

# 마켓 메이킹을 이용한 거래 자동화 시스템

김두종, 허준수

---

# Contents



1. 프로젝트 배경

2. 틱 데이터

3. GLFT 모델

4. 구현

5. 결과

**프로젝트 배경**

**01**

# 01 시장 미시구조란?

## 주가 자체가 어떠한 원리로 생성되는가?

- 거래 시장 내부에서 주가가 생성되기 위한 **요소** 등을 관찰하여 이것들의 **상호작용을 모형화하여** 주가가 생성되는 **원리를 규명**
- 주가 데이터가 아닌 주가 이전의 데이터, 각 주문의 유입 강도, 유동성 등의 데이터를 사용

Order Book		
		
Price(USDT)	Size(BTC)	Sum(BTC)
41499.6	0.001	4.914
41499.4	0.001	4.913
41499.3	0.031	4.912
41499.1	0.001	4.881
41498.9	0.030	4.880
41498.8	0.031	4.850
41498.7	0.009	4.819
41498.6	4.810	4.810
41498.6 ↑ 41500.5		
41498.5	9.051	9.051
41498.4	0.003	9.054
41497.7	0.012	9.066
41497.6	0.088	9.154
41497.3	0.080	9.234
41497.2	0.013	9.247
41497.0	0.030	9.277
41496.9	0.010	9.287

Trades		
Price(USDT)	Amount(U...	Time
41,515.4	1451503.0	21:14:49
41,514.5	3528.8	21:14:48
41,514.6	36034.7	21:14:48
41,514.5	1099843.7	21:14:47
41,502.4	7802.5	21:14:46
41,502.5	22452.9	21:14:46
41,502.4	19049.7	21:14:45

## 01 왜 시장 미시구조인가?

### 초과 수익을 찾아서

- 팩터란 수익률에 영향을 미치는 요소들
- 팩터로 초과 수익의 대부분을 설명할 수 있게 되면서 순수한 초과 수익을 내는 것이 더욱 어려워짐
- 보통 종목 선택, 팩터 타이밍 등을 이용하여 초과 수익을 내고는 하나 연구가 잘 안 된 **미지의 영역**이 존재한다면?



# 01 마켓 메이킹이란?

## 시장에 지속적으로 유동성을 제공하는 역할

- 매수/매도 호가를 동시에 제공하여, 판매자로부터 매수, 구매자로부터 매도하여 참여자들이 원하는 시점에 거래할 수 있는 환경을 제공
- 우리가 흔히 아는 중고차 딜러와 유사
- 중고차 딜러들을 통해 지금 타고 다니는 차를 언제든지 다시 팔 수 있고 원하는 중고차를 언제든지 살 수 있는 것처럼, 마켓메이커는 이러한 역할을 금융 시장에서 수행

Order Book		
Buy	Price	Sell
	27,501	Ask Price
	27,500 (Mid Price)	
Bid Price	27,499	

} Bid-Ask Spread

# 01 재고위험(Inventory Risk) 모델

---

## 마켓 메이킹도 재고 관리 위험이 존재

---

- 시장이 급격하게 변하는 경우, 매수/매도 호가가 거의 동시에 체결되는 것이 아니라 매수 호가 혹은 매도 호가만 체결되며 **한 방향으로 쏠릴 수 있음**
- 마켓메이커 **역시 자산을 팔지 못해 생기는 재고위험으로 인해 손실을 볼 수 있다**
- 중고차 딜러를 예시로 들면 중고차 딜러들이 차량을 매입한 직후 구매자가 없어 판매를 못하고 경기가 안 좋아 차 가격이 하락하면 중고차 딜러들은 더 낮은 가격에 판매를 해야 하는 경우가 생김

## 현실적인 제약들

---

- 보통 마켓 메이킹은 HFT(고빈도 매매)를 통해 구현되는데 지속적으로 유동성을 공급해야 하며, 동시에 양방향 주문을 체결하기 때문에 빠른 속도가 요구됨
- 빠르고 지속적으로 유동성을 공급하기 때문에 **거래가 굉장히 많이 일어남 = 많은 수수료**
- 현업에서는 기관들이 큰 돈(유동성)을 제공하기 때문에 거래소 차원에서 수수료를 감면해주거나 Rebate를 받음
- 개인은 수수료를 감당하기가 어려우니, **스프레드를 키워서 재고 위험을 조금 가져가면서, 거래 빈도를 줄여서 수익을 찾는 전략**

틱(호가) 데이터

02



## 02 틱 데이터

### 시장에 주문을 내는 방법

- 시장에 주문을 내는 방법에는 크게 두 가지, **시장가** 주문과 **지정가** 주문이 존재
- 시장가 주문은 주문을 던지는 즉시 **바로 체결**될 수 있도록 가격에 상관없이 **반대편 최우선 호가**에 무조건 체결을 시키는 **공격적인 스타일**의 주문
- 지정가 주문은 내가 **지정한 바로 그 가격에서만** 거래를 체결시키도록 하는 **방어적 스타일**의 주문을 의미
- 시장가 주문은 **시장의 유동성을 잡아먹게 되고**, 그 즉시 체결이 발생하기 때문에 이는 결국 **유동성 수요자(Liquidity Taker)**의 주문이라고 할 수 있으며, 반대로 지정가 주문은 **시장에 유동성을 공급**하므로 **Liquidity Provider**의 주문이라고 할 수 있다.

Order Book		
Price(USDT)	Size(BTC)	Sum(BTC)
41503.2	0.043	13.770
41503.1	0.057	13.727
41502.8	0.350	13.670
41502.7	0.088	13.320
41502.6	0.481	13.232
41502.2	0.541	12.751
41502.1	1.117	12.210
41502.0	11.093	11.093
41501.9↑ 41498.1		
41501.9	1.397	1.397
41501.8	0.006	1.403
41500.6	0.003	1.406
41500.4	0.160	1.566
41500.3	0.003	1.569
41499.9	0.008	1.577
41498.9	0.003	1.580
41498.8	0.003	1.583

Trades		
Price(USDT)	Amount(U...	Time
41,501.9	290.6	00:21:53
41,501.9	127078.9	00:21:53
41,501.9	42830.0	00:21:52
41,501.9	4689.8	00:21:52
41,501.9	590115.6	00:21:51
41,498.1	23903.0	00:21:50
41,498.0	1701.5	00:21:50

### 호가창

- 호가창은 유동성 공급자들에 의해 제출된 모든 지정가 주문들이 집적되어 있는 공간
- **틱 데이터**는 이러한 호가창에서 발생하는 모든 사건들에 대한 정보가 담겨있는 가장 높은 해상도의 장중 데이터 셋이며 체결 데이터와 호가 데이터 두 종류가 존재
- **체결 데이터(Trade data)**는 시장에서 매수와 매도가 만나 실제 매칭이 되어 발생한 거래에 대한 정보 (시각, 거래 가격, 거래량 등)
- **호가 데이터(Quote data)**란 호가창 상황의 변화를 포착하는 데이터로, 지정가 주문의 유입과 취소, 체결 발생에 의한 유동성 소멸을 실시간으로 반영
- 비트코인 데이터 수집, 하루 데이터 **5GB**라 여러 종목이나 오랜 기간 검증 불가

## 02 틱 데이터

### 데이터 수집

기존에는 로컬 환경에서 데이터 수집을 진행했지만, 예상하지 못한 이벤트로 인해 수집이 끊기는 것을 막기 위해 AWS를 사용

1. AWS EC2 인스턴스를 생성
2. 생성된 인스턴스에 git을 설치하고 데이터 수집에 필요한 코드를 git clone으로 복사
3. 로그아웃 하여 터미널을 빠져나가도 실행중인 프로그램이 종료되지 않고 계속 수행될 수 있게 nohup — ./shell.sh & 명령을 실행

```
[ec2-user@ip-172-31-14-180 raw_data]$ ls -al
total 224587848
drwxrwxrwx. 2 ec2-user ec2-user      16384 Dec  4 23:59 .
drwxr-xr-x. 3 ec2-user ec2-user         22 Nov  2 05:56 ..
-rwxrwxrwx. 1 ec2-user ec2-user 3792482932 Nov  2 23:59 btcusdt_20231102.dat
-rwxrwxrwx. 1 ec2-user ec2-user 4629265295 Nov  3 23:59 btcusdt_20231103.dat
-rwxrwxrwx. 1 ec2-user ec2-user 2924101701 Nov  4 23:59 btcusdt_20231104.dat
-rwxrwxrwx. 1 ec2-user ec2-user 3835840183 Nov  5 23:59 btcusdt_20231105.dat
-rwxrwxrwx. 1 ec2-user ec2-user 3970203635 Nov  6 23:59 btcusdt_20231106.dat
-rwxrwxrwx. 1 ec2-user ec2-user 4913858032 Nov  7 23:59 btcusdt_20231107.dat
-rwxrwxrwx. 1 ec2-user ec2-user 4403330524 Nov  8 23:59 btcusdt_20231108.dat
-rwxrwxrwx. 1 ec2-user ec2-user 7680001197 Nov  9 23:59 btcusdt_20231109.dat
-rwxrwxrwx. 1 ec2-user ec2-user 5112391645 Nov 10 23:59 btcusdt_20231110.dat
-rwxrwxrwx. 1 ec2-user ec2-user 3400082191 Nov 11 23:59 btcusdt_20231111.dat
-rwxrwxrwx. 1 ec2-user ec2-user 2871851842 Nov 12 23:59 btcusdt_20231112.dat
-rwxrwxrwx. 1 ec2-user ec2-user 4258812183 Nov 13 23:59 btcusdt_20231113.dat
-rwxrwxrwx. 1 ec2-user ec2-user 2766137081 Nov 14 23:59 btcusdt_20231114.dat
-rw-r--r--. 1 ec2-user ec2-user 3068389707 Nov 15 23:59 btcusdt_20231115.dat
-rw-r--r--. 1 ec2-user ec2-user 3439431477 Nov 16 23:59 btcusdt_20231116.dat
-rw-r--r--. 1 ec2-user ec2-user 2746791848 Nov 17 23:59 btcusdt_20231117.dat
-rw-r--r--. 1 ec2-user ec2-user 1528787008 Nov 18 23:59 btcusdt_20231118.dat
-rw-r--r--. 1 ec2-user ec2-user 1545937649 Nov 19 23:59 btcusdt_20231119.dat
-rw-r--r--. 1 ec2-user ec2-user 2921163345 Nov 20 23:59 btcusdt_20231120.dat
-rw-r--r--. 1 ec2-user ec2-user 5948728430 Nov 22 00:07 btcusdt_20231121.dat
-rw-r--r--. 1 ec2-user ec2-user 5286887657 Nov 22 23:59 btcusdt_20231122.dat
```

- 종목 : BTCUSDT 선물
- 기간 : 2023/11/02 ~ TBD
- 데이터 크기 : 2GB~7GB

**GLFT 모델**

**03**

## 03 GLFT 모델

### 최적의 매수/매도 호가를 결정하는 모델

- 「Dealing with the Inventory Risk. A solution to the market making problem」  
Olivier Guéant, Charles-Albert Lehalle, Joaquin Fernandez-Tapia (2012)
- 틱 데이터를 기반으로 다양한 파라미터들을 추정하여 최적 bid/ask quote을 결정

#### Dealing with the Inventory Risk

A solution to the market making problem

Olivier Guéant · Charles-Albert Lehalle ·  
Joaquin Fernandez-Tapia

This draft: July 2012

**Abstract** Market makers continuously set bid and ask quotes for the stocks they have under consideration. Hence they face a complex optimization problem in which their return, based on the bid-ask spread they quote and the frequency at which they indeed provide liquidity, is challenged by the price risk they bear due to their inventory. In this paper, we consider a stochastic con-

$$\delta_{\infty}^{b*}(q) \simeq \frac{1}{\gamma} \ln\left(1 + \frac{\gamma}{k}\right) + \frac{2q+1}{2} \sqrt{\frac{\sigma^2 \gamma}{2kA} \left(1 + \frac{\gamma}{k}\right)^{1+\frac{k}{\gamma}}}$$

$$\delta_{\infty}^{a*}(q) \simeq \frac{1}{\gamma} \ln\left(1 + \frac{\gamma}{k}\right) - \frac{2q-1}{2} \sqrt{\frac{\sigma^2 \gamma}{2kA} \left(1 + \frac{\gamma}{k}\right)^{1+\frac{k}{\gamma}}}$$

## 03 GLFT 모델

### 파라미터 추정

1. State Variables(상태 변수)
  - $\lambda$  : 거래 강도 (trading intensity)
  - $q$  : 내가 가지고 있는 물량(=재고, inventory)
  - $\sigma$  : 가격 변동성(volatility)
2. Parameters (추정하는 파라미터)
  - 가정 :  $\lambda = A \exp(-k\delta)$
  - $A$  : 유동성을 나타내는 양의 상수 (overall intensity of market orders)
  - $k$  : 유동성을 나타내는 양의 상수 (adjusts the intensity based on distance from reference price)
3. Hyper-parameters(사용자가 직접 설정해주는 파라미터)
  - $\gamma$  : Risk Aversion 계수 (위험 회피 계수)

$$\delta_{\infty}^{b*}(q) \simeq \frac{1}{\gamma} \ln\left(1 + \frac{\gamma}{k}\right) + \frac{2q+1}{2} \sqrt{\frac{\sigma^2 \gamma}{2kA} \left(1 + \frac{\gamma}{k}\right)^{1+\frac{k}{\gamma}}}$$

$$\delta_{\infty}^{a*}(q) \simeq \frac{1}{\gamma} \ln\left(1 + \frac{\gamma}{k}\right) - \frac{2q-1}{2} \sqrt{\frac{\sigma^2 \gamma}{2kA} \left(1 + \frac{\gamma}{k}\right)^{1+\frac{k}{\gamma}}}$$

구현

04

## 04 구현

### hftbacktest

- 수집된 데이터 : 10월 13일 ~ TBD
- Hftbacktest 오픈소스 소프트웨어를 활용하여 binance 마켓을 simulate
- 다양한 파라미터 값들을 시도해보며 최적의 값을 산출
- 기존 GLFT 모델의 파라미터에
  - Delta, adj1, adj2 등 새로운 파라미터를 추가하여 추정

### HftBacktest

code quality **A** CodeQL **passing** pypi package **1.7.1** downloads **13k** License **MIT** docs **passing** Stars **675**

#### High-Frequency Trading Backtesting Tool in Python

This Python framework is designed for developing high-frequency trading and market-making strategies. It focuses on accounting for both feed and order latencies, as well as the order queue position for order fill simulation. The framework aims to provide more accurate market replay-based backtesting, based on full order book and trade tick feed data.

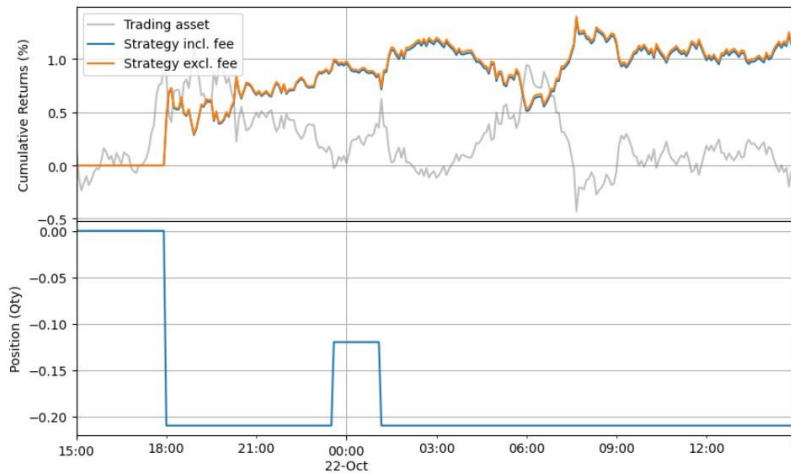
#### Key Features

- Working in [Numba](#) JIT function.
- Complete tick-by-tick simulation with a variable time interval.
- Full order book reconstruction based on L2 feeds(Market-By-Price).
- Backtest accounting for both feed and order latency, using provided models or your own custom model.
- Order fill simulation that takes into account the order queue position, using provided models or your own custom model.

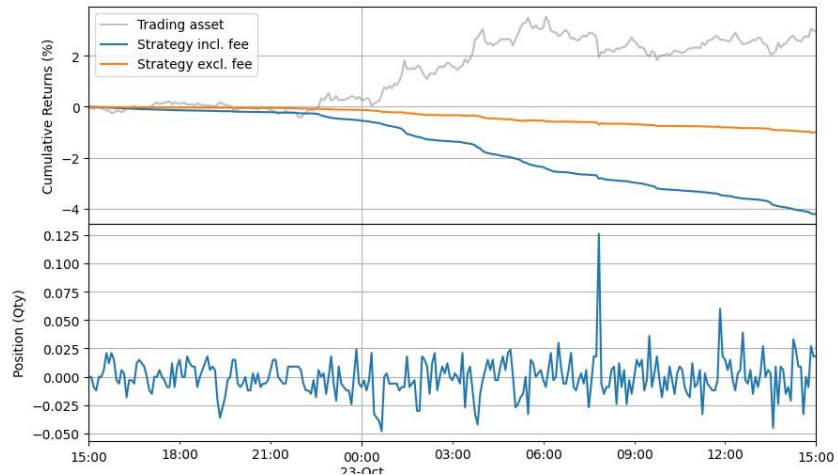
## 04 구현

### hftbacktest

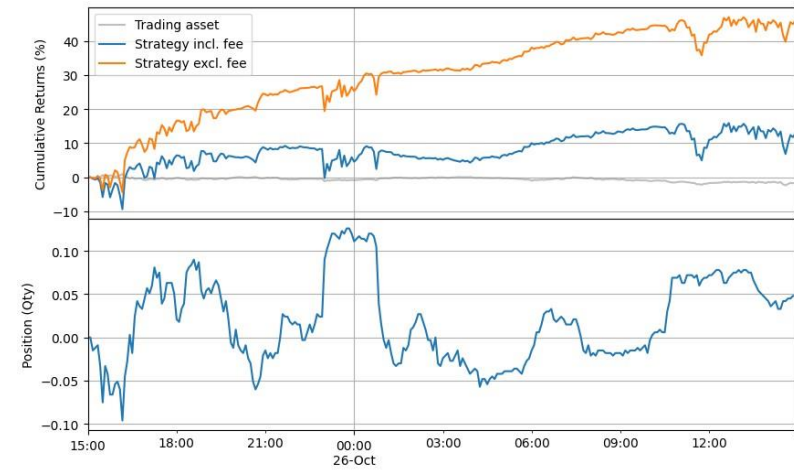
1. 수익은 좋지만 거래가 많이 이루어지지 않는 경우



2. 거래가 많이 이루어지지만 수익이 좋지 않은 경우



3. 균형을 이룬 경우





## 04 구현

### tradingbot

- 실제 Binance 거래소에 주문을 넣는 단계
- Binance에서 request를 거부하는 문제 발생
- 에러 메시지를 분석하여 최대한 오류가 덜 발생하도록 수정
- 만약 websocket이 끊기더라도 다시 연결할 수 있게 설정

에러 : Unhandled Error: 400, message='Bad Request'

Binance Response :

{'code': -2011, 'msg': 'Unknown order sent.'}

{'code': -1021, 'msg': 'Timestamp for this request is outside of the recvWindow.'}

{'code': -5022, 'msg': 'Due to the order could not be executed as maker, the Post Only order will be rejected. The order will not be recorded in the order history'}

S

BTCUSDT Perpetual

Cross

Closing PNL

0.29

Entry Price

39697.50

Avg. Close Price

39672.50

Max Open Interest

0.012 BTC

Opened

2023-12-04 00:36:50

Closed

2023-12-04 01:07:48

S

BTCUSDT Perpetual

Cross

Closing PNL

0.71

Entry Price

39593.33

Avg. Close Price

39566.66

Max Open Interest

0.024 BTC

Opened

2023-12-03 21:56:11

Closed

2023-12-03 23:56:33

B

BTCUSDT Perpetual

Cross

Closing PNL

-0.09

Entry Price

39420.00

Avg. Close Price

39390.00

Max Open Interest

0.003 BTC

Opened

2023-12-03 21:47:27

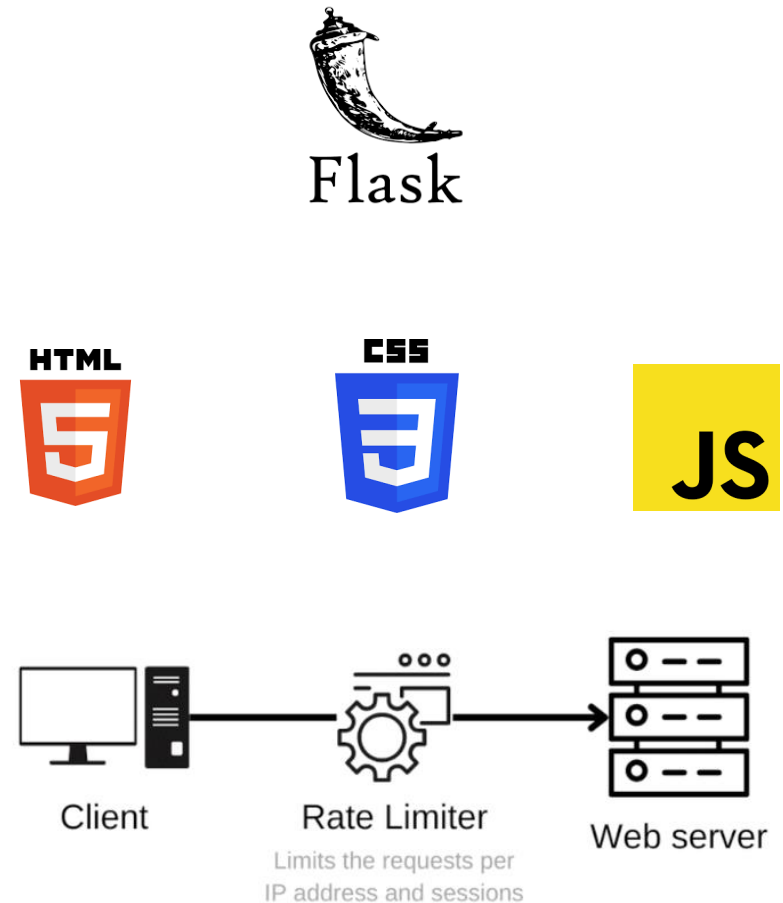
Closed

2023-12-03 21:53:50

## 04 구현

### web

1. Backend
  - Flask
2. Frontend
  - HTML, CSS, Javascript
3. 주요 기능
  - 실시간 비트코인 선물 1분봉 차트
  - 현재 계정의 total balance 와 PnL(수익률) 표시
  - 현재 보유한 포지션 출력
  - Open order(주문 내역) 출력
4. Challenges
  - Tradingbot과 같은 IP를 사용해서 실행하게 되면 Binance API의 IP Limit에 걸릴 수 있다



**결과**

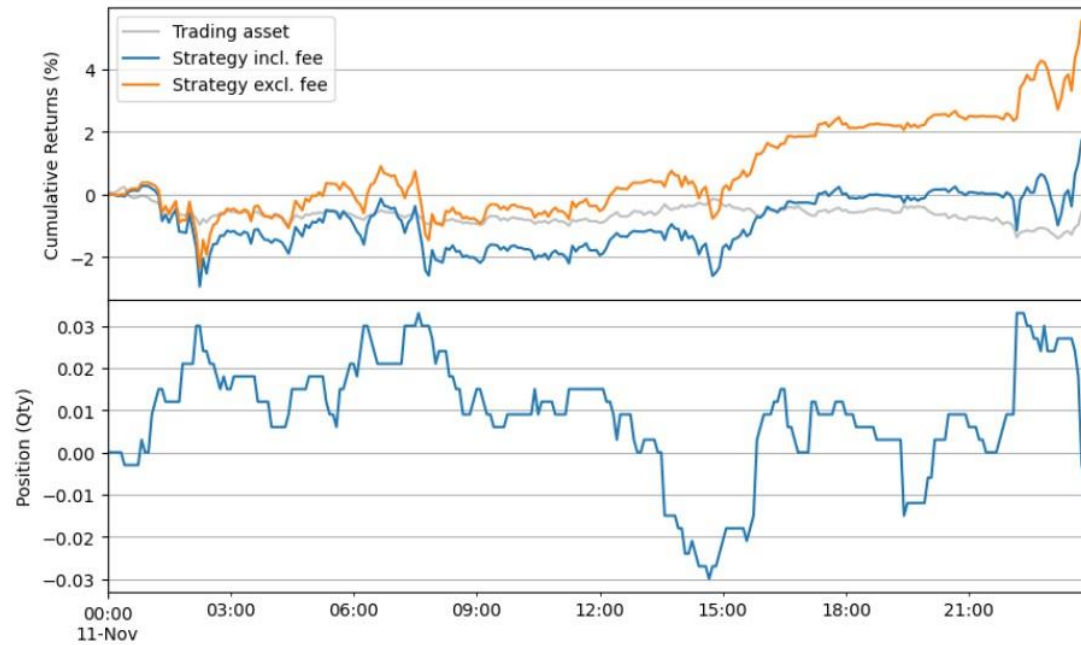
**05**

## 05 결과

### 최적의 파라미터 산출

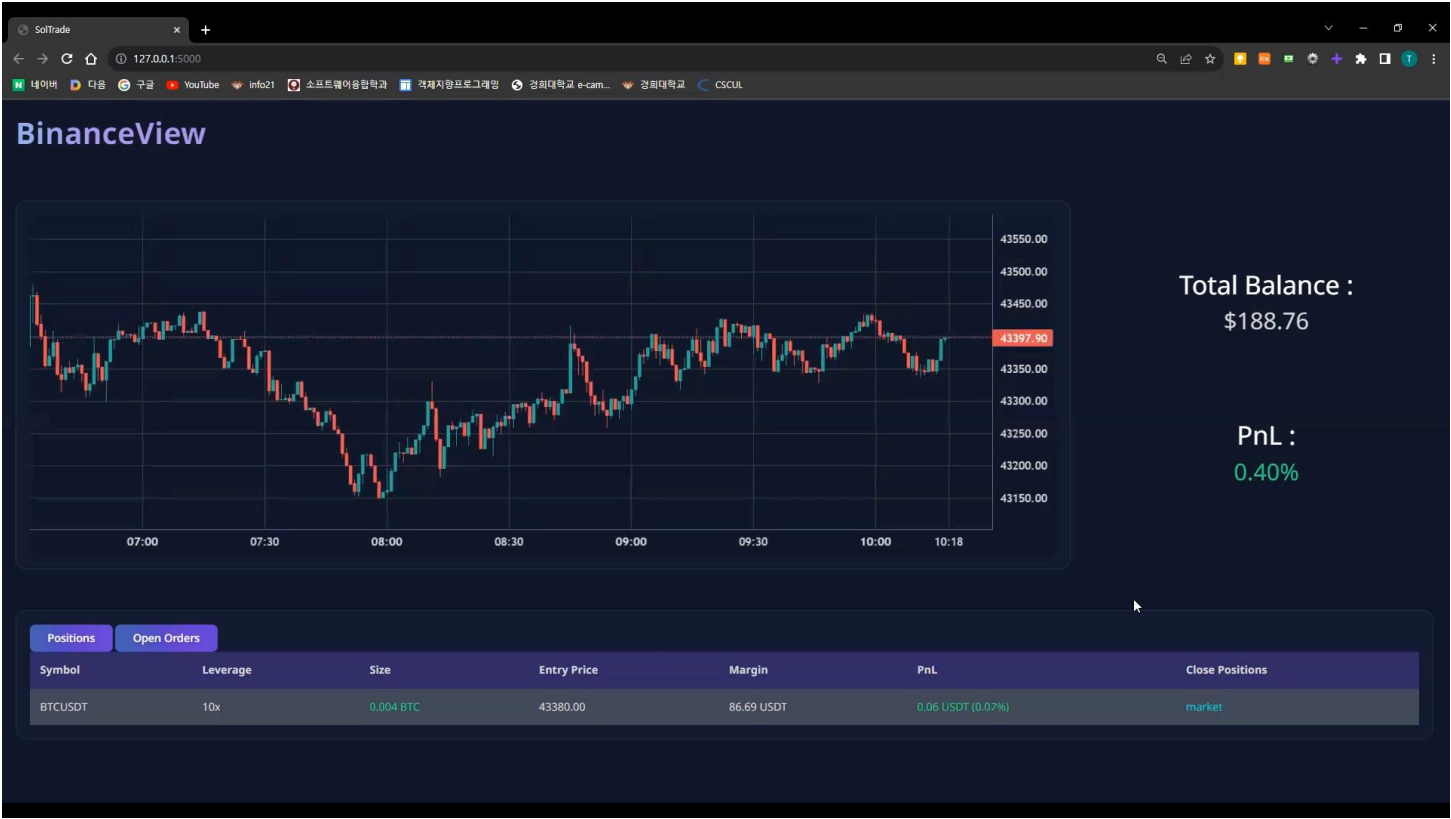
1.  $\gamma = 0.3$
2.  $\delta = 3$
3.  $\text{adj1} = 1.5$
4.  $\text{adj2} = 10$
5.  $\text{order\_qty} = 0.003$
6.  $\text{max\_position} = 0.12$
7.  $\text{grid\_num} = 4$
8.  $\text{lambda}[:40]$

Sharpe ratio: 6.9  
Sortino ratio: 9.0  
Risk return ratio: 184.9  
Annualised return: 594.09 %  
Max. draw down: 3.21 %  
The number of trades per day: 142  
Avg. daily trading volume: 0  
Avg. daily trading amount: 15899  
Max leverage: 14.83  
Median leverage: 2.23



# 05 결과

## 시연 영상



## 05 결과

### 실제 Binance Market에 모델을 적용



## 05 결과

---

### Limits

---

- 호가창 데이터는 가상화폐 시장에서만 구할 수 있고 실시간으로만 받을 수 있다
- 호가창 과거 데이터를 받을 수 없고 하루 데이터가 5GB라 오랜 기간에 대한 검증이 어렵다
- 실전에 투입한 지 얼마 되지 않아 여러 장 상황에 대한 성과 측정이 어렵다

## 05 결과

---

### Github & References

---

- 프로젝트 깃허브 링크
  - <https://github.com/tuchongkim/visualize-binance>
- 참고 문헌
  - *'High-frequency trading in a limit order book'*, Marco Avellaneda, Sasha Stoikov (2008)
  - *'Dealing with the Inventory Risk. A solution to the market making problem'*, Olivier Guéant, Charles-Albert Lehalle, Joaquin Fernandez-Tapia (2012)
- 참고 자료
  - Binance API Documentation
  - Hftbacktest by nkaz001
  - Sample-trading-bot by nkaz001





**Thank you**

감사합니다