

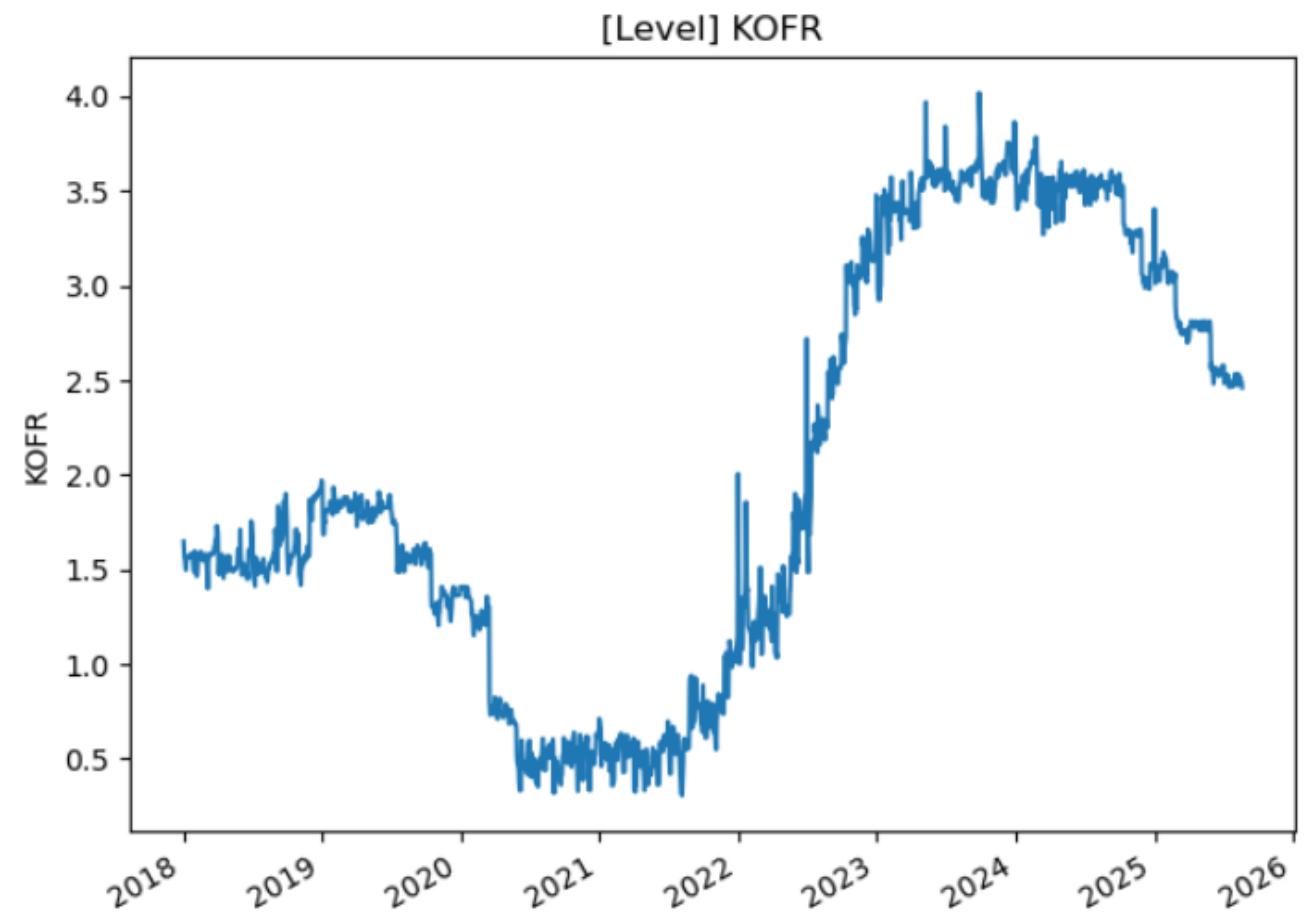
시계열 모델을 이용한 KOFR 예측

FIND- $\alpha$  11기 Team Gamma  
국준호 김산성 이수화 최정록

# 주제 선정

## KOFR 금리란?

- 한국 무위험 지표금리(Korean Overnight Financing Rate)
- 금융 시장의 중요한 지표



# 통계학적 접근: ARMA 모형

## AR(1) 모형

하루 전의 데이터가 당일 데이터에만 영향을 미친다고 가정하는 모형으로, OLS나 MLE 방법을 통해 회귀식을 만들 수 있음

## ARMA(1,1) 모형

AR 모델에 비해 잔차 충격에 대한 추가 회귀식이 포함된 보다 복잡한 모델

## AR(p) 모형

$$Y_t = c + \phi_1 Y_{t-1} + \phi_2 Y_{t-2} + \cdots + \phi_p Y_{t-p} + \varepsilon_t$$

## MA(q)모형

$$Y_t = \mu + \varepsilon_t + \theta_1 \varepsilon_{t-1} + \theta_2 \varepsilon_{t-2} + \cdots + \theta_q \varepsilon_{t-q}$$

## 금융공학적 접근: Vasicek 모형

### Vasicek 모형

금리의 움직임은 드리프트항(추세항)과 랜덤 항으로 나눌 수 있으며, 이산화하면 AR(1) 모형과 동일한 형태를 가짐.

Vasicek 모형을 추정하기 위해서는 통계학의 AR(1)을 활용할 수밖에 없음

→ 차이점: 회귀식에서 변동성, 장기 평균, 회귀속도를 역으로 추정하는 것

$$dr_t = \kappa(\theta - r_t)dt + \sigma dW_t$$

$$r_t = \alpha + \beta r_{t-1} + \epsilon_t$$



$$\bullet \quad \alpha = \kappa\theta\Delta t \quad \bullet \quad \beta = (1 - \kappa\Delta t) \quad \bullet \quad \epsilon_t = \sigma\sqrt{\Delta t}Z_t$$

## 두 접근법의 한계와 차이점

### Vasicek 모형 사용의 문제점

- 시계열 데이터의 특성을 무시
- AR(1) 모델을 활용해 무의미한 추정식 생성
- 다른 통계 모델을 사용 시 파라미터 추정이 의미 없게 됨

### 금융공학의 제약

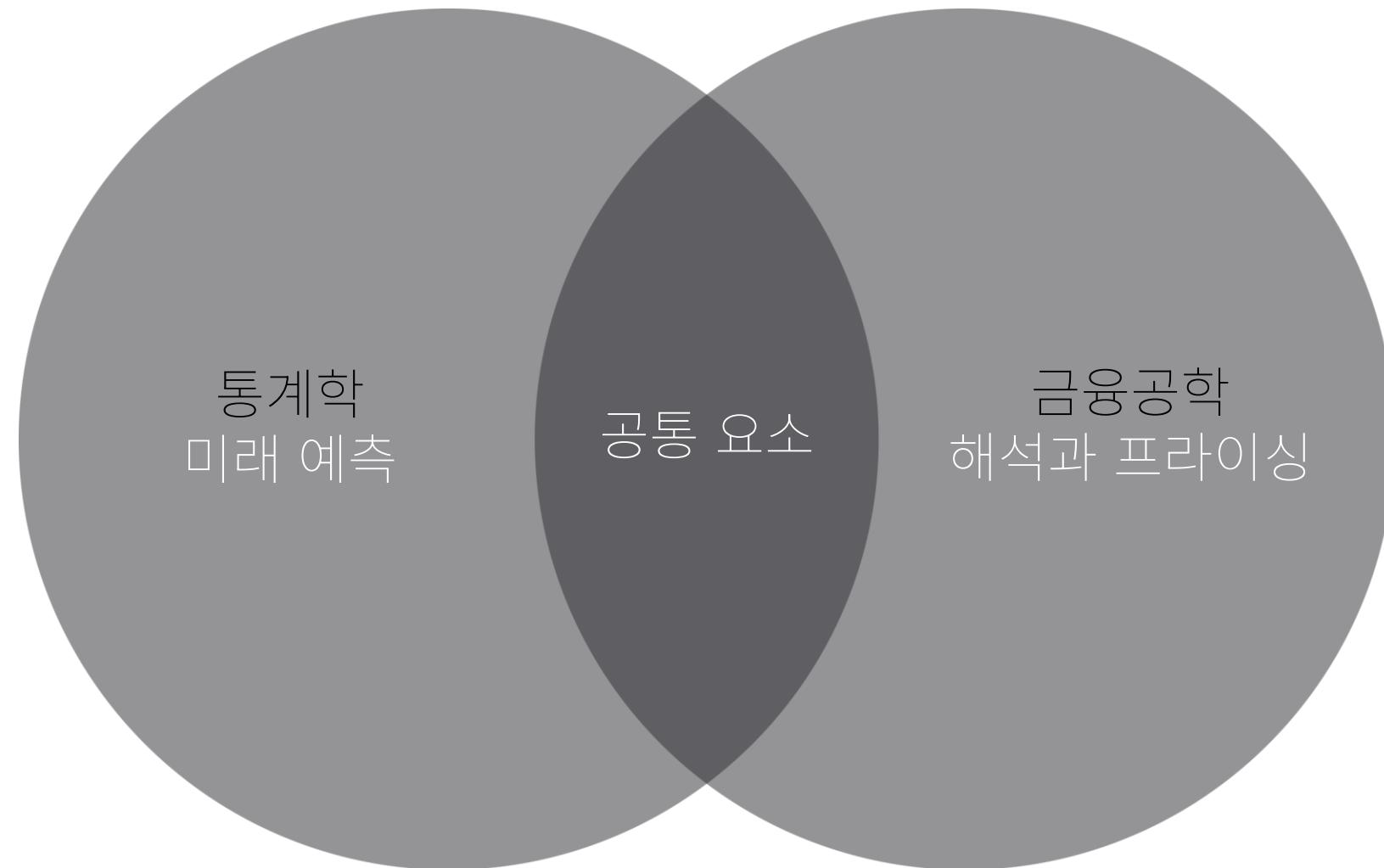
- 통계학에 비해 가정과 제약이 많음
- 모든 시계열 데이터는 각각의 특성이 다르므로 하나의 금융공학 이론으로 묶을 수 없음

### 금융공학의 장점

- 통계학에서 밝히지 못하는 데이터 해석 능력
- 장기평균, 변동성, 회귀 속도 등 사람이 해석할 수 있는 파라미터 제공

금융공학은 '차익거래 불능'이라는 명제를 활용하여 통계적 한계를 극복

## 결론: 예측과 해석의 균형



# 데이터 정상성 검정

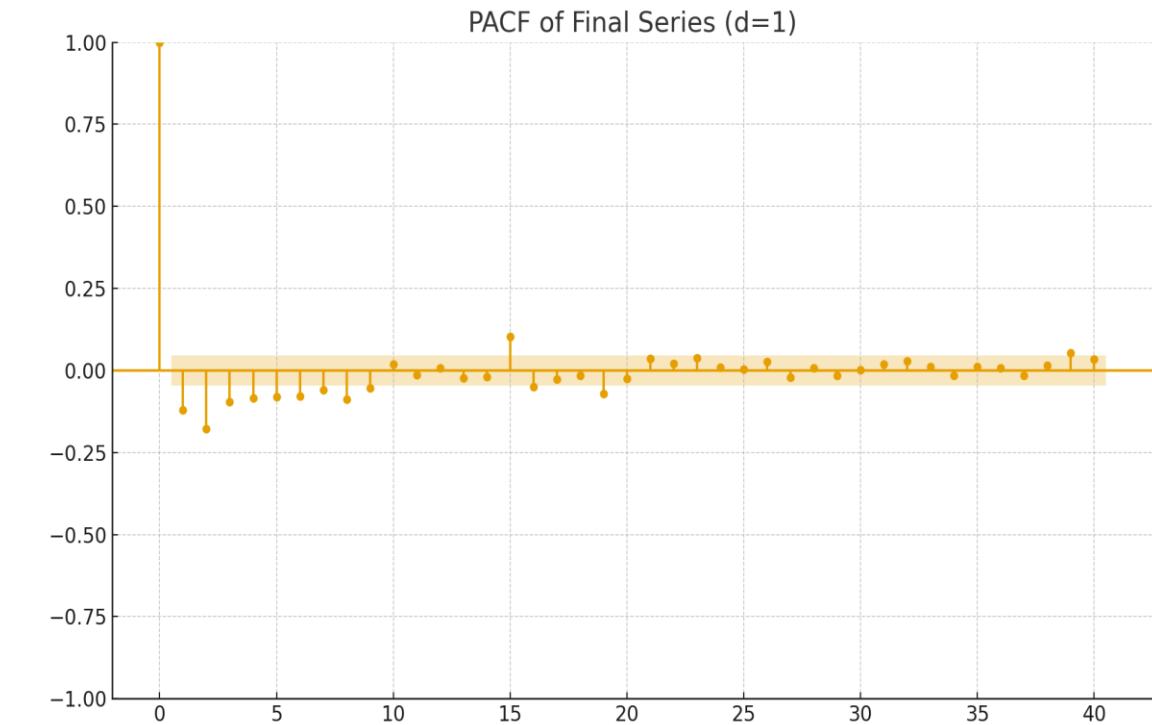
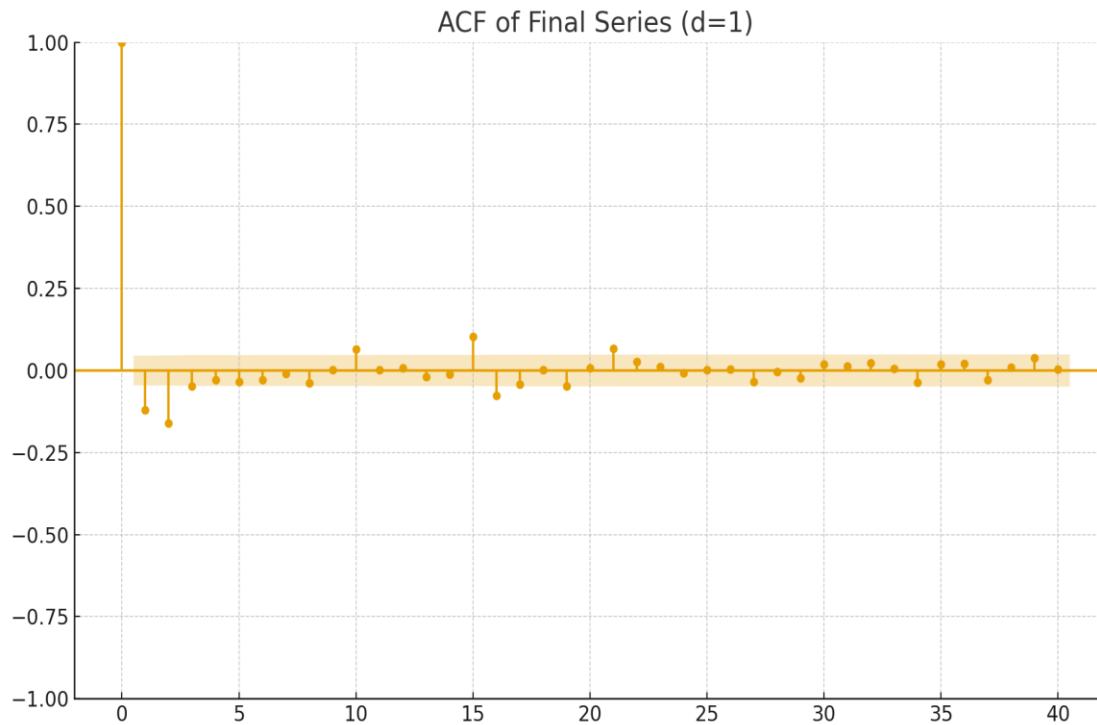
원자료 ADF 검정

p-value = 0.8715로 단위근 존재, 비정상 시계열

1차 차분 데이터 ADF 검정

p-value = 0.0000015로 정상성 확보

ACF, PACF 모두 시차 1시점에서 빠른 감쇠가 이루어짐. 이는 ARIMA 모델 (1,1,1)이 적합 가능성을 예상.



# ARIMA 모델 선택 및 학습

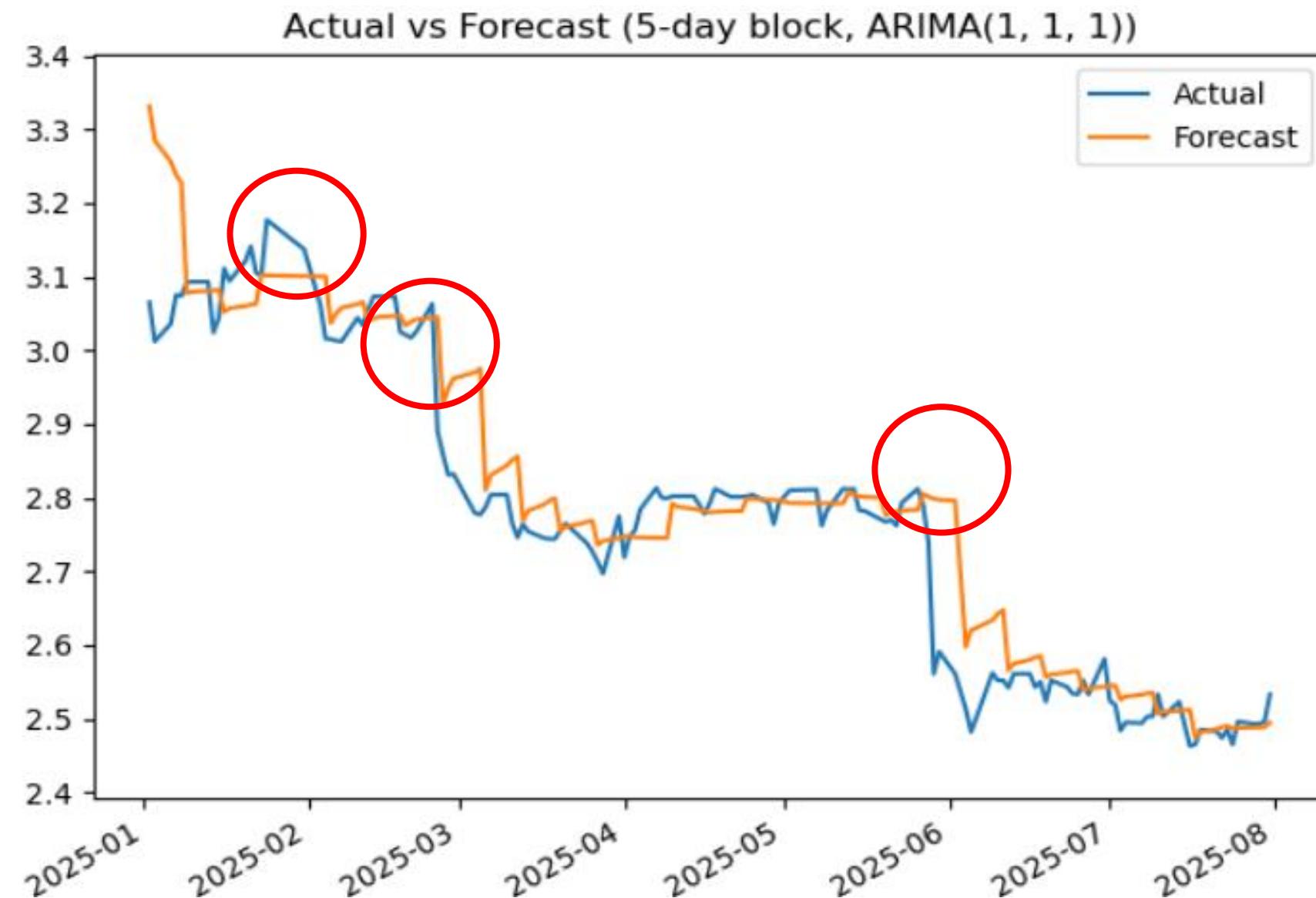
AIC, BIC 지표를 비교: 파라미터가 가장 적고 BIC가 가장 높은 ARIMA(1,1,1) 모델을 선택

p	d	q		AIC	BIC	k
0	1	1	1	-3865.509600	-3849.150653	3
1	2	1	1	-3870.859191	-3849.047262	4
2	1	1	2	-3868.937758	-3847.128148	4
3	2	1	2	-3867.777437	-3840.515424	5
4	3	1	1	-3866.603344	-3839.341332	5
5	1	1	3	-3864.533801	-3837.274690	5
6	0	1	3	-3855.801378	-3833.994089	4
7	3	1	2	-3865.646380	-3832.931965	6
8	0	1	4	-3858.465285	-3831.209077	5
9	2	1	3	-3862.804163	-3830.093229	6
10	4	1	1	-3862.651648	-3829.940715	6
11	1	1	4	-3860.104441	-3827.396991	6
12	0	1	2	-3842.268976	-3825.911769	3
13	2	1	4	-3861.530092	-3823.371400	7
14	3	1	3	-3861.057838	-3822.895083	7

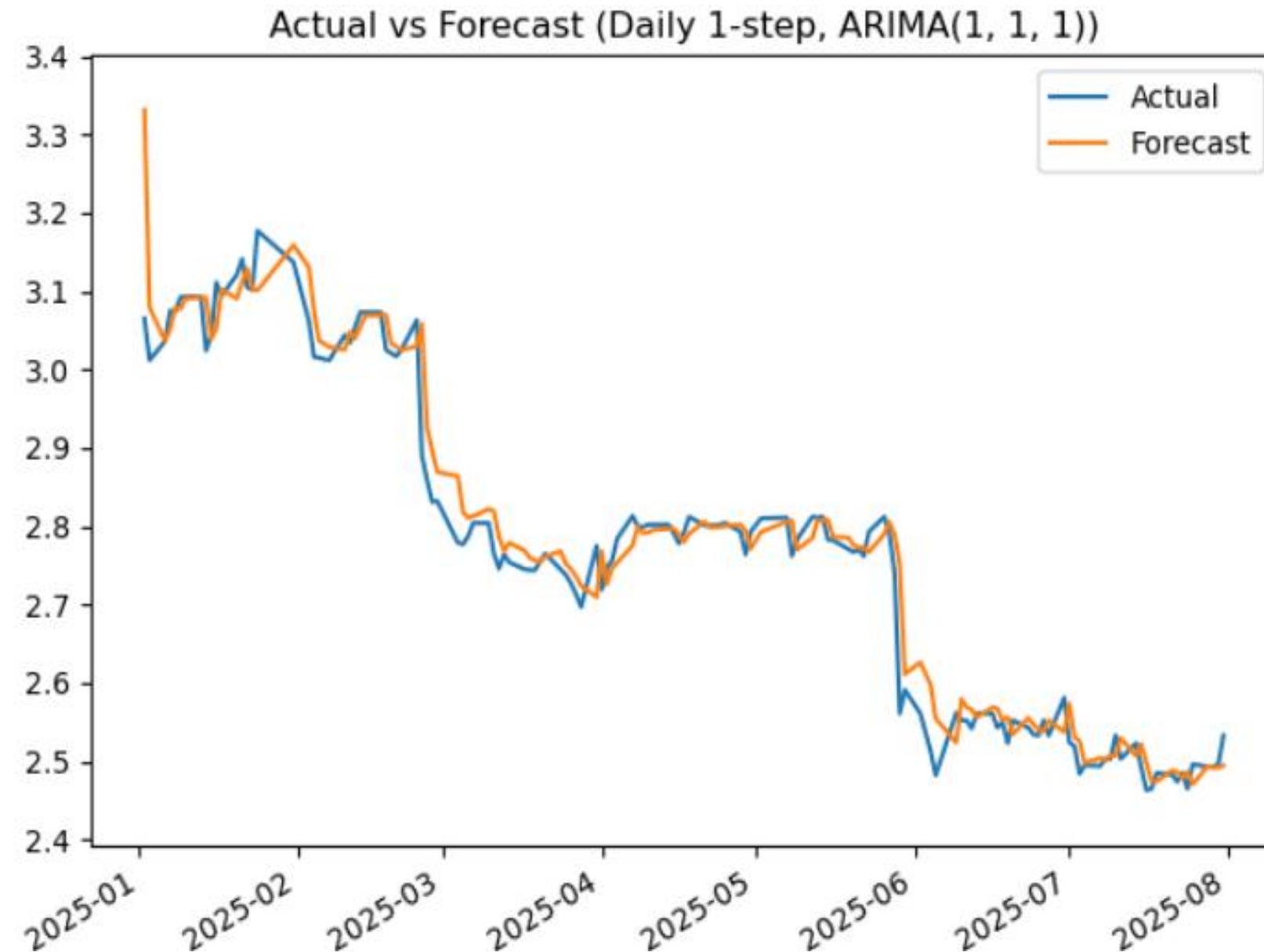
$$\Delta Y_t = 0.6200 \cdot \Delta Y_{t-1} - 0.8466 \cdot \epsilon_{t-1} + \epsilon_t$$

=====	=====	=====	=====	=====	=====	=====
Dep. Variable:	KOFR	No. Observations:	1728			
Model:	ARIMA(1, 1, 1)	Log Likelihood:	1935.755			
Date:	Mon, 01 Sep 2025	AIC:	-3865.510			
Time:	15:47:59	BIC:	-3849.151			
Sample:	0	HQIC:	-3859.458			
	- 1728					
Covariance Type:	opg					
=====	=====	=====	=====	=====	=====	=====
	coef	std err	z	P> z	[0.025	0.975]
-----	-----	-----	-----	-----	-----	-----
ar.L1	0.6200	0.028	22.081	0.000	0.565	0.675
ma.L1	-0.8466	0.023	-36.733	0.000	-0.892	-0.801
sigma2	0.0062	6.22e-05	99.781	0.000	0.006	0.006
=====	=====	=====	=====	=====	=====	=====
Ljung-Box (L1) (Q):	2.19	Jarque-Bera (JB):	64032.73			
Prob(Q):	0.14	Prob(JB):	0.00			
Heteroskedasticity (H):	2.04	Skew:	0.97			
Prob(H) (two-sided):	0.00	Kurtosis:	32.78			
=====	=====	=====	=====	=====	=====	=====

# 5일 예측 및 재학습 시스템 결과



# 하루 예측-재학습 시스템 결과 및 결론



# 결론

	Scheme	MAE	RMSE	OOS_R2_vs_RW	Theil_U2	MASE	sMAPE	Directional_Accuracy	n_obs
0	5-day block	0.044774	0.071199	-1.567454	1.602328	1.063886	0.015802	0.428571	141
1	Daily 1-step	0.025979	0.042640	0.079117	0.959626	0.617304	0.009279	0.435714	141

# ARX 모델 데이터 및 전처리

## 기간 설정

학습: 2018-01-02 ~ 2024-12-31

검증: 2025-01-02 ~ 2025-07-31

## 변수 정의

종속변수: KOFR

독립변수: 기준금리, CPI, 실업률,  
제조업 PMI, 업황실적BSI(제조업)

## 전처리 규칙

월/비일별 지표는 ffill 스텝-캐리

정보 누수 방지를 위해 외생변수 1  
영업일 지연

# 모델 사양 및 예측

모델 형태: ARX(OLS) with p=1

$$\mathbf{y}_t = \mathbf{c} + \boldsymbol{\phi}_1 \mathbf{y}_{t-1} + \boldsymbol{\beta}^T \mathbf{x}_{t-1} + \boldsymbol{\varepsilon}_t$$

$y_t$  : KOFR

$x_{t-1}$  : 자연된 외생변수

추정 및 예측

추정: OLS

(각 검증일 t마다 t-1까지의 데이터로 확장창 재적합)

예측: 워크포워드 1-스텝

(매일 1일 앞 예측)

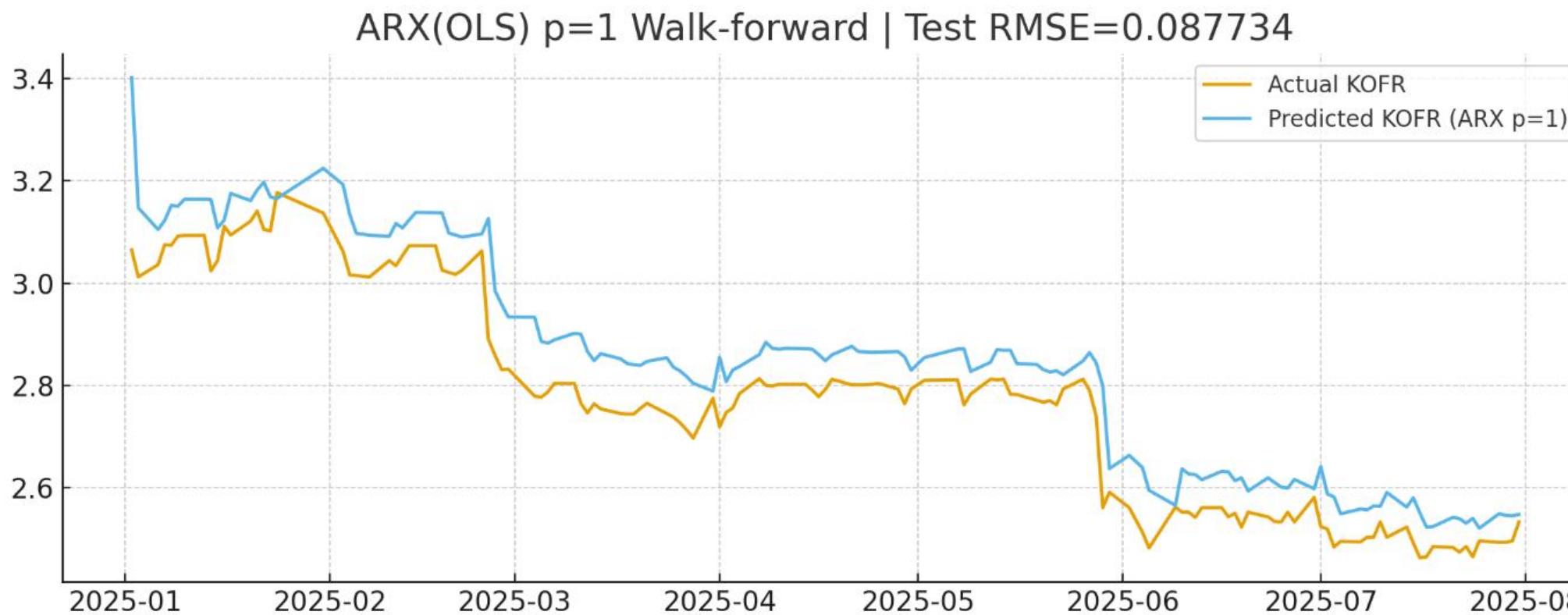
# 예측 결과 및 성능

RMSE (검증구간)

$\approx 0.0877$

Walk-forward Model ADF Test

$p = 0.0012$

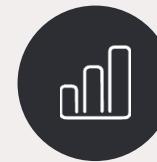


# Ljung-Box 검정



## 검정 설정

대상: 워크포워드 1-스텝 예측오차



## 요약 결과

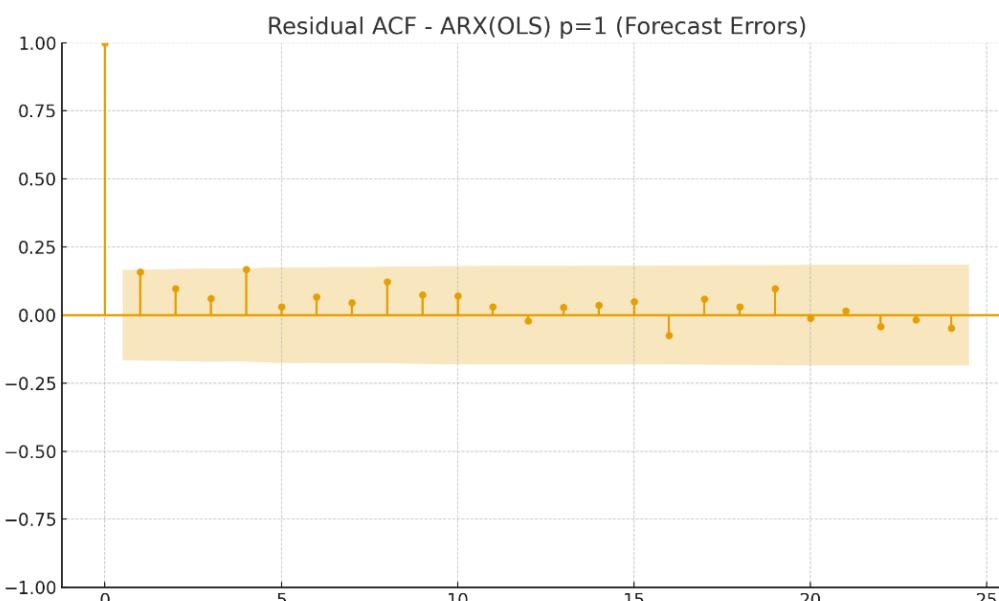
Lag 10: LB  $\approx 14.790$ , p  $\approx 0.1399$

잔차 ACF: 신뢰구간 내 빠른 감소



## 해석

유의수준 5%에서 귀무가설(자기상관 없음)을 기각하지 못함.



# TFT 시계열 모델 기반 다변량 KOFR 예측

## 주요 분석 과정

- 뉴스 제목 데이터 크롤링
- KR-FinBERT 모델 기반 감성분석 및 톤 점수 산출
- 감성분석 결과 및 외생변수 데이터 수집
- 데이터 전처리 및 모델링 예측



# 뉴스 제목 데이터 크롤링

## 목적

KOFR 예측에 필요한 독립변수 수집

거시경제 지표, 심리지표, 금리지표 외에도 KOFR 예측에 영향을 줄 비정형 이벤트 변수가 필요하다고 판단

## 세부 과정

1. BeautifulSoup 라이브러리 사용
2. 매일경제 홈페이지 접속 후 '금리' 검색
3. 2018-01-01~2025-07-31로 기간 지정
4. 각 날짜마다 상위 5개 금리 뉴스 제목 수집

```
import requests
from bs4 import BeautifulSoup
import pandas as pd
from datetime import datetime, timedelta
import time

query = "금리"
base_url = "https://www.mk.co.kr/search"

results = []

# 2018-01-01 ~ 2025-07-31
start_date = datetime(2018, 1, 1)
end_date = datetime(2025, 7, 31)

current_date = start_date
while current_date <= end_date:
    # 매일경제 필터링 검색 특성 상 하루 전날 ~ 당일 범위로 지정
    sdate = (current_date - timedelta(days=1)).strftime("%Y-%m-%d")
    edate = current_date.strftime("%Y-%m-%d")

    params = {
        "word": query,
        "sort": "desc",
        "dateType": "direct",
        "startDate": sdate,
        "endDate": edate,
        "searchField": "title",
        "newsType": "all"
    }

    resp = requests.get(base_url, params=params, headers={"User-Agent": "Mozilla/5.0"})
    soup = BeautifulSoup(resp.text, "html.parser")
```

# KR-FinBERT 모델 기반 감성분석 및 톤 점수 산출

목적: 비정형 데이터를 수치로 환산하여 KOFR 예측 모델링에 변수로 활용할 수 있는 형태로 만들기 위함

한국어 금융 텍스트 특화 모델 활용

KR-FinBERT 모델은 감성분석 수행 시 Label\_0(긍정),  
Label\_1(부정) 형태로 출력하고 긍정/부정 Score 계산

금리 관련 뉴스 매핑

긍정(Positive) → Hawkish (매파적, 금리 인상 시사, 긴축 기조)

부정(Negative) → Dovish (비둘기파적, 금리 인하, 완화 기조)

제목별 Tone Score 산출

Tone Score = Hawkish\_score - Dovish\_score

(양수 = 매파적, 음수 = 비둘기파적, 0 근처 = 중립)

날짜별 Tone Score 평균 계산

하나의 행에 하나의 날짜/Tone Score가 정렬되도록 제목별 Tone Score의 평균으로 계산

```
from transformers import AutoTokenizer, AutoModelForSequenceClassification
import torch
import torch.nn.functional as F
from tqdm import tqdm

# 빈 제목 제거
df = df.dropna(subset=["title"])

# 모델 로드 (KR-FinBERT)
model_name = "snunlp/KR-FinBERT"
tokenizer = AutoTokenizer.from_pretrained(model_name)
model = AutoModelForSequenceClassification.from_pretrained(model_name)

# 감성 분석 함수
def get_sentiment_scores(text):
    inputs = tokenizer(text, return_tensors="pt", truncation=True, padding=True)
    with torch.no_grad():
        outputs = model(**inputs)
        probs = F.softmax(outputs.logits, dim=-1).cpu().numpy()[0]

        hawkish_score = probs[0]    # LABEL_0 → Hawkish
        dovish_score = probs[1]     # LABEL_1 → Dovish
        tone_score = hawkish_score - dovish_score

    return hawkish_score, dovish_score, tone_score
```

# 감성분석 결과 및 외생변수 데이터 수집

목적: TFT 시계열 모델링에 활용할 데이터셋 구축

수집 기간: 2018.01.01~2025.07.31(daily)

## 금리 지표

'콜금리(1일)', 'KORIBOR(3개월)', 'KORIBOR(6개월)', 'KORIBOR(12개월)',  
'CD(91일)', '국고채(3년)', '국고채(10년)', 'KOFR', '기준금리', 'EFFR'

## 거시경제 지표

'CPI', 'Core CPI', '전산업생산지수', '원화실질실효환율', '실업률',  
'경제활동참가율', '제조업 PMI', '기대 인플레이션율'

## 심리 지표

'현재경기판단CSI', '향후1년경기전망CSI', '업황실적BSI(전산업)',  
'업황실적BSI(제조업)'

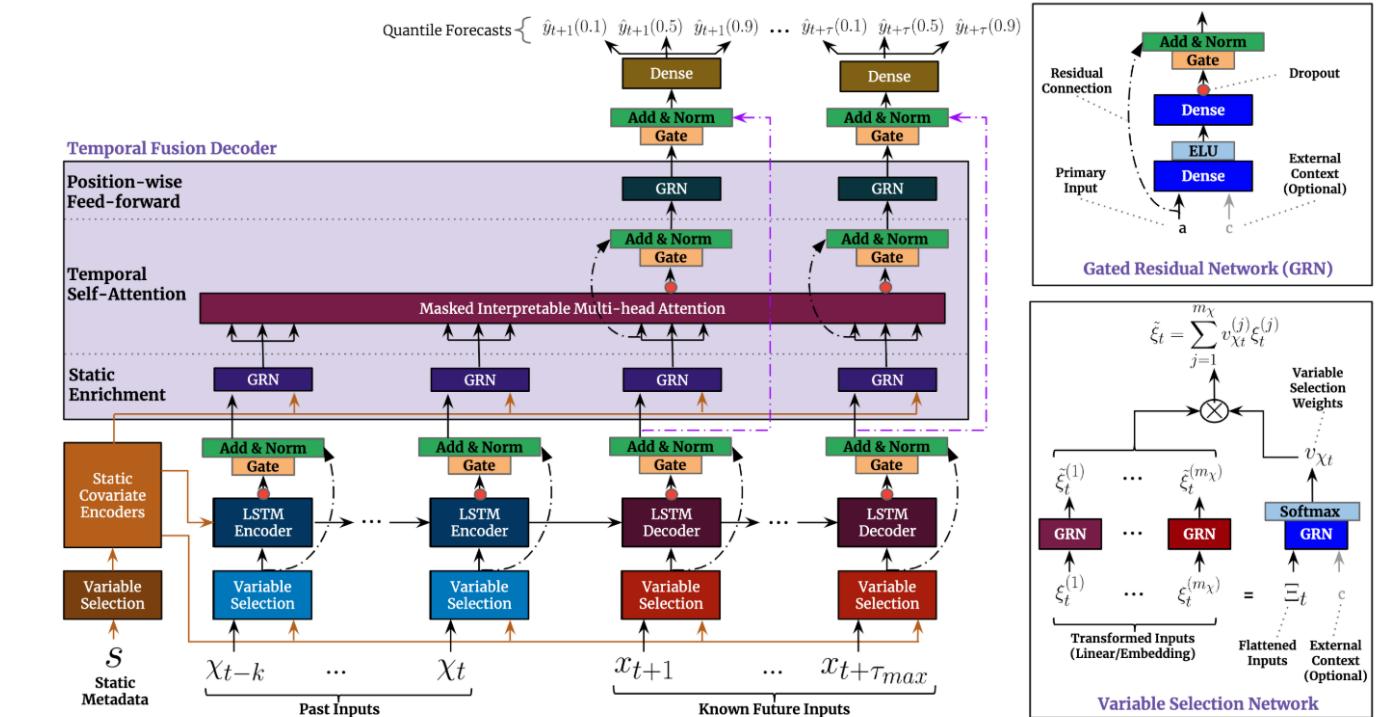
## 비정형 지표

'Tone\_score'

# TFT 시계열 예측 수행

## TFT 모델의 주요 특징

- 멀티 호라이즌 예측으로 여러 기간을 한 번에 예측
- 다중 변수와 외생 변수를 유연하게 처리
- 어텐션 매커니즘을 통해 중요 시점/변수에 가중치 적용
- 변수와 시점의 예측 기여도 해석 가능
- 게이트 구조로 불필요한 신호와 노이즈를 효율적 제거



# TFT 시계열 예측 수행

## 목적

2018~2024 데이터를 모델에 학습시켜 2025.01.01~2025.07.31까지의 예측값과 실제값을 비교  
모델의 성능을 파악해 추후 KOFR 지표 예측에 지속적으로 활용

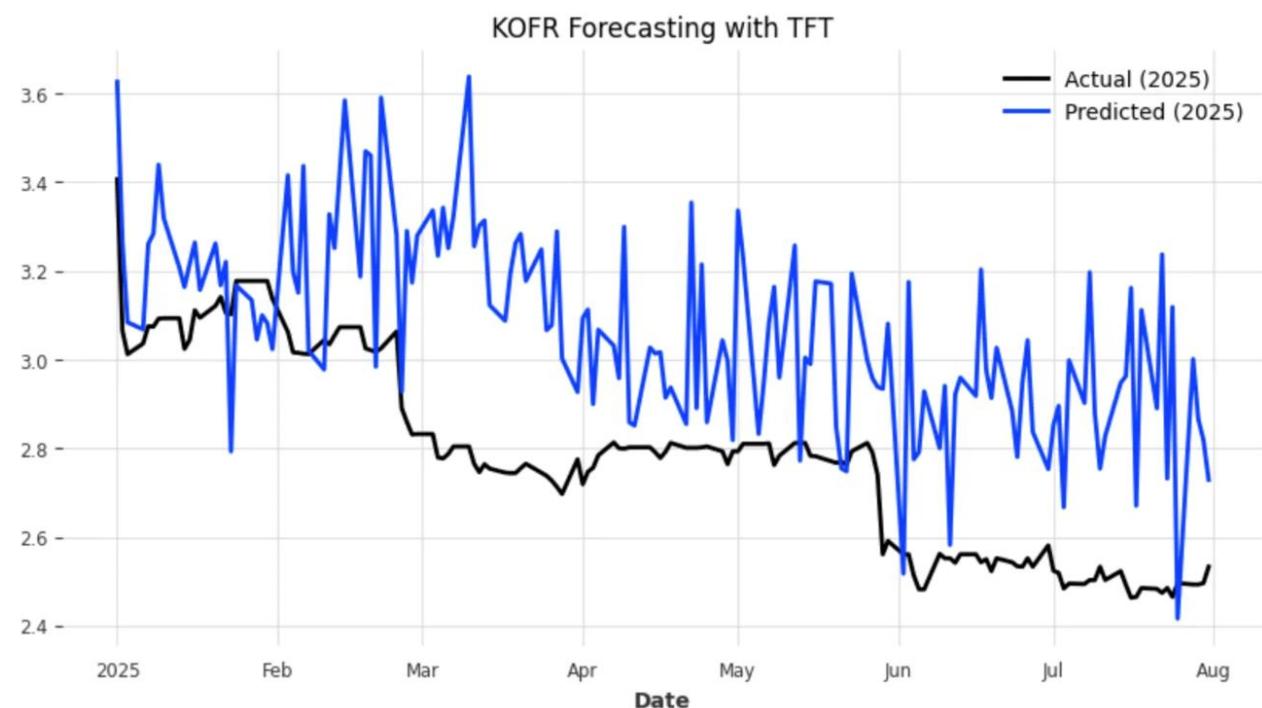
## 모델링 결과

Baseline ver: RMSE: 0.35196072 MAE: 0.30030718 R<sup>2</sup>: -1.6597626

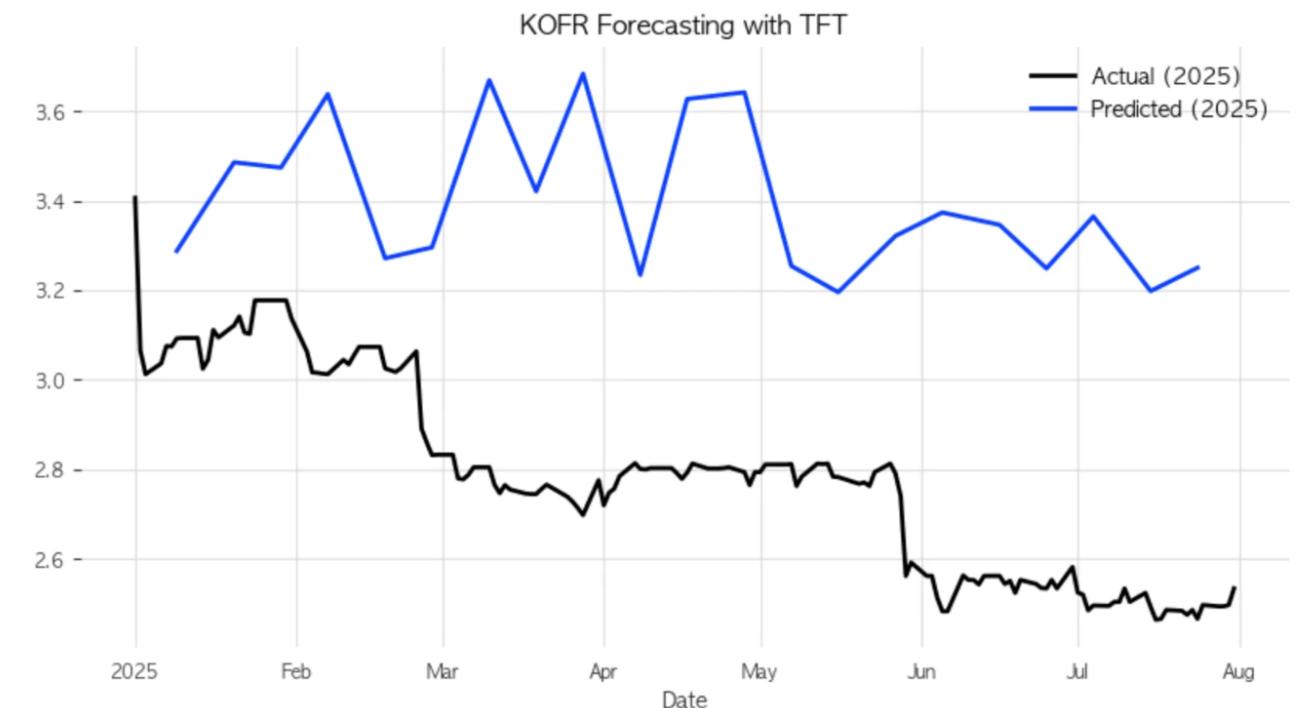
1. 2018부터 2024년까지의 데이터를 학습해서 2025-01-01부터 2025-07-31까지의 KOFR을 예측
2. 이미 2025-01-01~2025-07-31까지의 KOFR 값은 알고 있기 때문에, 예측값과 실제값을 비교
3. 전처리 과정에서 스케일링 및 정규화 진행, 결측치는 없어서 따로 처리X

Improve ver: RMSE: 0.6603136 MAE: 0.61808395 R<sup>2</sup>: -8.54697

1. 상관분석을 통해 KOFR과 관계성 높은 상위 10개 변수+Tone\_score(고정)으로 데이터셋 재구성
2. 모델 학습 복잡도 증가(입력 구간 확대, epoch 증가 등)
3. Optuna 하이퍼파라미터 튜닝



*Baseline ver*



*Improve ver*

# 결론 및 한계점/개선점

## 결론

1. baseline ver과 improve ver 모두 성능이 낮음
2. 오히려 improve ver에서 더 성능이 낮음

## 한계점/개선점

1. TFT 모델은 변수를 자동으로 선정해주는 데 불필요한 상관분석을 진행함
2. multi-head attention과 게이팅 메커니즘을 모델링 알고리즘에 잘 녹이지 못함
3. Pytorch forecasting lib 환경 호환성 이슈로 Darts lib 기반으로 진행함
4. TFT 모델을 단기간에 처음 다뤄본 것이 성능이 낮은 가장 큰 이유