

도와줘 소상공인!

신입 소상공인을 위한 시장 매력도 제공 서비스

시계열 1조

19기 이동준 임지영 최태순

20기 김지우 정준호 조미현 황민정

목차

1. 프로젝트 배경 및 개요
2. 데이터 설명
3. 지표 설정
4. 시계열 모델을 이용한 판매량 예측
5. DA를 이용한 판매량 예측
6. 결론 및 개선점

프로젝트 배경 및 개요

Q1

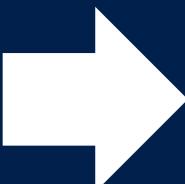
최근 이커머스 시장의
가파른 상승세와 함께
새롭게 시장에
진입하려는 소상공인
또한 가파르게 증가

Q2

어떤 상품군으로
진입해야 흑자 사업을
운영할 수 있는가
판단하기란
쉽지 않은 일

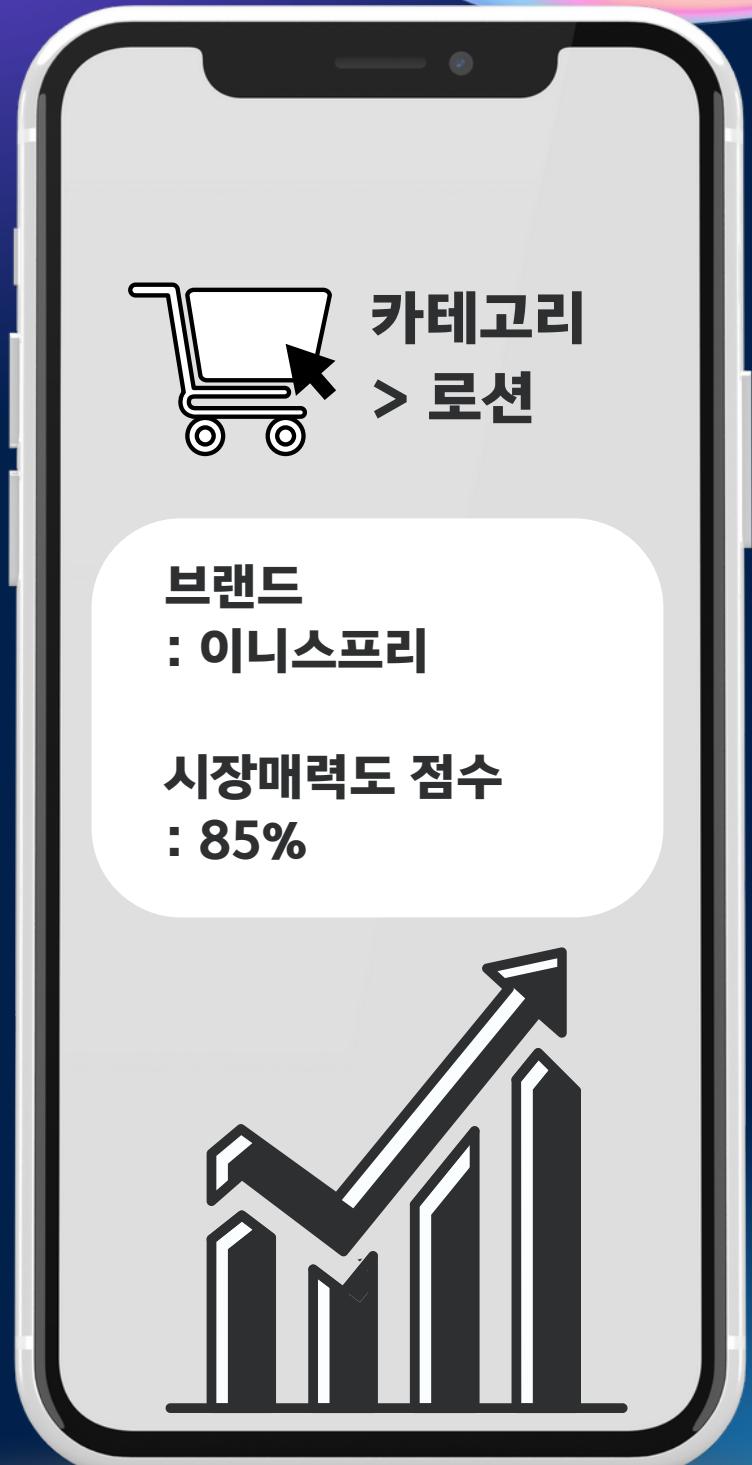
Q3

온라인 판매 시장에
뛰어들려는
신입 소상공인에게
진입하기 좋은
상품군을 추천해주는
서비스 필요



도와줘 소상공인!

신입 소상공인을 위한 시장 매력도 제공 서비스 필요



데이터 설명

LG Aimers 온라인 쇼핑몰 일별/제품별 판매량 데이터

<Column 구성>

- ID : 실제 판매되고 있는 고유 ID
- 제품 : 제품 코드
- 대분류 : 제품의 대분류 코드
- 중분류 : 제품의 중분류 코드
- 소분류 : 제품의 소분류 코드
- 브랜드 : 제품의 브랜드 코드
- 2022-01-01 ~ 2023-04-04 : 실제 일별 판매량

0	0	B002-00001-00001	B002-C001-0002	B002-C002-0007	B002-C003-0038	B002-00001	0	0	0	0	...
1	1	B002-00002-00001	B002-C001-0003	B002-C002-0008	B002-C003-0044	B002-00002	0	0	0	0	...
2	2	B002-00002-00002	B002-C001-0003	B002-C002-0008	B002-C003-0044	B002-00002	0	0	0	0	...
3	3	B002-00002-00003	B002-C001-0003	B002-C002-0008	B002-C003-0044	B002-00002	0	0	0	0	...
4	4	B002-00003-00001	B002-C001-0001	B002-C002-0001	B002-C003-0003	B002-00003	0	0	0	0	...
...
15885	15885	B002-03799-00002	B002-C001-0003	B002-C002-0008	B002-C003-0042	B002-03799	0	0	0	0	...
15886	15886	B002-03799-00003	B002-C001-0003	B002-C002-0008	B002-C003-0044	B002-03799	0	0	0	0	...
15887	15887	B002-03799-00004	B002-C001-0003	B002-C002-0008	B002-C003-0044	B002-03799	0	0	0	0	...
15888	15888	B002-03799-00005	B002-C001-0003	B002-C002-0008	B002-C003-0044	B002-03799	0	0	0	0	...
15889	15889	B002-03799-00010	B002-C001-0002	B002-C002-0004	B002-C003-0020	B002-03799	0	0	0	0	...

15890 rows x 12 columns

‘소분류’ nunique : 53

- 데이터 부족으로 충분히 많은 상품 항목 커버 불가
- 학습되지 못한 새로운 상품에 대한 예측 불가



‘대분류’로 학습을 진행하여 데이터 부족 문제를 해결하고
새로운 상품을 적용할 수 있도록 DA 적용

지표 설정

성장가능성 지표란?

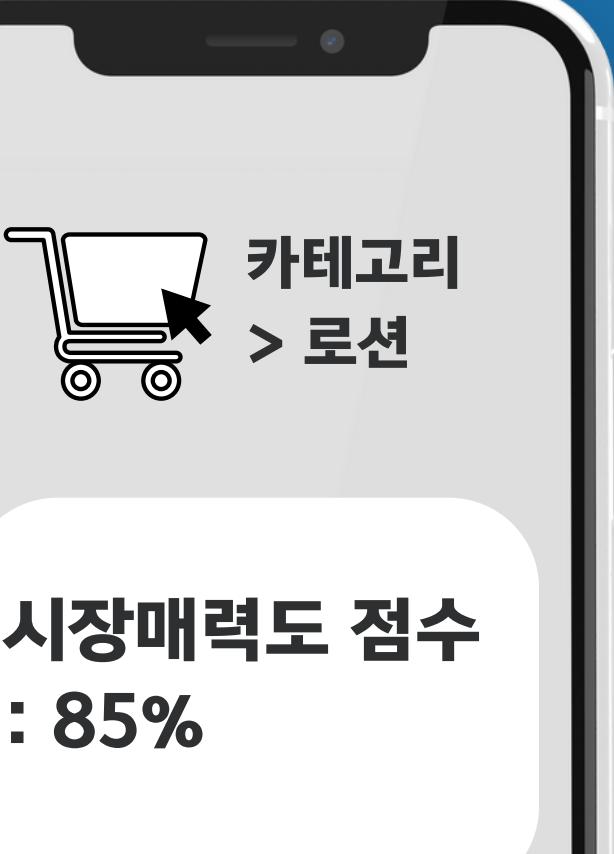


- 먼 미래에, 판매량이 잠재적으로 증가할 것인가?
• 성장가능성지표 = 미래 예측 판매량 / 현시점 예측 판매량

- 절대적인 판매량 수치는 상품군에 따른 차이로 인해 시장 현황에 대한 지표로 사용하기 어렵다고 판단
 - → 현시점·미래시점 판매량을 바탕으로 시장의 성장가능성을 지표로 제시
- 현재의 판매량과 경쟁률이 아닌 미래의 판매량과 경쟁력을 예측해주는 서비스가 부족한 상황
- 모델의 일반화 성능 향상(DA)을 통해 대분류의 흐름을 따르는 개별 상품의 판매량 예측

최종적으로 시장매력도 점수 (성장가능성 지표)

높게 나타난 항목을 추천 진입 항목으로 제시

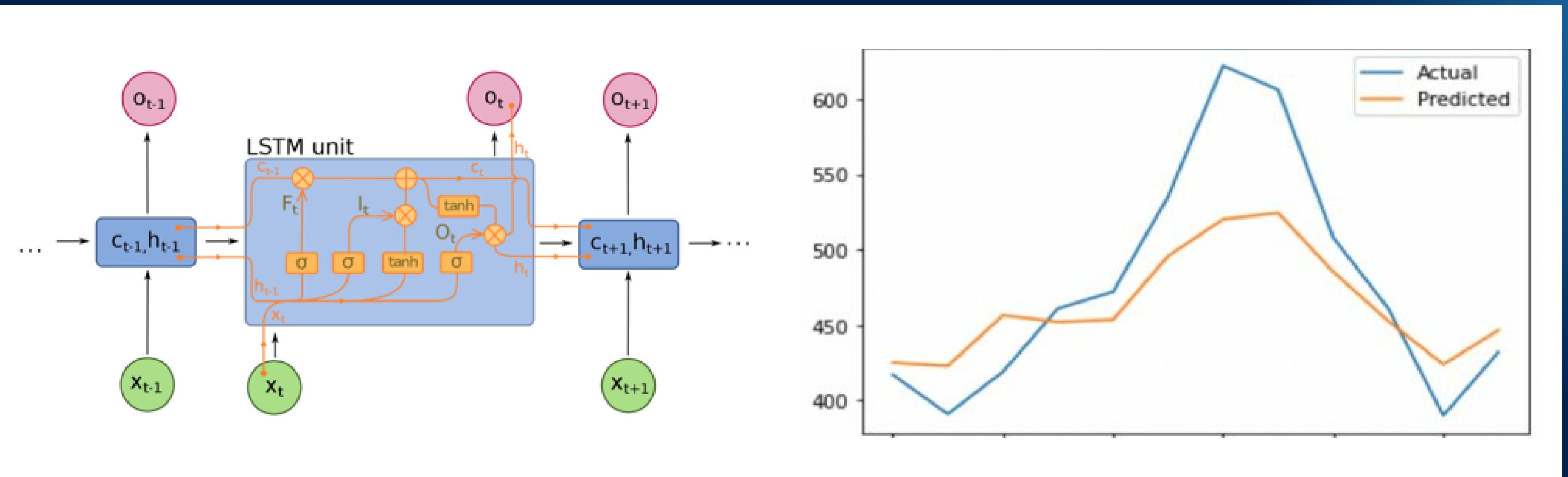


시계열 모델을 이용한 판매량 예측

1) LSTM

- RNN에서 출력과 먼 위치에 있는 정보를 기억하지 못하는 한계점을 개선하고 장기 기억능력을 발전시킨 모델
- 병렬처리 계산이 안돼 속도가 느리다는 단점

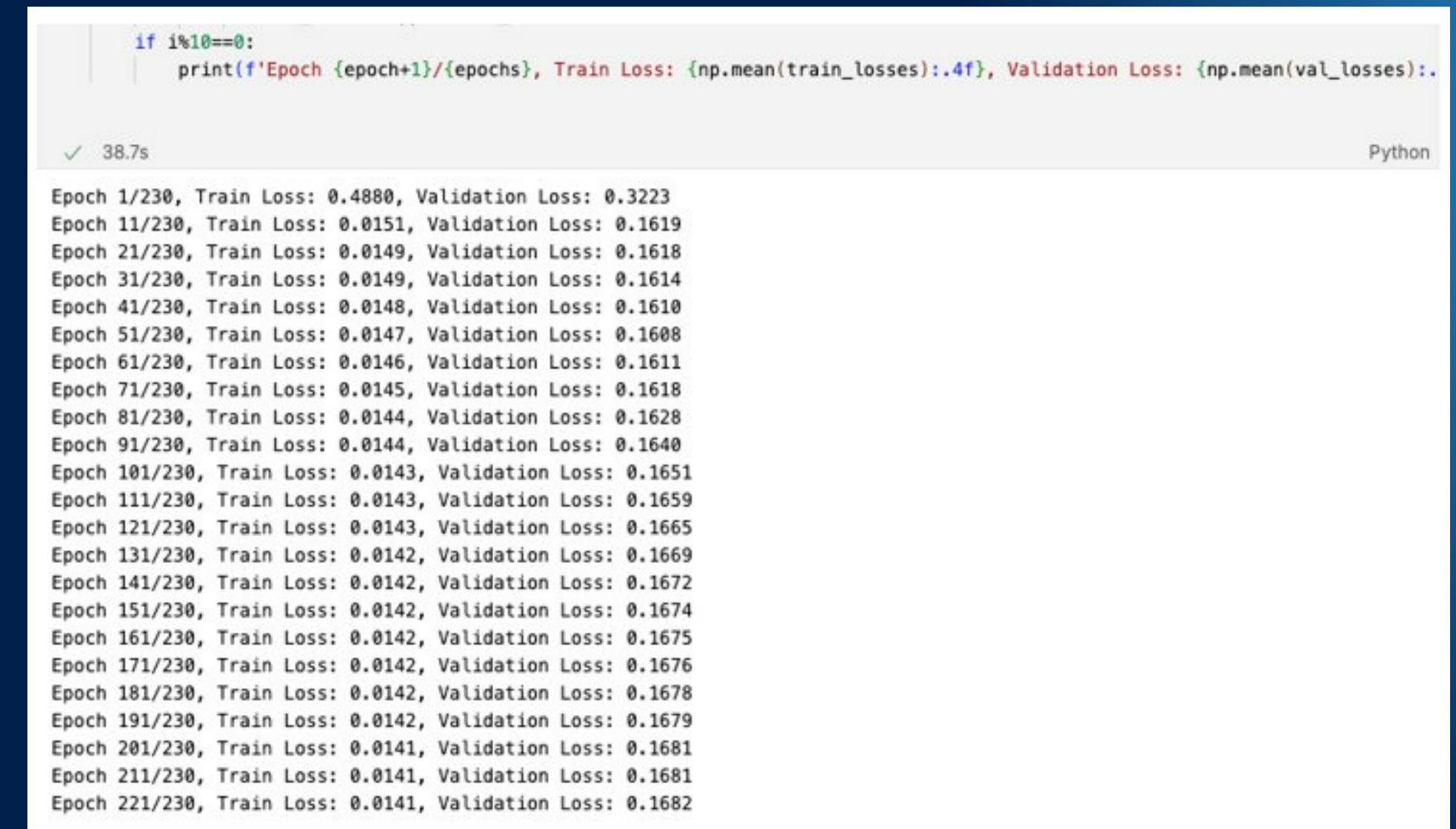
<Model Architecture>



1) LSTM

<Parameter>

- Window size : 30(30일치 학습)
- Forecast size : 10(10일치 예측)
- Batch size : 32
- Loss : MSELoss
- Optimizer : Adam
- Train / Valid / Test : 0.64 : 0.16 : 0.2



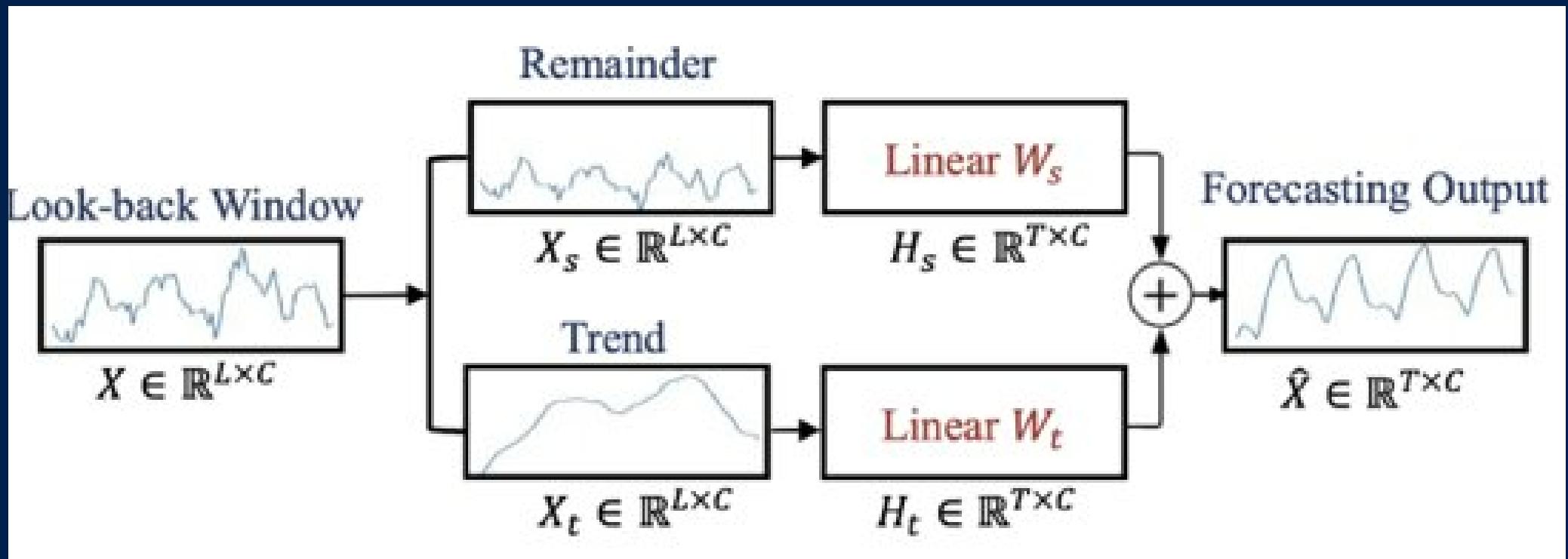
```
if i%10==0:  
    print(f'Epoch {epoch+1}/{epochs}, Train Loss: {np.mean(train_losses):.4f}, Validation Loss: {np.mean(val_losses):.4f}')  
  
38.7s
```

Epoch 1/230, Train Loss: 0.4880, Validation Loss: 0.3223
Epoch 11/230, Train Loss: 0.0151, Validation Loss: 0.1619
Epoch 21/230, Train Loss: 0.0149, Validation Loss: 0.1618
Epoch 31/230, Train Loss: 0.0149, Validation Loss: 0.1614
Epoch 41/230, Train Loss: 0.0148, Validation Loss: 0.1610
Epoch 51/230, Train Loss: 0.0147, Validation Loss: 0.1608
Epoch 61/230, Train Loss: 0.0146, Validation Loss: 0.1611
Epoch 71/230, Train Loss: 0.0145, Validation Loss: 0.1618
Epoch 81/230, Train Loss: 0.0144, Validation Loss: 0.1628
Epoch 91/230, Train Loss: 0.0144, Validation Loss: 0.1640
Epoch 101/230, Train Loss: 0.0143, Validation Loss: 0.1651
Epoch 111/230, Train Loss: 0.0143, Validation Loss: 0.1659
Epoch 121/230, Train Loss: 0.0143, Validation Loss: 0.1665
Epoch 131/230, Train Loss: 0.0142, Validation Loss: 0.1669
Epoch 141/230, Train Loss: 0.0142, Validation Loss: 0.1672
Epoch 151/230, Train Loss: 0.0142, Validation Loss: 0.1674
Epoch 161/230, Train Loss: 0.0142, Validation Loss: 0.1675
Epoch 171/230, Train Loss: 0.0142, Validation Loss: 0.1676
Epoch 181/230, Train Loss: 0.0142, Validation Loss: 0.1678
Epoch 191/230, Train Loss: 0.0142, Validation Loss: 0.1679
Epoch 201/230, Train Loss: 0.0141, Validation Loss: 0.1681
Epoch 211/230, Train Loss: 0.0141, Validation Loss: 0.1681
Epoch 221/230, Train Loss: 0.0141, Validation Loss: 0.1682

2) Dlinear

- 공급자를 위한 좋은 상품을 예측하기 위해 단기 뿐만 아니라 장기적인 흐름을 봄아하는 경우도 많음
→ 장기 시계열 예측에 transformer와 비교하여 좋은 성능을 보인 LTSF-dlinear 모델 역시 baseline model로 선정

<Model Architecture>



1. 데이터를 추세/주기성 데이터로 분해
2. 각 요소를 선형 레이어에 적합해 학습
3. 두 레이어를 합산해 최종 예측 계산

2) Dlinear

<Parameter>

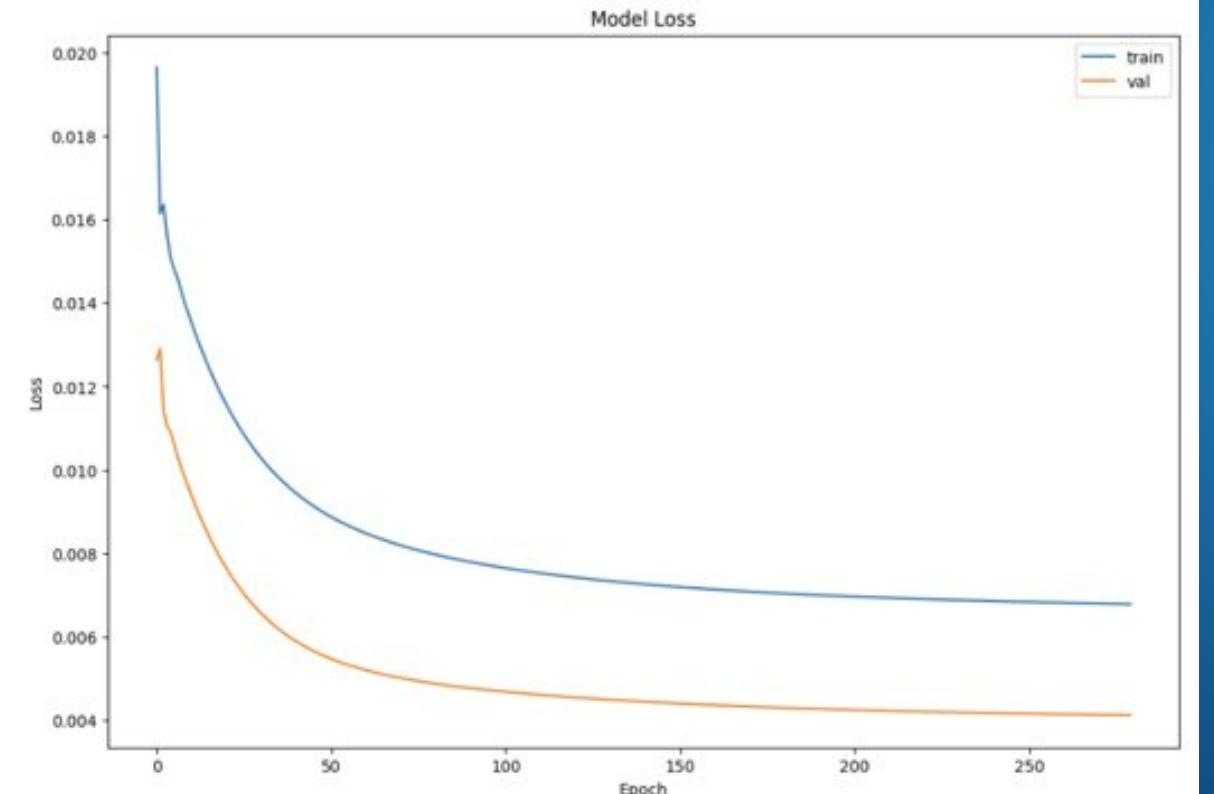
- Window size : 30(30일치 학습)
- Forecast size : 10(10일치 예측)
- Batch size : 32
- Epoch : 280
- Loss : MSELoss
- Optimizer : Adam
- Train : Valid : Test
= 0.64 : 0.16 : 0.20

```

if (epoch % 10 == 0):
    print("epoch = {}, train_loss : {:.3f}, valid_loss : {:.3f}".format(epoch, np.mean(loss_list), valid_loss))

 6%| 16/280 [00:00<00:07, 36.74it/s]epoch = 10, train_loss : 0.014, valid_loss : 0.010
 9%| 24/280 [00:00<00:07, 34.15it/s]epoch = 20, train_loss : 0.012, valid_loss : 0.008
11%| 32/280 [00:01<00:09, 27.50it/s]epoch = 30, train_loss : 0.010, valid_loss : 0.007
16%| 44/280 [00:01<00:08, 26.51it/s]epoch = 40, train_loss : 0.010, valid_loss : 0.006
19%| 53/280 [00:01<00:08, 26.85it/s]epoch = 50, train_loss : 0.009, valid_loss : 0.006
23%| 65/280 [00:02<00:08, 25.11it/s]epoch = 60, train_loss : 0.009, valid_loss : 0.005
25%| 71/280 [00:02<00:12, 16.09it/s]epoch = 70, train_loss : 0.008, valid_loss : 0.005
29%| 82/280 [00:03<00:18, 10.82it/s]epoch = 80, train_loss : 0.008, valid_loss : 0.005
33%| 92/280 [00:04<00:15, 11.88it/s]epoch = 90, train_loss : 0.008, valid_loss : 0.005
36%| 101/280 [00:05<00:09, 18.01it/s]epoch = 100, train_loss : 0.008, valid_loss : 0.005
40%| 111/280 [00:06<00:18, 9.10it/s]epoch = 110, train_loss : 0.008, valid_loss : 0.004
44%| 123/280 [00:06<00:09, 15.96it/s]epoch = 120, train_loss : 0.007, valid_loss : 0.004
48%| 133/280 [00:07<00:08, 17.83it/s]epoch = 130, train_loss : 0.007, valid_loss : 0.004
50%| 141/280 [00:07<00:06, 21.11it/s]epoch = 140, train_loss : 0.007, valid_loss : 0.004
55%| 153/280 [00:08<00:06, 19.95it/s]epoch = 150, train_loss : 0.007, valid_loss : 0.004
58%| 162/280 [00:08<00:05, 21.35it/s]epoch = 160, train_loss : 0.007, valid_loss : 0.004
62%| 174/280 [00:09<00:05, 21.14it/s]epoch = 170, train_loss : 0.007, valid_loss : 0.004
65%| 182/280 [00:10<00:07, 13.71it/s]epoch = 180, train_loss : 0.007, valid_loss : 0.004
69%| 194/280 [00:10<00:03, 24.80it/s]epoch = 190, train_loss : 0.007, valid_loss : 0.004
74%| 207/280 [00:10<00:02, 33.68it/s]epoch = 200, train_loss : 0.007, valid_loss : 0.004
77%| 215/280 [00:11<00:01, 35.94it/s]epoch = 210, train_loss : 0.007, valid_loss : 0.004
81%| 227/280 [00:11<00:01, 37.33it/s]epoch = 220, train_loss : 0.007, valid_loss : 0.003
84%| 235/280 [00:11<00:01, 36.15it/s]epoch = 230, train_loss : 0.007, valid_loss : 0.003
88%| 247/280 [00:11<00:00, 36.82it/s]epoch = 240, train_loss : 0.007, valid_loss : 0.003
91%| 256/280 [00:12<00:00, 38.20it/s]epoch = 250, train_loss : 0.007, valid_loss : 0.003
94%| 264/280 [00:12<00:00, 37.26it/s]epoch = 260, train_loss : 0.007, valid_loss : 0.003
99%| 276/280 [00:12<00:00, 36.03it/s]epoch = 270, train_loss : 0.007, valid_loss : 0.003
100%| 280/280 [00:12<00:00, 21.81it/s]epoch = 280, train_loss : 0.007, valid_loss : 0.003

```

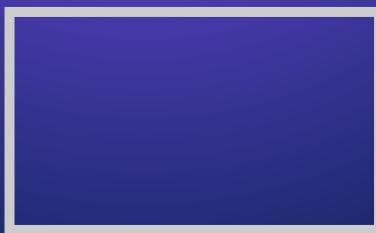


<판매금액 결과>

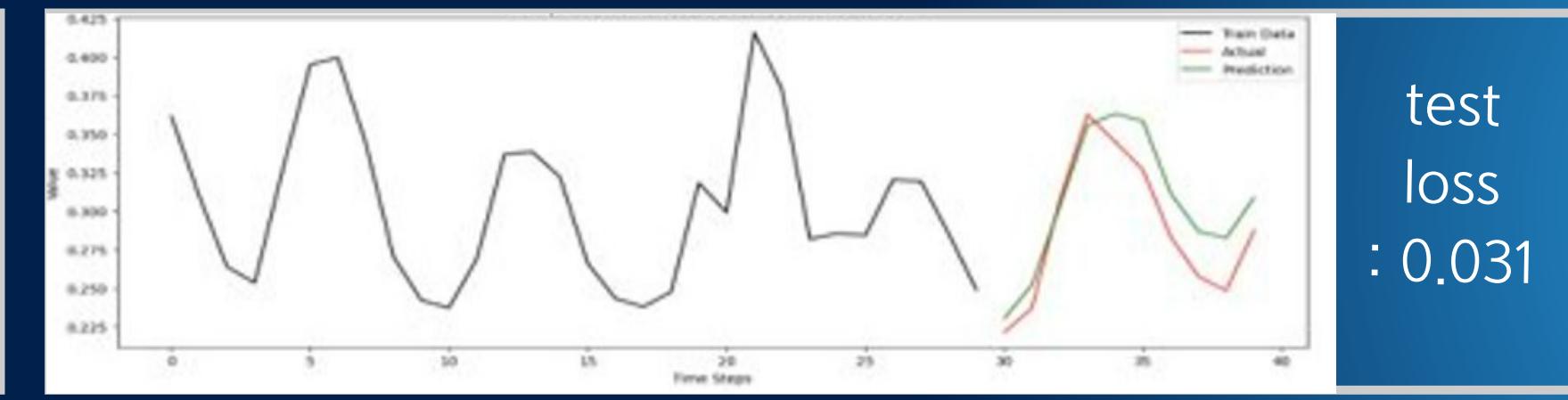
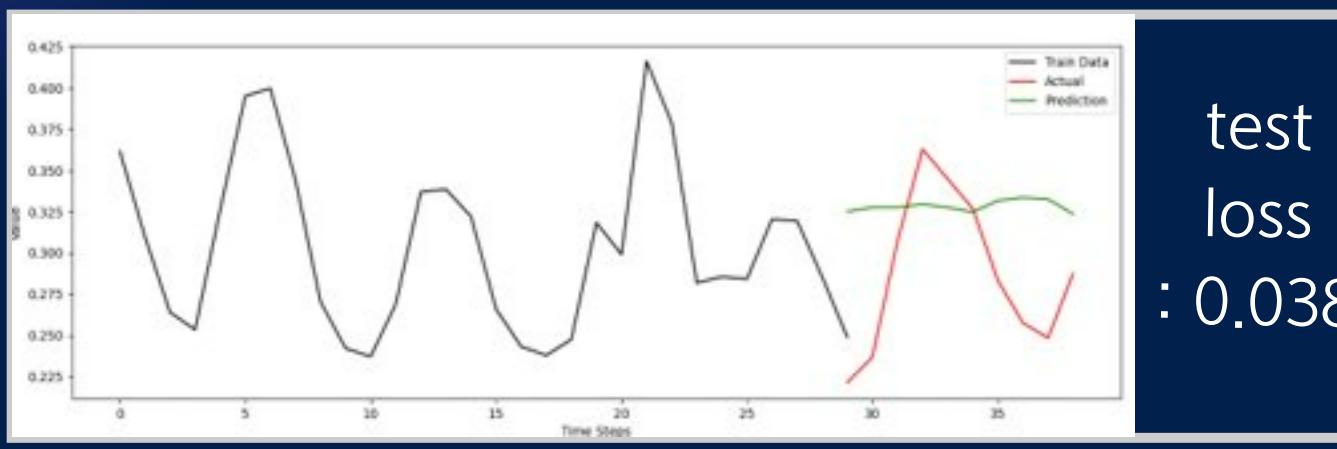
Train_loss : 0.009

Valid_loss : 0.002

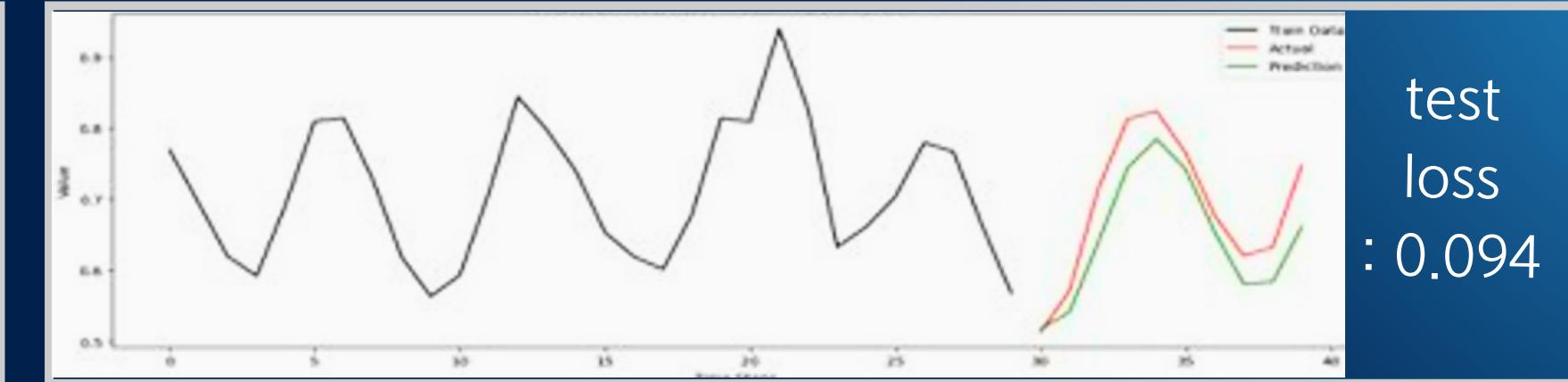
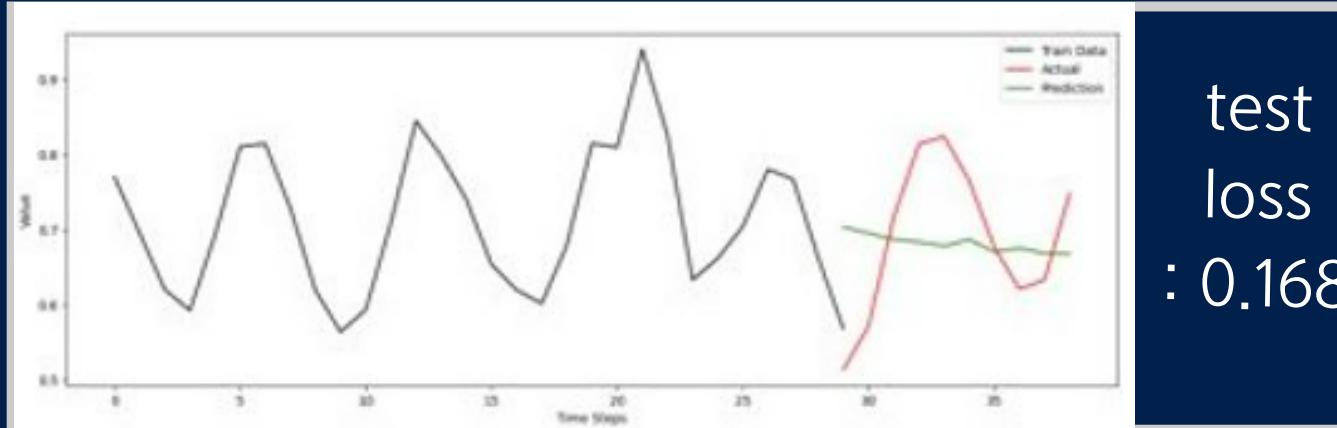
LSTM vs Dlinear



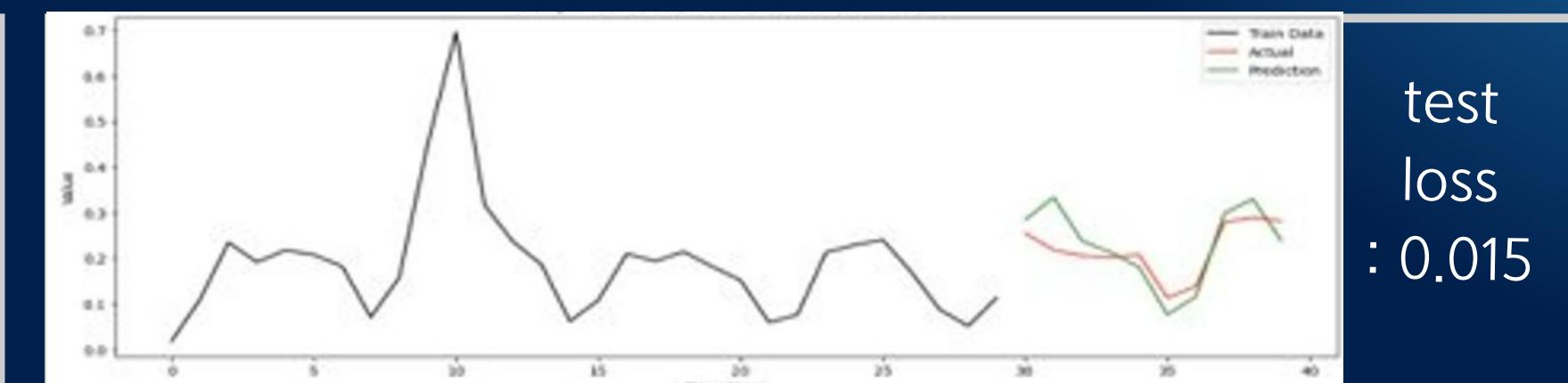
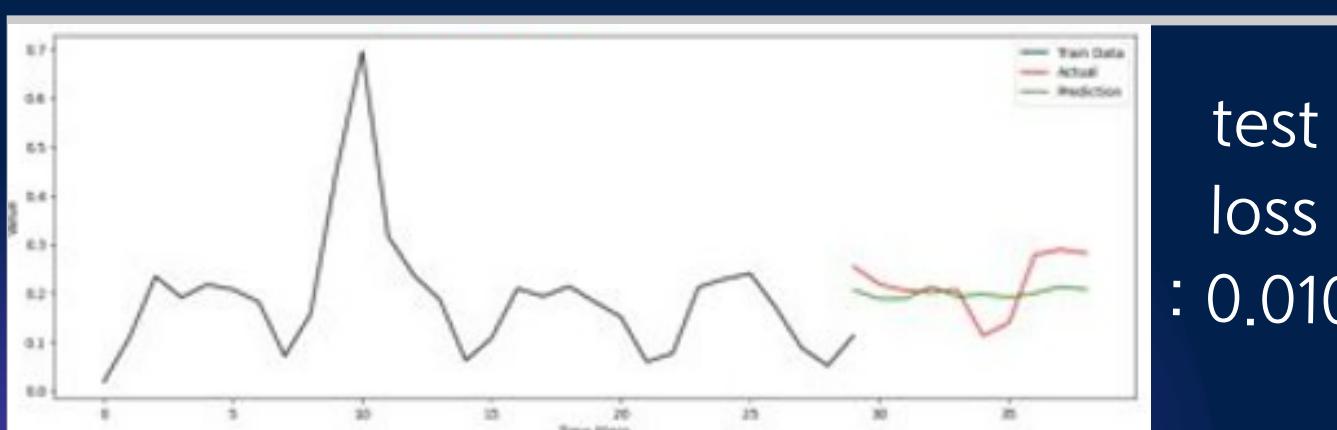
판매량



판매금액



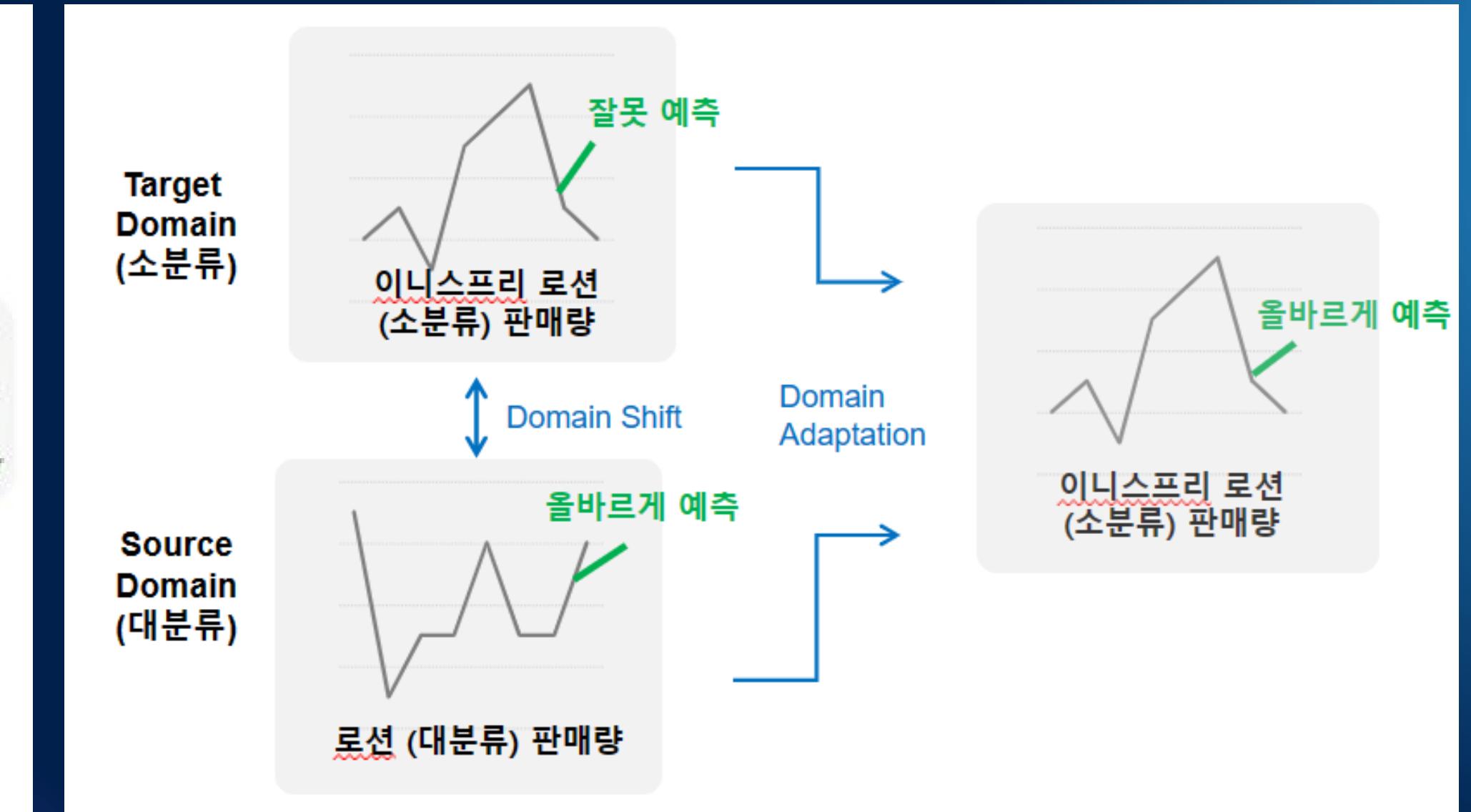
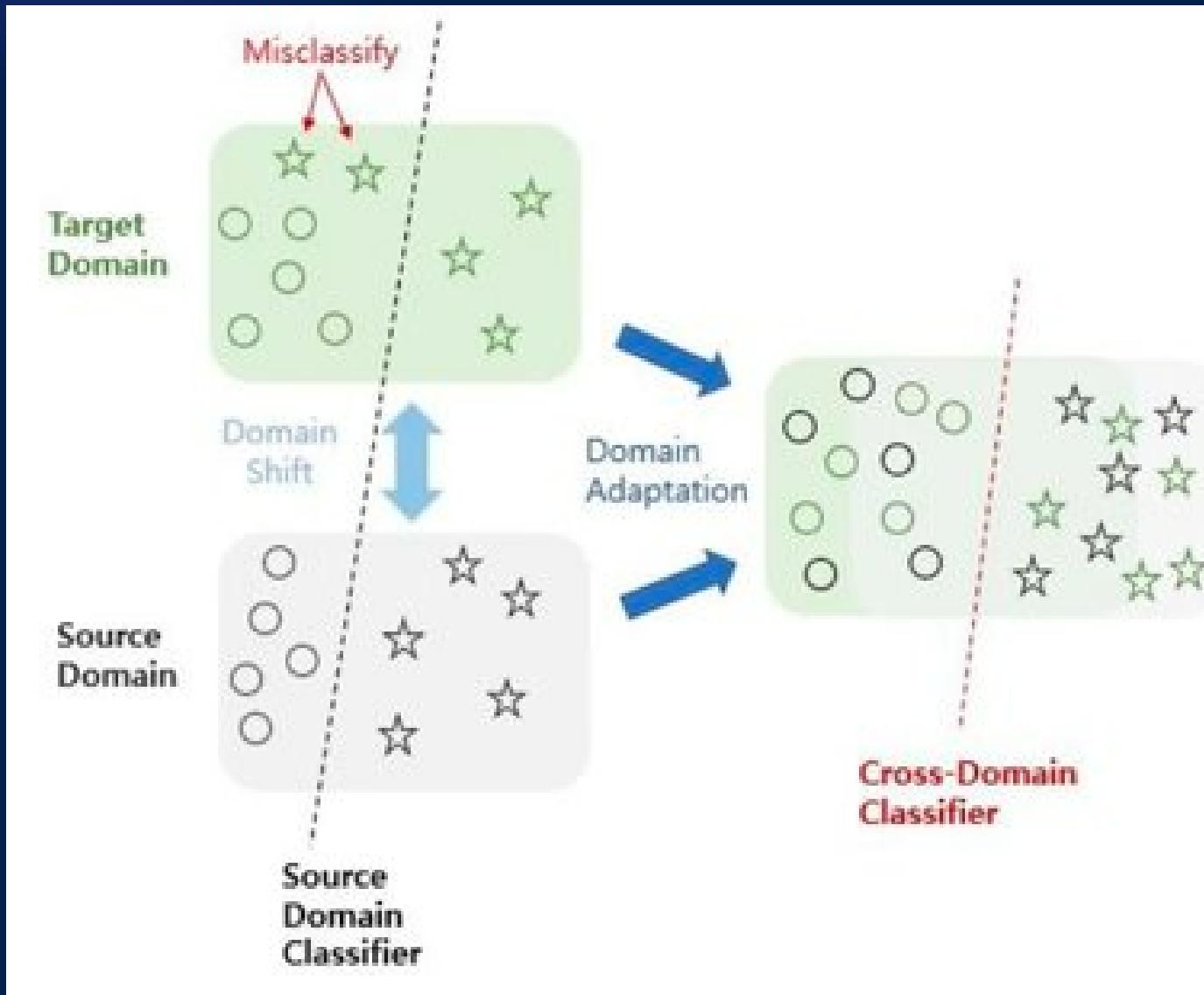
언급량



DA를 이용한 판매량 예측

DA(Domain Adaptation)란? :

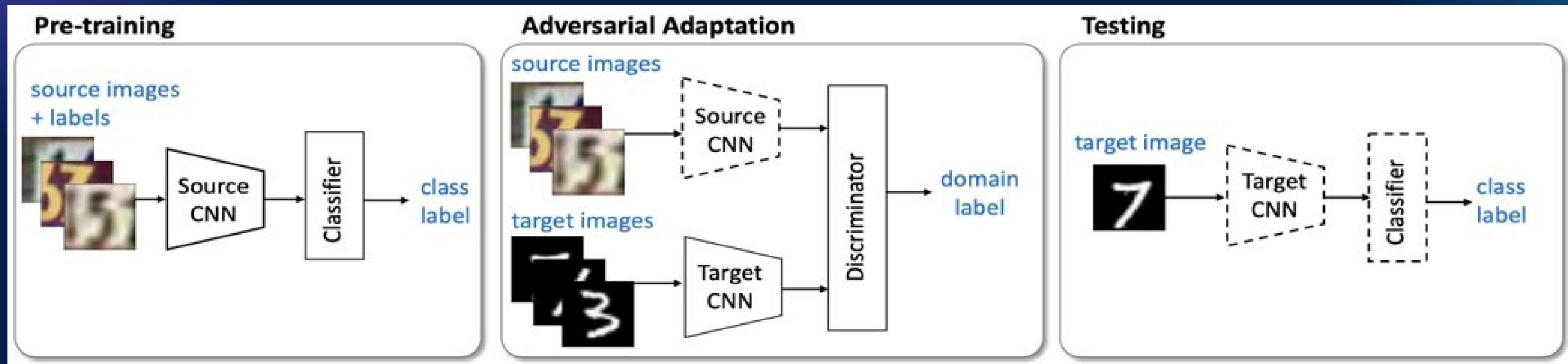
도메인이 다르지만 관련 있는 새로운 영역(distribution)에
기존 영역의 정보를 적응(adaptation) 시키는 것을 의미



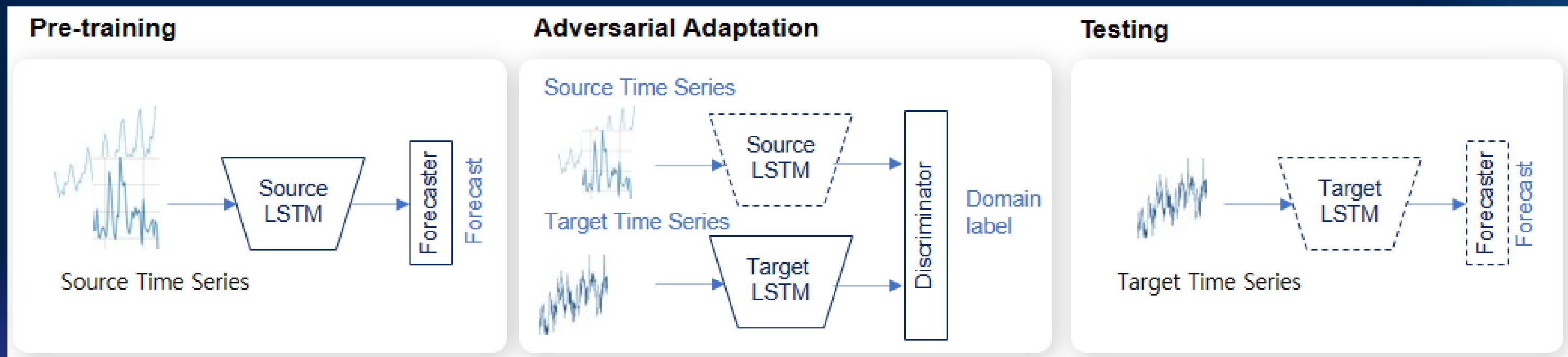
1) ADDA : Adversarial Discriminative Domain Adaptation 모델 변형

→ from image to time series

<기존 : image classification>



<변형 : time series forecasting>

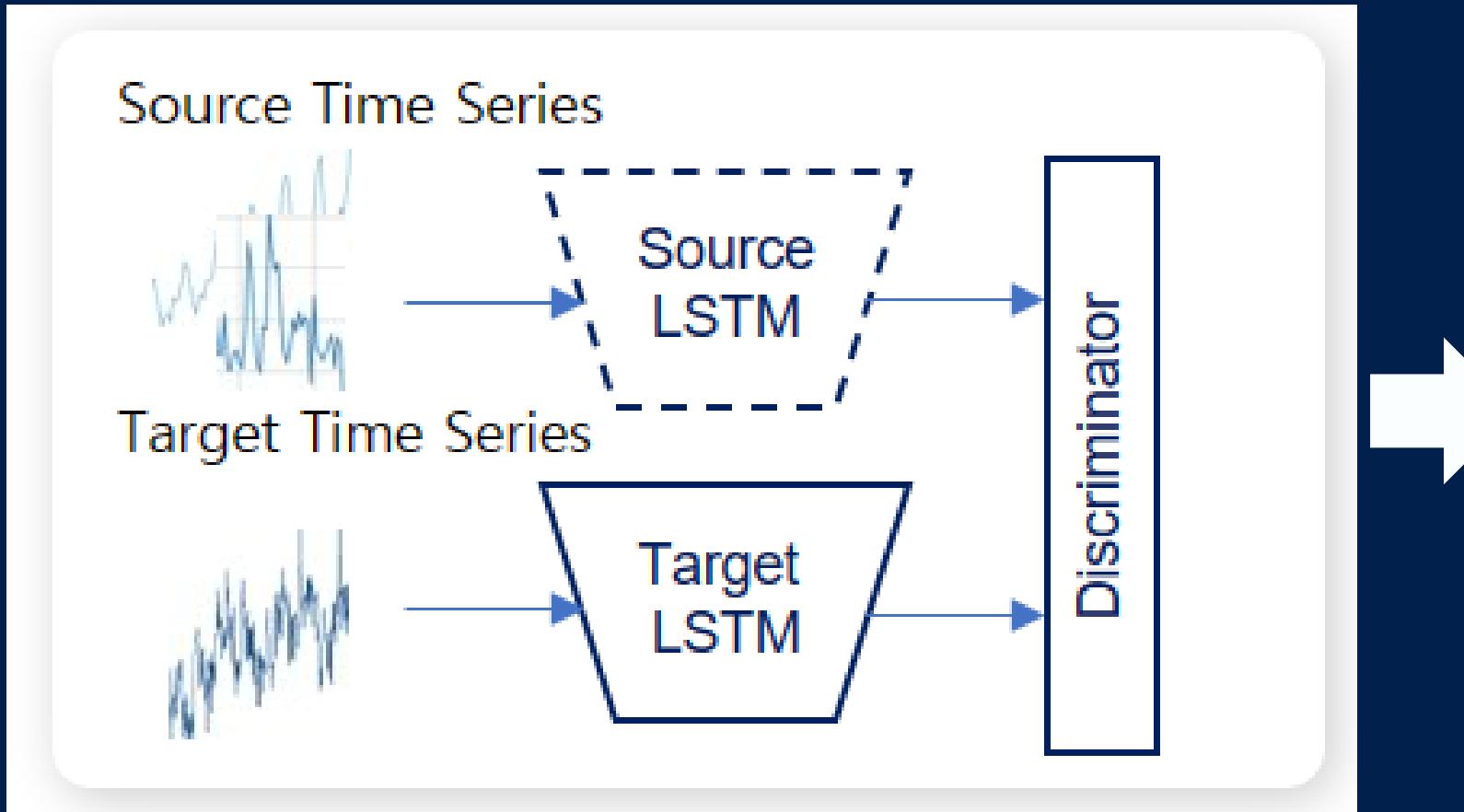


2) Reconstruction Error 추가

problem

Target 시계열의 정보를 활용하지 못하는 문제 발생

<기존 : time series ADDA>



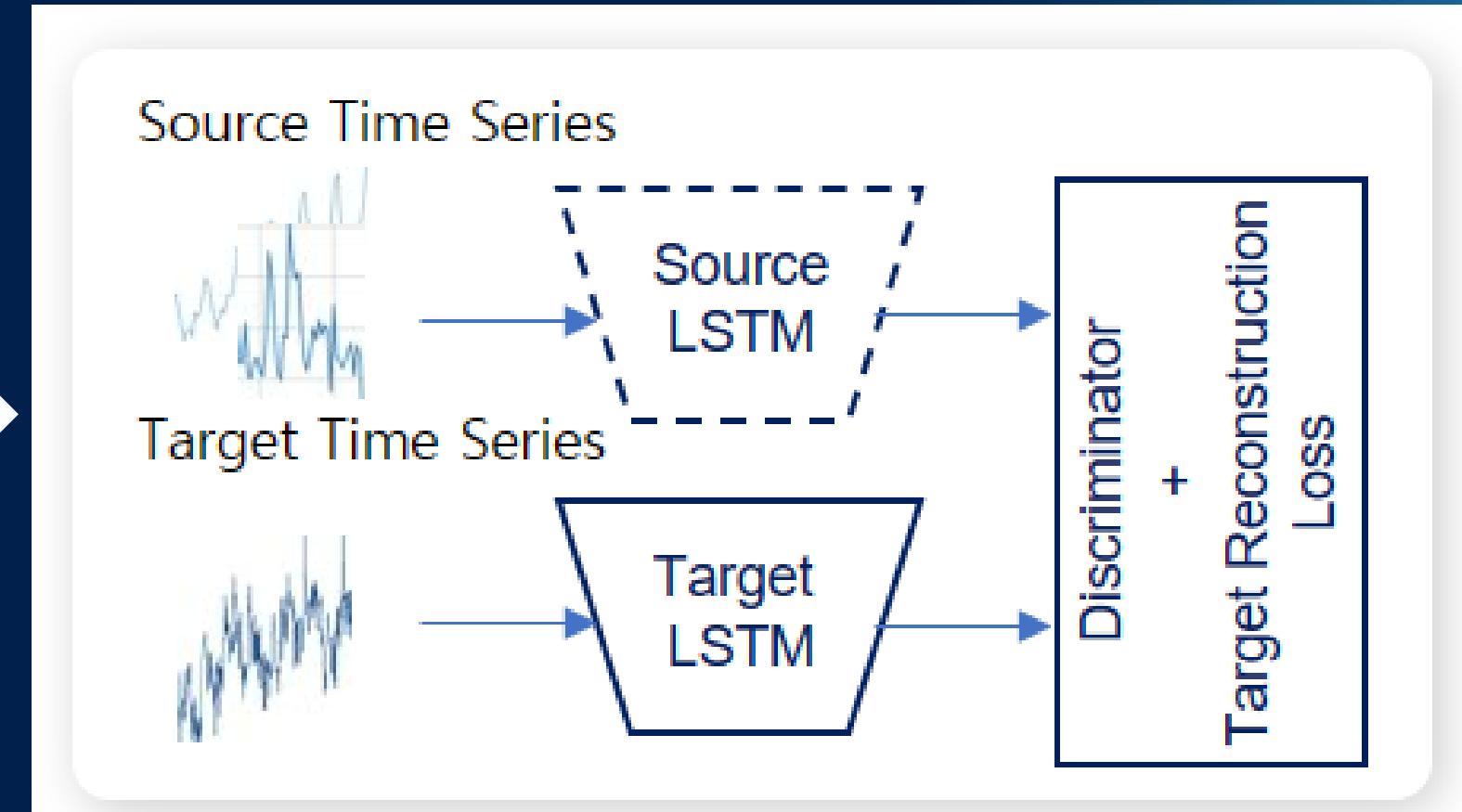
Source 훈련

→ GAN을 통해서만 target encoder 훈련
→ test

solution

시계열 데이터를 Reconstruction을 통해
target 시계열의 정보를 조금이라도 더 활용

<보완 : time series ADDA+Reconstruction Loss>



Source 훈련

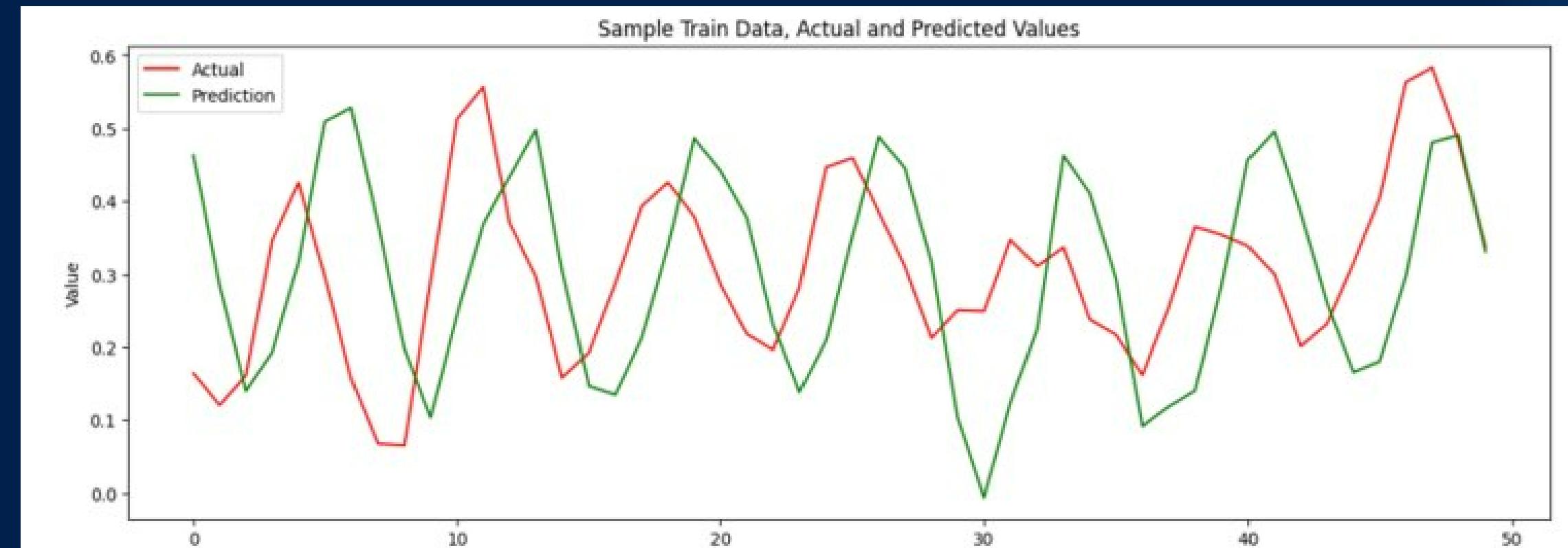
→ GAN + target 시계열의 reconstruction을 통한 훈련
→ test

3) DA 결과

Discriminator, Target Encoder, Source Encoder 동일

<Parameter>

- Window size : 100
 - (100일치 학습)
- Forecast size : 50
 - (50일치 예측)
- Learning rate : 0.001
- Epoch : 300
- Loss : MSELoss
- Optimizer : Adam



	DA 적용 x (source data만 활용)	ADDA	ADDA + Reconstruction Loss
대분류로 중분류 예측	0.499	0.464	0.439
대분류로 소분류 예측	0.257	0.255	0.243
중분류로 소분류 예측	0.417	0.401	0.400

3) DA 결과

problem

- source data에 대한 정보는 충분
 - 로션 판매량에 대한 정보는 충분히 확보하고 있음
- Target data에 대한 정보는 부족
 - 새롭게 팔기 시작한 이니스프리의 로션 판매량 데이터는 충분하지 않음

solution

- Test data의 절반만 가지고도 실험 진행
- 보다 도메인에 더 특화된 모델 구축 가능

	DA 적용 x (source data만 활용)	ADDA + Reconstruction Loss
대분류로 중분류 예측	0.346	0.347
대분류로 소분류 예측	0.302	0.279
중분류로 소분류 예측	0.423	0.410

결론 및 개선점

결론

- 일반적인 상품에 대해서는 여러 모델들의 훈련을 통해 판매량 예측
- DLinear이 일반적인 시계열 모델인 LSTM보다 효과적
- 데이터가 많이 없는 상품에 대해서는 Adversarial Loss에 기반한 Domain Adaptation 방법론을 시도하여, 정보가 많이 없는 상품에 대해서도 예측을 잘할 수 있도록 실험
- 실제로 Reconstruction Loss, Discriminator Loss를 통해 Target Encoder을 훈련시켰을 때, 조금 더 좋은 성능의 도메인에 특화 Target Encoder를 생성

**결론적으로 데이터 양이 충분한 상품, 부족한 상품 모두에 대한 예측치를 제공 가능
가게를 운영하는 소상공인에게 도움을 줄 수 있을 것으로 기대**

개선점

시계열 예측의 관점

- 더 다양한 요소(물가, 날씨 등)를 활용한 Multivariate Time series Forecasting 분석도 가능할 것
- SHAP, LIME 등 조금 더 해석 가능한 모델을 사용하여 조금 더 다채로운 예측을 할 수 있을 것

DA의 관점

- Domain Adaptation 문제를 시계열로 접근할 때 단순한 LSTM Encoder만 사용하여 진행한 것
- 계절성, 추세, 복잡한 시계열의 특성을 다양하게 반영할 수 있는 여타 모델 (CNN, Linear, Transformer 등)도 적용해볼 수 있을 것

감사합니다

제 17회 투빅스 컨퍼런스 시계열 1조