

호가창을 이용한 가상화폐 AI 시스템 트레이딩

프로젝트 중간발표

멘토: 진대한(신한은행)

산업시스템공학과

2018112519 김민재

통계학과

2018111707 권일준

글로벌무역학부

2017113439 이보성

생명과학과

2019111679 이승호

목차

01

■ 제안사항 및 변경점

02

■ 개발 현황

03

■ 향후 계획



추진 배경

"월급만으로 집 어떻게 사요"...밤마다 청년들 북적대는 곳 [2030 신부동산 공식⑥]

<https://www.hankyung.com/article/2024030873296>

부자는 월급이 아니라 투자로 된다. 자유몽 작가 인터뷰

| 입력 : 2019.08.20 10:19:22 | 최종수정 : 2019.08.21 16:18:29



<https://www.mk.co.kr/economy/view.php?sc=50000001&year=2019&no=643329>

필요성

투자 자동화 및 편리성에 대한 수요 증가

투자 과정을 자동화하여 데이터 기반 분석으로 일관된 투자 결정을 지원

리스크 관리 및 감정적 투자 배제

미리 설정된 알고리즘에 따라 감정적 요인을 배제하여 안정적인 수익률 기대 가능

기술 발전과 데이터 분석의 중요성 확대

이를 통해 일반 투자자도 전문가 수준의 분석과 전략으로 시장에 참여가 가능

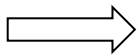
개요

OrderBookservator

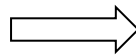
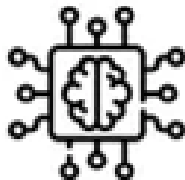
가상화폐 거래소의 **API**를 활용하여 호가창 데이터를 실시간으로 수집 및 분석하여 기계 학습 모델을 학습시켜 실시간 자동 매매 가능 시스템 개발



API 활용 데이터 수집



ML 모델 학습




호가창 데이터 분석 및 자동매매



연관된 특허

공개특허 10-2019-0104919

	(19) 대한민국특허청(KR) (12) 공개특허공보(A)	(11) 공개번호 10-2019-0104919 (43) 공개일자 2019년09월11일
(51) 국제특허분류(Int. Cl.) <i>G06Q 40/04</i> (2012.01) <i>G06N 3/08</i> (2006.01)	(71) 출원인 주식회사 리머스트레이딩 서울특별시 금천구 디지털로9길 68 ,1910호(가산동, 대림포스트타워5차)	
(52) CPC특허분류 <i>G06Q 40/04</i> (2013.01) <i>G06N 3/08</i> (2013.01)	(72) 발명자 이상덕 전라북도 장수군 장수읍 신천로 63, 309호(장수신천아파트)	
(21) 출원번호 10-2019-0083945	(74) 대리인 리앤록특허법인	
(22) 출원일자 2019년07월11일 심사청구일자 2019년07월11일		
전체 청구항 수 : 총 6 항		
(54) 발명의 명칭 주식 자동 매매 시스템 및 방법		

(57) 요약

본 발명은 수익률을 향상시킬 수 있는 주식 자동 매매 시스템 및 방법을 위하여, 사용자 디바이스 및 서비스 제공 서버를 포함하는 주식 자동 매매 시스템에 있어서, 상기 서비스 제공 서버는, 조건식을 통해 매수할 주식 종목을 추출하는 추출 모듈; 상기 추출 모듈에서 추출된 주식 종목을 저장하는 저장 모듈; 상기 저장 모듈에 저장된 주식 종목을 상기 사용자 디바이스로 전송하는 전송 모듈; 상기 추출 모듈에 의해 추출된 주식 종목을 매수하는 매수 모듈; 상기 매수 모듈에 의해 매수된 주식 종목의 1차 매도 가격을 설정하는 제1 설정 모듈; 딥 러닝 모델을 이용하여 상기 제1 설정 모듈이 설정한 상기 1차 매도 가격을 조절하여 2차 매도 가격을 설정하는 제2 설정 모듈; 및 상기 제2 설정 모듈을 통해 설정된 2차 매도 가격에 따라 상기 매수 모듈에 의해 매수된 주식 종목을 매도하는 매도 모듈;을 포함하는, 주식 자동 매매 시스템을 제공한다.

파제의 해결 수단

본 발명의 일 관점에 따르면, 사용자 디바이스 및 서비스 제공 서버를 포함하는 주식 자동 매매 시스템에 있어서, 상기 서비스 제공 서버는, 조건식을 통해 매수할 주식 종목을 추출하는 추출 모듈; 상기 추출 모듈에서 추출된 주식 종목을 저장하는 저장 모듈; 상기 저장 모듈에 저장된 주식 종목을 상기 사용자 디바이스로 전송하는 전송 모듈; 상기 추출 모듈에 의해 추출된 주식 종목을 매수하는 매수 모듈; 상기 매수 모듈에 의해 매수된 주식 종목의 1차 매도 가격을 설정하는 제1 설정 모듈; 딥 러닝 모델을 이용하여 상기 제1 설정 모듈이 설정한 상기 1차 매도 가격을 조절하여 2차 매도 가격을 설정하는 제2 설정 모듈; 및 상기 제2 설정 모듈을 통해 설정된 2차 매도 가격에 따라 상기 매수 모듈에 의해 매수된 주식 종목을 매도하는 매도 모듈;을 포함하는, 주식 자동 매매 시스템이 제공된다.

본 실시예에 있어서, 상기 조건식은, 거래량, 주가 비교, 등락률 및 거래량 비율 중 적어도 어느 하나를 포함하여 도출될 수 있다.

본 실시예에 있어서, 상기 딥 러닝 모델은, LSTM(Long Short-Term Memory), 웨이블릿 트랜스폼(Wavelet Transform) 및 오토인코더(Auto Encoder)를 결합한 모델일 수 있다.

본 실시예에 있어서, 상기 딥 러닝 모델은, 시장 상황을 반영하여 제2 매도 가격을 예측할 수 있다.

본 실시예에 있어서, 상기 1차 매도가격은 상기 2차 매도가격보다 낮거나 될 수 있다.

본 발명의 다른 관점에 따르면, 사용자 디바이스 및 서비스 제공 서버를 포함하는 주식 자동 매매 방법에 있어서, (1) 서비스 제공 서버가 조건식을 통해 매수할 주식 종목을 추출하는 단계; (2) 상기 단계 (1)에서 추출된 주식 종목을 저장하는 단계; (3) 상기 단계 (2)에서 저장된 주식 종목을 사용자 디바이스로 전송하는 단계; (4) 상기 단계 (1)에서 추출된 주식 종목을 매수하는 단계; (5) 상기 단계 (4)에서 매수된 주식 종목의 1차 매도 가격을 설정하는 단계; (6) 딥 러닝 모델을 이용하여 상기 단계 (5)에서 설정된 상기 1차 매도 가격을 조절하여 2차 매도 가격을 설정하는 단계; 및 (7) 상기 단계 (6)에서 설정된 2차 매도 가격에 따라 상기 단계 (4)에서 매수된 주식 종목을 매도하는 단계;를 포함하는, 주식 자동 매매 방법이 제공된다.

전술한 것 외의 다른 측면, 특징, 이점은 이하의 발명을 실시하기 위한 구체적인 내용, 청구범위 및 도면으로부터 명확해질 것이다.

알고리즘을 이용하여 매수할 주식을 결정하고 딥러닝을 이용해 매도 가격을 조정하는 시스템

선행 사례 비교

요소	OBservator	CCXT	pyupbit	gpt-bitcoin	BAMOWL
자동화 수준	조건부 자동화 지원	매매 API 제공	매매 API 제공	AI 기반의 제한적인 자동화	자동화 매매 및 백테스팅 결합
사용자 설정 개입	사용자가 Threshold 직접 설정 가능	X	X	기본 설정 가능	다양한 검증 가능
실시간 데이터	실시간 호가창 데이터 수집	제공	제공	제공	수집 및 제공
데이터 분석	실시간 분석 가능	수동	수동	AI 기반 실시간 분석	딥러닝 기반 예측 및 백테스팅
거래소 지원 범위	다양한 가상화폐 거래소 지원	전세계 주요 거래 소 지원	업비트 지원	제한적 거래소 지원	업비트 지원
안정성 및 성능	높은 처리량으로 인해 과부하 시 지연 가능성 존재	X	X	중간 수준	백테스팅 기반 성능 최적화

적법 여부

1. 거래소 규정 준수



암호화폐 거래소의 **API 사용**
규정을 준수해야 함

2. 투자자문 및 대리매매 관련 법률 (자본시장법 제6조 제1항, 제12조)



다른 사람의 자산을 대신 이용하거나
투자 자문을 제공하기 위해서는 **투자**
일임업으로 등록된 자만 가능함

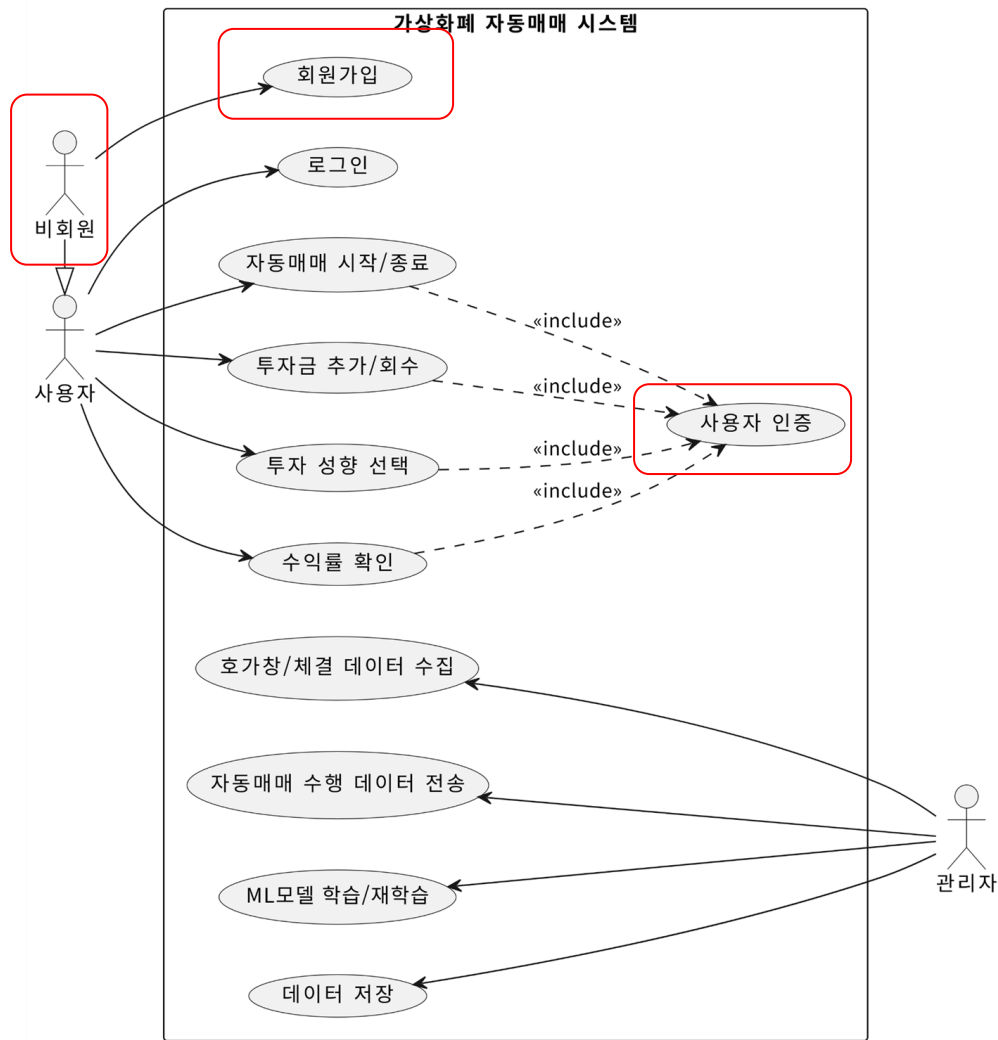
3. 세금 (소득세법 시행령 제 178조의 3)



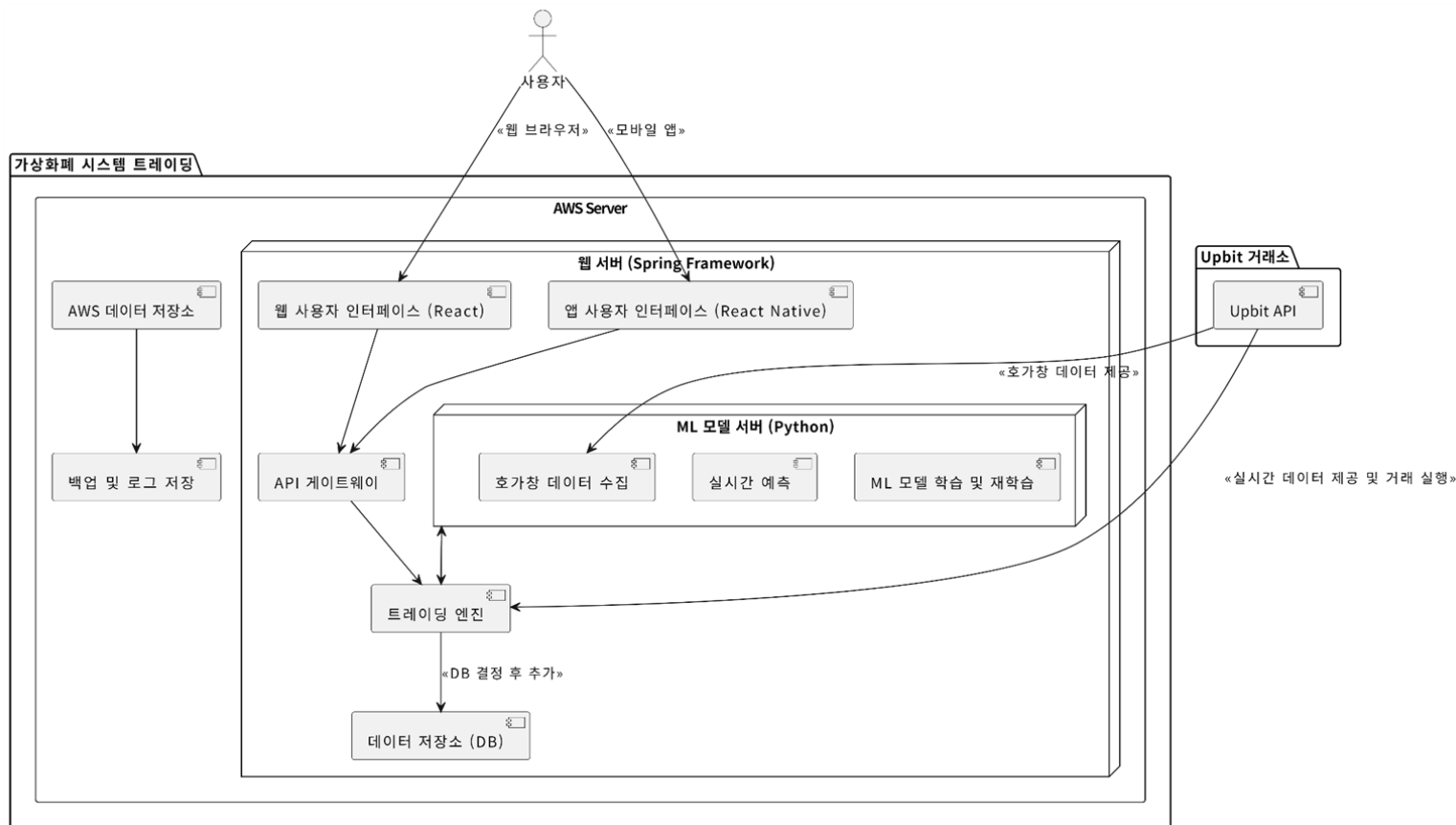
가상자산으로 연 250만원 이상의
소득 발생시 **20%의 세금 부과**
(2025년 1월부터 시행)

유스케이스 다이어그램

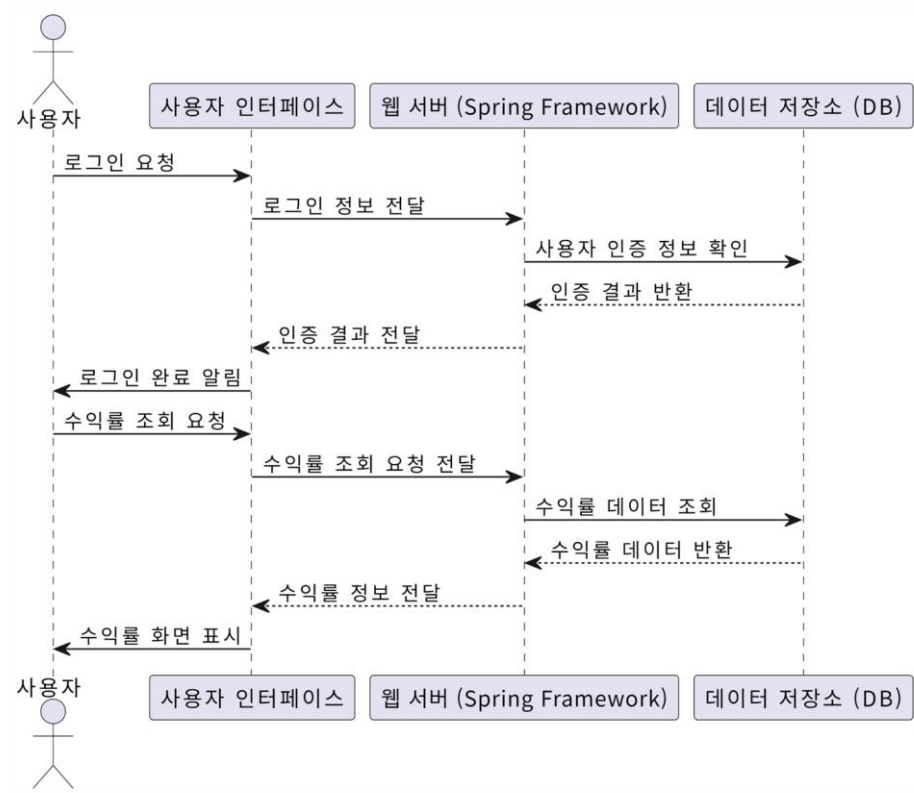
: 변경점



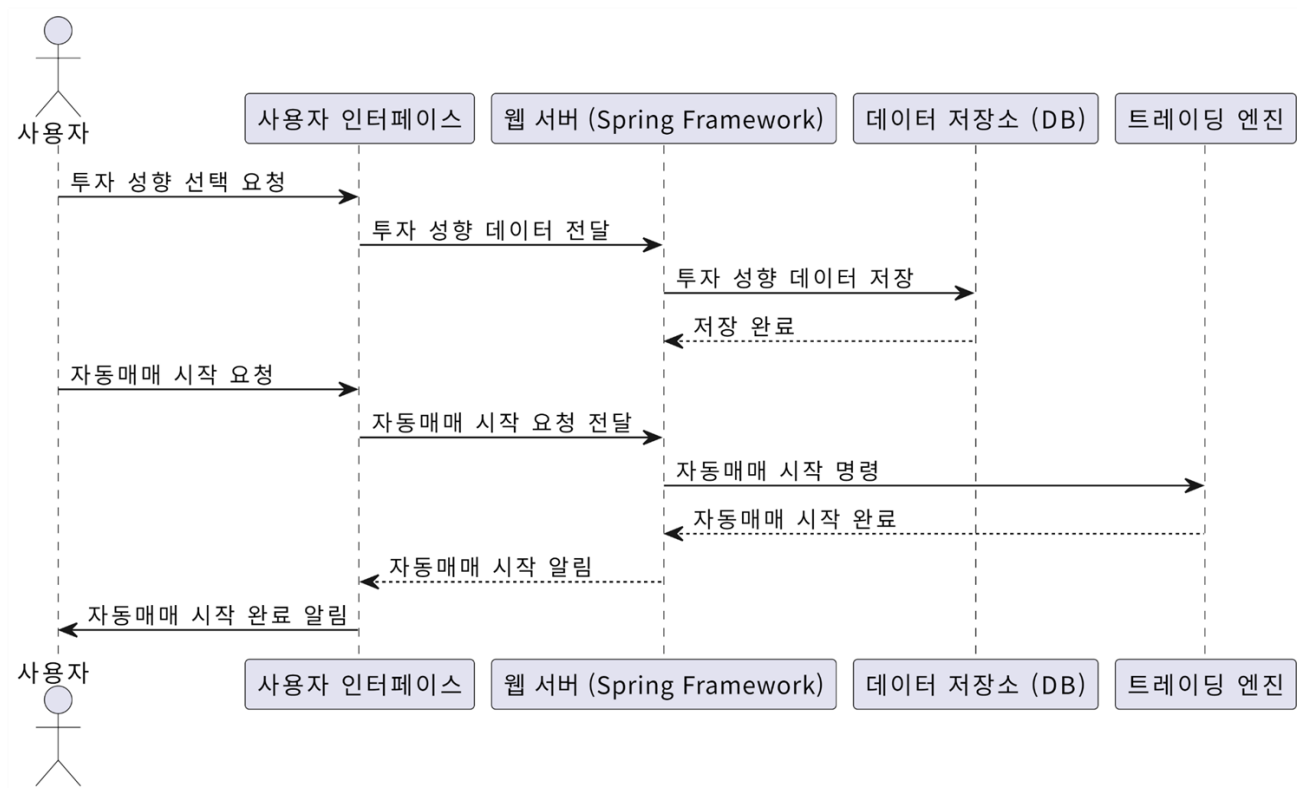
시스템 구성도



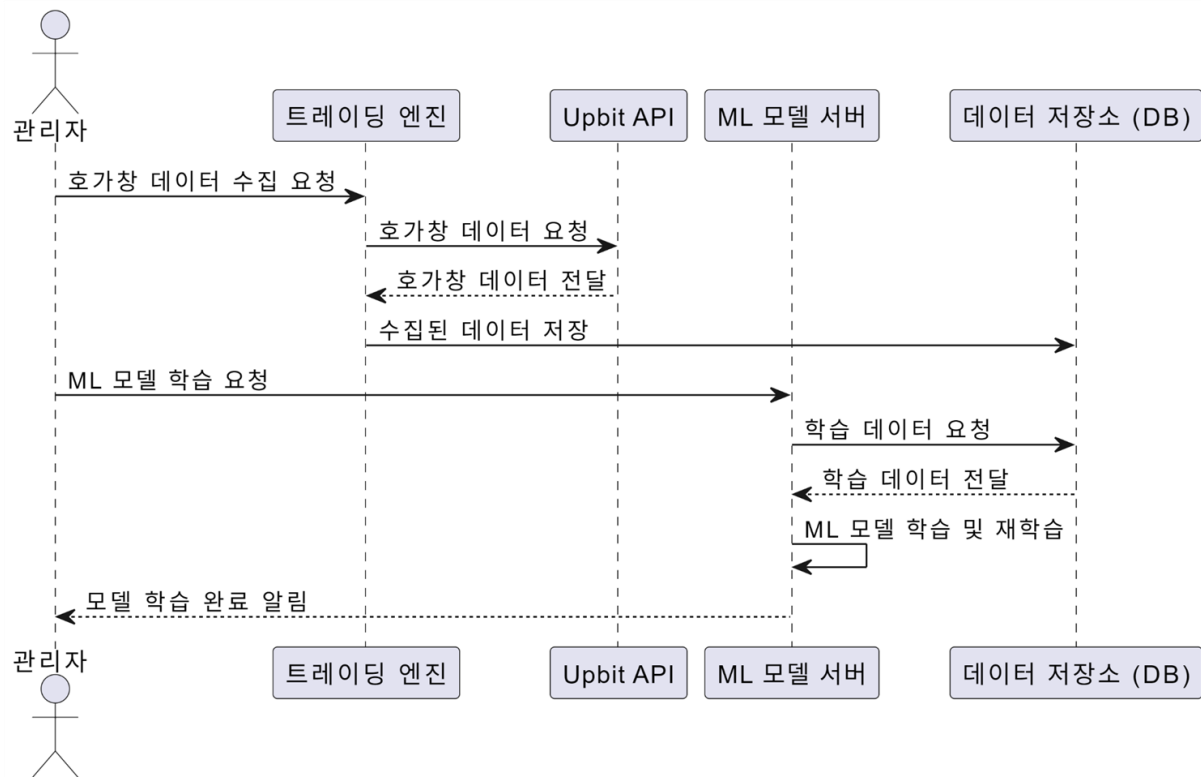
시퀀스 다이어그램 - 사용자 로그인 및 수익률 조회



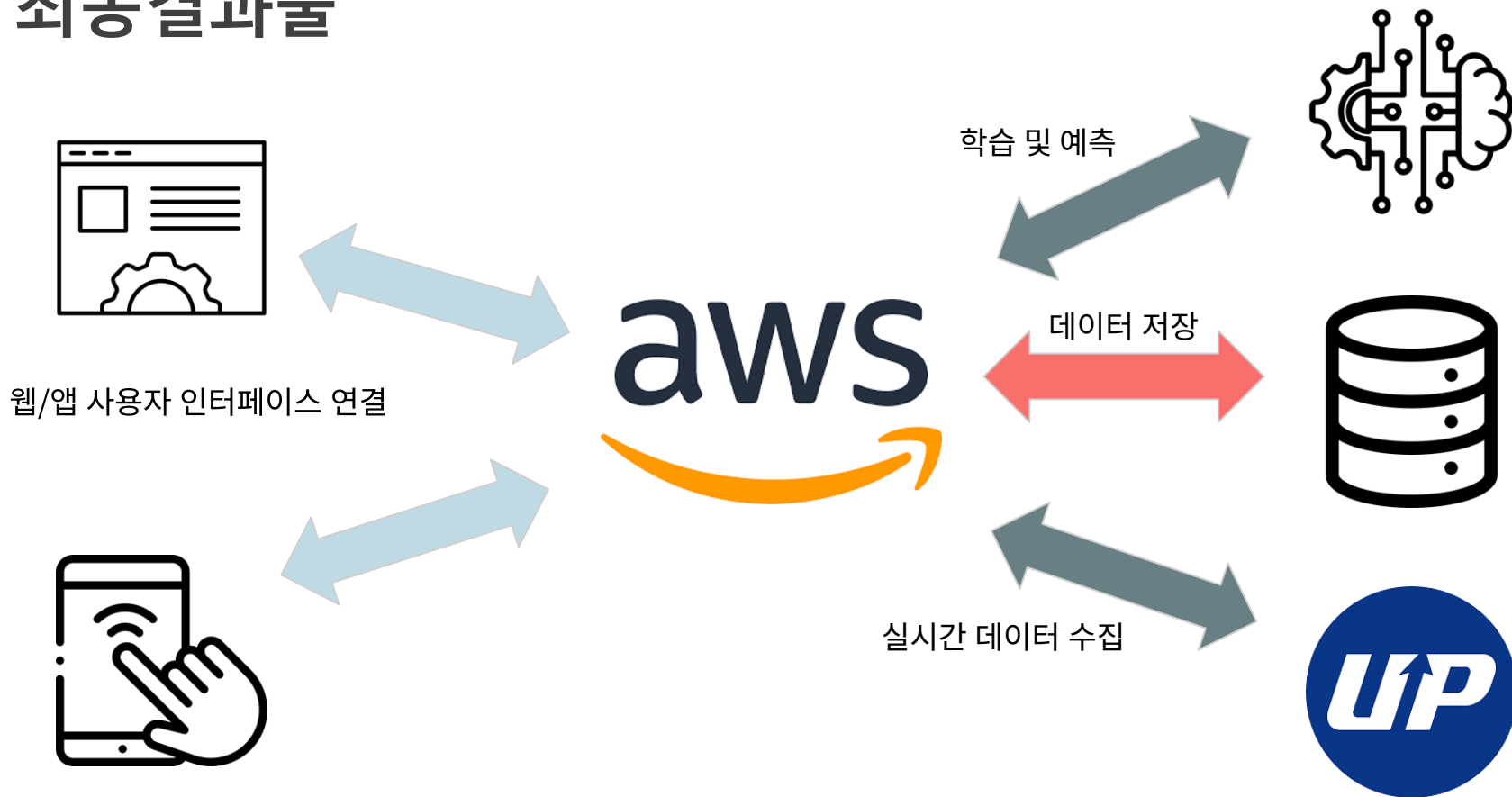
시퀀스 다이어그램 - 투자성향 선택 및 자동매매시작



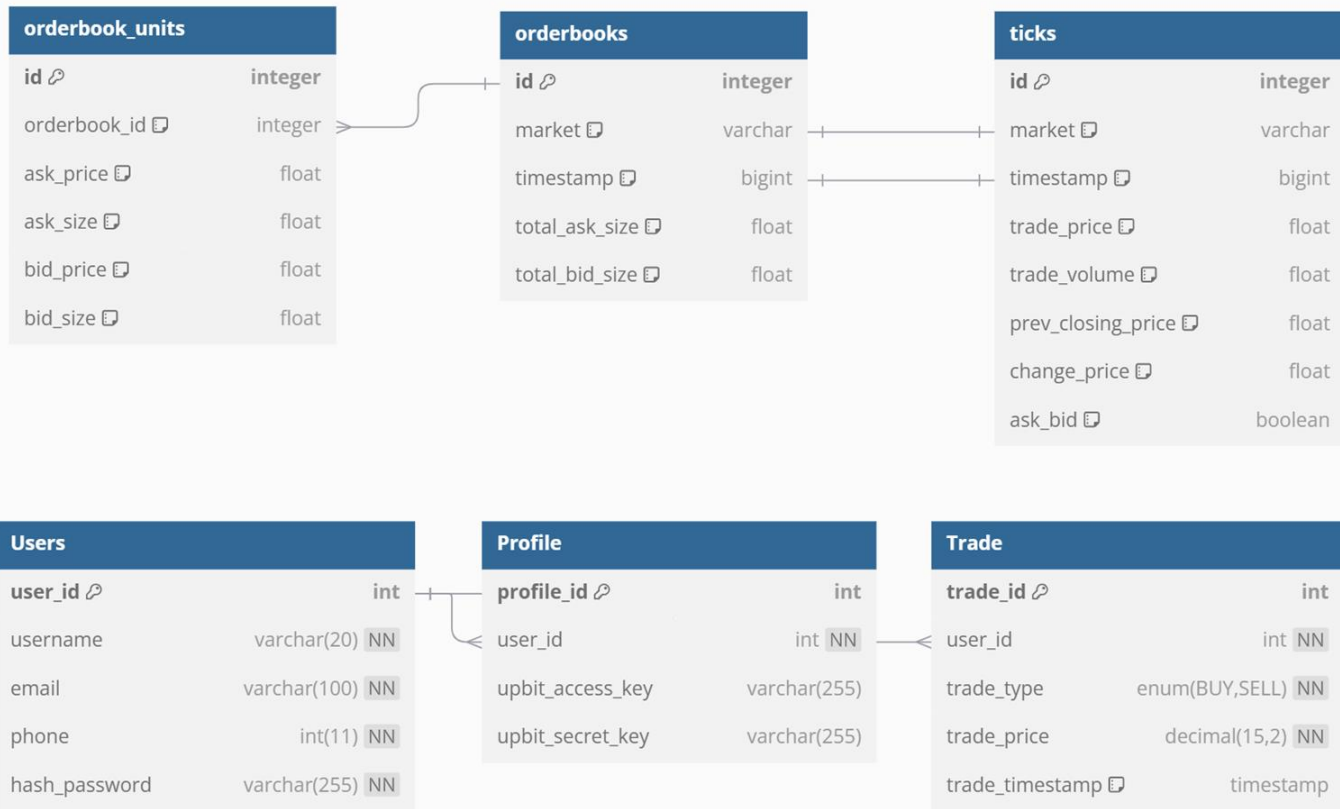
시퀀스 다이어그램 - 데이터 수집 및 ML모델 학습



최종결과물



데이터 스키마



API 명세서

index	대상	Method	URL	기능
1	사용자	POST	/users/join	회원가입
2	사용자	POST	/users/login	로그인
3	사용자	POST	/users/logout	로그아웃
4	사용자	GET	/chart	차트 보기
5	사용자	GET	/orderbook	호가창 보기
6	사용자	PUT	/trade/type	투자 성향 선택
7	사용자	PUT	/trade/start	자동매매시작
8	사용자	PUT	/trade/stop	자동매매종료
9	사용자	GET	/portfolio	개인 포트폴리오 확인
10	사용자	GET	/portfolio/check	수익률 확인
11	사용자	PATCH	/portfolio/addbudget	투자금 추가
12	사용자	PATCH	/portfolio/recievebudget	투자금 회수

ML 모델 비교

특징	LSTM	GRU	LightGBM	Random Forest	1D CNN
시계열 데이터 처리	우수	우수	보통	낮음	우수
비선형성 학습	우수	우수	우수	우수	우수
대용량 데이터 처리	보통	보통	우수	보통	우수
학습 속도	느림	보통	빠름	보통	빠름
실시간 예측 능력	보통	우수	우수	보통	우수
해석 용이성	낮음	낮음	낮음	우수	낮음

개발 환경

Frontend(Web)



Backend(Web)



Server



Moblie



ML model



Database



Amazon
S3



Amazon
RDS

Operating System



제안사항 - FE

사용자 피드백 수집

개발된 UI와 기능에 대한 실제 사용자 피드백을 받아야 함. 친구들이나 팀원들에게 간단한 피드백 요청해서 기능이 잘 작동하는지, 불편한 점이 있는지 확인할 필요 있음. 실사용자 피드백을 통해 개선점을 빠르게 찾을 수 있음.

디버깅 및 오류 추적 시스템 도입

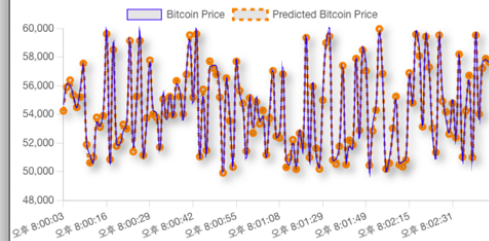
예상치 못한 오류가 발생했을 때 자동으로 기록하고 쉽게 확인할 수 있는 시스템을 도입해야 함. 단순히 `console.log` 찍는 것보다 Sentry 같은 오류 추적 도구를 사용하면 오류 위치와 원인을 쉽게 파악 가능함. 이를 통해 문제를 신속히 해결할 수 있고, 코드 안정성도 높아짐.

코드 리뷰와 정기적인 피드백

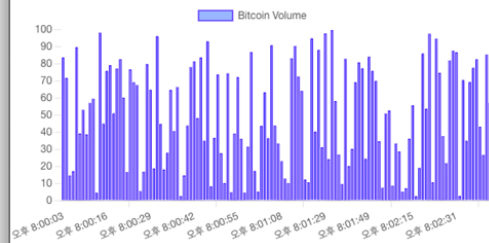
작성한 코드에 대해 팀원 간 피드백을 주고받는 코드 리뷰가 필요함. 코드 리뷰를 통해 문제점이나 개선할 부분을 발견하고, 더 나은 코드 작성법을 배울 수 있음. 정기적인 리뷰를 통해 코드 품질을 높이고, 피드백을 반영해 조금씩 개선해 나가는 것이 좋음.

테스트이보성님 안녕하세요

Bitcoin Price vs Predicted Price



Bitcoin Volume



Trade History (Dummy) - Latest 15

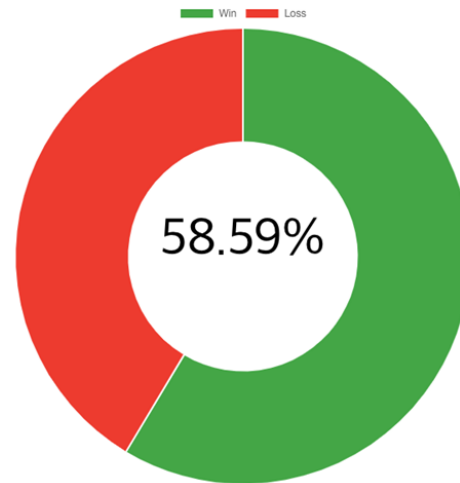
Time	Price (USD)	Volume	Action	Result
오후 8:02:28	54664.71	83.51	Buy	Win
오후 8:02:29	56100.82	71.56	Buy	Win
오후 8:02:30	56024.14	14.39	Buy	Win
오후 8:02:31	55216.53	17.03	Buy	Win
오후 8:02:33	54493.01	89.48	Sell	Lose
오후 8:02:35	55692.09	39.04	Sell	Lose
오후 8:02:37	57221.3	52.70	Sell	Lose
오후 8:02:39	51981.56	38.57	Buy	Win
오후 8:02:41	51121.99	56.73	Sell	Lose
오후 8:02:43	50656.64	59.23	Sell	Lose
오후 8:02:45	53420.33	4.35	Buy	Win
오후 8:02:47	53175.84	97.88	Buy	Win
오후 8:02:49	54359.29	44.82	Sell	Lose
오후 8:02:50	59848.71	75.59	Sell	Lose
오후 8:02:51	50619.73	79.01	Buy	Win

Trade Metrics

Win Rate: 58.59%

Average Holding Time: 28.63 minutes

Win Rate



개발현황 - FE

차트 UI 요소 개발 진행 상황 : 80% 완료

개발 주요 사항: Bitcoin Tracker 페이지의 주요 화면 요소 개발을 완료했고, Trade History와 Total Profit 부분을 수정 중.

이슈: Trade History에 '거래 결과' 표시 칸을 추가하고, Total Profit의 오르내림 기호 위치를 고정하는 작업이 필요
다음 작업: 디자인 검토 후 마지막 스타일링 작업을 마무리

데이터 상태 관리 진행 상황: 60% 완료

주요 작업: Redux 또는 Context API를 사용해 데이터를 관리하고, 컴포넌트 간 데이터 이동을 최적화

현재 이슈: 데이터가 변할 때 업데이트가 느려지는 문제

다음 작업: 데이터 업데이트 속도 개선

기능 테스트 및 오류 수정 진행 상황: 40% 완료

주요 작업: 각 기능이 잘 작동하는지 테스트 중, UI 요소와 데이터 연동 오류 수정

현재 이슈: 테스트 범위가 부족한 컴포넌트 존재

다음 작업: 테스트 범위 확대 및 통합 테스트 완료

향후 개발 일정 - FE

11월 1주차: 코드 리뷰 및 리팩토링

- 코드 리뷰 세션 진행: 팀원과 코드 리뷰를 통해 개선점 찾기
- 피드백 반영하여 코드 리팩토링 및 구조 개선
- 반복되는 코드 모듈화 작업 및 코드 정리

11월 2주차: 디버깅 및 성능 최적화

- 오류 추적 시스템 도입: **Sentry**와 같은 도구 설정 및 적용
- 코드 개선: 불필요한 코드 최적화, 데이터 업데이트 속도 개선
- 데이터 캐싱과 같은 성능 최적화 방법 도입 검토

11월 3주차: 테스트 범위 확대 및 기능 안정화

- 테스트 범위 확대: 기능별 테스트 코드 작성 및 통합 테스트 진행
- 발견된 버그 수정 및 기능 안정화 작업
- 모바일 및 다양한 기기에서 UI 테스트 및 최적화

11월 4주차: 최종 점검 및 배포 준비

- 최종 기능 점검: 모든 기능과 UI 요소가 정상 작동하는지 확인
- 배포 전 최종 테스트 진행
- 배포 준비 및 배포 완료

개발 현황 - ML (1) 학습에 사용한 데이터에 대한 설명

11/1 호가창 데이터의 일부

```
{'timestamp': 1730386800033, 'total_ask_size': 1.65789057, 'total_bid_size': 0.72023082, 'orderbook_units': [{'ask_price': 98679000, 'bid_price': 98600000, 'ask_size': 0.0238942, 'bid_size': 0.00951805}, {'ask_price': 98681000, 'bid_price': 98599000, 'ask_size': 0.02907236, 'bid_size': 0.00507099}, {'ask_price': 98682000, 'bid_price': 98597000, 'ask_size': 0.22824739, 'bid_size': 0.05060847}, {'ask_price': 98685000, 'bid_price': 98592000, 'ask_size': 0.24268735, 'bid_size':
```



0.2초에 한번씩 데이터를 호출 및 저장

11/1 거래 데이터의 일부

```
{'timestamp': 1730386799973, 'trade_price': 98679000.0, 'trade_volume': 0.00010133, 'prev_closing_price': 100571000.0, 'change_price': -1892000.0, 'ask_bid': 'BID'} {'timestamp': 1730386801723, 'trade_price': 98679000.0, 'trade_volume': 8.612e-05, 'prev_closing_price': 100571000.0, 'change_price': -1892000.0, 'ask_bid': 'BID'} {'timestamp': 1730386802424, 'trade_price': 98592000.0, 'trade_volume': 0.05900239, 'prev_closing_price': 100571000.0, 'change_price': -1979000.0, 'ask_bid': 'ASK'}
```



거래가 체결될 때 마다 데이터를 호출 및 저장

호가창 데이터

변수명	설명
timestamp	호가 정보가 생성된 시각을 밀리초 단위로 나타낸 값
total_ask_size	현재 매도 호가에 등록된 총 매도 잔량
total_bid_size	현재 매수 호가에 등록된 총 매수 잔량
orderbook_units	각 호가 단위의 상세 정보를 담고 있는 리스트로, 상위 15개의 매수/매도에 대한 정보를 포함
ask_price	해당 호가 단위의 매도 호가 가격
bid_price	해당 호가 단위의 매수 호가 가격
ask_size	해당 호가 단위에서의 매도 잔량
bid_size	해당 호가 단위에서의 매수 잔량

거래 데이터

변수명	설명
timestamp	체결 데이터가 생성된 시각을 밀리초 단위로 나타낸 값
trade_price	이번 체결의 거래 가격
trade_volume	이번 체결에서 거래된 암호화폐의 수량
prev_closing_price	이전 거래일의 종가 (24시간 전)
change_price	이전 종가 대비 현재 거래 가격의 변동액
ask_bid	체결 방향을 나타내며, 'ASK'는 매도, 'BID'는 매수를 의미

개발 현황 - ML (2) feature 추출

```
# orderbooks 데이터로부터 feature 추출 (가격 스프레드, 중간 가격, 호가창 불균형)
def preprocess_orderbook(orderbook_df):
    orderbook_df['spread'] = orderbook_df['orderbook_units'].apply(lambda x: x[0]['ask_price'] - x[0]['bid_price'])
    orderbook_df['mid_price'] = orderbook_df['orderbook_units'].apply(lambda x: (x[0]['ask_price'] + x[0]['bid_price']) / 2)
    orderbook_df['obi'] = (orderbook_df['total_bid_size'] - orderbook_df['total_ask_size']) / \
        (orderbook_df['total_bid_size'] + orderbook_df['total_ask_size'])
    return orderbook_df[['timestamp', 'spread', 'mid_price', 'obi']] # 필요한 feature만 반환

# ticks 데이터에서 추가적인 feature 및 target 추출 (가격 변동성, 매수/매도 비율, 체결 가격)
def preprocess_ticks(ticks_df):
    ticks_df['price_volatility'] = ticks_df['trade_price'].rolling(window=5).std()
    ticks_df['buy_sell_ratio'] = ticks_df['ask_bid'].apply(lambda x: 1 if x == 'BID' else 0).rolling(window=5).mean()
    return ticks_df[['timestamp', 'trade_price', 'price_volatility', 'buy_sell_ratio']] # 필요한 feature만 반환
```

Orderbook Features

Feature	설명
spread	최우선 매도 호가와 매수 호가의 차이로, 시장에서의 거래 가격 차이를 나타내는 값
mid_price	최우선 매도 호가와 매수 호가의 중간값으로, 현재 시장의 균형 가격을 나타내는 값
obi	호가창 불균형 지수(Order Book Imbalance)로, 매수와 매도 잔량의 균형 상태를 나타내며, 값이 클수록 매수 또는 매도의 강세를 의미

Ticks Features

Feature	설명
price_volatility	체결 가격의 표준 편차(rolling window=5)로, 일정 기간 동안의 가격 변동성을 나타내는 값
buy_sell_ratio	매수 체결의 비율로, 일정 기간 동안 매수(BID) 비율을 나타내며 매수와 매도의 강도를 비교하는 값

Target	설명
trade_price	각 체결의 거래 가격으로, 예측 대상이 되는 목표 값

개발 현황 - ML (3) hyperparameter 조정 및 학습

하이퍼파라미터 튜닝 범위

epoch_range = [10, 30, 50, 100]

hidden_size_range = [50, 100, 200]

num_layers_range = [1, 2, 3]

sequence_length_range = range(30, 121, 30)

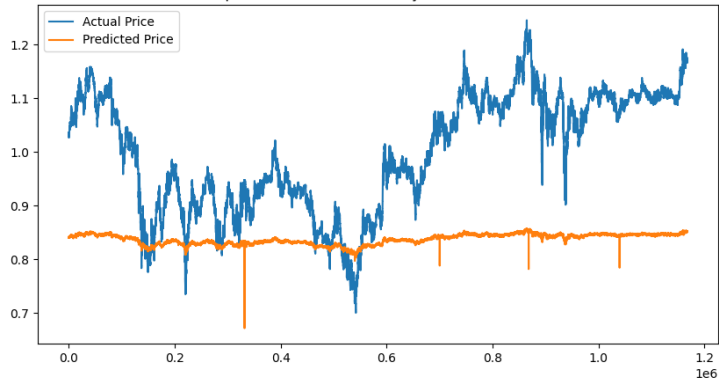
learning_rates = [0.001, 0.003, 0.005, 0.01, 0.03]

하이퍼파라미터	설명
epoch_range	모델 학습을 진행할 총 epoch 수
hidden_size	각 LSTM 레이어에 사용할 뉴런의 수
num_layers	LSTM 레이어의 층 수
sequence_length	미래 가격을 예측할 때 사용할 과거 데이터의 수
learning_rate	모델이 가중치를 업데이트할 때 이동하는 보폭의 크기

조합 시도 중 num_layers ≥ 2 인 경우 오차가 비교적 매우 큰 것을 확인 → num_layers=1로 고정

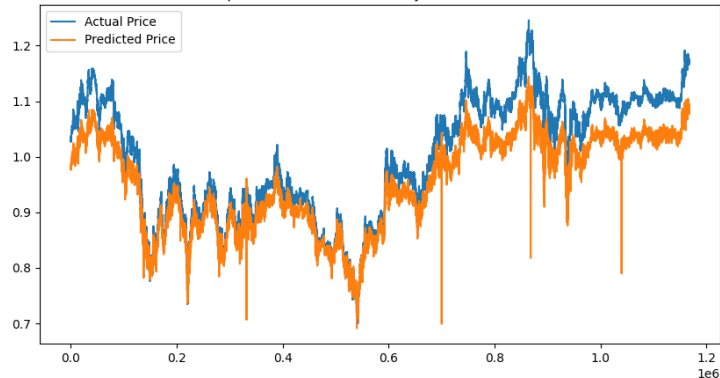
num_layer=2

Actual VS Predicted Price
Test Loss: 10.3436
Epochs: 10, Hidden: 50, Layers: 2, LR: 0.003



num_layer=1

Actual VS Predicted Price
Test Loss: 0.7043
Epochs: 10, Hidden: 50, Layers: 1, LR: 0.003



개발 현황 - ML (3) hyperparameter 조정 및 학습

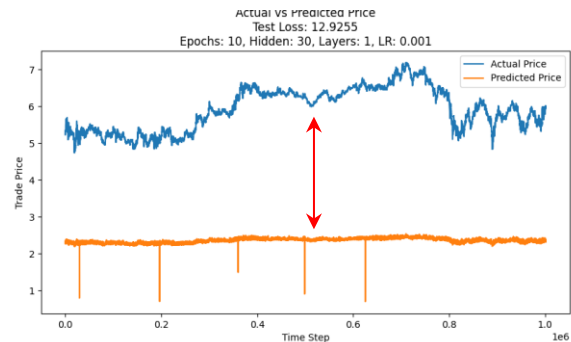
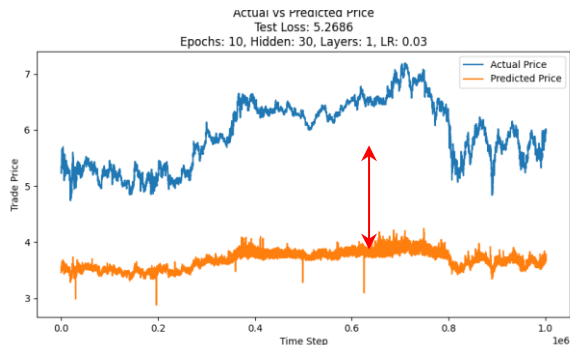
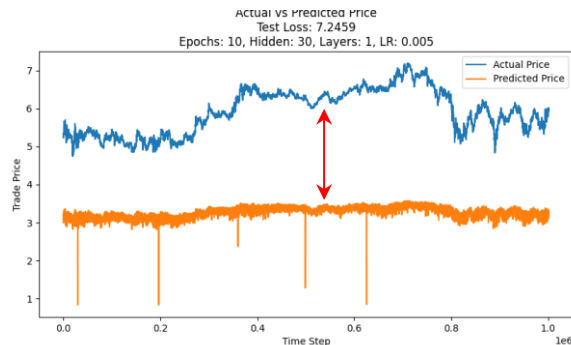
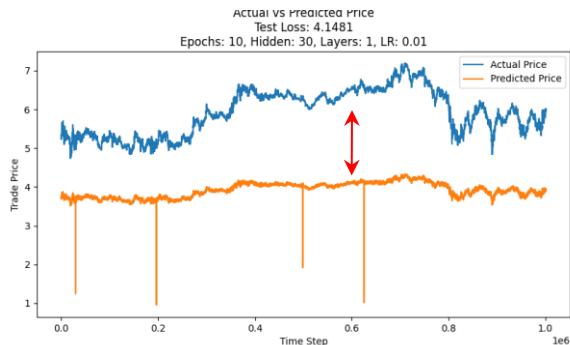
학습 결과 test loss가 가장 낮은 20개의 조합을 바탕으로 evaluation 진행

	file_suffix	test_loss
1	e100_s60_h100_l1_lr0.01_1028_robust	1.31E-05
2	e10_s30_h200_l1_lr0.003_1028_robust	1.35E-05
3	e100_s60_h200_l1_lr0.005_1028_robust	1.72E-05
4	e10_s90_h100_l1_lr0.01_1028_robust	1.97E-05
5	e10_s60_h200_l1_lr0.003_1028_robust	2.98E-05
6	e10_s90_h200_l1_lr0.003_1028_robust	3.28E-05
7	e30_s60_h50_l1_lr0.03_1028_robust	3.54E-05
8	e30_s30_h200_l1_lr0.005_1028_robust	4.53E-05
9	e30_s90_h100_l1_lr0.03_1028_robust	4.57E-05
10	e30_s90_h200_l1_lr0.01_1028_robust	5.00E-05
11	e30_s60_h200_l1_lr0.005_1028_robust	5.27E-05
12	e30_s30_h100_l1_lr0.03_1028_robust	6.09E-05
13	e10_s120_h200_l1_lr0.03_1028_robust	6.19E-05

개발 현황 - ML (4) 이슈 발생!

evaluation data에 대해 예측을 수행, 실제 데이터와 비교

전반적으로 가격을 낮게 예측하는 현상 발생!



개발 현황 - ML (5) 해결 시도 - input feature 변경

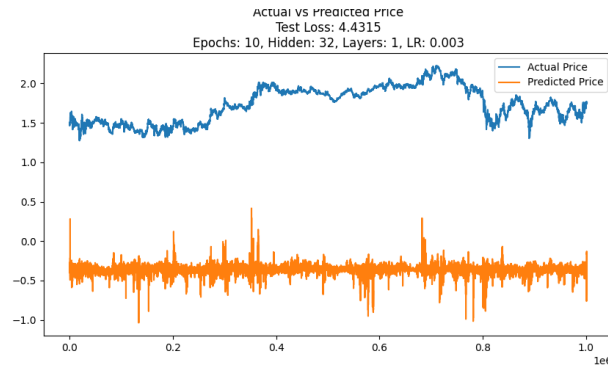
호가창에서 각 level별 호가 가격과 매수/매도 잔량을 곱해 feature로 직접 사용

```
# 각 레벨별로 ask_price * ask_size와 bid_price * bid_size를 추출하는 함수
def preprocess_orderbook(orderbook_df):

    for level in range(15):
        # 각 레벨의 ask와 bid에 대한 feature 생성
        orderbook_df[f'ask_price_size_level_{level+1}'] = orderbook_df['orderbook_units'].apply(
            lambda x: x[level]['ask_price'] * x[level]['ask_size'] if len(x) > level else 0
        )
        orderbook_df[f'bid_price_size_level_{level+1}'] = orderbook_df['orderbook_units'].apply(
            lambda x: x[level]['bid_price'] * x[level]['bid_size'] if len(x) > level else 0
        )
    orderbook_df['obi'] = (orderbook_df['total_bid_size'] - orderbook_df['total_ask_size']) / \
        (orderbook_df['total_bid_size'] + orderbook_df['total_ask_size'])

    # 각 레벨의 ask_price_size와 bid_price_size만 포함한 DataFrame 반환
    return orderbook_df[['timestamp', 'obi'] +
        [f'ask_price_size_level_{i+1}' for i in range(15)] +
        [f'bid_price_size_level_{i+1}' for i in range(15)]]
```

예측 성능이 더 하락, 경향성조차 따라가지 못함



feature에 문제가 있는 것이 아니라는 판단 → 이전 feature를 사용하기로 결정

개발 현황 - ML (6) 문제의 원인 파악

9/24~10/30 까지의 비트코인 가격



10/11 ~ 10/30 기간동안 비트코인 가격이 급상승



시계열 데이터 특성상 과거의 데이터를 바탕으로 미래 데이터를 예측



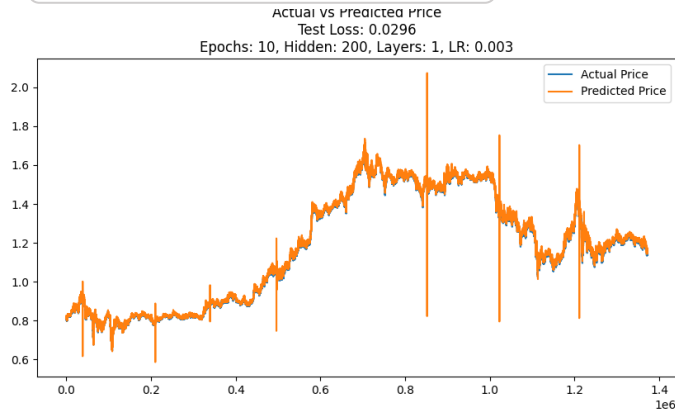
학습에 사용된 scaler가 감당하기에 가격이 과하게 상승



최대한 넓은 범위의 데이터가 쌓일 때까지 잠시 대기하기로 결정

개발 현황 - ML (7) 데이터 누적 후 재시도

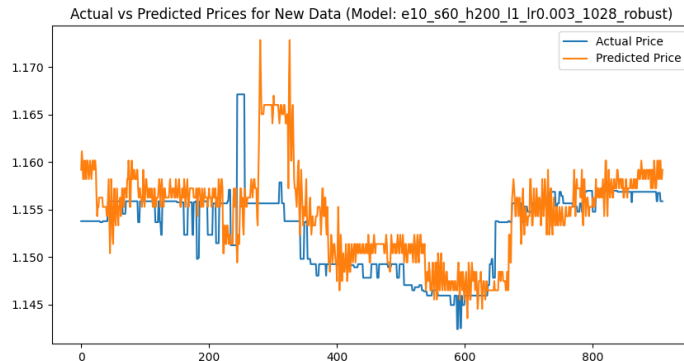
11/3 까지 데이터 누적 후 test 결과



Issue 1

전반적으로 예측을 잘 수행하지만, 값이 튀는 현상 발생

최근 1000개의 데이터만 plotting



Issue 2

Trend는 따라가나, 30 timestamp만큼 밀린상태로 예측

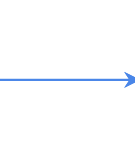
향후 계획 - ML

Issue 1

전반적으로 예측을 잘 수행하지만, 값이 튀는 현상 발생

Issue 2

Trend는 따라가나, 30 timestamp만큼 밀린상태로 예측

- 
1. 하이퍼파라미터 fine-tuning
 2. batch size 변경
 3. 변수 중요도 추출 및 가중치 부여
 4. 추가로 고려할 feature 식별

추가 개발사항

1. 현재는 5초 뒤의 가격만을 예측하지만, 예측 정확도 확보 후 10, 30, 60초 뒤의 가격을 추가로 예측하도록 알고리즘 변경
2. 비트코인의 가격이 급변하는 상황이 다시 발생하더라도 예측 정확도가 떨어지는 것을 최소화하는 방법 고안
3. 가격 상승을 예측했을 때 뿐만 아니라 가격 하락을 예측했을 때에도 숏 포지션을 활용해 추가적인 이익을 얻도록 거래 전략 다각화

개발현황 - BE

웹페이지 기본 기능

개발 주요 사항: 로그인/로그아웃/회원가입/정보입력 기능 추가, 로그인 시에만 투자확인 및 투자시작을 할 수 있도록 조정

다음 작업 : 별도로 개발중인 FE와 BE를 추후 연결 예정

Upbit API 활용 기능

개발 주요 사항: 암호화폐 매수/매도 및 잔고 확인 기능 추가

이슈: 자동매매 도입시 이미 주문을 대기하기 위해 준비되어 있는 명령 변경 방법 모색

다음 작업 : ML 모델에서의 신호를 받아 자동매매하는 기능 추가, 거래완료내역 조회 및 총 수익 확인 기능 추가

유저 데이터 저장

개발 주요 사항: 회원가입 및 입력 기능 사용시 정보 저장, 비밀번호의 경우 hashing을 통해 보안 강화

다음 작업 : 현재 테스트용 java 내 휘발성 저장소인 h2를 사용중에 있으며, 추후 DB 환경을 갖추어 연동 예정

로그인

아이디

비밀번호

☐ 아이디 저장

[아이디 찾기](#) | [비밀번호 찾기](#)

로그인

[회원가입](#)

회원가입

아이디

특수문자 없이 4자~20자로 입력해주세요.

이메일

@

도메인 선택

연락처: 010 - -

비밀번호



특수문자를 포함한 8~16자로 입력해주세요.

비밀번호 확인



☐ (필수) 개인정보 수집 및 활용 동의

[약관보기](#)

☐ (선택) 이메일 알림 서비스 수신 동의

가입하기

이미 계정이 있으신가요? [로그인](#)

Update Profile

Upbit Keys Set: No

Username:

Upbit Access Key:

Upbit Secret Key:

[Save Changes](#)

Upbit Balance Information

Please input your Upbit API keys to view balance.

[Go Back to Home](#)

Trade Bitcoin

Price to Buy (KRW):

[Buy Bitcoin](#)

Volume to Sell (BTC):

[Sell Bitcoin](#)

[Back to Home](#)

SKT 11:17

47%

< 주 비트코인(BTC/KRW)



94,644,000
0.38% ▲355,000



BINANCE 93,740,151 (€62,537.21) COINBASE 93,635,209 (€62,467.20)

주문 호가주문 차트 시세 정보

주문	호가주문	차트	시세	정보
0.39%	0.098		매수	매도
94,655,000	0.39%		미체결	체결
94,652,000	0.38%			
94,651,000	0.011			
94,650,000	0.324			
94,649,000	0.590			
94,648,000	0.086			
94,644,000	0.444			
94,632,000	0.010			
94,624,000	0.018			
94,623,000	0.005			
94,622,000	0.119			
94,621,000	0.000			
94,620,000	0.129			
94,619,000	0.026			
모아보기	수량			

거래내역

매수

매도

미체결

체결

하

매도

10.31 14:18

BTC/KRW

100,481,000

0.00010000

10,048

매수

10.31 14:11

BTC/KRW

100,445,000

0.00004977

5,000

매수

10.31 03:25

BTC/KRW

99,953,000

0.00005002

5,000

매수

10.17 18:04

BTC/KRW

91,950,000

0.00054350

49,975



개발현황 - 서버

서버 자동화

진행 상황: 90% 완료

개발 주요 사항: 호가창/체결 데이터 수집 및 1차 전처리, 업로드 자동화

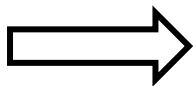
이슈: GPU의 부재(c5.large 인스턴스)로 인해 서버 자원으로는 ML 불가

다음 작업: 학습된 ML 모델 탑재, 웹 BE 연결



Upbit API

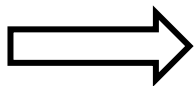
데이터 수집



Amazon EC2

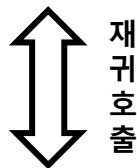
(1차 전처리 및 CSV 변환)

업로드(Crontab)

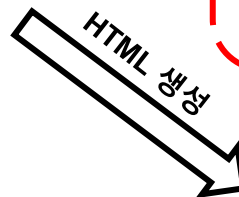


Amazon S3

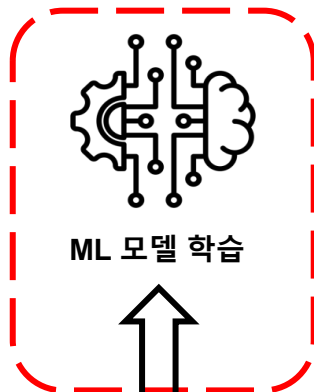
(raw, CSV 데이터 저장)



재귀 호출



HTML 생성



ML 모델 학습



데이터 다운로드 링크

개발현황 - 서버

서버 구조 (트리)

/home/ubuntu/project

```
├── codes
│   ├── etc
│   └── main
├── data
│   ├── csv
│   │   ├── orderbooks
│   │   └── ticks
│   ├── key
│   ├── queue
│   ├── raw
│   │   ├── orderbooks
│   │   └── ticks
└── logs
    ├── crontab.log
    └── date
```

Amazon S3 > 버킷 > observer-bucket

observer-bucket 정보

객체 | 속성 | 권한 | 지표 | 관리 | 액세스 지점

객체 (5) 정보 🔄 📄 S3 URI 복사 📄 URL 복사 📄 다운로드 📄 열기 삭제 작업 ▼ 폴더 만들기 업로드

객체는 Amazon S3에 저장되어 있는 기본 엔티티입니다. [Amazon S3 인벤토리](#)를 사용하여 버킷에 있는 모든 객체의 목록을 알 수 있습니다. 다른 사용자가 객체에 액세스할 수 있게 하려면 명시적으로 권한을 부여해야 합니다. [자세히 알아보기](#)

🔍 접두사로 객체 찾기 🔄 버전 표시 < 1 > ⚙️

<input type="checkbox"/>	이름	유형	마지막 수정	크기	스토리지 클래스
<input type="checkbox"/>	aws-programmatic-access-test-object	-	2024. 11. 4. pm 10:06:22 PM KST	4.0B	Standard
<input type="checkbox"/>	csv/	폴더	-	-	-
<input type="checkbox"/>	index.html	html	2024. 11. 4. pm 11:07:31 PM KST	19.3KB	Standard
<input type="checkbox"/>	raw/	폴더	-	-	-
<input type="checkbox"/>	report/	폴더	-	-	-


Lambda > 함수 > observer-s3index


observer-s3index


조정 ARN 복사 작업 ▼

▼ 함수 개요 정보 Application Composer로 내보내기 다운로드 ▼

다이어그램 템플릿

 observer-s3index

 Layers (0)

 S3

+ 대상 추가

+ 트리거 추가

설명

-

마지막 수정

41분 전

함수 ARN

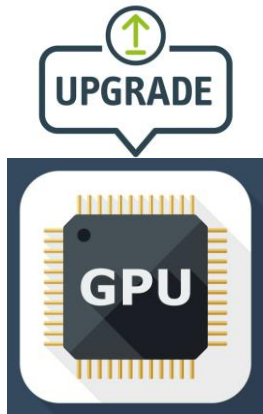
arn:aws:lambda:ap-northeast-2:637423438820:function:observer-s3index

함수 URL 정보

-

개선사항 - 서버

1. EC2 유형 업그레이드



현재 서버에 할당된 리소스로는 2개의 CPU뿐이기에 ML 모델 탑재를 위해서는 GPU가 포함된 유형으로의 인스턴스 업그레이드가 필요

2. Lambda 함수 최적화



S3 버킷의 파일목록을 계속해서 갱신하는 재귀호출이 반복되어 불필요한 로그 데이터가 계속 쌓이기에 최적화 필요

3. 웹 BE를 위한 DB 추가



BE 연결후, 가입한 회원들의 정보/거래를 위한 API 키와 비밀번호 등을 hashing한 값을 저장하는 DB 구축

제약조건 및 대안 방안

1. 하드웨어 성능 문제



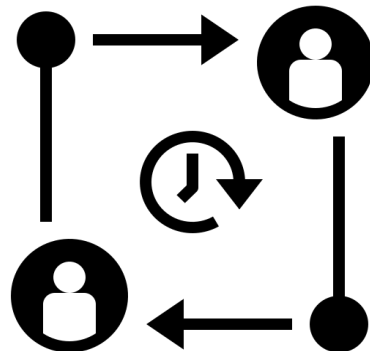
ML을 돌리기 위한 hardware이 부족한 경우 Google Cloud TPU 같은 **가상 하드웨어**를 구매해서 사용 가능

2. 과적합 문제



lap time을 활용하여 너무 세세하게 정보가 들어가지 않게 조절
lap time의 크기를 조절하여 가장 적절한 경우를 찾기

3. 실시간 상황 반영



매매 수치의 변경은 새로운 데이터를 모아 **일정 시간마다 시행**하여 실시간 처리의 지연을 해소

기대효과

1. 경제적 측면



일반 투자자들도 전문적인 투자 전략을 활용할 수 있게 하여 수익률 향상에 기여

2. 사회적 측면



일반 대중이 인공지능 기술을 일상 생활에 활용함으로써 기술에 대한 거부감이 줄고, 사회 전반의 기술 수용성이 높아짐

3. 리스크 관리 측면



AI를 활용한 정교한 분석으로 금융 시장의 위험 요소를 사전에 감지하고 대응

성과 도출 계획



현실적인 문제로 특허 진행이 어려울 수 있지만
멘토와 상의 후 검토할 예정

멘토&팀원 역할배분

김민재

팀장, PM
ML 모델개발

권일준

BE, 문서 작업
ML 모델 개발 지원

이보성

앱 개발, ML모델
개발 지원

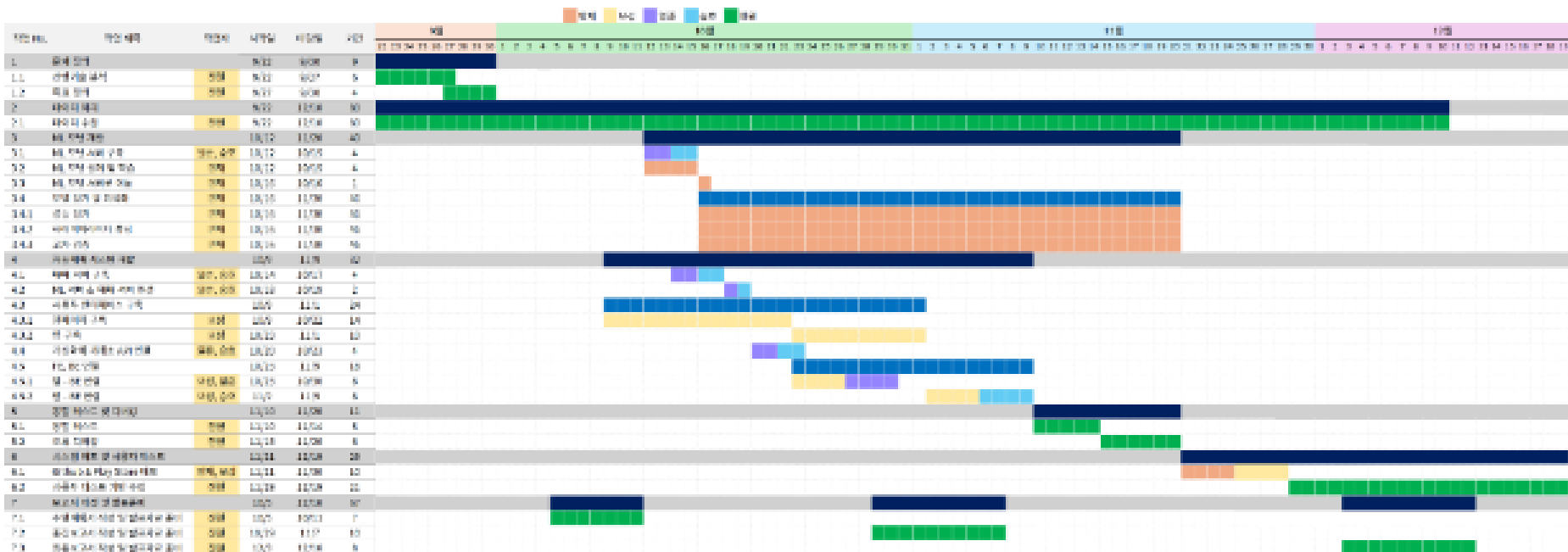
이승호

웹 BE 보조, 서버
ML 모델 개발 지원

진대한 멘토

전반적인 코드 리뷰, ML 모델 개발 지원

추진일정(Gantt Chart)



Q&A