

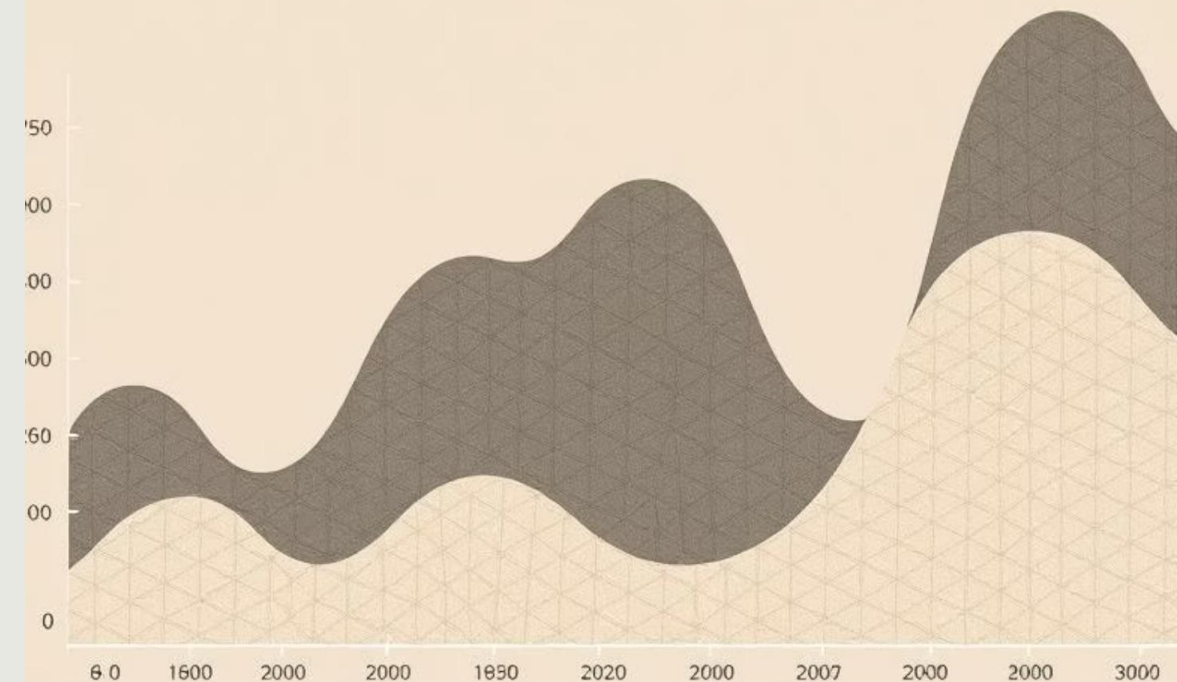
# 시계열 예측과 슬라이딩 윈도우

지도학습으로 바라보는 시계열 분석

과거 데이터의 패턴을 학습하여 미래를 예측하는 시계열  
분석을 지도학습의 관점에서 탐구

2026.01.21

ancestor9



# 시계열 예측이란?



## 시간의 흐름

시간에 따라 변화하는 데이터를 바탕으로 미래 값을 예측하는 문제입니다.



## 다양한 활용

기상 예보, 주가 예측, 기계 온도 모니터링 등 실생활의 모든 영역에서 활용됩니다.



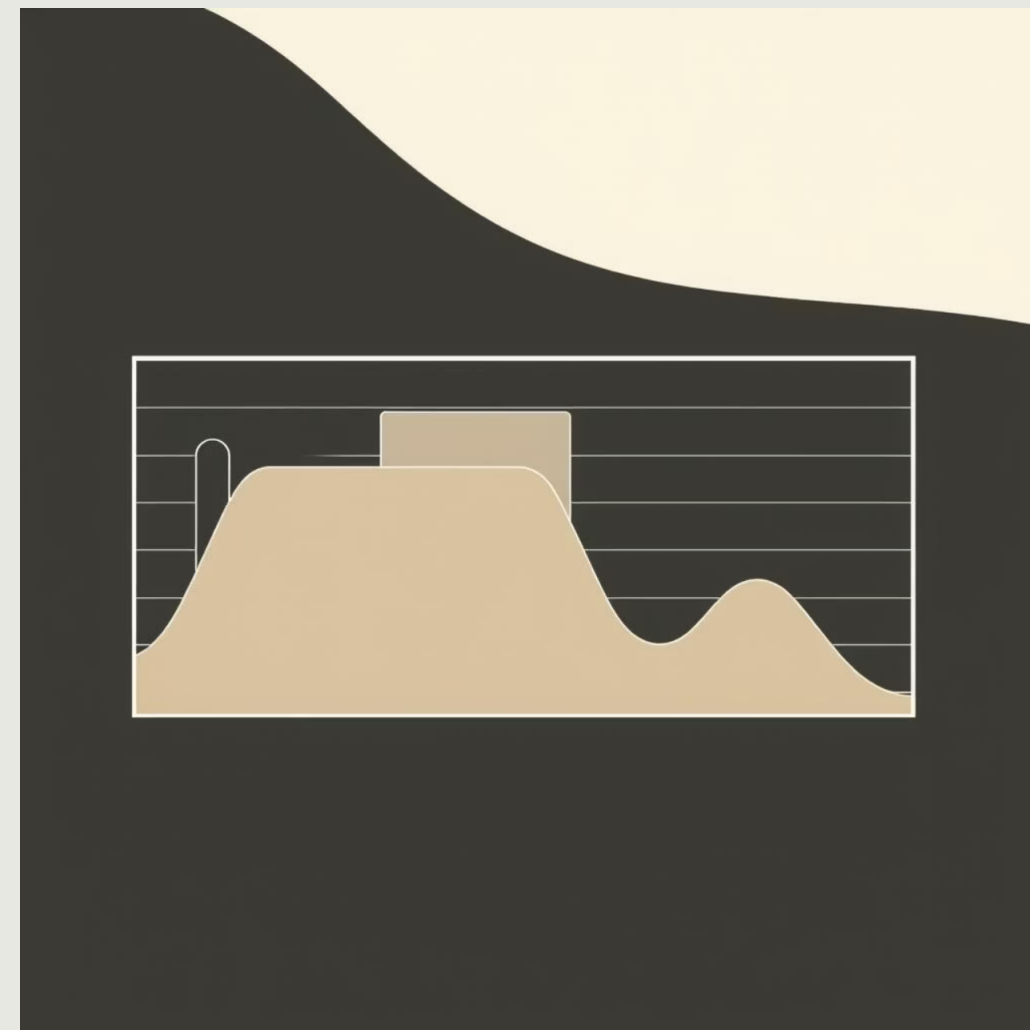
## 발전하는 기술

전통적 통계 모델부터 최신 머신러닝, 딥러닝까지 폭넓게 적용되고 있습니다.

# 시계열을 지도학습 문제로 재구성

시계열 데이터는 순차적 관측값들의 나열이지만, 슬라이딩 윈도우 기법을 통해 입력과 타겟 쌍을 생성할 수 있습니다.

과거 일정 구간의 데이터를 입력으로, 다음 시점 값을 타겟으로 설정하면 일반적인 머신러닝 모델에 적용이 가능해집니다.



# 시계열을 지도학습 문제로 재구성 - 1. 단변량

time,	measure
1,	100
2,	110
3,	108
4,	115
5,	120

Example of a small contrived  
time series dataset

$$Y = f(X)$$

X,	y
?,	100
100,	110
110,	108
108,	115
115,	120
120,	?

Example of time series dataset  
as supervised learning.

# 시계열을 지도학습 문제로 재구성 - 2. 다변량

time,	measure1,	measure2
1,	0.2,	88
2,	0.5,	89
3,	0.7,	87
4,	0.4,	88
5,	1.0,	90

$$Y = f(X)$$

X1,	X2,	X3,	y
?,	?,	0.2,	88
0.2,	88,	0.5,	89
0.5,	89,	0.7,	87
0.7,	87,	0.4,	88
0.4,	88,	1.0,	90
1.0,	90,	?,	?

Example of a small contrived multivariate time series dataset.

Example of a multivariate time series dataset as a supervised learning problem.

# 시계열을 지도학습 문제로 재구성 – 3. 단변량 Multistep

한 스텝 예측에서 더 나아가 여러 미래 시점을 동시에 예측

time	measure
1,	100
2,	110
3,	108
4,	115
5,	120

Example of a small contrived time series dataset.

$$Y = f(X)$$

X1,	y1,	y2
?	100,	110
100,	110,	108
110,	108,	115
108,	115,	120
115,	120,	?
120,	?,	?

Example of a univariate time series dataset as a multi-step or sequence prediction supervised learning problem.

# 슬라이딩 윈도우 기법 예시

단변량 시계열 데이터인 온도 데이터를 예로 살펴보겠습니다:  $[x_1, x_2, x_3, x_4, x_5, x_6, x_7]$

1

$[x_1, x_2, x_3] \rightarrow x_4$

첫 번째 입력-출력 쌍

2

$[x_2, x_3, x_4] \rightarrow x_5$

두 번째 입력-출력 쌍

3

$[x_3, x_4, x_5] \rightarrow x_6$

세 번째 입력-출력 쌍

4

$[x_4, x_5, x_6] \rightarrow x_7$

네 번째 입력-출력 쌍



**i** 윈도우 크기 3일 때: 과거 3일 데이터로 다음 날 온도를 예측

# 데이터 윈도우링과 학습 샘플 생성

시계열 길이  $T$

전체 데이터의 길이 = 10

윈도우 크기  $w$

입력으로 사용할 과거 시점 개수 = 3

예측 지평선  $\tau$

미래로 내다볼 시점 개수 = 1

총 학습 샘플 수:  $N = T - w - \tau + 1$

날짜	값
1월1일	1
1월2일	2
1월3일	3
1월4일	4
1월5일	5
1월6일	6
1월7일	7
1월8일	8
1월9일	9
1월10일	10

전처리된  
학습 샘플

$N = T - w - \tau + 1 = 10 - 3 - 1 + 1 = 7$

전처리된 학습 샘플		
샘플 번호	입력 $X$ (과거 $w$ 일)	출력 $Y$ (예측 $\tau$ 일 후)
1	(1월1일, 1월2일, 1월3일 → [1,2,3])	1월4일 → 4
2	(1월2일, 1월3일, 1월4일 → [2,3,4])	1월5일 → 5
3	(1월3일, 1월4일, 1월5일 → [3,4,5])	1월6일 → 6
4	(1월4일, 1월5일, 1월6일 → [4,5,6])	1월7일 → 7
5	(1월5일, 1월6일, 1월7일 → [5,6,7])	1월8일 → 8
6	(1월6일, 1월7일, 1월8일 → [6,7,8])	1월9일 → 9
7	(1월7일, 1월8일, 1월9일 → [7,8,9])	1월10일 → 10



# 데이터 윈도우링과 학습 샘플 생성

시계열 길이  $T$

전체 데이터의 길이 = 10

윈도우 크기  $w$

입력으로 사용할 과거 시점 개수 = 3

예측 지평선  $\tau$

미래로 내다볼 시점 개수 = 1

총 학습 샘플 수:  $N = T - w - \tau + 1$

날짜	값
1월1일	1
1월2일	2
1월3일	3
1월4일	4
1월5일	5
1월6일	6
1월7일	7
1월8일	8
1월9일	9
1월10일	10

전처리된  
학습 샘플



$N = T - w - \tau + 1 = 10 - 3 - 2 + 1 = 6$

샘플 번호	입력 $X$ (과거 $w$ 일)	출력 $Y$ (예측 $\tau$ 일 후, 이틀 뒤)
1	(1월1일, 1월2일, 1월3일 $\rightarrow$ [1,2,3])	1월5일 $\rightarrow$ 5
2	(1월2일, 1월3일, 1월4일 $\rightarrow$ [2,3,4])	1월6일 $\rightarrow$ 6
3	(1월3일, 1월4일, 1월5일 $\rightarrow$ [3,4,5])	1월7일 $\rightarrow$ 7
4	(1월4일, 1월5일, 1월6일 $\rightarrow$ [4,5,6])	1월8일 $\rightarrow$ 8
5	(1월5일, 1월6일, 1월7일 $\rightarrow$ [5,6,7])	1월9일 $\rightarrow$ 9
6	(1월6일, 1월7일, 1월8일 $\rightarrow$ [6,7,8])	1월10일 $\rightarrow$ 10

# 실습 : 삼양식품

시계열 길이  $T = 406$

```
# 삼양식품 (003230) 최근 3개월 주가
# pip install pykrx -q
from pykrx import stock
import pandas as pd
from datetime import datetime

# 종목 코드와 기간 설정
ticker = "003230" # 삼양식품 종목 코드
start_date = "20240101" # 시작 날짜 (YYYYMMDD)
end_date = "20250901" # 종료 날짜 (YYYYMMDD)

# 주가 데이터 가져오기 (시가, 고가, 저가, 종가, 거래량)
df = stock.get_market_ohlcw_by_date(fromdate=start_date,
to date=end_date, ticker=ticker)

# 데이터 확인
df.head()

# # 추가: 종목명 확인
# stock_name = stock.get_market_ticker_name(ticker)
# print(f"종목명: {stock_name}")
```

날짜	시가	고가	저가	종가	거래량	등락률
2024-01-02	221000	239500	220500	234500	236800	8.564815
2024-01-03	233000	233000	213500	217500	157348	-7.249467
2024-01-04	217000	222500	215500	220000	72555	1.149425
2024-01-05	218500	220000	209500	210500	122857	-4.318182
2024-01-08	212000	219000	211000	213500	79303	1.425178
...	...	...	...	...	...	...
2025-08-26	1429000	1505000	1415000	1501000	71888	4.891684
2025-08-27	1493000	1514000	1480000	1512000	50727	0.732845
2025-08-28	1511000	1642000	1510000	1597000	101756	5.621693
2025-08-29	1626000	1626000	1563000	1566000	33383	-1.941140
2025-09-01	1584000	1590000	1555000	1567000	24970	0.063857

406 rows × 6 columns

# 실습 : 삼양식품

윈도우 크기  $w$

입력으로 사용할 과거 시점 개수 = 5

Shift 크기  $s$

과거 데이터에서 건너 뛴 개수 = 2

예측 지평선  $\tau$

미래로 내다볼 시점 개수 = 2개

	시가	고가	저가	종가	거래량	등락률
날짜						
2024-01-02	221000	239500	220500	234500	236800	8.564815
2024-01-03	233000	233000	213500	217500	157348	-7.249467
2024-01-04	217000	222500	215500	220000	72555	1.149425
2024-01-05	218500	220000	209500	210500	122857	-4.318182
2024-01-08	212000	219000	211000	213500	79303	1.425178
2024-01-09	214000	217500	208000	215000	86674	0.702576
2024-01-10	215000	220000	212500	215000	53653	0.000000
2024-01-11	214500	215000	208500	211000	62178	-1.860465
2024-01-12	212500	212500	207500	209500	34035	-0.710900
2024-01-15	208500	211000	205000	205000	45261	-2.147971
2024-01-16	204500	208000	202000	203000	36721	-0.975610
2024-01-17	203000	204500	196500	199700	55644	-1.625616
2024-01-18	197000	199500	193700	195800	90344	-1.952929
2024-01-19	196100	199200	192600	193800	53712	-1.021450
2024-01-22	193800	199200	190300	197900	71673	2.115583

Rolling

2일 예측 데이터:

입력 데이터 크기: (398, 5, 6)

출력 데이터 크기: (398, 2)

총 샘플 수: 398

2day 예측 샘플:

샘플 1:

입력 시퀀스 (종가들): [234500, 217500, 220000, 210500, 213500.]

예측 타겟 (2일간 종가): [211000, 209500.]

샘플 2:

입력 시퀀스 (종가들): [217500, 220000, 210500, 213500, 215000.]

예측 타겟 (2일간 종가): [209500, 205000.]

샘플 3:

입력 시퀀스 (종가들): [220000, 210500, 213500, 215000, 215000.]

예측 타겟 (2일간 종가): [205000, 203000.]

# 실습 : 삼양식품

윈도우 크기  $w$

입력으로 사용할 과거 시점 개수 = 5

Shift 크기  $s$

과거 데이터에서 건너 뛴 개수 = 2

예측 지평선  $\tau$

미래로 내다볼 시점 개수 = 3개

	시가	고가	저가	종가	거래량	등락률
날짜						
2024-01-02	221000	239500	220500	234500	236800	8.564815
2024-01-03	233000	233000	213500	217500	157348	-7.249467
2024-01-04	217000	222500	215500	220000	72555	1.149425
2024-01-05	218500	220000	209500	210500	122857	-4.318182
2024-01-08	212000	219000	211000	213500	79303	1.425178
2024-01-09	214000	217500	208000	215000	86674	0.702576
2024-01-10	215000	220000	212500	215000	53653	0.000000
2024-01-11	214500	215000	208500	211000	62178	-1.860465
2024-01-12	212500	212500	207500	209500	34035	-0.710900
2024-01-15	208500	211000	205000	205000	45261	-2.147971
2024-01-16	204500	208000	202000	203000	36721	-0.975610
2024-01-17	203000	204500	196500	199700	55644	-1.625616
2024-01-18	197000	199500	193700	195800	90344	-1.952929
2024-01-19	196100	199200	192600	193800	53712	-1.021450
2024-01-22	193800	199200	190300	197900	71673	2.115583

Rolling

3day 분할 결과:

학습 세트: 316 샘플

테스트 세트: 80 샘플

3day 예측 샘플:

샘플 1:

입력 시퀀스 (종가들): [234500, 217500, 220000, 210500, 213500.]

예측 타겟 (3일간 종가): [209500, 205000, 203000.]

샘플 2:

입력 시퀀스 (종가들): [217500, 220000, 210500, 213500, 215000.]

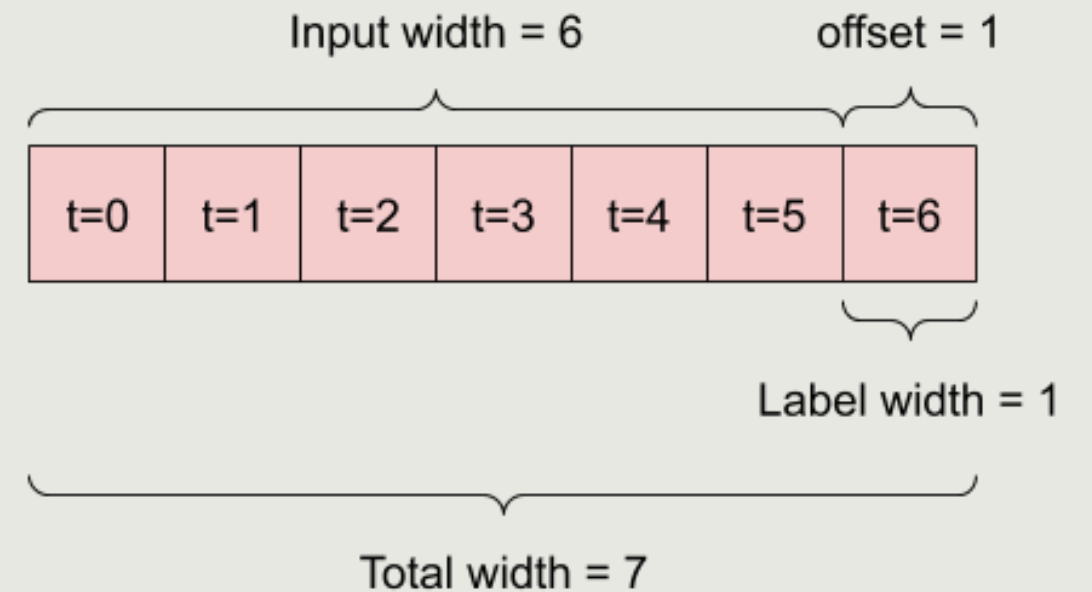
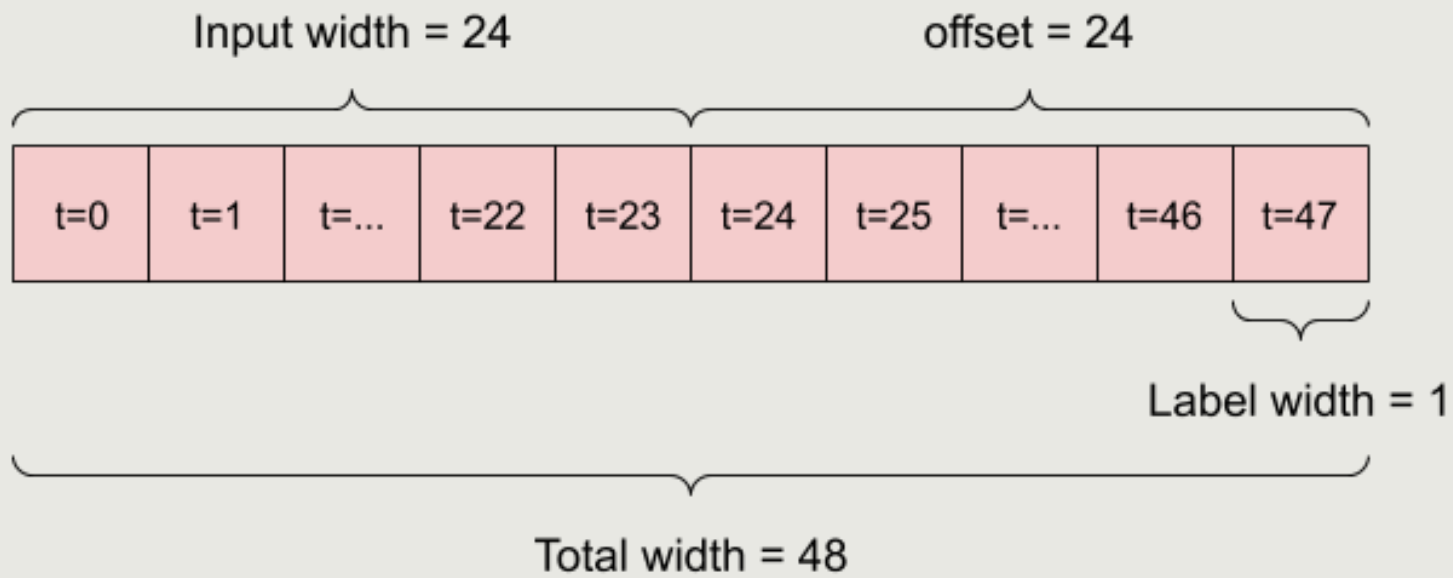
예측 타겟 (3일간 종가): [205000, 203000, 199700.]

샘플 3:

입력 시퀀스 (종가들): [220000, 210500, 213500, 215000, 215000.]

예측 타겟 (3일간 종가): [203000, 199700, 195800.]

# 실습 : Python 프로그래밍

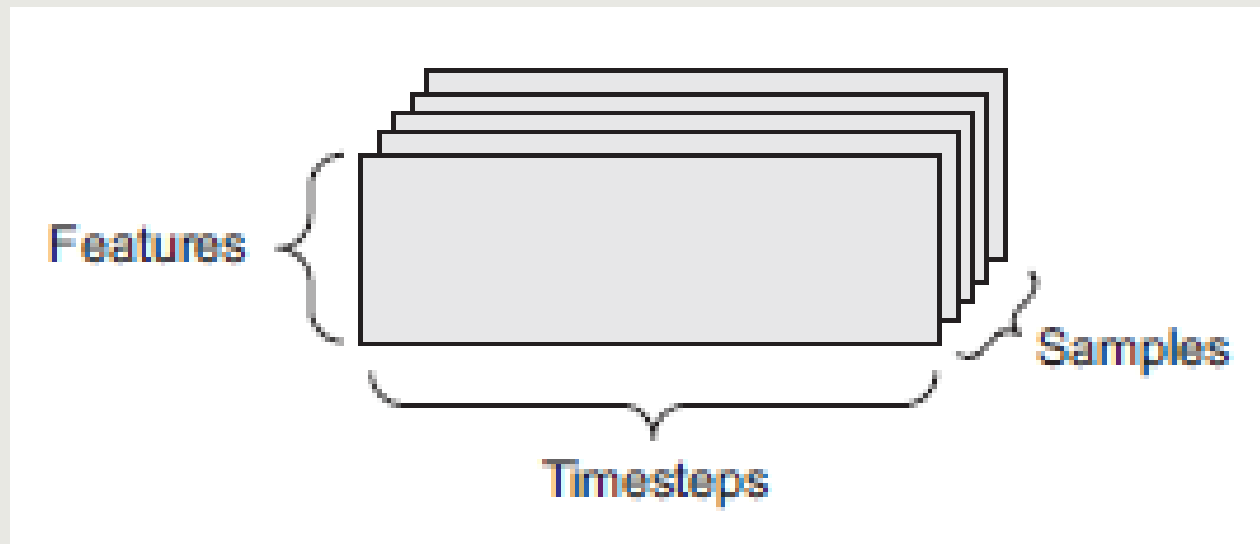


[https://www.tensorflow.org/tutorials/structured\\_data/time\\_series?hl=ko](https://www.tensorflow.org/tutorials/structured_data/time_series?hl=ko)

# 3D tensor : Timeseries data or sequence data

시계열자료의 시간 축은 관례적으로 항상 두 번째 축(인덱스 1)으로 설정

- 데이터는 시간 축(두 번째 축)을 기준으로 텐서 형태로 구조화되며, 샘플 단위(예: 하루 또는 한 트윗)로 저장



# 3D tensor : Timeseries data or sequence data

## 1. 주식 가격 데이터셋:

- 매분마다 주식의 현재 가격, 지난 1분간의 최고 가격, 최저 가격을 저장.
- 1분은 3차원 벡터로 표현되고, 하루 거래 시간(390분)은 (390, 3) 형태의 2D 텐서로 저장.
- 250일치 데이터는 (250, 390, 3) 형태의 3D 텐서로 저장.
- 여기서 각 샘플은 하루치 데이터에 해당.

## 2. 트윗 데이터셋:

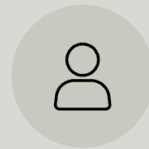
- 각 트윗은 128개 고유 문자로 구성된 280자 시퀀스로 인코딩.
- 각 문자는 128 크기의 이진 벡터로 표현(해당 문자 인덱스에 1, 나머지는 0).
- 하나의 트윗은 (280, 128) 형태의 2D 텐서로 저장.
- 100만 개의 트윗은 (1000000, 280, 128) 형태의 텐서로 저장.

# 슬라이딩 윈도우 기법의 장점



## 패턴 모델링

시계열의 자기상관 구조를 효과적으로 학습하고 모델링할 수 있습니다.



## 자기 지도 학습

외부 라벨 없이도 과거 데이터만으로 학습이 가능합니다.



## 광범위한 활용

ARIMA부터 LSTM, GRU까지 다양한 알고리즘에 적용 가능합니다.



## 유연한 적용

단변량, 다변량, 단일/다중 스텝 모든 상황에 대응합니다.



# 결론

## 슬라이딩 윈도우와 감독학습

### 강력한 변환 방법

시계열 데이터를 감독학습 문제로 변환하는 효과적인 접근법입니다.

### 유연한 대응

다양한 변수와 예측 범위에 맞춰 적용할 수 있는 범용성을 제공합니다.

### 성능 향상 기대

머신러닝 모델 적용을 통해 예측 정확도 개선이 기대됩니다.

여러분의 데이터에 맞는 **윈도우 크기와 예측 지평선** 설정이 성공의 핵심입니다!