

VectorBT를 활용한 벡터 기반 백테스팅

VectorBT와 VectorBT Pro Tutorial

2025-1-25

벡터 기반 백테스팅

- 과거 시계열 데이터를 벡터 형태로 사용
- 벡터의 계산으로 백테스팅을 시뮬레이션
- 빠르게 백테스팅을 수행할 수 있다.
 - 많은 설정 값들에 대해서 백테스팅을 할 수 있다
- 이벤트 기반 백테스팅에 비해서 정확도가 떨어질 수 있다
- 이전 책 'Python for Finance Cookbook'의 12장의 앞 부분에 해당하는 내용
 - https://github.com/restful3/ml4t/blob/main/archive/Python_for_Finance_Cookbook/chapter_12_song.ipynb

VectorBT (1/3)

- Numpy, Pandas, Numba를 활용해서 벡터 기반 백테스팅을 할 수 있는 Library
- 코드는 Python으로 작성하되 실제 수행은 대부분 기계 코드로 변환해서 수행할 수 있다.
 - Numpy를 활용해서 수행되거나
 - Numba를 사용해서 Python 코드를 기계 코드로 변환한 다음에 수행한다.
- Numba
 - [GitHub - numba/numba: NumPy aware dynamic Python compiler using LLVM](#)
 - Numba는 Python에서 사용되는 동적 컴파일러로, NumPy와 호환되며 LLVM을 사용하여 기계 코드를 생성
 - Numba는 수치 계산에 중점을 둔 Python 코드를 컴파일할 수 있다
 - 자동 병렬화, GPU 가속 코드 생성, ufuncs 및 C 콜백 생성 등을 지원

VectorBT (2/3)

- 데이터가 구조화 되어 있거나 시계열을 다루는 부분은 Pandas를 활용하여 쉽게 다룰 수 있다
- Pandas를 확장하는 방식으로 구현하여 실제 내부에서 계산은 Numpy와 Numba를 활용하도록 되어 있다
 - Dataframe과 Series에서 vbt Accessor 사용하는 부분
 - Pandas 함수를 VectorBT에서 구현한 것들
 - '_nb'로 끝나는 함수들
 - Numba로 변환된 함수들
- 예)
 - `df = pd.DataFrame({'a': range(10), 'b': range(9, -1, -1)})`
 - `df.vbt.rolling_mean(3)`

VectorBT (3/3)

- 백테스팅을 위한 여러 기능 추가
 - Data 기능
 - 데이터를 저장, 다운로드, 업데이트 및 관리하는 기능
 - 지원하는 다운로드 방법
 - Alpaca Finance, Binance, CCXT, Yahoo Finance
 - Indicator 기능
 - Indicator를 쉽게 만들 수 있는 IndicatorFactory
 - Finance에 쓰이는 여러 지표 구현
 - ATR, BBANDS, MA, MACD, RSI
 - Portfolio 기능
 - 포트폴리오 성능을 모델링하고 다양한 위험 및 성과 지표를 계산하는 데 사용

VectorBT 기본 사용법 (1/4)

1. 필요한 데이터 다운로드

- Pandas 데이터 형태 (Dataframe, Series)로 다운로드 받을 수 있다
 - 지원하는 다운로드 방법
 - Alpaca Finance, Binance, CCXT, Yahoo Finance
 - 데이터 생성
 - GBM (Geometric Brownian motion)
- 데이터 관련 추가 기능
 - Updating, Merging, Indexing, Saving & Loading, Stat, Plot

2. 매수/매도 데이터 생성

- 다운받은 데이터로 여러 지표 생성 가능
 - ATR, BBANDS, MA, MACD, MSTD, OBV, RSI, STOCH
- Signal을 생성하거나 Order를 생성

3. 매수/매도 데이터에 따른 성과 평가

- 여러 평가를 계산 가능
 - Alpha, beta, sharpe ratio, sortino ratio, omega ratio, tail ratio, ...
- 'group_by'를 이용하여 그룹별로 평가 가능

VectorBT 기본 사용법 (2/4)

- 필요한 데이터 다운로드

```
>>> # Prepare data
>>> start = '2019-01-01 UTC' # crypto is in UTC
>>> end = '2020-01-01 UTC'
>>> btc_price = vbt.YFData.download('BTC-USD', start=start, end=end).get('Close')
```

VectorBT 기본 사용법 (3/4)

- 매수/매도 데이터 생성

```
>>> fast_ma = vbt.MA.run(btc_price, 10, short_name='fast')
>>> slow_ma = vbt.MA.run(btc_price, 20, short_name='slow')

>>> entries = fast_ma.ma_crossed_above(slow_ma)
>>> entries

>>> exits = fast_ma.ma_crossed_below(slow_ma)
>>> exits
```


VectorBT 기본 사용법 (4/4)

- 매수/매도 데이터에 따른 성과 평가

```
>>> pf = vbt.Portfolio.from_signals(btc_price, entries, exits)
>>> pf.total_return()
fast_window  slow_window
10           30           0.848840
20           30           0.543411
Name: total_return, dtype: float64
```

VectorBT의 Accessor 확인

- Pandas 클래스에 Custom Accessor를 붙여서 확장
 - <https://pandas.pydata.org/docs/development/extending.html#registering-custom-accessors>
- Pandas 클래스 별로 확장 구현 확인
 - `pd.Series/DataFrame.vbt.*`
 - > `vbt.base.accessors.BaseSR/DFAccessor`
 - `pd.Series/DataFrame.vbt.*`
 - > `vbt.generic.accessors.GenericSR/DFAccessor`
 - `pd.Series/DataFrame.vbt.signals.*`
 - > `vbt.signals.accessors.SignalsSR/DFAccessor`
 - `pd.Series/DataFrame.vbt.returns.*`
 - > `vbt.returns.accessors>ReturnsSR/DFAccessor`
 - `pd.DataFrame.vbt.ohlcv.*` and `pd.DataFrame.vbt.ohlcv.*`
 - > `vbt.ohlcv.accessors.OHLCVDFAccessor`
 - `pd.DataFrame.vbt.px.*`
 - > `vbt.px.accessors.PXAccessor`

VectorBT Pro

- VectorBT의 확장된 기능을 가진 버전
 - AlphaVantage와 Polygon에서 데이터 가져오기
 - 합성 데이터 생성기
 - 멀티스레딩, 스톱 계단 설정, 시간 스톱,
 - RiskFolio-Lib 및 PyPortfolioOpt를 활용한 포트폴리오 최적화 등 다양한 기능을 제공
- 비용
 - 1개월 \$25(\$40에서 discount) subscription
 - 평생 \$500
 - Github Sponsor로 구입
- VectorBT와 같이 로딩되면 문제 발생

VectorBT Pro 사용법

1. 필요한 데이터 다운로드

- 멀티 스레딩 기능을 사용하여 데이터 다운로드

2. 매수/매도 데이터 생성

- IndicatorFactory를 사용해서 Custom Indicator 구현
- ATR를 활용하는 SuperTrend Indicator 구현
- Entry / Exit 신호 생성

3. 매수/매도 데이터에 따른 성과 평가

- Entry / Exit 신호로 Portfolio 평가

SuperTrend 설명

- ATR과 '중간 가격(median price)'를 이용해서 상단/하단 밴드를 정의
- Basic Band
 - $\text{Median} \pm \text{Multiplier} * \text{ATR}$
- Final Band
 - 밴드를 벗어나지 않으면
 - 처음 상단으로 뚫으면 하단 밴드를 최대 값으로 유지
 - 처음 하단으로 뚫으면 상단 밴드를 최소 값으로 유지
- Super Trend
 - 최근 뚫은 것이 상단이면 하단 밴드, 아니면 상단 밴드
- 상단 뚫을 때 매수, 하단 뚫을 때 매도