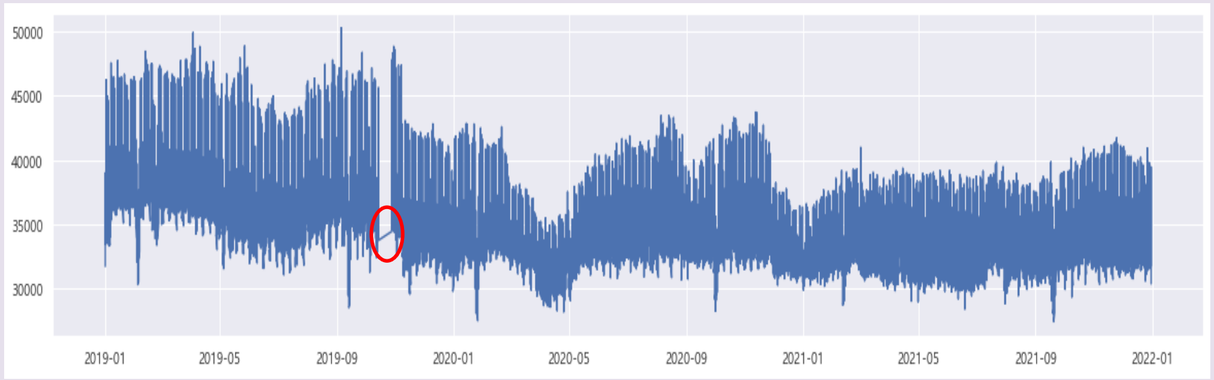
고려 요건
1. 새해 첫날~ 12시까지는 2일, 3일 그래프에 비해 사람이 적다
2. 설 연휴동안 감소한다
3. 코로나로 인해 해마다 그래프가 줄어든 것을 고려

4. 구현 :

1. 예측하고자 하는 1월1일 시작은 토요일이므로 가장 가까운 토요일인 2021-10-23 부터 데이터 뽑아서 shift & roll
2. 선형모델로 나온 결과에 새해 첫날 감소, 명절 감소, 코로나 감소 가중치를 곱하기

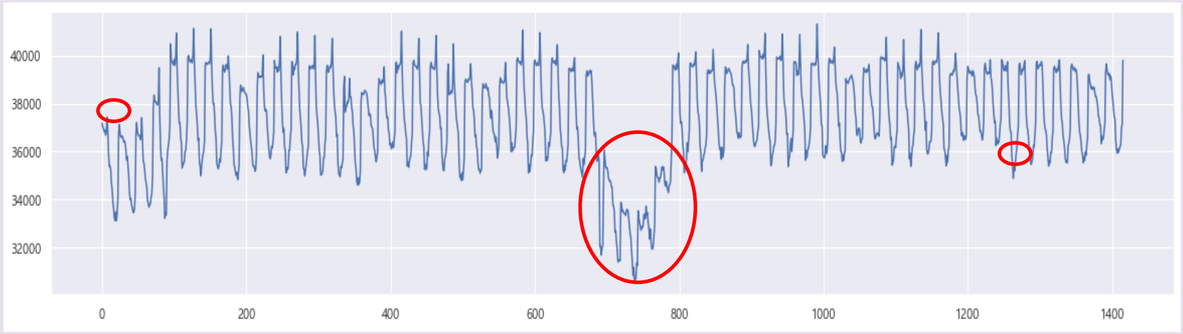


1. 데이터 전처리 :



19~21년 데이터 사용, 결측치 무시
1주일, 1달, 3달, 9달, 1년 단위로 shift 및 roll

2. 모델 학습 및 후처리:



전처리한 데이터로 학습, 테스트는 21년도 1월 2월 데이터로
진행
후처리
공휴일인 1월 1일과 설날 연휴 전날(2022,1,28) 오후 6시 ~ 연휴 마지막 날(2022,2,2)
오후 6시, 코로나 거리두기 정책 반영 -

3. 핵심 아이디어:

1. 예측하고자 하는 데이터는 코로나의 영향을 받았기 때문에 학습 데이터 또한 코로나의 영향을 받은 데이터만을 사용한다면 좋은 결과 있을 거라고 생각함.(전처리)
2. 1월과 2월에는 공휴일이 있기 때문에 생활인구에 영향을 주었다고 생각함.(후처리)

3. 1월과 2월의 코로나 거리두기 방침 또한 생활인구에 영향을 주었을 거라고 생각함.(후처리)

4. 구현 :

모델 : 선형회귀-LinearRegression()
점수 : 824

모델의 한계

코로나가 심각한 1~2월은 예측을 잘할 수 있지만, 학습 데이터가 적어지기 때문에 시간이 지남에 따라 전체적으로 인구가 올라가는지 줄어드는지 등에 대한 추세를 잘 반영하지 못한다는 것.

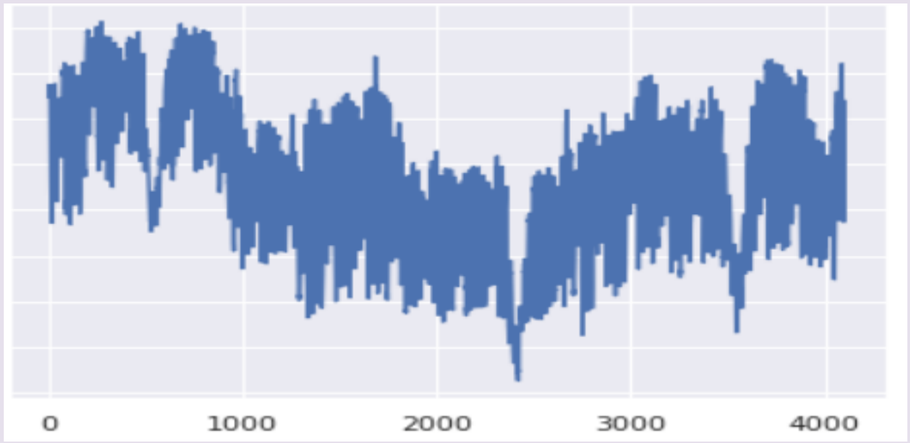
1. 데이터 전처리 :

```
q3 = df_total['총생활인구수'].quantile(0.75)
q1 = df_total['총생활인구수'].quantile(0.25)
iqr = q3-q1
a = (df_total['총생활인구수'] > q3 + 1.5*iqr) | (df_total['총생활인구수'] < q1 - 1.5*iqr)
idx = df_total[a].index
for i in (idx):
  print(i)
  df_total.loc[i,'총생활인구수'] = np.nan
  print(df_total.loc[i,'총생활인구수'])
```

사분위수 편차를 이용하여, 이상치를 최대한 제거하여
그래프를 일반화 시키도록 함.

전반적인 추세를 반영하기에는 크게 분할 하는 게
효과적일 거라 판단하여, 1년 단위로 shift

2. 데이터 그리기 :



이상치를 제거했음에도, 주기적으로 아래로
값들이 꺼지는 시점이 존재하는 걸 확인

3. 핵심 아이디어:

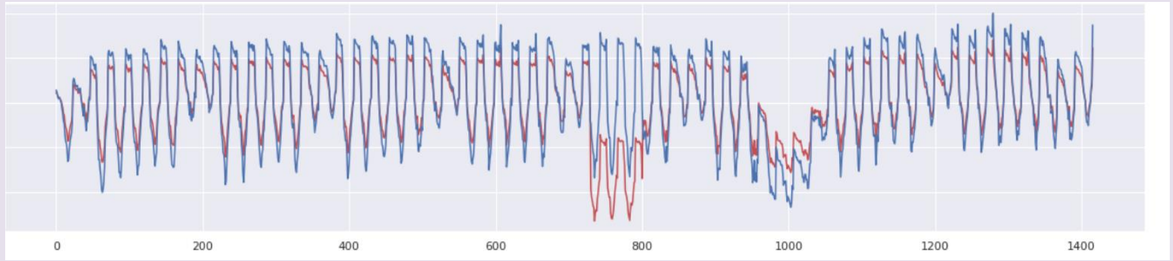
1. 선형회귀는 결국 매년 생활인구수의 변동을 일반화해서
보여주기 때문에, 오버피팅을 피하면서도, 그래프처럼 바닥으로
꺼지는 지점을 찾을 필요가 있음.

2. 특정 지점에 가중치를 줌으로써 해결할 수 있지 않을까?

3. 2022년과 2021년은 사회적 거리두기에 의한 차이 때문에,
2022년이 생활인구가 상대적으로 높을 것을 감안해야한다.

4.하지만 발병 초기 경직된 분위기의 2020년과 코로나 발병
이전년도 보다는 2021년이 가장 유사한 그래프를 그리는
데이터라 판단.

4. 구현 :



1.2021년 데이터 요일 주기를 2022년과 일치시키기 위해서 2021년 1월 2일
부터의 총생활인구수를 x값으로 둠. 특정 지점에 가중치를 부여하면서,
가장 직관적으로 사용할 수 있는 선형회귀를 통해, 예측값 추출.

2.2021년과 예측값이 최대한 일치할수록 정답에 가까울거라 생각하여, 저
변폭이 큰 부분을 최대한 맞추면 정확도가 높아질 거라 생각하지만
아이디어를 찾기 전에 마감되어 아쉬움.

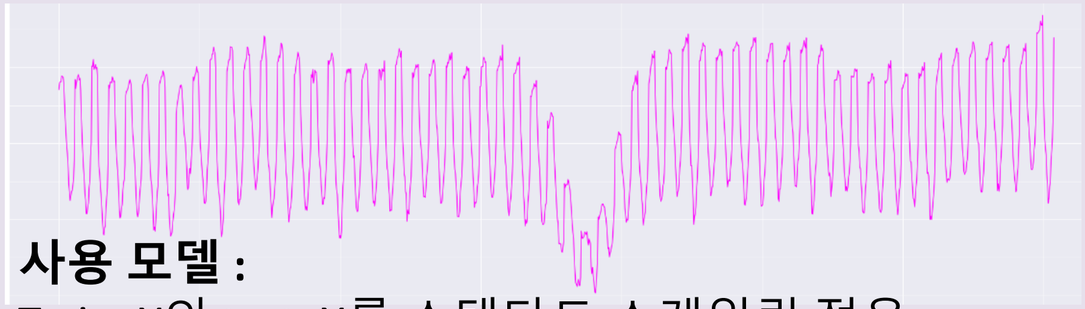
1. 데이터 전처리 :

Train의 목표 = 2021년도 1~2월 생활인구 예측

2021 1~2월 음력 월일	2020 총생활인구수	2019 총 생활인구수	2018 총생활인구수
1118(2021.01.01 음력) ~ 0117(2021.02.28 음력)	음력 월일을 인덱스로 삼아, 연도별 동일한 음력날짜에 대한 총생활인구수		

Test의 목표 = 2022년도 1~2월 생활인구 예측

2022 1~2월 음력 월일	2020 총생활인구수	2019 총 생활인구수	2018 총생활인구수
1129(2022.01.01 음력) ~ 0128(2022.02.28 음력)	음력 월일을 인덱스로 삼아, 연도별 동일한 음력날짜에 대한 총생활인구수		

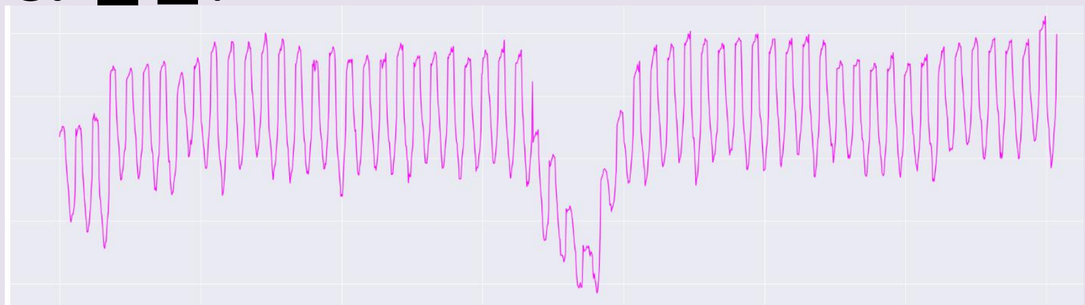


사용 모델 :

Train_X와 test_X를 스탠다드 스케일링 적용

linear regression를 보완해서 나온 **LASSO**

3. 결론:



최종스코어 820~830점대

2. 데이터 후처리 : (한계점 반영)

사용한 데이터는 18, 19, 20, 21 의 1~2월

= 코로나로 인한 생활인구수 감소를 잘 반영하지 못했을 것

➢ 전체 데이터 * 0.99

=1,2월의 데이터만 이용-> 전날의 데이터를 반영하지 못함

2021년도 12.28~12.31 평균적으로 3만 4천, 바로 다음 날인 모델이 예측한 1.1~1.3일 데이터는 평균적으로 3만 8천

➢ 바로 전날의 데이터를 반영해주기 위해 [1.1~1.3]*0.95

=설 연휴 이동 인구 수가 더 증가했을 것

[설날3일전 ~ 설날] *0.95

1. 데이터 전처리 :

Train의 목표 = 2021년도 1~2월 생활인구 예측

Test의 목표 = 2022년도 1~2월 생활인구 예측

2021 1~2월 음력 월일	2019 생활인구수	2018 총생활인구수
1118(2021.01.01 음력 ~ 0117(2021.02.28 음력	을 인덱스로 삼아, 날짜에 대한 총생활인구수	

21년도 1~2월 예측 위한 train_X

월일	음력	총생활인구수	2년전음력	2년전총생활인구수	3년전음력	3년전총생활인구수	시간대구분
1230	20191230	32461.4457	20181230	33644.9237	20171230	32397.8976	16
1230	20191230	32572.9124	20181230	33300.5551	20171230	32174.5247	17
1230	20191230	32764.0829	20181230	33657.0220	20171230	31870.6307	18
1230	20191230	32663.6476	20181230	34039.2241	20171230	31811.6000	19
1230	20191230	31875.8291	20181230	34125.4904	20171230	31627.5609	20
1230	20191230	32099.6727	20181230	33938.4749	20171230	31798.4512	21
1230	20191230	32530.3060	20181230	34461.7506	20171230	31880.4333	22
1230	20191230	35951.5437	20181230	35279.0704	20171230	33354.0436	23
0101	20200101	36103.4259	20190101	35465.0305	20180101	33454.3142	0
0101	20200101	35998.0441	20190101	35709.0414	20180101	33527.1741	1
0101	20200101	35904.1479	20190101	35917.4943	20180101	33550.2985	2
0101	20200101	35939.2294	20190101	36021.6756	20180101	33562.4396	3
0101	20200101	36016.1200	20190101	36365.8707	20180101	33654.8505	4

반영)

영하지 못했을 것

사용 모델 :

Train_X와 test_X를

linear regression를

를 반영하지 못함

바로 다음 날인

으로 3만 8천

[1.1~1.3]*0.95

3. 결론:

최종스코어 820~830점대

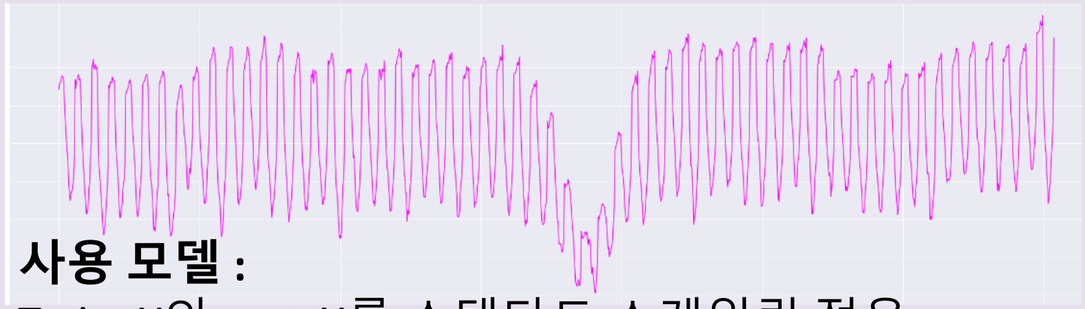
1. 데이터 전처리 :

Train의 목표 = 2021년도 1~2월 생활인구 예측

2021 1~2월 음력 월일	2020 총생활인구수	2019 총 생활인구수	2018 총생활인구수
1118(2021.01.01 음력) ~ 0117(2021.02.28 음력)	음력 월일을 인덱스로 삼아, 연도별 동일한 음력날짜에 대한 생활인구수		

Test의 목표 = 2022년도 1~2월 생활인구 예측

2022 1~2월 음력 월일	2020 총생활인구수	2019 총 생활인구수	2018 총생활인구수
1129(2022.01.01 음력) ~ 0128(2022.02.28 음력)	음력 월일을 인덱스로 삼아, 연도별 동일한 음력날짜에 대한 생활인구수		

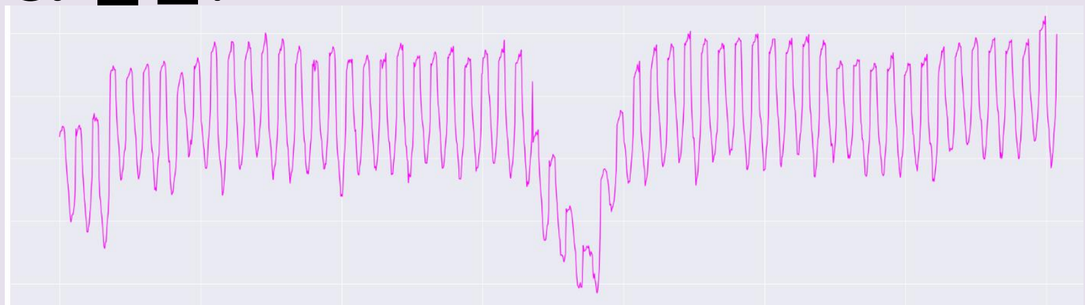


사용 모델 :

Train_X와 test_X를 스탠다드 스케일링 적용

linear regression를 보완해서 나온 **LASSO**

3. 결론:



최종스코어 820~830점대

2. 데이터 후처리 : (한계점 반영)

사용한 데이터는 18, 19, 20, 21 의 1~2월

= 코로나로 인한 생활인구수 감소를 잘 반영하지 못했을 것

➢ 전체 데이터 * 0.99

=1,2월의 데이터만 이용-> 전날의 데이터를 반영하지 못함

2021년도 12.28~12.31 평균적으로 3만 4천, 바로 다음 날인

모델이 예측한 1.1~1.3일 데이터는 평균적으로 3만 8천

➢ 바로 전날의 데이터를 반영해주기 위해 [1.1~1.3]*0.95

=설 연휴 이동 인구 수가 더 증가했을 것

[설날3일전 ~ 설날] *0.95

The end

감사합니다!