FinRL Contest Task II

Elena Zhao*
Nikolaus Holzer*
elena.zhao@columbia.edu
nh2677@columbia.edu
Columbia University
New York, NY, USA

1 INTRODUCTION

The provided Python script outlines an advanced framework for executing trading strategies using three key financial analytics methods: Bollinger Bands, Z-Score, and Volume Weighted Average Price (VWAP). These methods are widely recognized in the trading community for their effectiveness in market analysis and decision-making. The document demonstrates an approach to incorporating these methodologies into a practical trading class.

2 TRADING STRATEGIES

(1) **Bollinger Bands:**
   - **Concept:** Bollinger Bands are a type of price envelope developed by John Bollinger. They consist of a middle band being an N-period simple moving average (SMA), and two outer bands calculated from the standard deviation of the price.
   - **Application in Script:** The script utilizes Bollinger Bands to identify overbought and oversold conditions in the market, informing buy or sell decisions based on the relative position of the asset price to these bands.
(2) **Z-Score:**
   - **Concept:** The Z-Score is a statistical measurement that describes a value's relationship to the mean of a group of values, measured in terms of standard deviations. In trading, it's used to identify price anomalies.
   - **Application in Script:** The script employs the Z-Score to gauge the extent to which an asset's price deviates from its historical norm, facilitating decisions on potential mean-reversion trades.
(3) **VWAP (Volume Weighted Average Price):**
   - **Concept:** VWAP is the average price a security has traded at throughout the day, based on both volume and price. It

is important because it provides traders with insight into both the trend and value of a security.
   - **Application in Script:** By integrating VWAP, the script can execute trades that align with the market's overall direction, capitalizing on the weighted average price as a benchmark.

3 ALGORITHM IN CODE

Listing 1: Buy and Sell Signals

```python
if current_price <= lower_band and z_score < -2
    and current_price < vwap and
    self.current_capital >= current_price:
    # The price is below the lower Bollinger
        Band, Z-score is significantly low, and
        below VWAP -> buy signal
    quantity = self.current_capital //
        current_price
    self.place_market_order('BUY', ticker,
        quantity, current_price)
elif current_price >= upper_band and z_score > 2
    and current_price > vwap and
    self.position.get(ticker, 0) > 0:
    # The price is above the upper Bollinger
        Band, Z-score is significantly high, and
        above VWAP -> sell signal
    self.place_market_order('SELL', ticker,
        self.position[ticker], current_price)
```

Listing 2: Calculating VWAP

```python
def calculate_vwap(self, ticker: str):
    if not self.price_history[ticker]:
        return
    prices = self.price_history[ticker]
    volumes = self.volume_history[ticker]
    df = pd.DataFrame({'price': prices, 'volume':
        volumes})
    df['cumulative_quantity'] =
        df['volume'].cumsum()
    df['cumulative_turnover'] = (df['price'] *
        df['volume']).cumsum()
    vwap_value =
        df['cumulative_turnover'].iloc[-1] /
        df['cumulative_quantity'].iloc[-1]
    self.vwap[ticker] = vwap_value
```

3.1 Conclusion

Our script addresses the outlined task of creating a Strategy class which implements an order execution strategy. The class structure,

as designed, is equipped with essential functions to facilitate efficient and strategic market trading. This includes the executing trades based on the defined strategies using Bollinger Bands, Z-Score, and VWAP metrics. Next steps could include incorporating more advanced statistical analysis such as regression models built from historical data.