

재벌집막내조

2022 UOS 빅데이터 알고리즘 경진대회

목차

A

NEW IDEA

B

ARIMA , SARIMAX

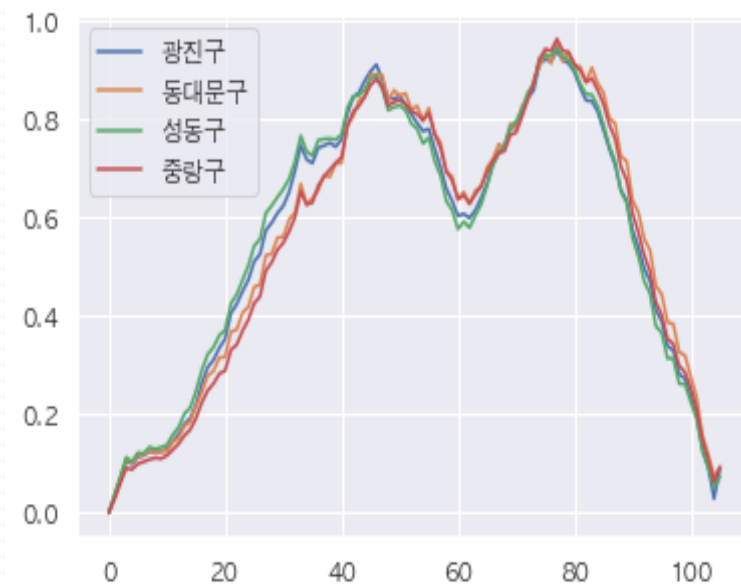
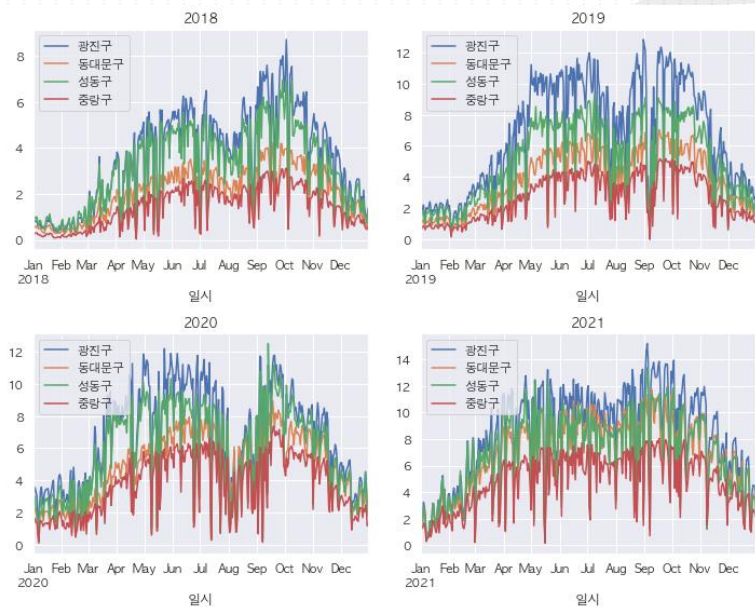
C

DEEP LEARNING

NEW IDEA

LAST WEEK

이동 평활



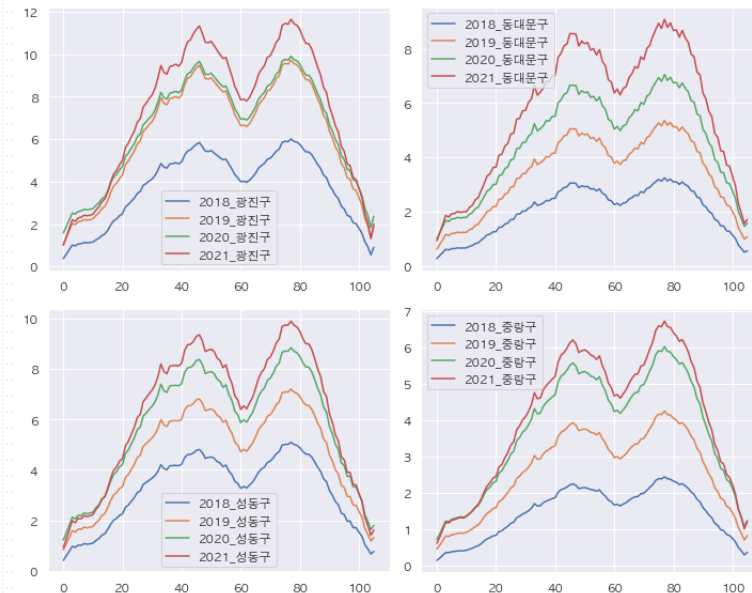
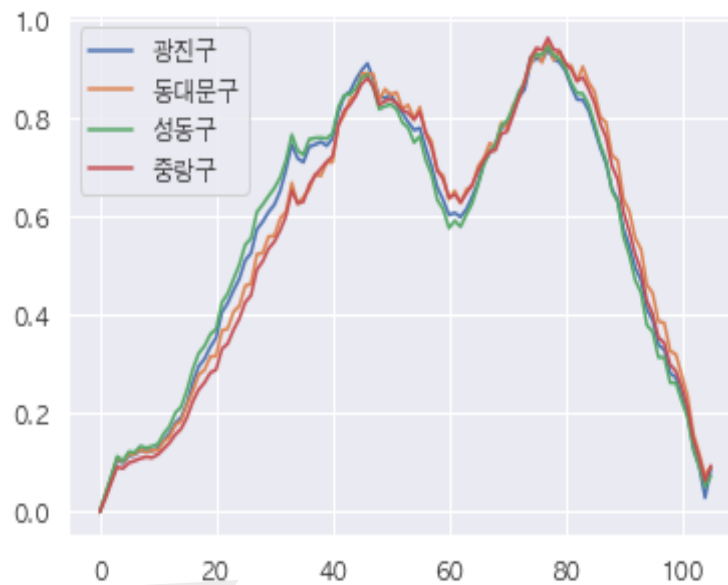
Year : MinmaxScaler

NEW IDEA

Year : Inverse_transform



지역 기준 groupby



Check

연도 기준 plot -> 지역 기준 groupby plot-> 지역 기준 plot

NEW IDEA

Question

2022년 대여량 = 2021년 대여량 + 지역구의 연도별 대여량의 차이의 평균?

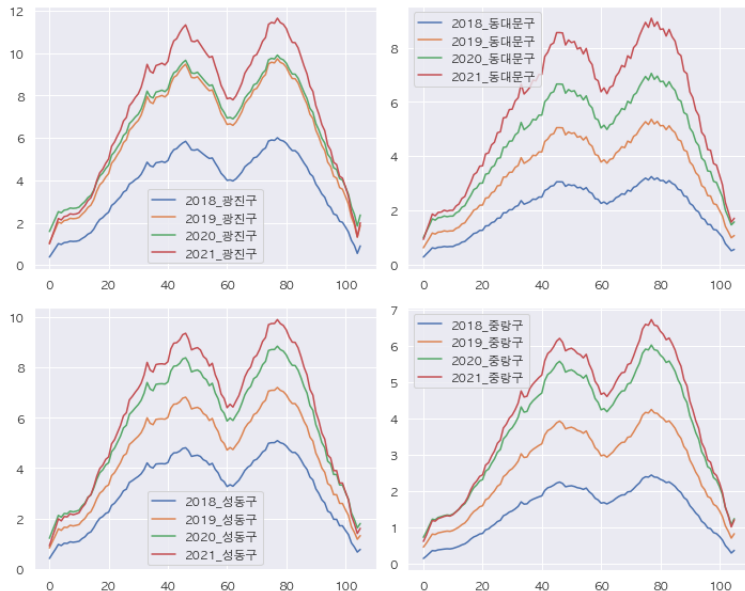
EX) 광진구의 2022년 대여량 = 광진구 21년 대여량 데이터 + mean((21년 대여량 - 20년 대여량),(20년 대여량 - 19년 대여량),(19년 대여량 - 18년 대여량))

Question

증가분이 어느 정도 인가? 수리적 계산이 가능한가?

Answer

추론의 영역...? 가능할 수도?



X2, X3



NEW IDEA

NEW IDEA

LAST WEEK

NEW IDEA

Year : Inverse transform

NEW IDEA

Question

2022년 대여량 = 2021년 대여량 + 지역구의 연도별 대여량의 차이의 평균?

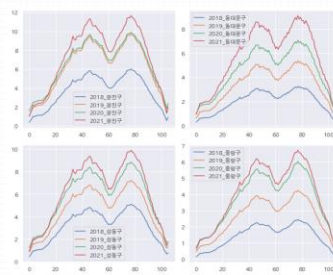
EX) 광진구의 2022년 대여량 = 광진구 21년 대여량 데이터 + $\text{mean}((21\text{년 대여량} - 20\text{년 대여량}), (20\text{년 대여량} - 19\text{년 대여량}), (19\text{년 대여량} - 18\text{년 대여량}))$

Question

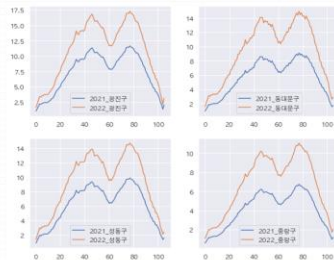
증가분이 어느 정도 인가? 수리적 계산이 가능한가?

Answer

추론의 영역...? 가능할 수도?



X2, X3



Check

원본 데이터 - 추세(시계열 분해)

-> 시계열 예측

-> 예측 결과 + 추세 (시계열 분해)와 비슷한 과정

LAST WEEK

ARIMA, SARIMAX

시계열 성분 분석

4P

30일 기준 / 가법 분해

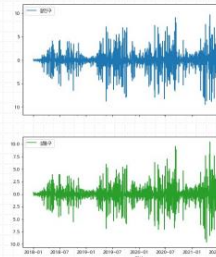
```
from statsmodels.tsa.seasonal
    gu = ['광진구', '동대문구', '
plt.rcParams["figure.figsize"]
# 시계열 성분 분해
def time_series_decomposition
    for i in list:
        seasonal_decompose(df[i],
        return 0
time_series_decompositi
#한달 기준
```

트렌드가 일
잔차



ACF, PACF 확인

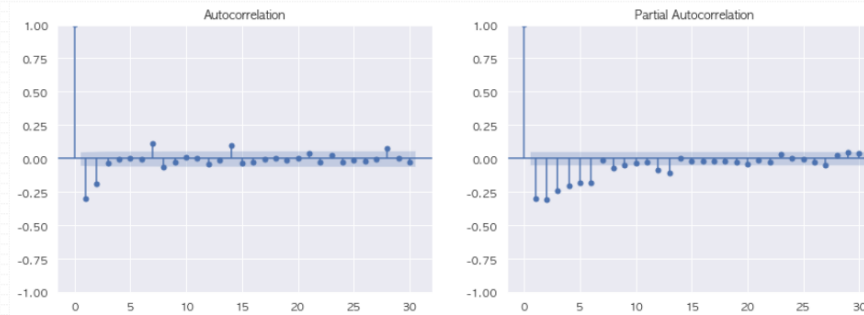
4P



차분
중간에

ACF, PACF 확인

4P



ACF : 1 이후로 0으로 수렴

PACF : 2 이후로 0으로 수렴

=> ARIMA 수행시 AR(1), MA(2)과 관련한 조합으로 하면 성능이 좋을 것이라 추측

Check

시계열 성분 분석

차분 & log 변환

ACF, PACF 확인

정상 시계열로 변환 완료

ARIMA, SARIMAX

ARIMA

```
def modeling_arima(modeling_data, fore_data, col):  
    model_arima = sm.tsa.arima.ARIMA(modeling_data[col], order = pdq[0])  
    history = model_arima.fit()  
    history.summary()  
    pred = history.get_forecast(len(fore_data[col]))  
    return pred
```

```
from sklearn.metrics import mean_absolute_error  
mean_absolute_error(ml_test[col][1:], rev_pred)
```

2.9421685536577615

SARIMAX

```
def modeling_sarimax(modeling_data, fore_data, col):  
    model_arima = sm.tsa.SARIMAX(modeling_data[col], order = pdq[0])  
    history = model_arima.fit()  
    history.summary()  
    pred = history.get_forecast(len(fore_data[col]))  
    return pred
```

```
from sklearn.metrics import mean_absolute_error  
mean_absolute_error(ml_test[col][1:], rev_pred)
```

3.270039794878407

Check

ARIMA MAE < SARIMAX MAE

ARIMA, SARIMAX

2 / 1000

최신순

점수순

	제목	제출 일시	public점수 private점수	제출선택
808809	sub_pred (1).csv SARIMAX edit	2023-02-21 14:33:55	4.6703845002 5.6753834257	<input type="checkbox"/>
808801	sub_pred.csv ARIMA edit	2023-02-21 14:24:45	4.4008179763 5.1630478622	<input checked="" type="checkbox"/>

Check

ARIMA : 90등

DEEP LEARNING

A

LSTM

B

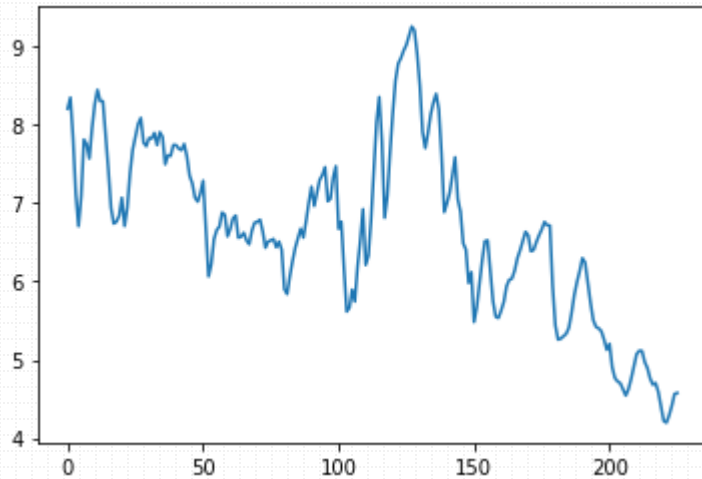
CNN

C

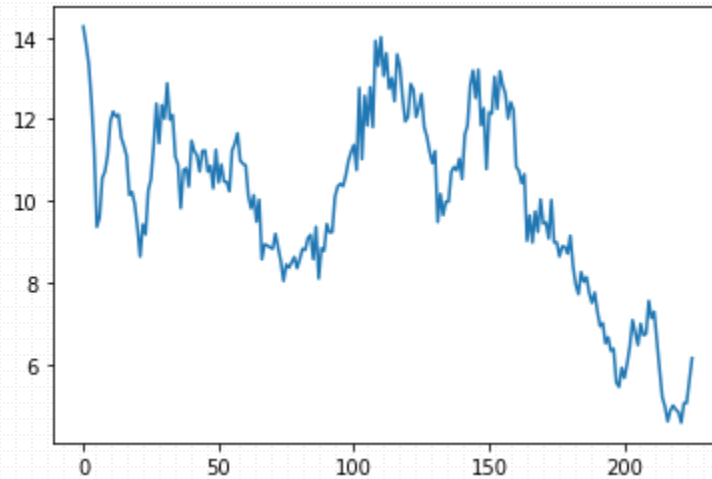
CNN-LSTM

DEEP LEARNING

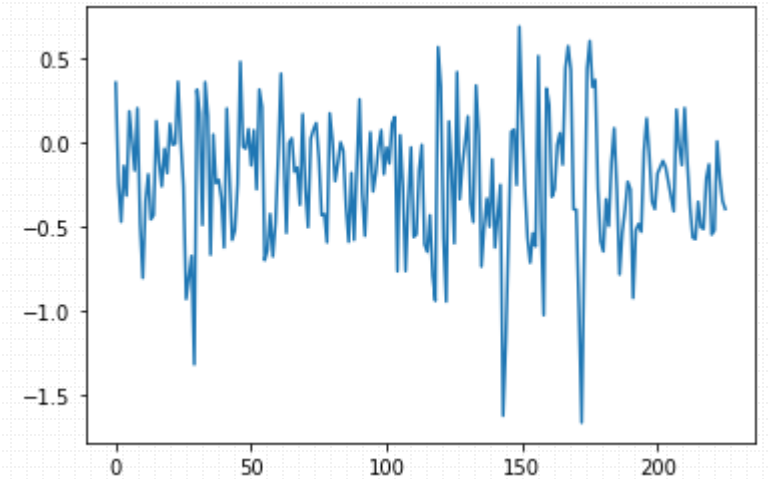
60일 기준 7일 예측 : 모델 성능



LSTM MAE : 3.244



CNN MAE : 2.000



CNN-LSTM MAE : 9.509

Check

CNN < LSTM < CNN-LSTM

NEXT

A

RNN, GRU

B

ARIMA, SARIMAX

C

NEW IDEA



THANK YOU

재벌집막내조

최지혁 신주연 김진호 박은비