



KQR Final Project 3조
중형주 다중 팩터 전략

박민서, 엄제윤, 이화원, 허지원

목차

- 001 **변화사항**
 - 어닝서프라이즈(OUT), RSI(OUT)
- 002 **백테스트 코드 구현**
 - 수익률확인 기능 추가
- 003 **전략 구체화**
 - MACD 매수/매도
 - DMI추가
 - MACD 기간 수정
- 004 **계획**
- 005 **한계점**

1. 변화사항

-어닝서프라이즈(OUT), RSI(OUT)

변화사항



Track1.

종목선정

10개



Track2.

매수/매도

중소형주
전략

어닝
서프라이즈

MACD

RSI

변화사항



Track1.

종목선정

10개

Track2.

매수/매도

중소형주
전략

OUT

MACD

OUT

Earning Surprise

Tra

종목선정



매수/매도

1. 어닝서프라이즈 지표 효과 불안정
2. 적합한 데이터 수집 실패
=보완 위한 ROE 입력변수 추가

중소형주

전

MACD

OUT

OUT

변화사항



Track1.

종목선정

10개

Track2.

매수/매도

RSI

중소형주
전략

타이밍

MACD

(한 발 늦는 지표)

OUT

OUT

변화사항



두산인프라코어 (042670)

변화사항



MACD값>MACDsignal
=매수타이밍

두산인프라코어 (042670)

변화사항



BUT 잡아내지 못하는
RSI

MACD값>MACDsignal
=매수타이밍

두산인프라코어 (042670)

변화사항



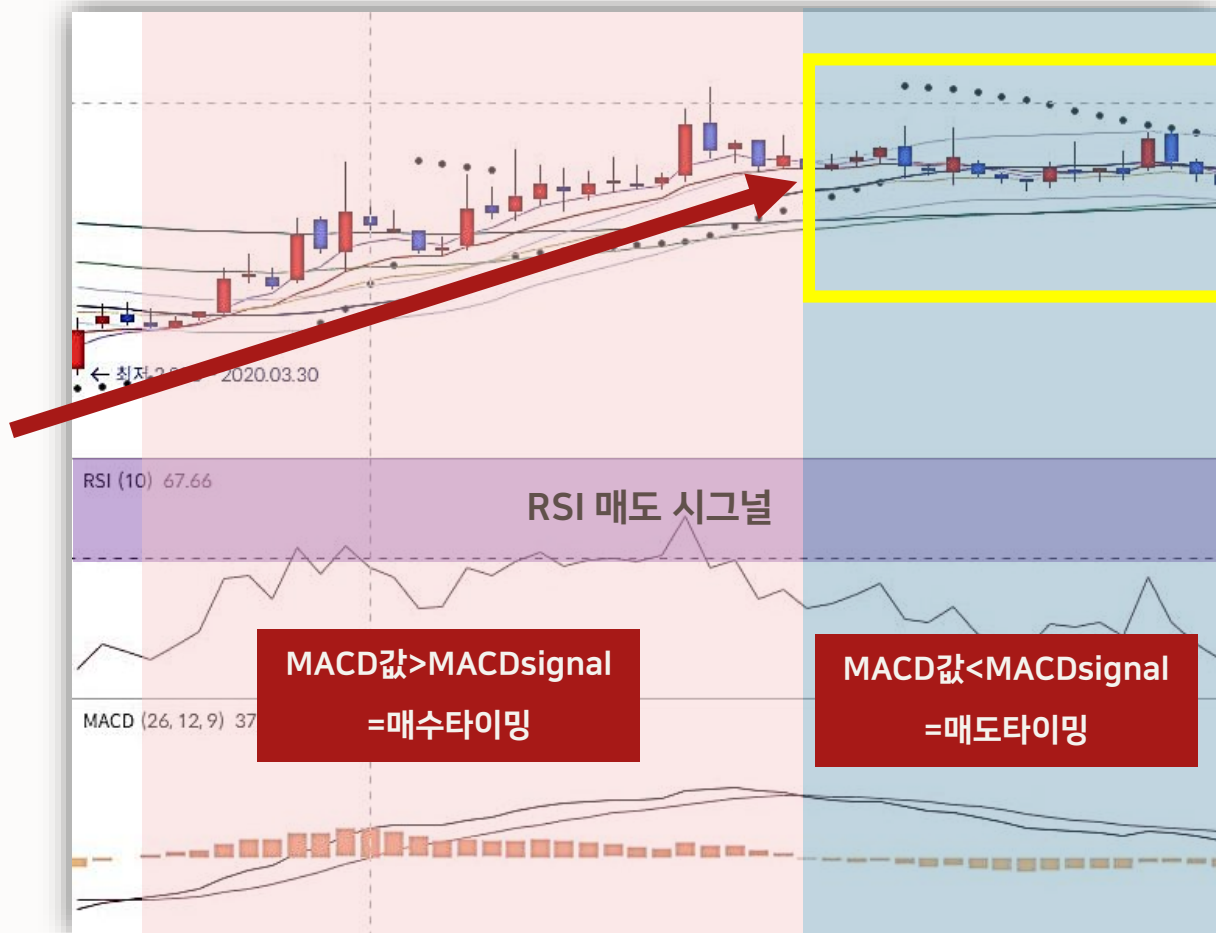
두산인프라코어 (042670)



잡아내지 못하는 RSI

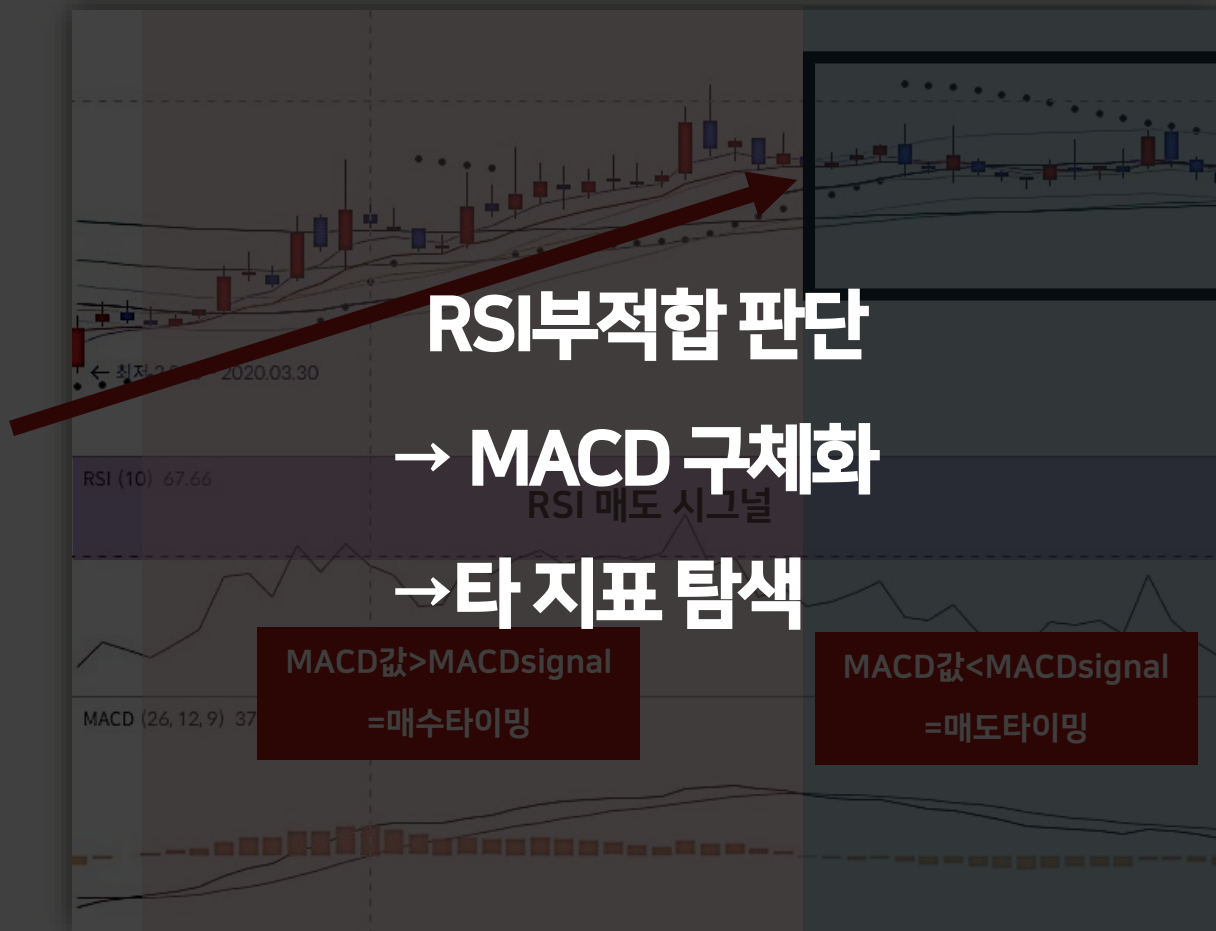
$MACD_{값} < MACD_{signal}$
=매도타이밍

변화사항



두산인프라코어 (042670)

전략 소개



두산인프라코어 (042670)

2. 백테스트 코드 구현

-수익률 확인 기능 추가

백 테스트 코드 구현



지난주 마무리, 최대 과제 백테스트

NEW BACKTEST CODE	
기간	2018년 01월 01일 - 2020년 12월 30일
매수/매도	당일 종가
수수료	아직 고려X
종목	3개월 마다 청산 및 변경

백테스트 코드 구현



기간별 백테스트 결과 함수

```
def period_result(total_reset1, total_reset2):
    total_info1 = pd.DataFrame(total_reset1.sum(axis=1))
    total_info2 = pd.DataFrame(total_reset2.sum(axis=1))
    total_info_df = pd.concat([total_info1, total_info2], axis=1)
    total_info_df.columns = ['자산', '수익률']

    total_info_df['수익률'] = (total_info_df['자산'] / total_info_df['자산'] [0] - 1)
    total_info_df['누적수익률'] = (total_info_df['수익률'] + 1).cumprod() - 1
    total_info_df['수익률'] = total_info_df['수익률'] * 100
    total_info_df['누적수익률'] = total_info_df['누적수익률'] * 100

    # 전고점 mdd
    list_period = list(total_info_df['자산'])

    maxlist = []

    for num, kes in enumerate(list_period):
        if num == 0:
            maxlist.append(i)
        else:
            maxlist.append(max(i, maxlist[-1]))
    total_info_df['전고점'] = maxlist
    total_info_df['MDD'] = ((total_info_df['자산'] - total_info_df['전고점']) / total_info_df['전고점']) * 100

    trade_day = int(len(total_info_df.index))
    cagr = ((total_info_df['자산'][-1] / total_info_df['자산'][0]) ** (1 / trade_day)) - 1

    # 주요 지표 구하기
    backtest_list = {}
    backtest_list['Start date'] = start_date
    backtest_list['End date'] = end_date
    backtest_list['Start Money'] = start_money
    backtest_list['Total returns'] = str(round(min(list(backtest_list['MDD'])), 2))
    backtest_list['CAGR'] = str(round(cagr, 2))
    backtest_list['MDD'] = str(round(min(list(backtest_list['MDD'])), 2))
    backtest_list['Periods'] = trade_day

    # 결과물 프레임워크 지정
    result = pd.DataFrame(backtest_list)
    result = result.transpose()
    # result.columns = ['Algorithm']

    file_name1 = start_date[1] + 'total_info'
    total_df.to_csv(file_name1)

    file_name2 = start_date[1] + 'total_info'
    result.to_csv(file_name2)

    return total_info_df
```

전체 백테스트 결과 함수

```
def period_result2(kampa):
    kampa['수익률'] = (kampa['자산'] / kampa['자산'] [0] - 1)
    kampa['누적수익률'] = (kampa['수익률'] + 1).cumprod() - 1
    kampa['수익률'] = kampa['수익률'] * 100
    kampa['누적수익률'] = kampa['누적수익률'] * 100

    # 전고점 mdd
    list_period2 = list(kampa['자산'])

    maxlist2 = []

    for num, kes in enumerate(list_period2):
        if num == 0:
            maxlist2.append(kes)
        else:
            maxlist2.append(max(kes, maxlist2[-1]))

    kampa['전고점'] = maxlist2
    kampa['MDD'] = ((kampa['자산'] - kampa['전고점']) / kampa['전고점']) * 100

    trade_day = int(len(kampa.index))
    cagr = ((kampa['자산'][-1] / kampa['자산'] [0]) ** (1 / trade_day)) - 1

    # 주요 지표 구하기
    backtest_list = {}
    backtest_list['Start date'] = '20180101'
    backtest_list['End date'] = '20210331'
    backtest_list['Start Money'] = 100000000
    backtest_list['Total returns'] = str(round(kampa['누적수익률'][-1], 2)) + "%"
    backtest_list['CAGR'] = str(round(cagr, 2)) + "%"
    backtest_list['MDD'] = str(round(min(list(kampa['MDD'])), 2)) + "%"
    backtest_list['Periods'] = trade_day

    # 결과물 프레임워크 지정
    result = pd.DataFrame(backtest_list, index = [0])
    result = result.transpose()
    # result.columns = ['Algorithm']

    file_name1 = 'total_df.csv'
    total_df.to_csv(file_name1)

    file_name2 = 'total_result.csv'
    result.to_csv(file_name2)

    total_df2 = kampa

    return total_df2
```

종목별 수익 보기

```
pd.set_option('display.max_rows', None)
```

```
profit_df = pd.DataFrame([bula, boil]), T
```

```
minus = profit_df[profit_df [1] < 0]
plus = profit_df[profit_df [1] > 0]
```

```
print(len(plus), len(minus))
```

종목별 / 3개월 / 전체

세밀한 전략분석틀제공
전략방향성 수립 위한 데이터 확보

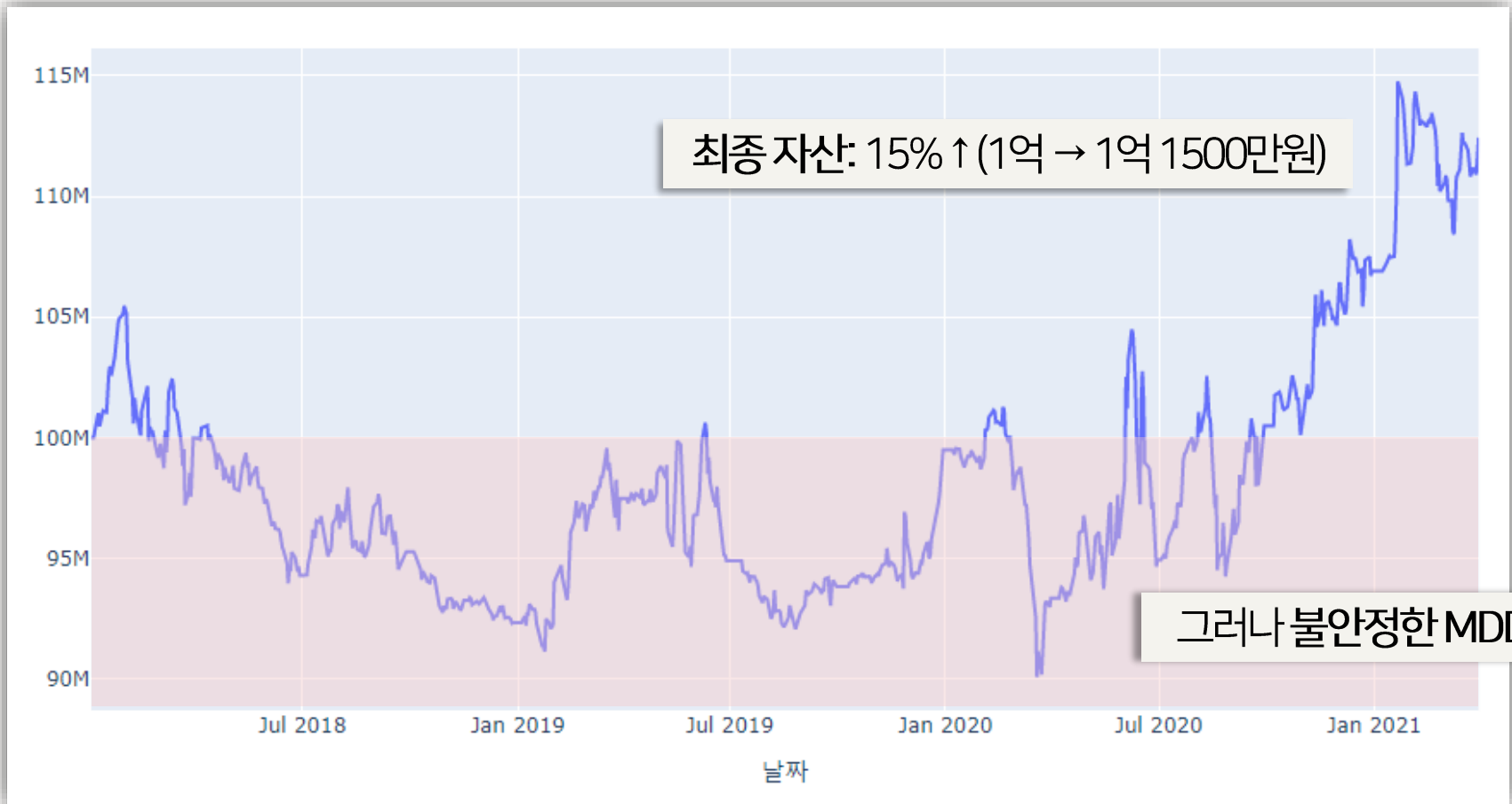
3. 전략구체화

- MACD 매수/매도
- DMI추가
- MACD 기간 수정

전략구체화



1. RSI + MACD 적용



전략구체화



1. RSI + MACD 적용



동기간 **삼성전자** 수익률: 117%

삼전에다 투자한 것과 다를 바 X
(수수료 고려하면 삼전이 이득)

MDD 방어도
삼전 단독투자가 더 높은 상황

전략구체화



1. RSI + MACD 적용



전략구체화



1. RSI + MACD 적용



전략구체화



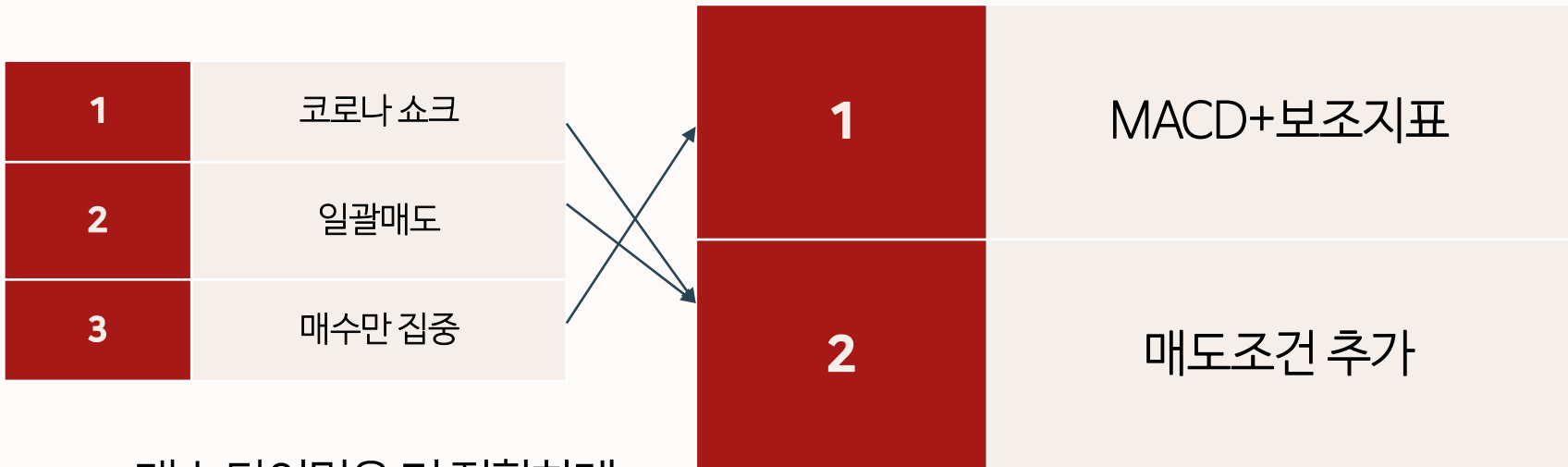
문제점 도출

1	코로나 쇼크
2	일괄매도
3	매수만 집중

전략구체화



해결책 도출



매수 타이밍은 더 정확하게
매도로 MDD 방어
=CAGR 극대화

전략구체화



1) MACD>0



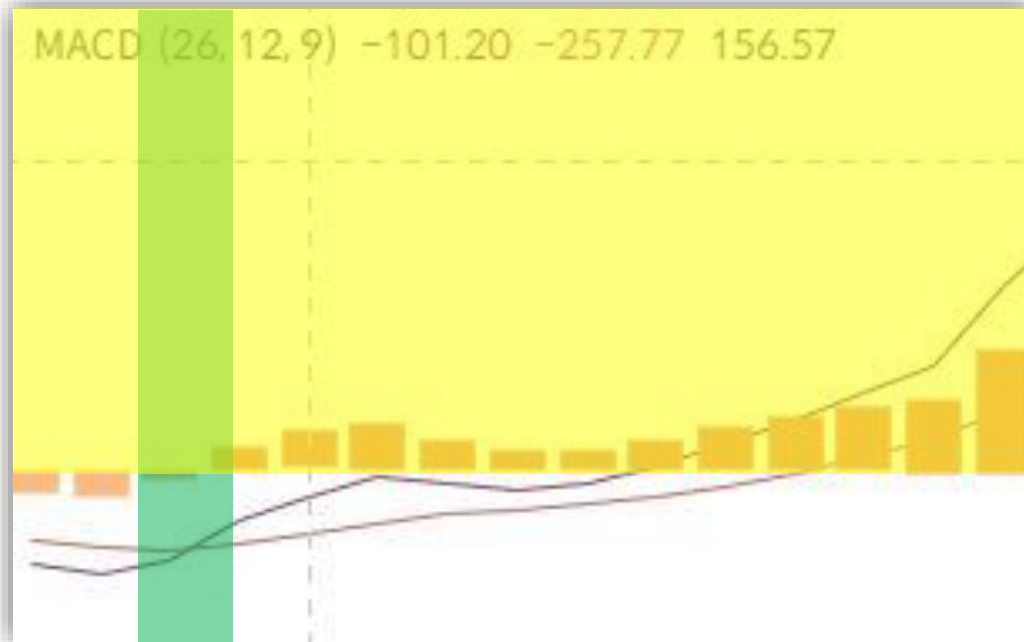
기존 매수전략

```
df['MACD'] > df['Signal_Line']
```


전략구체화



1) MACD>0



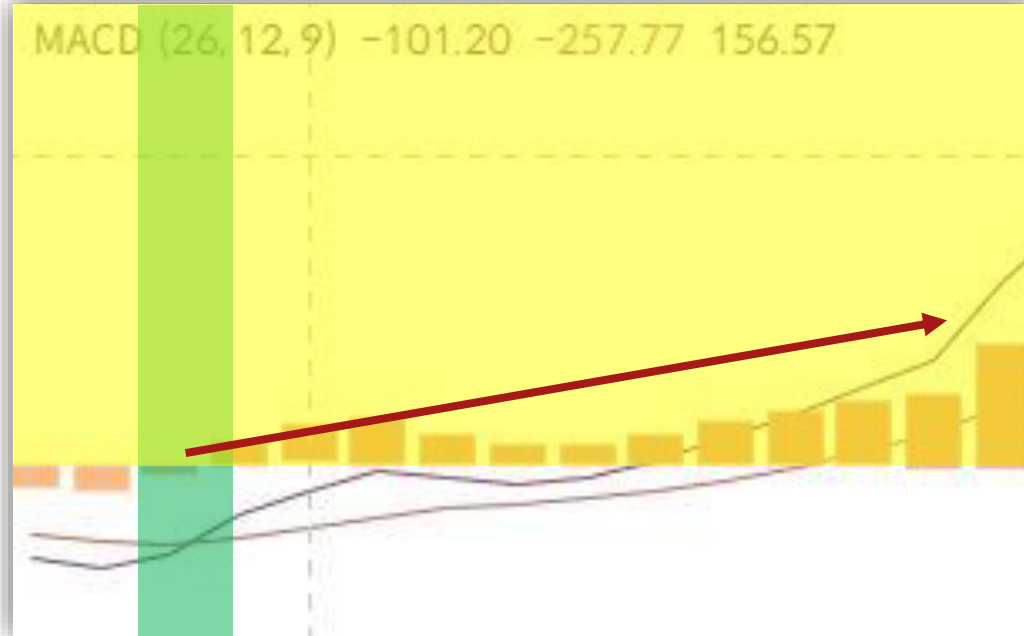
수정 매수 전략

```
df['MACD'] > df['Signal_Line'] & (df['MACD'] > 0)
```

전략구체화



1) $MACD > 0$, 그 전날 시그널보다 클때



수정 매수 전략

```
df['MACD'] > df['Signal_Line'] & (df['MACD'] > 0)
& ((df['MACD'] - df['Signal_Line']) >
(df.shift(1)['MACD'] - df.shift(1)['Signal_Line']))
```

전략구체화



1) $MACD > 0$, 그 전날 시그널보다 클 때



전략구체화



1) $MACD > 0$, 그 전날 시그널보다 클때



전략구체화



1) $MACD > 0$, 그 전날 시그널보다 클때



전략구체화



2) 매도조건추가

매도 조건추가

```
df['MACD매도'] = (df['MACD'] < df['Signal_Line']) & (df['MACD'] < 0) &  
((df['MACD'] - df['Signal_Line']) < (df.shift(1)['MACD'] -  
df.shift(1)['Signal_Line']))
```

전략구체화



2) 매도조건추가



매도 조건 추가

```
df['MACD매도'] = (df['MACD'] < df['Signal_Line']) & (df['MACD'] < 0) &  
((df['MACD'] - df['Signal_Line']) < (df.shift(1)['MACD'] -  
df.shift(1)['Signal_Line']))
```

3) DMI 지표 추가

DMI란?

Directional Movement Index

시장의 방향성과 추세의 강도를 수치로 나타낸 지표
고점과 저점을 이용한 강도 탐지 지표

전략구체화



3)DMI지표추가

DM

$P(+)\text{DM} = \text{오늘 고가} - \text{어제 고가}$

$M(-)\text{DM} = \text{오늘 저가} - \text{어제 저가}$

TR

1) 오늘 고가 - 오늘 저가

2) 오늘 고가 - 전일 종가

3) 오늘 저가 - 전일 종가

$$DI(\text{Directional Indicator}) = DM / TR$$

전략구체화



3)DMI지표추가

+DI(추세상승)

$$\frac{P(+)\text{DM의 } m\text{일 지수이동평균}}{\text{TR의 } m\text{일 지수이동평균}}$$

-DI(추세하락)

$$\frac{M(-)\text{DM의 } m\text{일 지수이동평균}}{\text{TR의 } m\text{일 지수이동평균}}$$

매수: +DI > -DI

(상승 추세가 하락 추세보다 강하게 나타나면 사라)

매도: +DI < -DI

(하락 추세가 상승 추세보다 강하게 나타나면 팔아라)

전략구체화

3)DMI지표추가



왜 DMI?

MACD는 종가를 기준으로한 이평선

DMI는 고가와 저가 이용한 지표

3) DMI지표추가



```
df['DI매수'] = (df['PMI'] > df['DMI']) & ((df['MACD'] - df['Signal_Line']) > (df.shift(1)['MACD'] -  
df.shift(1)['Signal_Line']))
```

```
df['DI매도'] = (df['PMI'] < df['DMI']) & ((df['MACD'] - df['Signal_Line']) < (df.shift(1)['MACD'] -  
df.shift(1)['Signal_Line']))
```

전략구체화



4) MACD기간조정

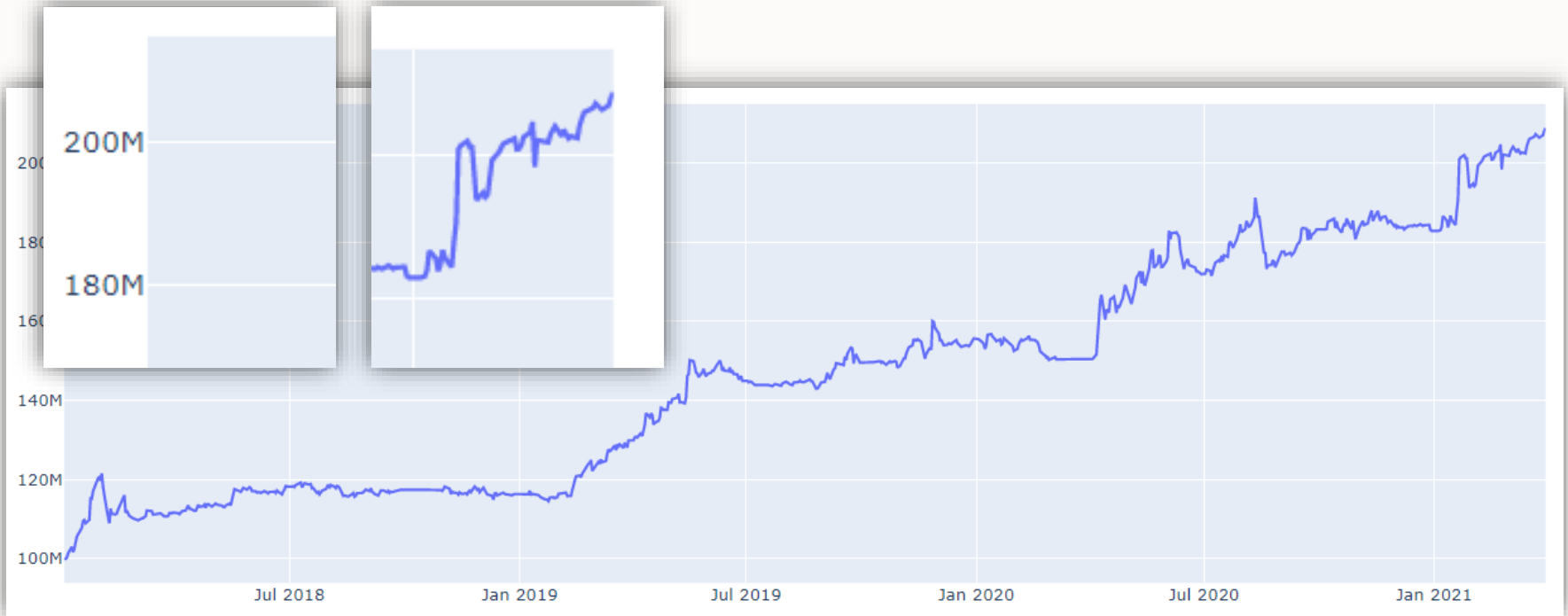
기간 조정			
MACD?	Short-Long		
Short	12	→	6
Long	26	→	13
MACD Signal	9	→	4

단기에 빠른 주가 상승
중소형주 특징 반영

전략구체화



4) MACD기간조정

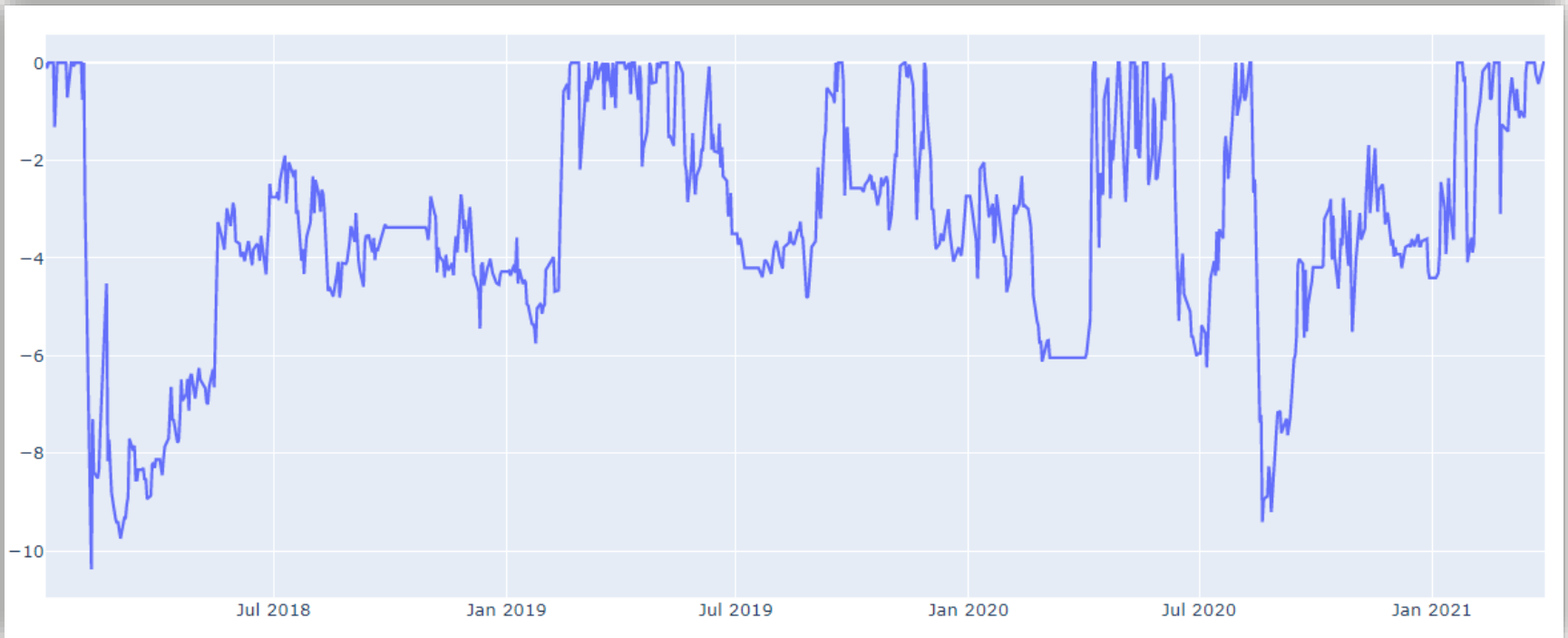


자산가격 2배 돌파

전략구체화



현재까지 최종본



MDD: -10.38%
CAGR: +25.42%
원금대비 2배 이상 자산가격 상승

4. 계획

계획

- 1) 거래량 지표
- 2) Slow Stochastic
- 3) ADX(DMI 강도 확인용)
- 4) 그래프 기울기 활용한 매수/매도 전략

5. 한계점

한계점

데이터 부족

(2017년 이전 데이터 부재)

구매? 다른 방안?



감사합니다