به نام خدا

بخش اول

در این قسمت سعی داریم تا الگوریتم مورد نظرمان را پیاده سازی کنیم تا بر روی داده های مورد نظرمان تست کنیم.

برای الگوریتم مورد نظرمان نیاز به شاخص های MACD و RSI داریم.
برای محاسبه ی این دو تابع مربوط به هرکدام را مینویسی

```
def calculate_macd(close_prices):
    short_period=12
    long_period=26
    signal_period=9

    short_ema = close_prices.ewm(span=short_period, adjust=False).mean()
    long_ema = close_prices.ewm(span=long_period, adjust=False).mean()

    macd_line = short_ema - long_ema
    signal_line = macd_line.ewm(span=signal_period, adjust=False).mean()

    return macd_line, signal_line
```

تابع calculate_macd با گرفتن مجموعه داده close_prices، با توجه به پارامترهای استاندارد macd، ابتدا ema های مربوطه را روی close_prices حساب میکند.(با استفاده از متد ewm برای pada seriesو پارامتر های بازه ها (۱۲و ۲۶ به ترتیب برای short و long))
حال سیگنال macd که حاصل تفاضل این دو میباشد را حساب میکنیم.
در ادامه سینگنال لاین را که همان ema بازه ی ۹ میباشد نیز حساب میکنیم و این دو را برمیگردانیم

```
def calculate_rsi(close_prices):
    period=14
    delta = close_prices.diff()
    gain = (delta.where(delta > 0, 0)).rolling(window=period).mean()
    loss = (-delta.where(delta < 0, 0)).rolling(window=period).mean()
    rs = gain / loss
    rsi = 100 - (100 / (1 + rs))
    return rsi
```

این تابع برای محاسبه‌ی شاخص قدرت نسبی یا RSI طراحی شده است. RSI با عددی بین ۰ تا ۱۰۰، قدرت حرکت‌های قیمت در یک دوره زمانی مشخص (معمولاً ۱۴ روز) را نشان می‌دهد. این شاخص به معامله‌گران کمک می‌کند تا نقاط اشباع خرید (overbought) یا اشباع فروش (oversold) را شناسایی کنند.

مراحل محاسبه در این تابع:

1. **تعیین دوره**: دوره را برابر ۱۴ (به‌صورت پیش‌فرض برای RSI) تنظیم می‌کند.
2. **محاسبه‌ی تغییرات روزانه**: `delta` که اختلاف روزانه قیمت‌ها (`close_prices.diff()`) است، محاسبه می‌شود.
3. **جداسازی سود و زیان**:

- `gain`: سود را با انتخاب تغییرات مثبت (و صفر کردن بقیه) و محاسبه میانگین آن‌ها برای هر دوره ۱۴ روزه، به‌دست می‌آورد.
- `loss`: زیان را با انتخاب تغییرات منفی (و صفر کردن بقیه) و محاسبه میانگین آن‌ها برای هر دوره ۱۴ روزه، به‌دست می‌آورد.

4. **محاسبه‌ی RS و RSI**:
   - **RS**: نسبت سود به زیان (`gain / loss`).
   - **RSI**: استفاده از فرمول RSI، یعنی RSI=100- 100/1+RS برای تبدیل RS به عددی بین ۰ تا ۱۰۰.

نتیجه‌ی نهایی این تابع، سری زمانی RSI است.

```python
def backtest_strategy(close_prices: pd.Series, initial_capital: float):
    # indicators
    macd_line, signal_line = calculate_macd(close_prices)
    rsi = calculate_rsi(close_prices)

    # Initialize variables for tracking position and equity
    in_position = False  # True means we are in a position, False means no position
    equity = pd.Series(data=[0]*len(close_prices), index=close_prices.index)
    equity.iloc[0] = initial_capital  # Starting with initial capital on all days

    # Loop through each day to simulate trading
    for i in range(1, len(close_prices)):
        # Entry condition: MACD crosses above Signal and RSI > 48
        if (not in_position and macd_line.iloc[i] > signal_line.iloc[i]) and (macd_line.iloc[i - 1] <= signal_line.iloc[i - 1]) and (rsi.iloc[i] > 48):
            in_position = True  # Enter position

        # Exit condition: MACD crosses below Signal
        elif (in_position and macd_line.iloc[i] < signal_line.iloc[i]) and (macd_line.iloc[i - 1] >= signal_line.iloc[i - 1]):
            in_position = False  # Exit position

        # Calculate daily equity based on position
        if in_position:
            equity.iloc[i] = equity.iloc[i - 1] * (1 + ((close_prices.iloc[i] - close_prices.iloc[i-1]) / close_prices.iloc[i-1]))  # Gain or loss if in position
        else:
            equity.iloc[i] = equity.iloc[i - 1]  # Capital remains the same if out of position
```

این تابع یک استراتژی ما که مبتنی بر شاخص MACD و RSI را می باشد را شبیه‌سازی می‌کند و عملکرد آن را با استفاده از سرمایه‌ی اولیه بررسی می‌کند. هدف این تابع این است که تعیین کند با پیروی از این استراتژی چه میزان بازدهی یا سود دهی به‌دست می‌آید.

گام‌های تابع:

1. **محاسبه‌ی شاخص‌های فنی**:
   - ابتدا MACD و `signal line` (خط سیگنال MACD) را با تابع `calculate_macd` محاسبه می‌کند.
   - سپس RSI را با استفاده از تابع `calculate_rsi` به‌دست می‌آورد.
2. **متغیرهای وضعیت معاملاتی**:
   - `in_position`: اگر مقدار `True` باشد، یعنی یک موقعیت معاملاتی باز داریم؛ اگر `False` باشد، یعنی هیچ موقعیت بازی نداریم.
   - `equity`: سری زمانی برای ارزش دارایی مان در طول روزهای معاملاتی می باشد. در ابتدا مقدار آن برابر با سرمایه‌ی اولیه `initial_capital` تنظیم شده است.
3. **تکرار معامله در هر روز به ازای هر مقدار close_prices**:
   - این قسمت تابع در طول روزهای معاملاتی، استراتژی را به‌صورت زیر بررسی و اعمال می‌کند:
     - **شرایط ورود به معامله**: وقتی که خط MACD بالای خط سیگنال قرار می‌گیرد (تقاطع صعودی) و RSI بیشتر از ۴۸ باشد، یک معامله‌ی خرید باز می‌شود.
     - **شرایط خروج از معامله**: وقتی که خط MACD زیر خط سیگنال قرار می‌گیرد (تقاطع نزولی)، موقعیت معاملاتی بسته می‌شود.

- **به‌روزرسانی ارزش دارایی**: اگر در آن روز در موقعیت معاملاتی بودیم، ارزش دارایی در انتهای روز، بر اساس تغییرات قیمت به‌روزرسانی می‌شود(ارزش دارایی روز قبل در سود امروز به علاوه ی یک ضرب میشود تا دارایی امروزمان حاصل شود)؛ در غیر این‌صورت ارزش دارایی ثابت باقی می‌ماند.

4. **خروجی نهایی**:
   - سری زمانی `equity`، که شامل ارزش دارایی روزانه مان بعد از backtest می باشد، به‌عنوان نتیجه بازگشت داده می‌شود و نشان می‌دهد که استراتژی به چه صورت عمل کرده و چه مقدار بازدهی ایجاد شده است.

بخش دوم

در این قسمت سعی داریم تا معیارهای سنجش الگوریتم مان را پیاده سازی کنیم تا با استفاده از آن ها به دید بهتری نسبت به عملکرد الگوریتممان برسیم.

```
def calculate_netProfit(equity):
    return equity.iloc[-1] - equity.iloc[0]
```

این تابع، **سود خالص** استراتژی را محاسبه می‌کند، که تفاوت بین اولین و آخرین مقدار سرمایه (`equity`) است.

- **ورودی**: سری زمانی `equity` که ارزش سرمایه را در طول دوره معاملاتی نشان می‌دهد.
- **خروجی**: مقدار سود خالص.

```
def calculate_max_drawdown(equity):
    peak = equity.cummax()
    drawdown = (equity - peak) / peak
    return drawdown.min()*100
```

این تابع برای محاسبه **بیشترین افت سرمایه** (Max Drawdown) طراحی شده است. بیشترین افت سرمایه، بزرگترین درصد افت از یک نقطه اوج (حداکثر) به پایین‌ترین نقطه بعدی در طول زمان است.

- **مراحل**:
  1. **پیدا کردن نقاط اوج**: `peak` که حداکثر تجمعی سرمایه است و با استفاده از `equity.cummax()` محاسبه می‌شود.
  2. **محاسبه‌ی افت سرمایه**: با استفاده از فرمول equity - peak / peak برای به دست آوردن نسبت افت به پیک قبلی.
  3. **بیشترین افت سرمایه**: کمترین مقدار از این افت‌ها به عنوان بیشترین افت سرمایه بازگشت داده می‌شود، که در اینجا به درصد تبدیل شده است.
- **خروجی**: بیشترین افت سرمایه به صورت درصدی.

```python
def calculate_drawdown_duration(equity_curve):
    cumulative_max = equity_curve.cummax()
    drawdown = equity_curve / cumulative_max - 1
    in_drawdown = drawdown < 0
    drawdown_durations = in_drawdown.astype(int).groupby((in_drawdown != in_drawdown.shift()).cumsum()).cumsum()

    max_duration = drawdown_durations.max()
    return max_duration, drawdown_durations
```

این تابع **دوره‌های افت سرمایه و همچنین طولانی ترین مدت افت سرمایه** را محاسبه می‌کند، یعنی مدت زمانی که سرمایه در یک دوره افت قرار داشته است.

- **مراحل**:
  - **محاسبه نقاط اوج**: `cumulative_max` به‌عنوان بالاترین ارزش تجمعی سرمایه.
  - **محاسبه افت سرمایه**: نسبت سرمایه جاری به اوج‌های قبلی.
  - **شناسایی دوره‌های افت**: `in_drawdown`، که نشان می‌دهد آیا سرمایه در یک دوره افت قرار دارد یا خیر.
  - **محاسبه مدت زمان افت**: مدت زمان هر دوره افت را با شمارش روزهای متوالی که در حالت افت هستند محاسبه می‌کند.
- **خروجی**: دو مقدار:
  - `max_duration`: طولانی‌ترین دوره افت.
  - `drawdown_durations`: دوره‌های زمانی افت برای هر بازه ای که در افت (drawdown) قرار دارد.

```python
def calculate_annual_sharp_ratio(equity, N=255, rf=0.04):
    returns = equity.pct_change()
    mean = returns.mean() * N -rf
    sigma = returns.std() * np.sqrt(N)
    return mean / sigma
```

این تابع **نسبت شارپ سالانه** را محاسبه می‌کند که معیاری برای اندازه‌گیری بازده به ازای ریسک (نوسان) است. نسبت شارپ بالاتر (معمولا بیشتر از ۰.۷ خوب است) نشان‌دهنده بازده ای بیشتر نسبت به ریسک پذیرفته شده است.

- **ورودی‌ها**:
  1. `equity`: سری زمانی ارزش سرمایه.
  2. `N`: تعداد روزهای معاملاتی در سال (معمولاً ۲۵۵).
  3. `rf`: نرخ بهره بدون ریسک (معمولاً ۰.۰۴ یا ۴٪).
- **مراحل**:
  1. محاسبه **بازده روزانه** (`returns`).
  2. محاسبه **بازده میانگین سالانه تعدیل‌شده با نرخ بدون ریسک** (`mean`).
  3. محاسبه **انحراف معیار سالانه بازده‌ها** (`sigma`).
  4. تقسیم میانگین بازده ها بر انحراف معیار سودها به عنوان نسبت sharpe ratio.
- **خروجی**: sharpe ratio سالانه.

```python
def calculate_annual_sortino_ratio(equity, N=255,rf=0.04):
    returns = equity.pct_change()
    mean = returns.mean() * N -rf
    std_neg = returns[returns<0].std()*np.sqrt(N)
    return mean/std_neg
```

این تابع **نسبت سورتینو سالانه** را محاسبه می‌کند. نسبت سورتینو شباهت زیادی به نسبت شارپ دارد، با این تفاوت که در این نسبت فقط به نوسانات منفی عملکرد الگوریتم مان توجه می‌شود. این نسبت به ما کمک می‌کند تا بازدهی را به ازای ریسک نزولی ارزیابی کنیم.

- **ورودی‌ها**:
  1. `equity`: سری زمانی ارزش سرمایه حاصل از الگوریتم مان.
  2. `N`: تعداد روزهای معاملاتی در سال (۲۵۵).
  3. `rf`: نرخ بهره بدون ریسک(۰.۰۴ به عنوان داده ی مسئله).
- **مراحل**:
  1. محاسبه **بازده روزانه** (`returns`).
  2. محاسبه **بازده میانگین سالانه تعدیل‌شده با نرخ بدون ریسک** (`mean`).
  3. محاسبه **انحراف معیار نوسانات منفی سالانه** (`std_neg`) (تنها برای بازده هایی که منفی باشند).
  4. تقسیم میانگین بازده ها بر انحراف معیار منفی به عنوان sortino ratio.
- **خروجی**: sortino ratio سالانه.

بخش سوم

در این قسمت داده های مربوط به رمزارزهای دلخواهمان را استخراج میکنیم تا بتوانیم الگوریتم مان را روی آن ها تست کنیم

```
import yfinance as yf
import pandas as pd

tickers = {"BTC-USD", "ETH-USD", "DOGE-USD"}

start_date = "2022-10-01"
end_date = "2024-10-01"

crypto_data = {}

# Download data for each cryptocurrency
for ticker in tickers:
    data = yf.download(ticker, start=start_date, end=end_date)
    crypto_data[ticker] = data['Adj Close']
Bitcoin_adj_closes = pd.Series(crypto_data['BTC-USD'].values[:,0])
Ethereum_adj_closes = pd.Series(crypto_data['ETH-USD'].values[:,0])
Dogecoin_adj_closes = pd.Series(crypto_data['DOGE-USD'].values[:,0])
```

برای سه رمزارز بیت کوین، اتریوم و دوج کوین داده های آن را از yahoo finance برای بازه های 2022-10-01 تا 2024-10-01 دانلود

میکنیم و سپس ستون Adj Close (معادل Adjusted Close) را به عنوان داده های سری زمانی موردنیازمان برای هر رمز ارز در آرایه متناظر آن رمز ارز به عنوان panda series (برای سهولت کار) ذخیره میکنیم.

```
equity4Bitcoin = backtest_strategy(Bitcoin_adj_closes, 100)
equity4Ethereum = backtest_strategy(Ethereum_adj_closes, 100)
equity4Dogecoin = backtest_strategy(Dogecoin_adj_closes, 100)
```

الگوریتم را روی این داده ها ران و بک تست میکنیم و نتایج را در سری زمانی های متناظر ذخیره میکنیم تا با استفاده از شاخص های ارزیابی مناسبی از آن ها داشته باشیم.

```
def calculate_measures(equity, crypto):
  Net_proit = calculate_netProfit(equity)
  Maximum_drawdown = calculate_max_drawdown(equity)
  Maximum_drawdown_period, Duration_of_drawdown = calculate_drawdown_duration(equity)
  Annual_sharp_ratio = calculate_annual_sharp_ratio(equity)
  Annual_sortino_ratio = calculate_annual_sortino_ratio(equity)
  print(f"Net profit for {crypto} is {Net_proit}$")
  print(f"Maximum drawdown for {crypto} is {Maximum_drawdown}")
  print(f"Duration of drawdown for {crypto} is \n{Duration_of_drawdown.values}")
  print(f"Maximum drawdown period for {crypto} is {Maximum_drawdown_period} days")
  print(f"Annual sharpe ratio for {crypto} is {Annual_sharp_ratio}")
  print(f"Annual sortino ratio for {crypto} is {Annual_sortino_ratio}")
```

با استفاده از این تابع مقدار شاخص های مختلف که در بخش قبل پیاده سازی کردیم را میتوانیم محاسبه و خروجی دهیم.

برای بیتکوین

Bitcoin

```
calculate_measures(equity4Bitcoin, "Bitcoin")
```

```
Net profit for Bitcoin is 218.97841234318886$
Maximum drawdown for Bitcoin is -8.247514008869924
Duration of drawdown for Bitcoin is
[  0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
   0   0   0   0   0   1   0   0   1   2   0   1   2   3   4   5   0   0
   1   2   3   4   5   6   7   8   9  10  11  12  13  14  15  16  17  18
  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36
  37  38  39  40  41  42  43  44  45  46  47  48  49  50  51  52  53  54
  55  56  57  58  59  60  61  62  63  64  65  66  67  68  69  70  71  72
Maximum drawdown period for Bitcoin is 162 days
Annual sharpe ratio for Bitcoin is 1.9337744784941078
Annual sortino ratio for Bitcoin is 1.9484377903853445
```

برای اتریوم

Ethereum

```
calculate_measures(equity4Ethereum, "Ethereum")
```

```
Net profit for Ethereum is 404.5220009721855$
Maximum drawdown for Ethereum is -9.53664752493234
Duration of drawdown for Ethereum is
[  0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
   0   0   0   0   0   1   0   0   1   2   0   1   2   3   4   5   0   1
   2   3   4   5   6   7   8   9  10  11  12  13  14  15  16  17  18  19
  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37
  38  39  40  41  42  43  44  45  46  47  48  49  50  51  52  53  54  55
  56  57  58  59  60  61  62   0   1   0   0   0   0   0   0   0   0   0
Maximum drawdown period for Ethereum is 144 days
Annual sharpe ratio for Ethereum is 2.103952768637083
Annual sortino ratio for Ethereum is 2.334873850381769
```

برای دوج کوین

Dogecoin

```
calculate_measures(equity4Dogecoin, "Dogecoin")
```

```
Net profit for Dogecoin is 430.767189652539745$
Maximum drawdown for Dogecoin is -19.944663332826362
Duration of drawdown for Dogecoin is
[  0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
   0   0   0   0   1   0   0   1   2   3   4   5   6   7   8   9  10  11
  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29
  30  31  32  33  34  35  36  37  38  39  40  41  42  43  44  45  46  47
Maximum drawdown period for Dogecoin is 165 days
Annual sharpe ratio for Dogecoin is 1.547797256156177
Annual sortino ratio for Dogecoin is 1.4377622955396172
```

بخش چهارم

ارزیابی معیارها:

- **Net profit:** دوج‌کوین بیشترین سود خالص را دارد (۴۳۰.۷۶)، سپس اتریوم (۴۰۴.۵۲) و بیت‌کوین (۲۱۸.۹۱). در اینجا بیت کوین سود بسیار کمتری نسبت به دو ارز دیگر دارد که پوینت منفی برایش حساب میشود.
- **Max drawdown:** بیت‌کوین کمترین افت سرمایه را دارد (-۸.۲۴%)، که نشان‌دهنده ریسک کمتر نسبت به اتریوم (-۹.۵۳%) و دوج‌کوین (-۱۹.۹%) است. در اینجا دوج کوین اختلاف بسیار بیشتری با بقیه دارد که یک نمره ی منفی برایش محسوب میشود.
- **Max drawdown period:** اتریوم کوتاه‌ترین دوره افت سرمایه را دارد (۱۴۴ روز)، یعنی زمان کمتری را در فاز افت سرمایه نسبت به بیت‌کوین (۱۶۲ روز) و دوج‌کوین (۱۶۵ روز) گذرانده است.در اینجا اتریوم نیز با اختلاف بیشتری نسبت به رقبایش، دوره افت کمتری دارد پس امتیاز مثبت برایش حساب میشود.
- **Sharpe ratio:** اتریوم بالاترین نسبت شارپ را دارد (۲.۱۰)، که نشان‌دهنده بازدهی بهتری نسبت به ریسک در مقایسه با بیت‌کوین (۱.۹۳) و دوج‌کوین (۱.۵۴) است. در اینجا نیز دوج کوین شارپ کمتری نسبت به رقبایش دارد که امتیاز منفی دیگری برایش تلقی می شود، همچنین برای اتریوم که بالاترین مقدار را دارد امتیاز مثبت حساب میشود.
- **Sortino ratio:** اتریوم همچنین بالاترین نسبت سورتینو را دارد (۲.۳۳)، که بیانگر بازدهی بهتری نسبت به ریسک‌های نزولی در مقایسه با بیت‌کوین (۱.۹۴) و دوج‌کوین (۱.۴۳) است. در اینجا اتریوم نسبت سورتینوی بیشتری نسبت به دو ارز دیگر دارد(امتیاز مثبت) و دوج کوین کمترین را دارد(امتیاز منفی)

تحلیل:

- **ریسک و پایداری:** بیت‌کوین کمترین افت سرمایه را دارد، که برای سرمایه‌گذاران ریسک‌گریز مثبت است، اما سود کمتر و نسبت‌های بازدهی به ریسک پایین‌تر، آن را در مقایسه با اتریوم کمتر جذاب می‌کند.
- **سود در مقابل بازدهی به ریسک:** با اینکه دوج‌کوین بالاترین سود خالص را دارد، اما افت سرمایه زیاد (-۱۹.۹%) و نسبت‌های شارپ و سورتینو پایین‌تر، نشان می‌دهد که این سود با ریسک بالاتر و پایداری کمتری حاصل شده است.
- **عملکرد متوازن:** اتریوم با داشتن بالاترین نسبت‌های شارپ و سورتینو، سود نسبتاً بالا (در مقایسه با بیت کوین و نزدیک دوج کوین)، افت سرمایه متوسط، و کوتاه‌ترین دوره افت سرمایه، عملکرد متوازن و بهتری دارد.

نتیجه‌گیری: اتریوم بهترین ارز برای دوره مشخص‌شده است. علی‌رغم سود کمی بیشتر دوج‌کوین، ترکیب سود بالا، بازدهی قوی به ریسک (sharpe ratio و sortino ratio)، افت سرمایه متوسط (نزدیک بیت کوین) و زمان کوتاه‌تر بهبودی افت سرمایه(max drawdown period) در اتریوم نشان می‌دهد که این ارز دیجیتال عملکردی با تعادل مناسب بین ریسک و بازدهی ارائه می‌دهد.

پس بهترین انتخاب در بین این رمز ارز ها اتریوم می باشد.