

주식 투자자를 위한 딥러닝 기반 주가 예측 '**Fin Sight**' 솔루션

팀명: 핀사이트(Fin Sight)

팀원: 정영균, 박영진, 백승주

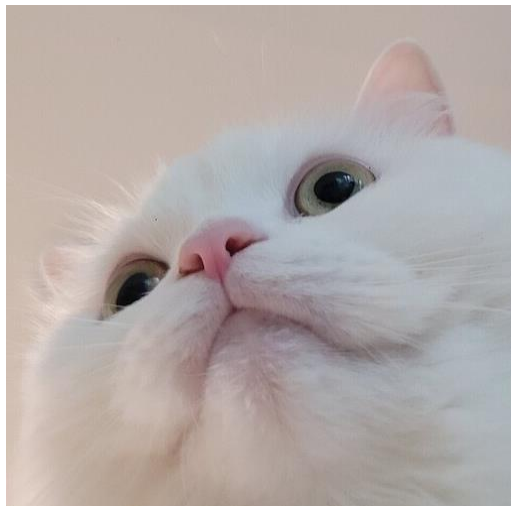
팀원소개



정영균 (팀장)

TA

- 프로젝트 리더
- 모델 구현 기획 및 구현
- 서비스 봇 생성



박영진

DA

- 데이터셋 수집
- 메인 모델 구현
- 웹 서비스 구현



백승주

AA

- 발표 문서 작성
- 서비스 봇 기획
- 아키텍처 작성

I. 기획 배경

1. 제안 배경 및 필요성
2. 아키텍처

II. 프로젝트 수행

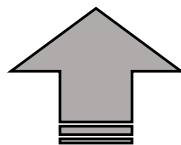
1. 데이터 수집 및 EDA
2. 데이터셋 분리
3. 종목 선정
4. 주가 예측 모델 구현
 - (1)모델 구성
 - (2)기술 플로우
 - (3)종목별 최적 모델 결과
 - (4)하이퍼파라미터 튜닝
 - (5)결론

III. 솔루션

1. 솔루션 소개
2. 주요 기능
3. 주요 모듈
4. 서비스
 - (1)웹 서비스
 - (2)챗봇 서비스
5. 데모 시연
6. 특징 및 장점
7. 기대효과
8. 비즈니스

I. 기획 배경_제안 배경 및 필요성

“종목 별 주가 예측 모델 구현”



1. 주가 예측의 중요성: 오랜 기간 연구되었으며, 금융 시장에서 투자 전략의 핵심 도구임

2. 현대 금융 시장의 복잡성: 종목에 따라 가격 변동성이 큼.

3. 예측 정확도의 중요성: 금융 데이터 분석에 딥러닝 모델을 도입

I. 기획 배경_아키텍처

데이터 수집

데이터 구축

모델 비교 분석

서비스 구현

서비스

주가 데이터: 종가

기술 지표: RSI,MACD,Bollinger Bands,
Volume

외부 요인: 금리, 달러 인덱스, 장단기 금리차

모델

데이터
(D1, D2, D3)

Fin Sight 웹 서비스

모델 학습 및 평가



최적의 모델과 데이터셋 선정

Fin Sight 챗봇

기술

'yfinance' library

FRED API

XGBoost

LSTM

GRU

Transformer

Streamlit

하이퍼파라미터 튜닝

RMSE 값 기반 모델 선택



최적의 모델과 데이터셋 선정

BotFather API

python-telegram-bot

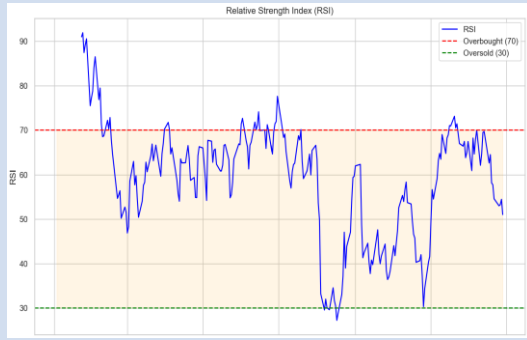
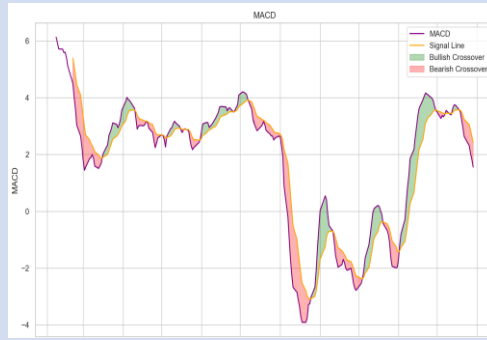
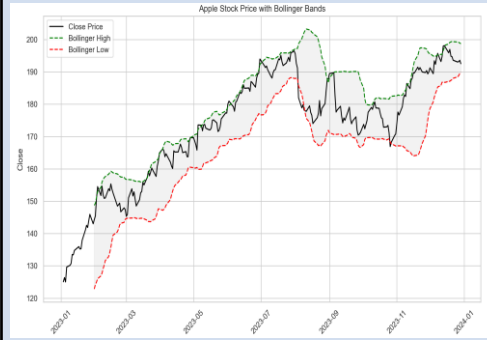
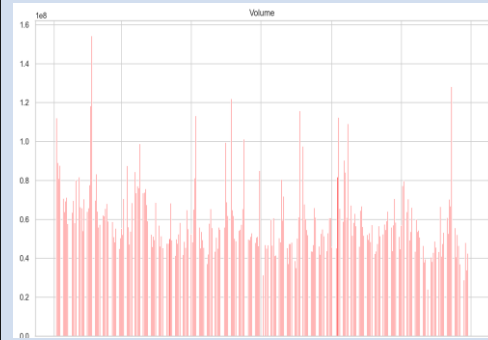
II. 프로젝트 수행_데이터 수집 및 EDA

분류	데이터명	데이터 수집 과정	코드
주가 데이터	종가	1. "yfinance"라이브러리를 이용하여 주가 데이터 다운로드 (2014.06.30~2024.06.27 일봉 데이터) 2. 필요한 열 선택: '종가', '거래량'	<pre>import yfinance as yf import pandas as pd # 애플(AAPL) 주식 데이터 다운로드 ticker = "AAPL" start_date = "2014-06-01" end_date = "2024-06-28" # 데이터 다운로드 (일봉) data = yf.download(ticker, start=start_date, end=end_date, interval='1d') # 필요한 열만 선택 (고가, 저가, 종가, 거래량) data = data[['High', 'Low', 'Close', 'Volume']]</pre>
기술 지표	Volume		
	RSI	주가의 일일 변동의 14일간 평균 상승폭과 하락폭을 계산	<pre>import pandas as pd # 종가의 변화량 계산 delta = data["Close"].diff() # RSI 계산 period = 14 gain = delta.where(delta > 0, 0) loss = delta.where(delta < 0, 0) avg_gain = gain.rolling(window=period, min_periods=1).mean() avg_loss = loss.rolling(window=period, min_periods=1).mean() rs = avg_gain / avg_loss data["RSI"] = 100 - (100 / (1 + rs)) # MACD 계산 short_period = 12 long_period = 26 signal_period = 9 ema_short = data["Close"].ewm(span=short_period, adjust=False).mean() ema_long = data["Close"].ewm(span=long_period, adjust=False).mean() data["MACD"] = ema_short - ema_long data["Signal"] = data["MACD"].ewm(span=signal_period, adjust=False).mean() data["MACD_Hist"] = data["MACD"] - data["Signal"] # 볼린저밴드 계산 band_period = 20 std_dev = 2 data["Middle_Band"] = data["Close"].rolling(window=band_period).mean() data["Std_Dev"] = data["Close"].rolling(window=band_period).std() data["Upper_Band"] = data["Middle_Band"] + (data["Std_Dev"] * std_dev) data["Lower_Band"] = data["Middle_Band"] - (data["Std_Dev"] * std_dev) data.drop(columns=["Std_Dev"], inplace=True)</pre>
	MACD	12일과 26일 이동 평균의 차이로 MACD 선을 계산하고, 9일 이동 평균으로 신호선을 설정하여 추세 변화를 분석	
	Bollinger Bands	20일 이동 평균을 중심선으로, 표준 편차를 이용해 상·하단 밴드를 계산하여 주가 변동성 계산	
외부 요인	금리	1. API: FRED API를 사용하여 미국 달러 인덱스(DXY), 미국 기준 금리(Federal Funds Rate), 장단기 금리차(Yield Spread) 데이터 수집.	<pre>import pandas as pd import yfinance as yf from fredapi import Fred # FRED API key 설정 fred_api_key = 'API_KEY' fred = Fred(api_key=fred_api_key) # 데이터 수집 기간 설정 start_date = '2014-06-30' end_date = '2024-06-28' # 달러 인덱스 데이터 (DXY) usd_index = yf.download('DX-Y.NYB', start=start_date, end=end_date)['Close'] # 미국 금리 데이터 (Federal Funds Rate) interest_rate = fred.get_series('FEDFUNDS', start_date, end_date) # Yield Spread 데이터 (10-Year vs 2-Year Treasury) yield_spread = fred.get_series('T10Y2Y', start_date, end_date) # 데이터 병합 (달러 인덱스를 기준으로 날짜 맞추기) data = pd.DataFrame({ 'USD_Index': usd_index, 'Interest Rate': interest_rate.reindex(usd_index.index), 'Yield Spread': yield_spread.reindex(usd_index.index) }) # 데이터 확인 (컬럼이 있는지 확인) print(data.head(10)) # 첫 10개 행 출력</pre>
	달러 인덱스		
	장단기 금리차	2. 데이터 처리: 수집된 데이터를 날짜 기준으로 정렬하고, 하나의 데이터프레임으로 통합.	

II. 프로젝트 수행_데이터 수집 및 EDA

(1) 기술 지표

: 시장 다양성을 고려하여 중복을 피하며, 계산도 간단해 모델에 적용하기 쉬운 지표를 선정함.

	RSI (상대 강도 지수)	MACD (이동 평균 수렴 발산 지수)	Bollinger Bands (볼린저 밴드)	Volume (거래량)
정의	주가의 상승과 하락을 측정하여 과매수/과매도 상태 파악.	단기와 장기 이동 평균의 차이로 주가의 추세 방향 파악.	이동 평균선 주위에 설정된 밴드를 통해 가격의 변동성을 시각화.	주가 움직임에 대한 시장 참여도를 측정.
지표 해석 방법	<ul style="list-style-type: none">· 과매수 신호(70 이상)· 과매도 신호(30 이하)	<ul style="list-style-type: none">· 매매 신호 기준선을 기준으로 상승/하락 가능성	<ul style="list-style-type: none">· 변동성 상/하한선	<ul style="list-style-type: none">· 종목의 거래가 얼마나 활발한 지를 나타냄.
사진 자료				

※ 사진 자료: 애플 1년치 주가 시각화(23.01.01 ~ 24.01.01)

II. 프로젝트 수행_데이터 수집 및 EDA

(2) 외부 요인

: 7개의 외부 요인에 대한 상관 관계 분석 진행 결과, '금리', '달러 인덱스', '장단기 금리차' 선정

< 외부 요인과 S&P 500 간의 상관관계 >



외부 요인

1. 금리 (Interest Rate)
2. 달러 인덱스 (Dollar Index)
3. 실업률 (Unemployment Rate)
4. GDP 성장률 (GDP Growth Rate)
5. 장단기 금리차 (Industrial Production Index)
6. 산업 생산지수 (Industrial Production Index)
7. 초기 실업수당 청구건수 (Initial Jobless Claims)

< 코드 >

```
import seaborn as sns
import matplotlib.pyplot as plt

# S&P 500 증가와 경제 지표들 간의 상관관계 계산
correlation_matrix = final_data[['Close'] + list(economic_indicators.keys())].corr()

# Close 열과 나머지 지표들 간의 상관관계만 추출
correlation_with_close = correlation_matrix.loc[list(economic_indicators.keys()), ['Close']]
```


II. 프로젝트 수행_데이터 수집 및 EDA

(2) 외부 요인

금리



- 경제 안정화와 성장 관리의 핵심.
- 미국 기준 금리는 연방준비제도(Fed)에 의해 결정됨.

달러 인덱스



- 미국 달러 가치를 주요 6개 통화에 대해 비교한 지수.
- 국제 무역 및 자본 흐름에 영향을 줌.

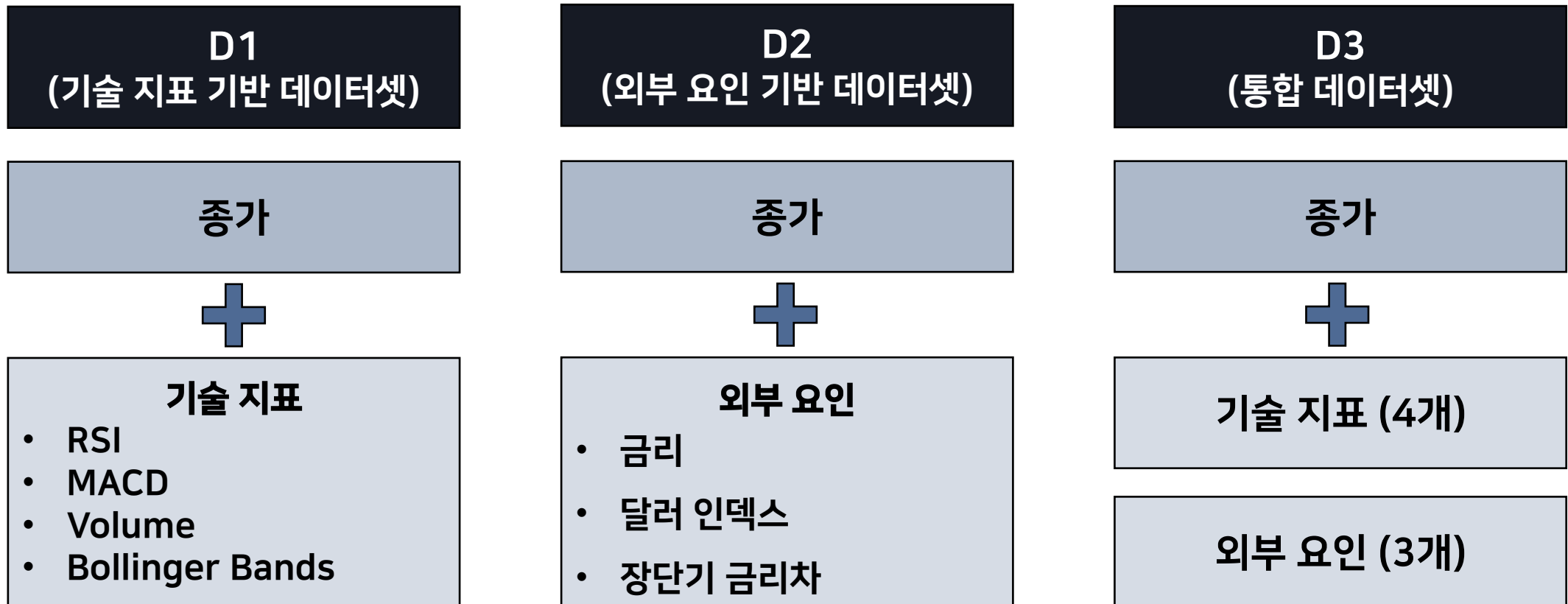
장단기 금리차



- 장단기 금리차는 미래의 경제 상황에 대한 투자자들의 기대를 반영하는 것.
- 경제 성장 및 경기 침체에 대한 신호.

II. 프로젝트 수행_데이터셋 분리

- 종목별 **주가 예측에 더 큰 영향을 미치는 데이터셋을 분석하기 위해** 3개의 데이터셋을 생성.
- **데이터 기간이 예측 정확도에 미치는 영향을 평가**하기 위해 각 데이터셋을 5년치와 10년치로 나눠 총 6개의 데이터셋으로 진행.



II. 프로젝트 수행_데이터셋 분리

- 주가 데이터 분리 (train, validation, test)

TRAIN	VALIDATION	TEST
60%	20%	20%

- 전체 데이터를 6:2:2 비율로 분할
- Train(모델 학습용), Validation(하이퍼파라미터 튜닝), Test(모델 성능 평가용)

슬라이딩 윈도우 기법 예시

1	2	3	4	5	6	7	8	9	10	11	12
1	2	3	4	5	6	7	8	9	10	11	12
1	2	3	4	5	6	7	8	9	10	11	12
1	2	3	4	5	6	7	8	9	10	11	12
1	2	3	4	5	6	7	8	9	10	11	12

- 과거 데이터를 누적하여 다음 시점의 값을 예측
- ex) window size = 5,
- prediction offset = [1,2,3]
- 최근 5일 데이터로 1일, 2일, 3일 후를 예측
- 매일 새 데이터로 업데이트
- 연속적 학습: 최신 경향 반영

실제 모델에는 window size = 20,
prediction offset = [1,2,3,4,5] 적용

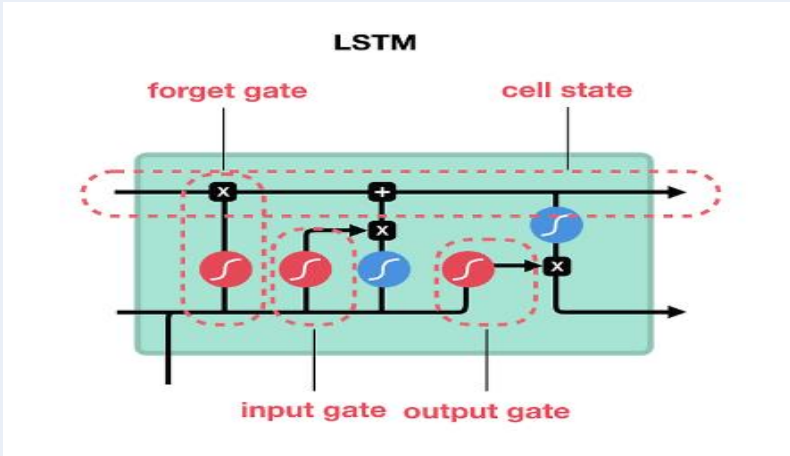
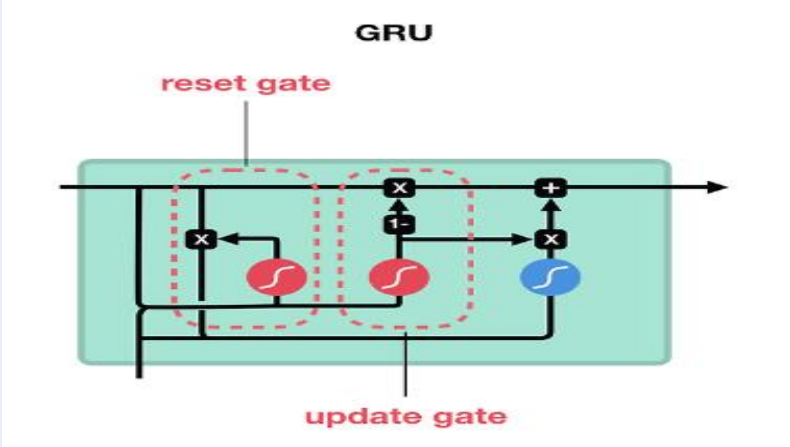
II. 프로젝트 수행_종목 선정

- 각 산업을 대표하고 있는 기업을 주가 예측 종목으로 선정.

산업	기업	선정 이유 및 기업 설명
IT	Apple Inc. (AAPL)	-소비자 가전, 소프트웨어, 서비스 분야 선도 -기술 혁신 주도, 글로벌 시장 영향력 큼
필수 소비재	The Coca-Cola Co. (KO)	-오랜 역사, 강력한 브랜드 파워 -경기 변동에 상대적으로 안정적, 글로벌 시장 점유율 높음
헬스케어	Johnson & Johnson (JNJ)	-의약품, 의료기기, 소비자 건강 제품 다각화 -안정적 성장, 다양한 제품 포트폴리오로 리스크 분산
금융	JPMorgan Chase & Co. (JPM)	-미국 최대 은행 중 하나, 다양한 금융 서비스 제공 -경제 지표 및 금리 변화와 밀접한 연관성, 금융 시장의 전반적 동향 반영

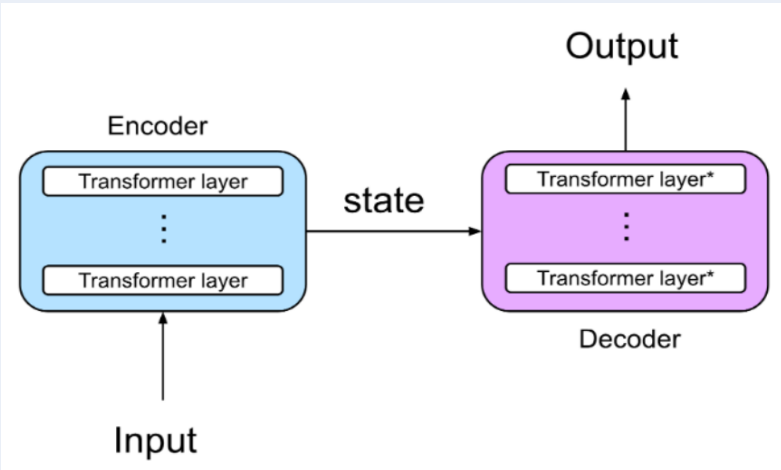
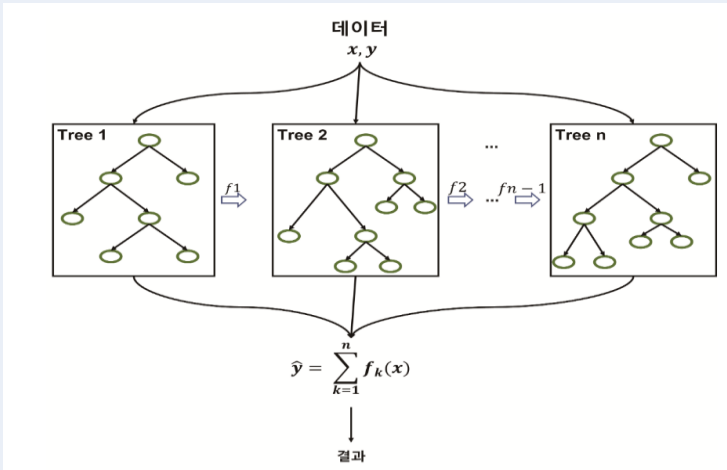
II. 프로젝트 수행_주가 예측 모델 구현

(1) 모델 구성

모델	LSTM	GRU
사진 자료		
특징	<ul style="list-style-type: none">• 장기 의존성 해결, Forget gate, input gate, output gate로 구성	<ul style="list-style-type: none">• LSTM과 유사, 더 단순한 구조• Update gate, reset gate로 구성
하이퍼 파라미터	<ul style="list-style-type: none">• Units(층별 뉴런 수)• LSTM_Layers(모델 깊이)	<ul style="list-style-type: none">• Units(층별 뉴런 수)• GRU_Layers(모델 깊이)

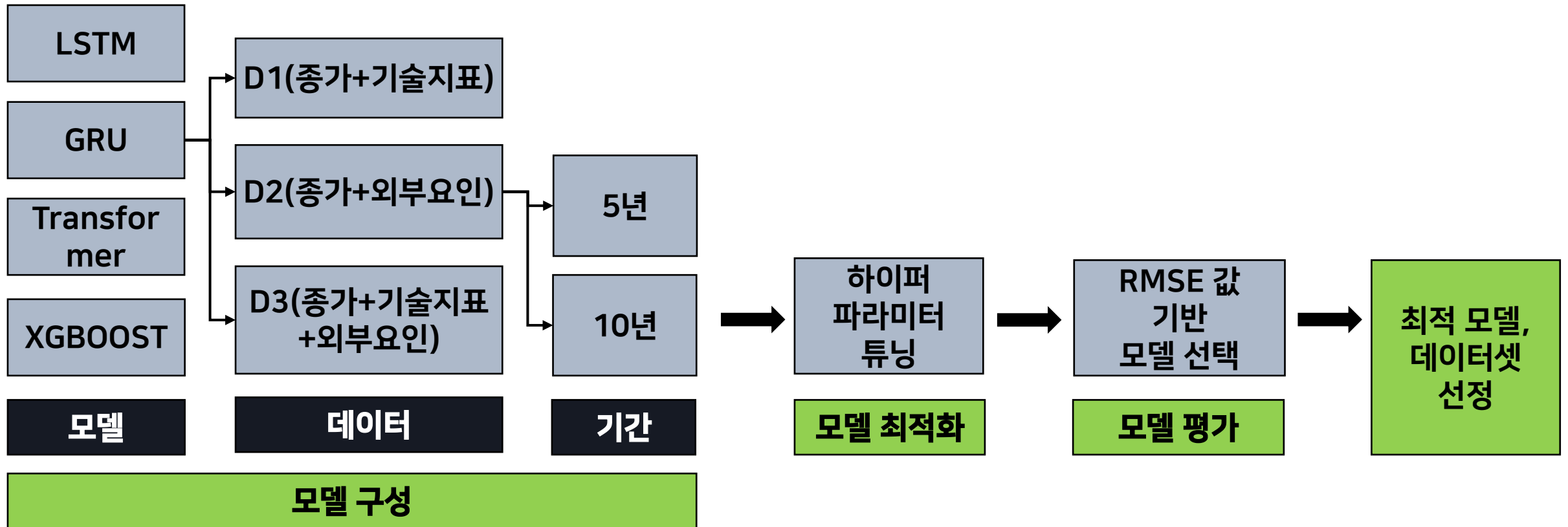
II. 프로젝트 수행_주가 예측 모델 구현

(1) 모델 구성

모델	Transformer	XGBOOST
사진 자료		
특징	<ul style="list-style-type: none">• Self attention 매커니즘 사용• 병렬 처리 가능	<ul style="list-style-type: none">• Gradient 알고리즘,• 결정 트리 앙상블 방식
하이퍼 파라미터	<ul style="list-style-type: none">• Head_size• Num_Heads• FF_Dim(피드포워드 차원)	<ul style="list-style-type: none">• n_estimators(트리의 수)• Max_depth(최대 깊이)• Subsample(데이터 비율)

II. 프로젝트 수행_주가 예측 모델 구현

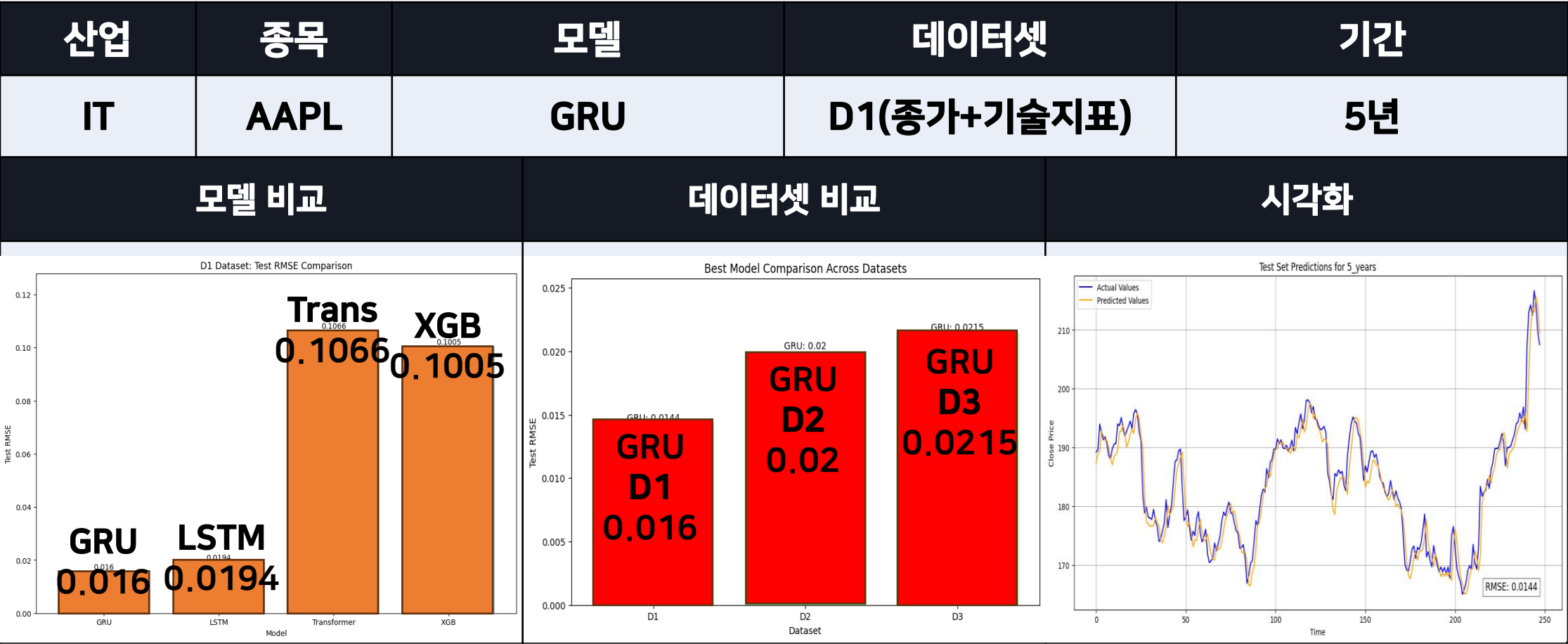
(2) 기술 플로우



II. 프로젝트 수행_주가 예측 모델 구현

(3) 종목별 최적 모델 결과

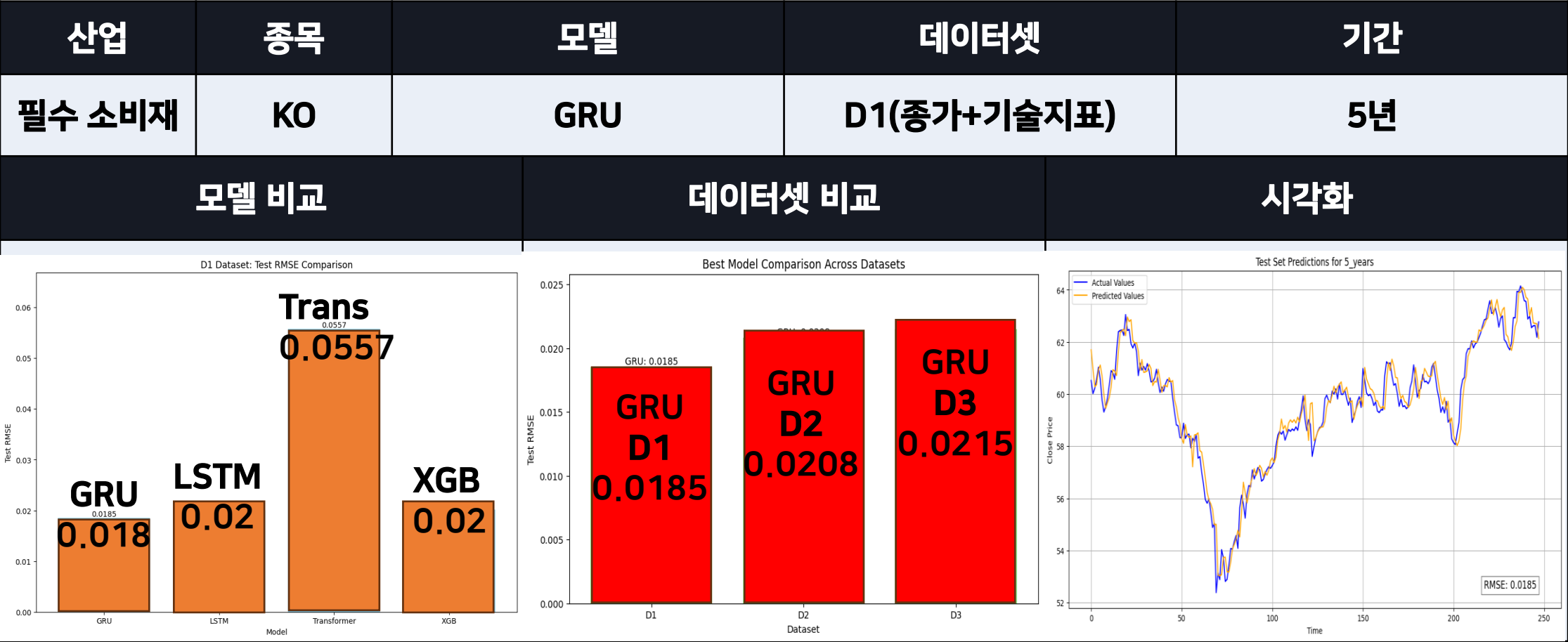
- IT 산업의 '애플' 종목: GRU + D1 + 5년



II. 프로젝트 수행_주가 예측 모델 구현

(3) 종목별 최적 모델 결과

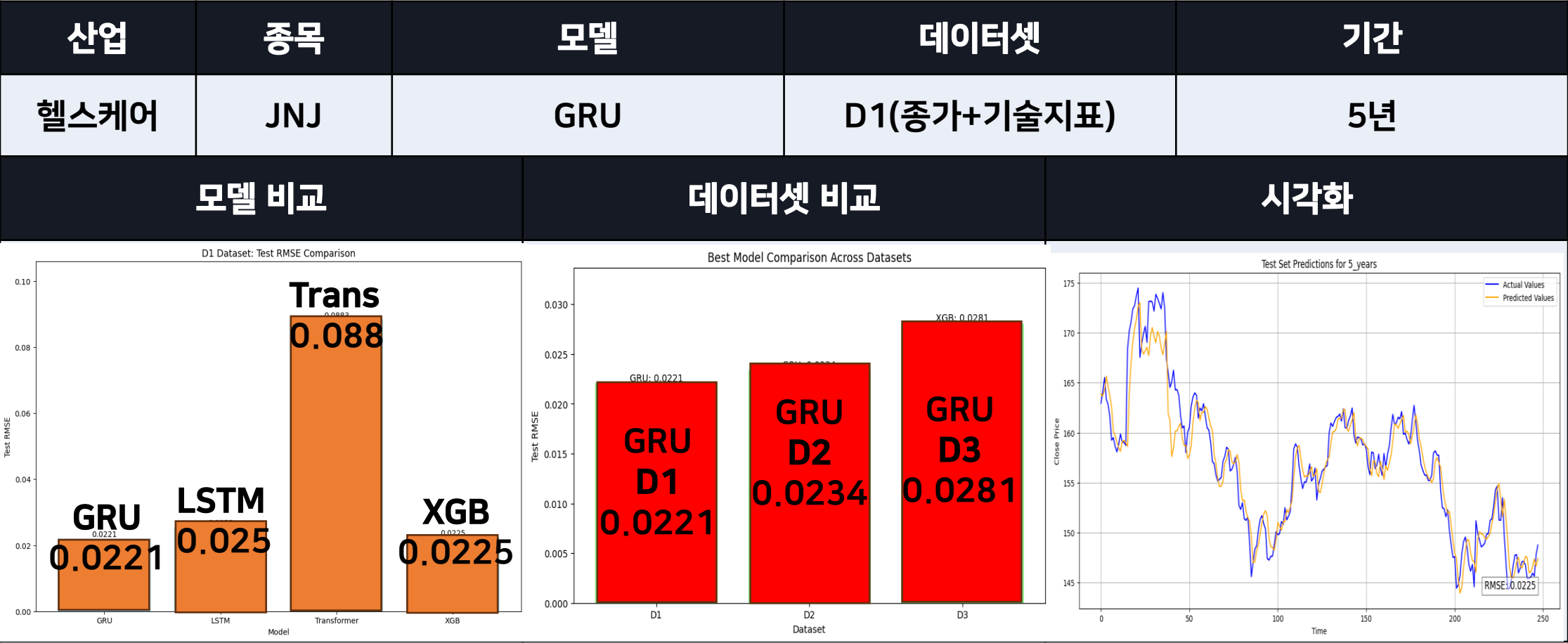
- 필수 소비재 산업의 '코카콜라' 종목: GRU + D1 + 5년



II. 프로젝트 수행_주가 예측 모델 구현

(3) 종목별 최적 모델 결과

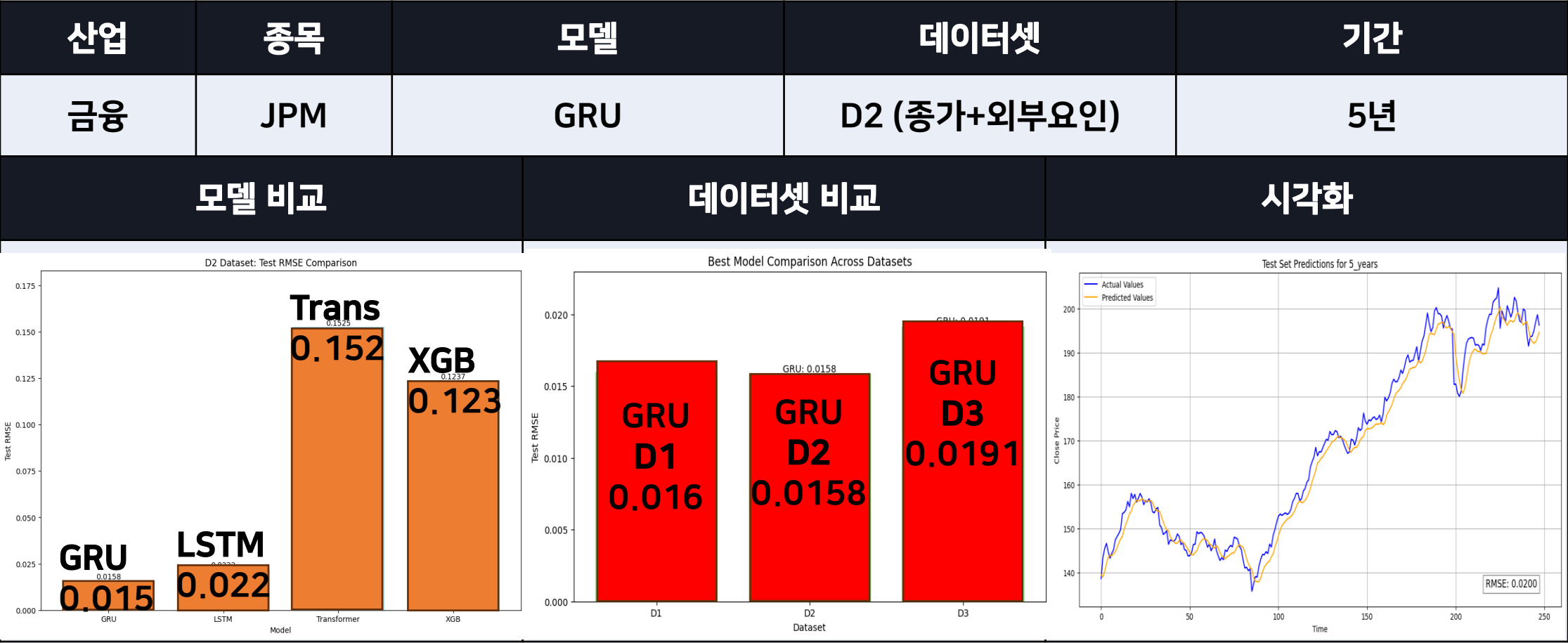
- 헬스케어 산업의 '존슨앤존슨' 종목: GRU + D1 + 5년



II. 프로젝트 수행_주가 예측 모델 구현

(3) 종목별 최적 모델 결과

- 금융 산업의 'JP모건 체이스' 종목: GRU + D2 + 5년



II. 프로젝트 수행_주가 예측 모델 구현

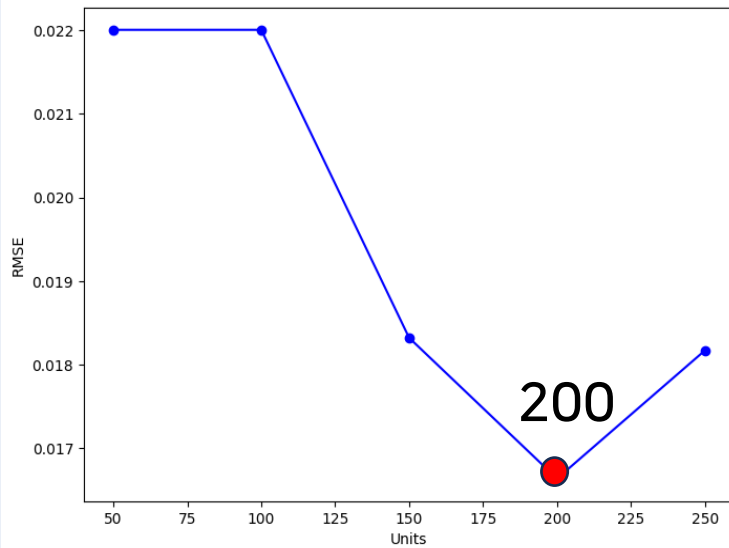
(3) 종목별 최적 모델 결과

종목	AAPL	KO	JNJ	JPM
모델	GRU			
기간	5년			
데이터셋	D1(종가+기술지표)	D1(종가+기술지표)	D1(종가+기술지표)	D2(종가+외부요인)
RMSE	0.0144	0.0185	0.0225	0.0200

II. 프로젝트 수행_주가 예측 모델 구현

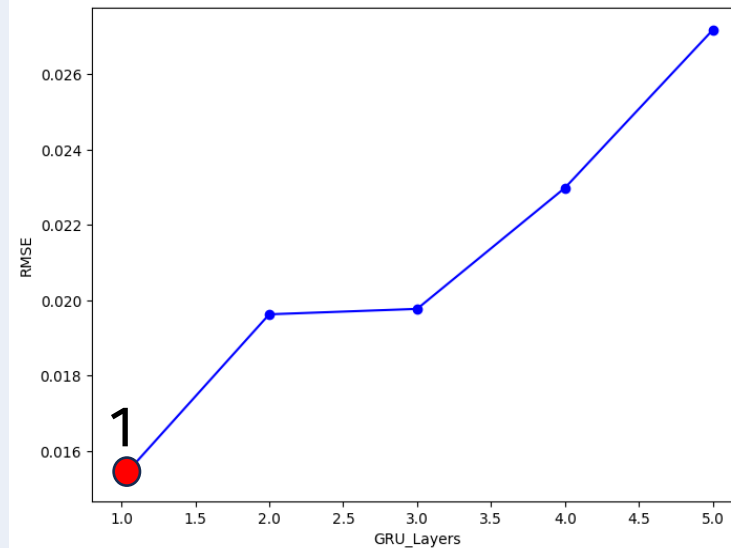
(4) AAPL 하이퍼파라미터 튜닝

Units(층별 뉴런 수)



- 유닛 수 증가로 모델 표현력이 향상되나, 과도하면 과적합 위험

GRU_Layers(모델 깊이)

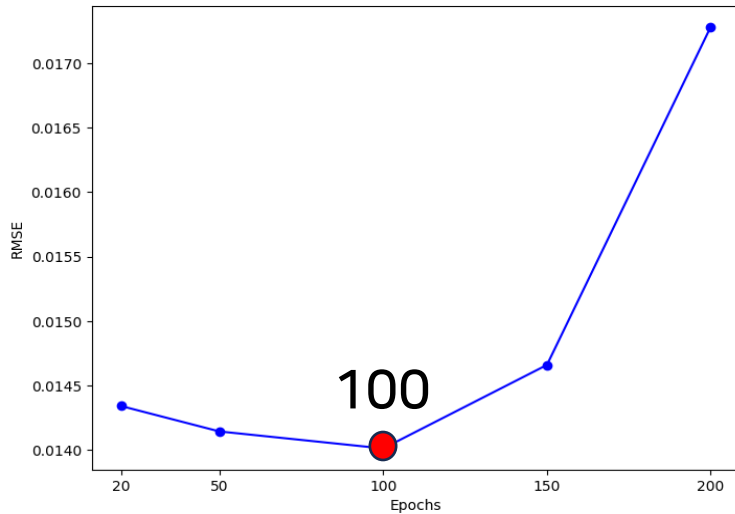


- 단순한 구조가 효과적

II. 프로젝트 수행_주가 예측 모델 구현

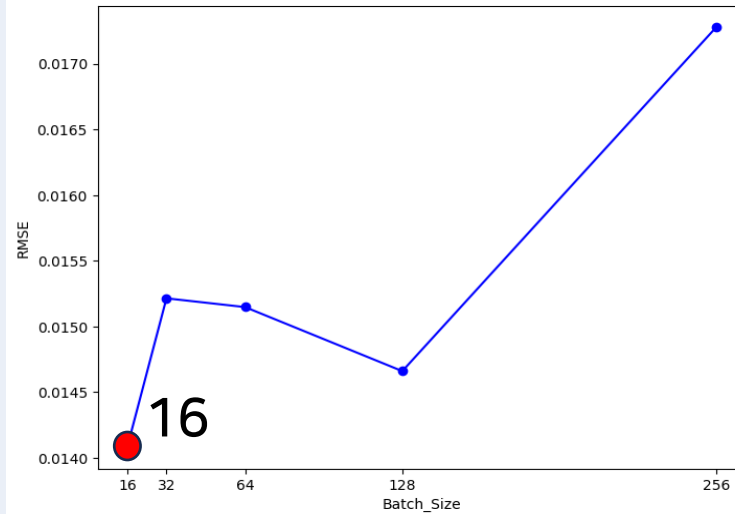
(4) AAPL 하이퍼파라미터 튜닝

Epochs (학습 횟수)



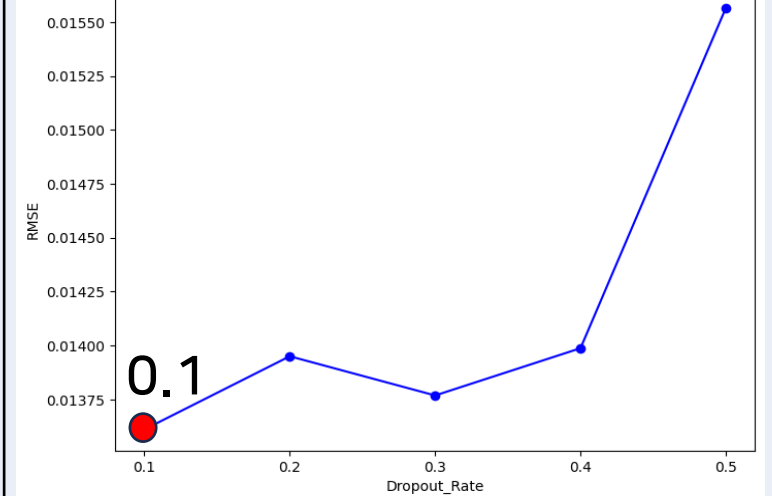
- 적은 Epoch은 학습 부족으로 성능 저하, 큰 Epoch은 과적합으로 일반화 능력 감소

Batch Size (학습 데이터 크기)



- Batch size 값이 커질수록 모델 업데이트 빈도가 감소하여 학습이 세밀하지 않음

Dropout_Rate(노드 비활성화 비율)



- dropout이 너무 높을 경우 모델이 많은 정보를 잃어 학습 저하

II. 프로젝트 수행_주가 예측 모델 구현

(4) 전체 종목 하이퍼파라미터 튜닝 결과

	종목	AAPL	KO	JNJ	JPM
하이퍼 파라미터	Units	200	200	150	150
	GRU Layers	1	2	1	1
	Batch Size	16	32	16	16
	Epochs	100	100	100	100
	Dropout Rate	0.1	0.1	0.2	0.1

II. 프로젝트 수행_주가 예측 모델 구현

(5) 결론

최적 모델	GRU	시계열 데이터의 장기 의존성 효과적으로 포착, 복잡한 주기 패턴 학습에 적합
최적 기간	5년	최근의 시장 트렌드를 잘 반영하면서도 과적합을 방지
최적 데이터셋	AAPL,KO,JNJ : D1(종가+기술지표) JPM : D2(종가+외부요인)	JPM의 경우 금융 섹터 특성상 거시경제 지표 등 외부 요인의 영향이 큼

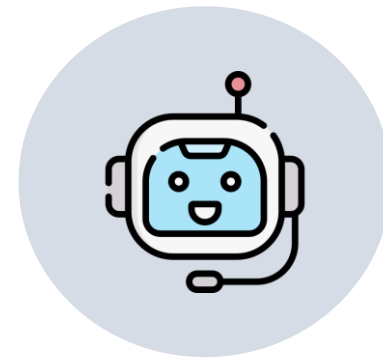
III. 솔루션_솔루션 소개

"주식 투자자를 위한 주가 예측 솔루션 Fin Sight"

투자자들에게 종목별 맞춤형 주가 예측을 제공하여 손쉽게 접근가능한 플랫폼 구축을 위함.



웹 서비스



챗봇 서비스

III. 솔루션_주요 기능

Fin Sight 웹 서비스

기능	설명
모델 소개	GRU 모델 구조 설명 및 다른 모델과의 성능 비교 시각화
데이터셋 소개	주가 예측에 사용되는 데이터셋 유형 설명 및 시각화
예측	사용자가 선택한 주식의 현재 가격 정보와 향후 5일간의 예상 주가 제공

Fin Sight 챗봇 서비스

기능	설명
주가 예측 제공	<ul style="list-style-type: none">• 사용자가 선택한 카테고리에 따라 해당 주식의 현재 가격과 향후 5일간의 예상 주가를 제공.• 각 요일별로 예측된 가격을 텍스트로 출력
자동 메시지 발송	<ul style="list-style-type: none">• 매주 일정 시간에 예측 정보 제공: 매주 일요일에 사용자에게 향후 5일간의 예상 주가를 자동으로 발송

III. 솔루션_주요 모듈

서비스	기능/모듈	과정	코드
웹	모델, 데이터셋 소개	1. GRU 모델 구조와 성능 설명, 모델별 성능 비교 2. 주가 예측에 사용되는 데이터셋의 주요 특성을 실제 데이터를 활용한 다양한 차트와 그래프로 시각화하여 표현	<pre>def dataset_exploration(): # 데이터셋의 주요 특성을 설명 # 주가 예측을 위해 세 가지 주요 데이터를 활용합니다. 각 데이터셋은 서로 다른 측면의 정보를 제공하며 # 데이터 예측 성능을 향상시킵니다. """ 1. 현재 데이터: 과거주가 stock = df[['stock', 'open']] end_date = datetime.now() start_date = end_date - timedelta(days=365) # 지난 1년 data = stock.history(start=start_date, end=end_date) 2. 미래 데이터: 목표 # header: (1) - 주가 - 거래처지표 # columns: 주가의 과거와 다양한 기술적 지표 # 데이터셋은 주가의 과거와 다양한 기술적 지표를 포함합니다. 이 데이터셋은 주로 주가의 과거 패턴과 # 주가를 분석하는 데 사용됩니다. """</pre>
	예측	1. 사용자가 선택한 현재 주식의 가격 정보를 표시하고, 향후 5일간의 예상 주가를 그래프로 시각화하여 제공 2. 다음 거래일의 예상 가격 변동을 강조하여 표시	<pre>with col2: st.subheader("주가 예측 결과") next_day_prediction = predictions[0] delta = next_day_prediction - current_price st.metric("다음 거래일 예상 가격", f"{next_day_prediction:.2f}", delta=f"{delta:.2f}") if delta > 0: st.success("주가 상승할 것으로 예상됩니다.") else: st.error("주가 하락할 것으로 예상됩니다.") # 주가 예측 시각화 fig = go.Figure() fig.add_trace(go.Scatter(x=prediction_dates, y=predictions, name='예측 주가', line=dict(color='red', width=2))) fig.add_trace(go.Scatter(x=prediction_dates, y=[monday_price] * 5, name='현재 주가 (월요일)', mode='markers', marker=dict(color='blue', size=10))))</pre>
챗봇	주가 예측 제공	1. 사용자가 특정 주식을 선택하면, predict 함수 호출 2. 선택한 주식의 현재 가격과 다음 주의 예상 주가 조회 3. 예상 주가는 월요일부터 금요일까지의 날짜와 함께 메시지로 제작되며, 사용자에게 전송	<pre># 주식 선택 후 예측 결과 전송 usage.new * async def predict(update: Update, context: ContextTypes.DEFAULT_TYPE): query = update.callback_query await query.answer() stock = query.data.split('@')[1] # stock_apple -> apple current_price = current_prices[stock] # current 가격 조회 predictions = predictions_data[stock] # 예측 데이터 조회 # 각 요일과 날짜를 함께 출력 days = ["월요일", "화요일", "수요일", "목요일", "금요일"] response = f"{friday_date.strftime('%m월 %d일')}: 금요일 가격: \${current_price:.2f}\n\n" response += "다음 5일간의 예측 가격:\n\n" for day, date, pred in zip(days, next_week_dates, predictions): response += f"{date.strftime('%m월 %d일')}: {day}: \${pred}\n\n" # "읽음" 버튼 추가 (해당 기능은 주가 선택 화면에서 정의함) category = get_category_from_stock(stock) keyboard = [[InlineKeyboardButton(text="읽음", callback_data=f'category_{category}'))] reply_markup = InlineKeyboardMarkup(keyboard) await query.edit_message_text(response, reply_markup=reply_markup)</pre>
	자동 메시지 발송	1. 특정 요일과 시간에 자동으로 메시지를 전송하도록 스케줄 설정 2. 설정된 시간에 send_weekly_predictions 함수가 호출되어, 각 주식의 현재 가격과 다음 주 예상 주가 조회 3. 조회된 예상 주가는 사용자에게 메시지로 생성되어 자동으로 전송	<pre># 자동 메시지 발송 함수 usage.new * async def send_weekly_predictions(context: ContextTypes.DEFAULT_TYPE): # 각 사용자에게 예측 데이터를 전송하는 로직 for user_id in context.job.data['user_ids']: for stock, current_price in current_prices.items(): predictions = predictions_data[stock] days = ["월요일", "화요일", "수요일", "목요일", "금요일"] response = f"{friday_date.strftime('%m월 %d일')}: 금요일 가격: \${current_price:.2f}\n\n" response += "다음 5일간의 예측 가격:\n\n" for day, date, pred in zip(days, next_week_dates, predictions): response += f"{date.strftime('%m월 %d일')}: {day}: \${pred}\n\n" await context.bot.send_message(chat_id=user_id, text=response)</pre>

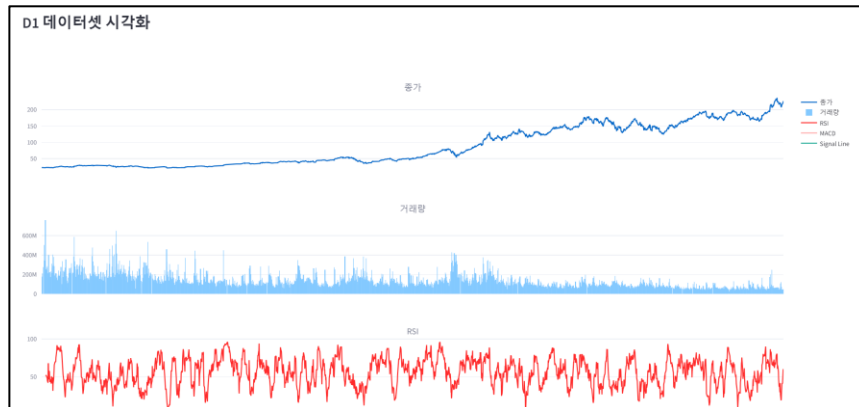
III. 솔루션_서비스

(1) Fin Sight 웹 화면

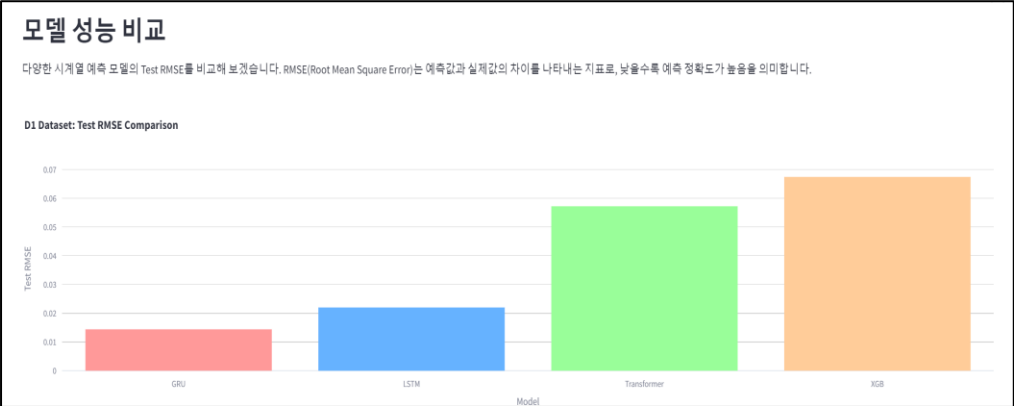
: 스트림릿을 통해 구현한 주가 예측 서비스



홈 화면



데이터셋 소개



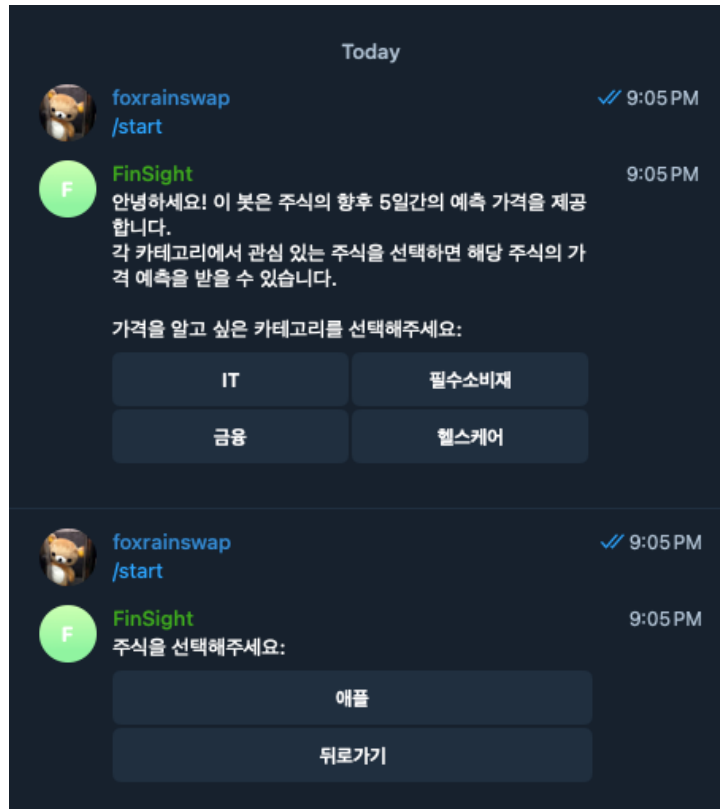
모델 소개



주가 예측

III. 솔루션_서비스

(2) Fin Sight 봇 화면



주가 예측 제공



자동 메시지 발송

III. 솔루션_데모 시연

“주식 투자자를 위한 주가 예측 서비스”

1. 웹 서비스 2. 텔레그램 챗봇 서비스

III. 솔루션_특징 및 장점

종목별 맞춤형 예측

종목별로 최적화된 모델과 데이터셋으로,
사용자에게 보다 **정확하고 신뢰성 높은 주가 예측** 정보를 제공.

접근성

웹과 챗봇 **두 가지의 플랫폼을 통해 손쉽게 접근**할 수 있어,
사용자는 언제 어디서나 주가 예측 서비스를 이용 가능함.

편리성

챗봇으로 **직관적이고 사용자 친화적인 인터페이스**를 제공하고,
자동 메시지 발송 기능으로 **사용자 편의 증가**

III. 솔루션_기대 효과

투자 결정 도움

종목별 최적화된 모델을 사용하여
사용자의 투자 결정에 도움

사용자 만족도 증가

챗봇과 자동화된 주가 예측 메시지 발송 기능을 통해
사용자에게 높은 편리성을 제공하여 서비스 만족도 증가

III. 솔루션_비즈니스

수익/비용 구조

- 수익 구조: **모델 제공 (예측 결과 및 구현 모델)**
- 비용 구조: 서비스 유지 보수 비용

고객 및 가치 제안

- 고객: 개인 투자자, 자산 관리자 등
- 가치 제안: 투자 결정 지원

핵심 활동/자원

- 핵심 활동: 데이터 처리, 서비스 운영
- 핵심 자원: 데이터셋, 예측 모델, 주요 모듈 관리자

참고문헌

- '주가 예측을 위한 Deep Learning: LSTM 및 Transformer 모델' - 이정렬, 구명완
- 'LSTM을 이용한 주가 예측: 금리를 중심으로' - 김은솔, 전재기
- '기술 분석과 환경요소를 이용한 주가 예측률 향상을 위한 딥러닝 병렬 모델' - 황주훈, 김창복
- '감정분석과 GRU 모델 기반의 주가지수 예측 및 분석: 코스피 200 에너지/화학을 중심으로' - 양혁진, 최종민, 박찬정

Q&A