

## [Bollinger Bands 전략]

Bollinger Bands 란 주가의 이동평균선(moving average(ma)). 표준편차 범위를 상-하로 설정한 밴드. 주가가 하단 밴드보다 낮으면 매수하고 상단밴드보다 높으면 매도하는 방식. 즉 “추세”를 따르는 전략이다.

Backtrader 안에는 Bollinger Bands 를 제공하기에 해당 인자를 불러와서 사용할 수 있었다. 해당 인자에서 설정해야 할 것은 이동평균기간과 표준편차였다. 멘토링 수업에서 이동평균기간은 50~100 일로 설정하고 표준편차는 보통 1 로 설정한다고 배웠기에 해당 방식을 적용하였다.

```
1 # Bollinger Bands 전략 클래스
2 class BollingerBandsStrategy(bt.Strategy):
3     params = dict(
4         ma_period=50, # 이동평균 기간
5         dev_factor=1 # 표준편차 계수
6     )
7
8     def __init__(self):
9         self.bbands = bt.ind.BollingerBands(period=self.p.ma_period, devfactor=self.p.dev_factor)
10        self.order = None # 주문 상태
```

기본 세팅을 완료했고 이제 어떠한 액션을 취할것인지에 대해 코딩을 해야했다. 우선 코드를 보고 이해를 해본다.

```
1 def next(self):
2     if not self.position: # 포지션이 없을 때
3         if self.data.close[0] < self.bbands.lines.bot[0]: # 하단 밴드 아래
4             order_size = math.floor(self.broker.get_value() / self.data.close)
5             self.order = self.buy(size=order_size) # 매수
6         elif self.data.close[0] > self.bbands.lines.top[0]: # 상단 밴드 위
7             self.order = self.sell(size=self.position.size) # 매도
```

포지션이 없을 때 -> 가지고 있는 것이 없다. -> 즉 매수를 하는 타이밍을 의미.

포지션이 없을 때, 현재의 가격이 하단 밴드 아래에 위치한다면 사야한다.

반대로 포지션이 있는데 -> 무언가를 손에 들고 있을 때 -> 팔 수 있음을 의미.

주식이나 코인을 홀드하고 있을 때 지수가 상단 밴드 위로 간다면 팔아야한다.

그 후의 진행사항은 저번에 했던 backtrader 코드와 같다.

야후파이낸스에서 2011 년 1 월 1 일 ~ 2013 년 12 월 31 일까지의 msft 의 주가 데이터를 가져와서 사용하는데 볼린저밴드 전략의 효율이 궁금했다. 그냥 2011 년 1 월 1 일에 사서 마지막날까지 들고 있는 전략과, 추세전략을 통해 매수/매도를 반복적으로 했다면 수익면에서 어떠한 차이가 있었을지 알아보기로 했다.

```
1  # 1. Buy and Hold 전략
2  class BuyAndHold(bt.Strategy):
3      def __init__(self):
4          pass
5
6      def next(self):
7          # 포지션이 없을 경우 매수, 처음 매수 후에는 아무것도 하지 않음
8          if not self.position:
9              order_size = math.floor(self.broker.get_value() / self.data.close)
10             self.buy(size=order_size)
```

```
Buy and Hold 최종 자본: 133407.55
Bollinger Bands 전략 최종 자본: 145450.37
Buy and Hold 수익률: 33.41%
Bollinger Bands 수익률: 45.45%
```

볼린저 밴드의 전략이 수익이 높았다. 하지만 초기 자본을 한번에 사서 묵혀두는 것과 정해진 기간을 두고 추가매수를 하는 것은 또 수익이 다를 것이다. 즉 분할매수 전략을 추가해보았다. 초기 자산을 10 만으로 설정하였으니 5 번에 나눠서 구매하는 전략으로 구현해보았다.

```

1  # 2. 분할 매수 전략 (5번에 나눠서 매수)
2  class DollarCostAveraging(bt.Strategy):
3      def __init__(self):
4          self.buy_counter = 0 # 매수 횟수 카운터
5          self.max_buys = 5 # 최대 매수 횟수
6          self.buy_amount = self.broker.getcash() / self.max_buys # 매수 금액 (전체 현금의 1/5)
7          self.next_buy_date = None # 다음 매수일
8
9      def next(self):
10         # 매수 횟수가 최대에 도달하지 않았을 때
11         if self.buy_counter < self.max_buys:
12             # 첫 매수 또는 일정 간격 후 매수 실행
13             if self.next_buy_date is None or self.datas[0].datetime.date(0) >= self.next_buy_date:
14                 order_size = math.floor(self.buy_amount / self.data.close[0]) # 매수 가능한 수량 계산
15                 self.buy(size=order_size) # 매수 실행
16                 self.buy_counter += 1 # 매수 횟수 증가
17                 # 3개월 후로 다음 매수일 설정
18                 self.next_buy_date = self.datas[0].datetime.date(0) + timedelta(days=90)

```

**Buy and Hold 최종 자본: 133407.55**

**분할 매수 전략 최종 자본: 142747.14**

**Bollinger Bands 전략 최종 자본: 145450.37**

**Buy and Hold 수익률: 33.41%**

**분할 매수 전략 수익률: 42.75%**

**Bollinger Bands 수익률: 45.45%**

전체적인 우상향 기간이었어서 그런지 다 수익률이 높고 분할매수 전략 또한 볼린저밴드와 수익률이 비슷한 모습을 볼 수 있었다.