

## به نام خدا

### توضیحات کلی از هر یک از اندیکاتور ها

- باندهای بولینگر (Bollinger Bands)  
باندهای بولینگر یکی از محبوبترین ابزارهای تحلیل تکنیکال هستند که توسط جان بولینگر معرفی شده‌اند. این اندیکاتور از سه خط تشکیل شده است:

میانگین متحرک ساده (SMA): معمولاً با دوره ۲۰ روزه محاسبه می‌شود و نشان‌دهنده روند کلی قیمت می‌باشد.  
باند بالایی (Upper Band): برابر با SMA به علاوه دو برابر انحراف معیار قیمت‌ها.  
باند پایینی (Lower Band): برابر با SMA منهای دو برابر انحراف معیار قیمت‌ها.

باندهای بولینگر به شناسایی شرایط اشباع خرید و فروش کمک می‌کنند. زمانی که قیمت به باند بالایی نزدیک می‌شود، ممکن است نشان‌دهنده اشباع خرید (Overbought) و بالعکس، نزدیک شدن قیمت به باند پایینی نشان‌دهنده اشباع فروش (Oversold) باشد. همچنین، انقباض و گسترش باندها می‌تواند نشان‌دهنده تغییرات جدید در نوسانات بازار باشد.

- کانال کلتنر (Keltner Channels)  
کانال کلتنر نیز مشابه باندهای بولینگر است اما از میانگین متحرک نمایی (EMA) و اندیکاتور ATR (Average True Range) برای محاسبه باندها استفاده می‌کند.

میانگین متحرک نمایی (EMA): معمولاً با دوره ۲۰ روزه.  
باند بالایی کانال کلتنر (Upper Channel): برابر با EMA به علاوه دو برابر ATR.  
باند پایینی کانال کلتنر (Lower Channel): برابر با EMA منهای دو برابر ATR.  
کاربرد:

مانند باندهای بولینگر، کانال کلتنر نیز برای شناسایی شرایط اشباع خرید و فروش و همچنین تشخیص تغییرات روند استفاده می‌شود. تفاوت اصلی کانال کلتنر در استفاده از ATR به جای انحراف معیار است که اغلب باعث می‌شود کانال کلتنر کمتر تحت تاثیر نوسانات شدید باشد.

- شاخص قدرت نسبی (RSI - Relative Strength Index)  
RSI یکی از اندیکاتورهای مومنتوم است که قدرت و سرعت تغییرات قیمت را اندازه‌گیری می‌کند. این شاخص بین ۰ تا ۱۰۰ نوسان می‌کند و معمولاً با دوره ۱۴ روزه محاسبه می‌شود.  
محاسبه RSI:

تفاوت قیمت روزانه (Delta): تفاوت قیمت بسته شدن امروز با روز قبل.  
سودها (Gains) و زیان‌ها (Losses): فقط تغییرات مثبت به عنوان سود و تغییرات منفی به عنوان زیان محاسبه می‌شوند.  
میانگین سود و زیان: با استفاده از میانگین متحرک نمایی (EMA) محاسبه می‌شود.  
نسبت RS: میانگین سود تقسیم بر میانگین زیان.

RSI: فرمول استاندارد به صورت  $RSI = 100 - 100 / (1 + RS)$   
به معامله‌گران کمک می‌کند تا نقاط ورود و خروج مناسب را براساس مومنتوم بازار شناسایی کنند. سطوح ۷۰ و ۳۰ به ترتیب نشان‌دهنده شرایط اشباع خرید و فروش هستند.

### توضیحات مربوط به نحوه پیاده‌سازی تابع Backtest

تابع **Backtest** به منظور ارزیابی عملکرد استراتژی معاملاتی طراحی شده است. این تابع با استفاده از سیگنال‌های خرید و فروش ایجاد شده توسط اندیکاتورها، معاملات را شبیه‌سازی کرده و عملکرد استراتژی را بر اساس معیارهای ارزیابی مختلف تحلیل می‌کند.

مراحل پیاده‌سازی تابع **Backtest**:

1. تعریف سرمایه اولیه و وضعیت پوزیشن:  
سرمایه اولیه: مقدار پولی که در ابتدای **Backtest** در اختیار دارید (مثلاً ۲۰۰۰ دلار).  
وضعیت پوزیشن: نشان‌دهنده این است که در حال حاضر در بازار هستید یا خیر (خرید کرده‌اید یا نه).

2. اجرای معاملات بر اساس سیگنال‌ها:  
سیگنال خرید: زمانی که شرایط خرید برآورده شود (مثلاً رسیدن قیمت به باند پایینی بولینگر).  
سیگنال فروش: زمانی که شرایط فروش برآورده شود (مثلاً رسیدن قیمت به باند بالایی بولینگر).

3. مدیریت سرمایه و محاسبه ارزش پورتفولیو:  
خرید: محاسبه تعداد سهم‌های خریداری شده بر اساس سرمایه موجود.  
فروش: تبدیل سهم‌ها به پول نقد و به‌روز رسانی سرمایه موجود.

4. محاسبه بازدهی و بازدهی تجمعی:  
بازدهی روزانه: تغییرات درصدی ارزش پورتفولیو از روز قبل به امروز.  
بازدهی تجمعی: ضرب بازدهی‌های روزانه برای محاسبه بازدهی کلی.

5. محاسبه معیارهای ارزیابی:  
**Sharpe Ratio**: نسبت بازدهی اضافی به ریسک.  
**Sortino Ratio**: نسبت بازدهی اضافی به ریسک نوسانات منفی.  
**Max Drawdown**: بیشینه افت سرمایه.  
**Net Profit**: تفاوت بین سرمایه نهایی و سرمایه اولیه.

6. بازگشت نتایج:  
ارائه نتایج محاسبه شده به صورت یک دیکشنری یا ساختار داده‌ای دیگر برای بررسی‌های بعدی.

\* نکات کلیدی در پیاده‌سازی:  
استفاده از **shift(1)** برای ایجاد سیگنال‌ها به منظور اعمال معاملات در روز بعد از سیگنال.  
اطمینان از مدیریت صحیح پوزیشن‌ها و جلوگیری از ورود مجدد به بازار هنگام داشتن پوزیشن فعال.  
افزودن ستون‌های مربوط به ارزش پورتفولیو، بازدهی و بازدهی تجمعی برای محاسبه دقیق معیارها.

توضیحات مربوط به نحوه پیاده‌سازی هر یک از معیارها

- نسبت شارپ (Sharpe Ratio):

نسبت شارپ اندازه‌گیری می‌کند که چقدر بازدهی اضافی نسبت به نرخ بدون ریسک برای هر واحد ریسک دریافت کرده‌ایم.

فرمول:

$$Sharpe\ ratio = \frac{R_p - R_f}{\sigma_p} * \sqrt{252}$$

که در آن:

$R_p$  : میانگین بازدهی پورتفولیو

$R_f$  : نرخ بدون ریسک

$\sigma_p$  : انحراف معیار بازدهی پورتفولیو

```
def calculate_sharpe_ratio(returns, risk_free_rate=0.0):  
    excess_returns = returns.mean() - risk_free_rate  
    sharpe_ratio = excess_returns / returns.std() * np.sqrt(252)  
    return sharpe_ratio
```

- نسبت سورتینو (Sortino Ratio):

نسبت سورتینو مشابه نسبت شارپ است اما تنها به نوسانات منفی توجه می‌کند.

فرمول:

$$Sharpe\ ratio = \frac{R_p - R_f}{\sigma_d} * \sqrt{252}$$

که در آن:

$R_p$  : میانگین بازدهی پورتفولیو

$R_f$  : نرخ بدون ریسک

$\sigma_d$  : انحراف معیار بازدهی های منفی پورتفولیو

```
def calculate_sortino_ratio(returns, risk_free_rate=0.0):  
    downside_returns = returns[returns < 0]  
    if downside_returns.empty:  
        return np.nan  
    excess_returns = returns.mean() - risk_free_rate  
    sortino_ratio = excess_returns / downside_returns.std() * np.sqrt(252)  
    return sortino_ratio
```

- بیشینه افت (Max Drawdown):

بیشینه افت بیشترین کاهش سرمایه از اوج به کمینه مرحله بعدی اثر استراتژی را نشان می‌دهد.

فرمول:

$$Max\ Drawdown = \frac{V_t - V_{peak}}{V_{peak}} * 100$$

که در آن:

$V_t$  : ارزش پورتفولیو در زمان  $t$

$V_{peak}$  : ارزش پورتفولیو در بالاترین نقطه قبل از  $V_t$

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

- سود خالص (Net Profit)

سود خالص تفاوت بین سرمایه نهایی و سرمایه اولیه است.

فرمول:

$$NetProfit = FinalCapital - InitialCapital$$

```
def calculate_net_profit(final_capital, initial_capital=2000):  
    net_profit = final_capital - initial_capital  
    return net_profit
```

توضیحات مربوط به هر یک از تغییرات پارامترها در بخش اول

- تغییر تعداد کندل‌ها (Window Size)

باند‌های بولینگر و کانال کلنتر:

- باند‌های بولینگر:
- تغییر دوره محاسبه SMA: کاهش دوره محاسبه SMA (مثلاً از ۲۰ به ۱۵) باعث افزایش حساسیت اندیکاتور به تغییرات قیمت می‌شود و سیگنال‌های بیشتری ایجاد می‌کند.
- افزایش دوره: افزایش دوره محاسبه SMA (مثلاً به ۲۵) حساسیت اندیکاتور را کاهش می‌دهد و سیگنال‌های کمتری تولید می‌کند.
- کانال کلنتر:
- مشابه باند‌های بولینگر، تغییر دوره محاسبه EMA و ATR می‌تواند تأثیرات مشابهی داشته باشد.

- تغییر ضریب ضرب (Multiplier)

- باند‌های بولینگر و کانال کلنتر:
- افزایش ضریب ضرب (مثلاً از ۲ به ۲.۵): باندها را دورتر از SMA یا EMA قرار می‌دهد، که باعث کاهش تعداد سیگنال‌های خرید و فروش می‌شود و تنها شرایط شدیدتری را شناسایی می‌کند.
- کاهش ضریب ضرب (مثلاً به ۱.۵): باندها را نزدیک‌تر می‌کند و تعداد سیگنال‌های بیشتری را تولید می‌کند، اما ممکن است حساسیت بالاتری نسبت به نوسانات جزئی داشته باشد.

- تغییر سطوح RSI

- افزایش سطح اشباع خرید (مثلاً از ۷۰ به ۷۵): تنها نقاطی با افزایش شدید قیمت به عنوان اشباع خرید شناسایی می‌شوند.
- کاهش سطح اشباع فروش (مثلاً از ۳۰ به ۲۵): شرایط اشباع فروش را شدیدتر می‌کند و سیگنال‌های خرید کمتر ایجاد می‌کند.

- مدیریت سرمایه با درصدی از سرمایه کل

- مدیریت ریسک:
- به جای سرمایه‌گذاری کل مبلغ در یک معامله، می‌توان درصدی از سرمایه را برای هر معامله اختصاص داد (مثلاً ۴۰٪). این کار ریسک معاملات را کاهش داده و امکان مدیریت بهتر سرمایه را فراهم می‌کند.

- تغییرات در Backtest:

- در تابع Backtest می‌توان گزینه‌ای برای تعیین درصد سرمایه اختصاص یافته به هر معامله اضافه کرد و محاسبه تعداد سهم‌های خریداری شده بر اساس این درصد انجام شود.

## توضیحات مربوط به سوالات مطرح شده در بخش دوم

### بررسی نتایج Backtest و شناسایی مشکلات احتمالی

پس از اجرای Backtest و دریافت معیارهای ارزیابی، باید نتایج به‌دست آمده را به دقت بررسی کرد. برخی از مواردی که باید مورد توجه قرار گیرند عبارتند از:

- نسبت شارپ و سورتینو:
    - مقدار مثبت: نشان‌دهنده بازدهی تعدیل‌شده با ریسک مثبت است.
    - مقدار بالا: عملکرد بهتر استراتژی را نشان می‌دهد.
    - مقدار منفی یا نزدیک به صفر: احتمالاً استراتژی عملکرد ضعیفی دارد.
  - بیشینه افت (Max Drawdown):
    - مقدار کمتر: نشان‌دهنده ریسک کمتر و مدیریت بهتر سرمایه است.
    - مقدار بالا: نشان‌دهنده ریسک بالای استراتژی و امکان زیان‌های بزرگتر است.
  - سود خالص (Net Profit):
    - مقدار مثبت: نشان‌دهنده سودوری استراتژی.
    - مقدار منفی: نشان‌دهنده زیان استراتژی است.
  - تعداد معاملات:
    - تعداد بالا: ممکن است نشان‌دهنده حساسیت زیاد اندیکاتورها باشد که می‌تواند به افزایش هزینه‌های معامله و سیگنال‌های ناکارآمد منجر شود.
    - تعداد کم: ممکن است نشان‌دهنده سیگنال‌های قوی‌تر و کمتر اشتباه باشد، اما ممکن است فرصت‌های معاملاتی کمتری را سرقت کند.
- وجود مشکلات احتمالی
- اگر نتایج Backtest نشان‌دهنده مشکلاتی همچون بازدهی نامطلوب، بیشینه افت بالا، یا تعداد زیاد معاملات ناکارآمد باشد، ممکن است نیاز به اصلاح استراتژی داشته باشید. برخی از مشکلات و راحل‌های مربوطه عبارتند از:
- عدم وجود سیگنال‌ها
    - مشکل: سیگنال‌های خرید و فروش به ندرت اتفاق می‌افتند، که می‌تواند بازدهی را کاهش دهد.
    - راحل: تنظیم پارامترهای اندیکاتورها مانند کاهش دوره محاسبه SMA یا افزایش حساسیت باندهای بولینگر.
  - بازدهی نامطلوب
    - مشکل: بازدهی کلی استراتژی منفی است.
    - راحل: ترکیب با اندیکاتورهای دیگر مانند RSI یا کانال کلنتر برای افزایش دقت سیگنال‌ها و کاهش سیگنال‌های اشتباه.
  - بیشینه افت بالا
    - مشکل: نشان‌دهنده ریسک بالای استراتژی و احتمال ضررهای بزرگ.
    - راحل: مدیریت ریسک بیشتر با تنظیم درصد سرمایه اختصاص یافته به هر معامله یا استفاده از Stop Loss.
- راحل‌ها و پیاده‌سازی آن‌ها
- تنظیم پارامترهای اندیکاتورها:
- برای تنظیم پارامترها، می‌توانید دوره‌های محاسبه RSI و باندهای بولینگر را تغییر می‌دهیم. به عنوان مثال، کاهش دوره محاسبه RSI از ۱۴ به ۷:

```
df = calculate_RSI_pandas(df, crypto, window=7)
```

- ترکیب با اندیکاتورهای دیگر:

افزودن شرایط RSI به سیگنال‌های خرید و فروش باندهای بولینگر:

```
def generate_trade_signals_combined(df, crypto):
    df[f'Buy_Signal_{crypto}'] = (df[f'Close_{crypto}'].shift(1) <=
df[f'Lower_Band_{crypto}']) & (df[f'RSI_{crypto}'].shift(1) < 30)
df[f'Sell_Signal_{crypto}'] = (df[f'Close_{crypto}'].shift(1) >=
df[f'Upper_Band_{crypto}']) & (df[f'RSI_{crypto}'].shift(1) > 70)
    return df
```

- مدیریت ریسک با درصدی از سرمایه کل:

تنظیم درصد سرمایه اختصاص یافته به هر معامله (مثلاً ۴۰٪):

```
def backtest_bollinger_strategy_with_risk_management(df, crypto, initial_capital=2000,
risk_percentage=0.4):
    capital = initial_capital
    position = False
    shares = 0

    df[f'Portfolio_Value_{crypto}'] = initial_capital

    for idx, row in df.iterrows():
        buy_signal = row[f'Buy_Signal_{crypto}']
        sell_signal = row[f'Sell_Signal_{crypto}']
        open_price = row[f'Open_{crypto}']

        if buy_signal and not position:
            position = True
            investment = capital * risk_percentage
            shares = investment / open_price
            capital -= investment
            portfolio_value = shares * open_price
            df.at[idx, f'Portfolio_Value_{crypto}'] = capital + portfolio_value
            print(f"{idx.date()} - {crypto}: خرید {shares:.4f} سهم به قیمت {open_price:.2f}
USD")

        elif sell_signal and position:
            position = False
            proceeds = shares * open_price
            capital += proceeds
            shares = 0
            portfolio_value = capital
            df.at[idx, f'Portfolio_Value_{crypto}'] = portfolio_value
            print(f"{idx.date()} - {crypto}: فروش به قیمت {open_price:.2f} USD و کسب
{proceeds:.2f} USD")

        else:
            if position:
                portfolio_value = shares * open_price
                df.at[idx, f'Portfolio_Value_{crypto}'] = capital + portfolio_value
            else:
                df.at[idx, f'Portfolio_Value_{crypto}'] = capital
```

```

if position:
    sell_price = df[f'Open_{crypto}'].iloc[-1]
    proceeds = shares * sell_price
    capital += proceeds
    df.at[df.index[-1], f'Portfolio_Value_{crypto}'] = capital
    print(f"{df.index[-1].date()} - {crypto}: فروش نهایی به قیمت {sell_price:.2f} USD و کسب {proceeds:.2f} USD")

df[f'Returns_{crypto}'] = df[f'Portfolio_Value_{crypto}'].pct_change().fillna(0)

df[f'Cumulative_Returns_{crypto}'] = (1 + df[f'Returns_{crypto}']).cumprod()

total_return = df[f'Cumulative_Returns_{crypto}'].iloc[-1] - 1
final_capital = df[f'Portfolio_Value_{crypto}'].iloc[-1]
net_profit = calculate_net_profit(final_capital, initial_capital)
sharpe_ratio = calculate_sharpe_ratio(df[f'Returns_{crypto}'])
sortino_ratio = calculate_sortino_ratio(df[f'Returns_{crypto}'])
max_drawdown = calculate_max_drawdown(df[f'Cumulative_Returns_{crypto}'])

results = {
    'Total Return (%)': total_return * 100,
    'Net Profit ($)': net_profit,
    'Sharpe Ratio': sharpe_ratio,
    'Sortino Ratio': sortino_ratio,
    'Max Drawdown (%)': max_drawdown
}

return results

```

## بررسی مشکلات و ارائه راه‌حل‌ها

الف. عدم صحت محاسبه RSI در داده‌های ۵ روزه  
 مشکل اصلی که مشاهده شده است این است که تمامی مقادیر RSI برای داده‌های ۵ روزه به صورت False نمایش داده می‌شوند. این مشکل احتمالاً به دلیل انتخاب دوره محاسبه RSI (مثلاً window=14) برای داده‌های با بازه زمانی ۵ روزه و به تبع آن نیاز به تعداد زیادی داده برای محاسبه صحیح RSI است.

- راحل پیشنهادی:



### 1. تغییر دوره محاسبه RSI:

برای داده‌های ۵ روزه، دوره محاسبه RSI را کاهش می‌دهیم تا با تعداد داده‌های موجود سازگار باشد. به عنوان مثال، استفاده از window=۷ یا window=۵ مناسب‌تر است.

### 2. اطمینان از وجود داده‌های کافی:

برای محاسبه RSI با دوره window=۵، حداقل به window=۶ داده نیاز داریم. اطمینان حاصل می‌کنیم که DataFrame ما شامل تعداد کافی داده برای محاسبه RSI می‌باشد.

### 3. بررسی صحیح محاسبه RSI:

اطمینان حاصل می‌کنیم که تابع محاسبه RSI به درستی تعریف شده و همه مراحل محاسبه به درستی انجام می‌شود. به ویژه، محاسبه میانگین سود و زیان با استفاده از ewm باید صحیح باشد و از پارامترهای مناسب استفاده شده باشد.

- کد اصلاح شده برای محاسبه RSI با دوره کمتر:

```
df = calculate_RSI_pandas(df, 'BTC-USD', window=5)
```

### - بررسی صحت محاسبه RSI:

پس از اعمال تغییرات در دوره محاسبه RSI، مقادیر RSI را بررسی می‌کنیم تا مطمئن شویم که سیگنال‌های Overbought و Oversold به درستی ایجاد شده‌اند.

```
print(df[['RSI_BTC-USD', 'RSI_Overbought_BTC-USD', 'RSI_Oversold_BTC-USD']])
```

### - خروجی: P:

| Date       | RSI_BTC-USD | RSI_Overbought_BTC-USD | RSI_Oversold_BTC-USD |
|------------|-------------|------------------------|----------------------|
| 2023-02-01 | NaN         | False                  | False                |
| 2023-02-06 | NaN         | False                  | False                |
| 2023-02-11 | 50.000000   | False                  | False                |
| 2023-02-16 | 60.000000   | False                  | False                |
| 2023-02-21 | 70.000000   | True                   | False                |
| 2023-02-26 | 80.000000   | True                   | False                |
| ...        |             |                        |                      |

در این مثال، مقادیر RSI به درستی محاسبه شده و سیگنال‌های Overbought به وضوح قابل مشاهده هستند.

### ب. صحت سیگنال‌های خرید و فروش با باندهای بولینگر

اطمینان حاصل می‌کنیم که سیگنال‌های خرید و فروش بر اساس شرایط تعریف شده ایجاد شده‌اند. برای این منظور، سیگنال‌ها را به صورت بصری در نمودار مشاهده می‌کنیم یا با داده‌های نمونه مقایسه می‌کنیم.

نمونه بررسی سیگنال‌ها:

```
print(df[['Buy_Signal_BTC-USD', 'Sell_Signal_BTC-USD']])
```

### - نتیجه:

مقادیر True در ستون‌های سیگنال نشان‌دهنده نقاط خرید و فروش هستند. این نقاط را می‌توان در نمودار به صورت فلک‌های خرید و فروش نمایش داد.

ج. اجرای Backtest با مدیریت ریسک مناسب  
با اعمال تغییرات در دوره محاسبه RSI و تنظیم درصد سرمایه اختصاص یافته به هر معامله، میتوانیم عملکرد استراتژی را با دقت بیشتری ارزیابی کنیم.

- مثال اجرای Backtest با مدیریت ریسک:

```
results = backtest_bollinger_strategy_with_risk_management(df,
'BTC-USD', initial_capital=2000, risk_percentage=0.4)

print("\nبا مدیریت ریسک Backtest معیارهای ارزیابی:")
for metric, value in results.items():
    print(f"{metric}: {value:.2f}")
```

- نتایج مورد انتظار:
- Sharpe Ratio: نسبت بازدهی اضافی به ریسک مثبت.
- Sortino Ratio: نسبت بازدهی اضافی به ریسک نوسانات منفی مثبت.
- Max Drawdown: میزان بیشینه افت سرمایه.
- Net Profit: سود خالص مثبت یا منفی.

د. بررسی نمودار سیگنال‌های معامله  
با رسم نمودار سیگنال‌های خرید و فروش بر روی نمودار قیمت، به طور بصری عملکرد استراتژی را بررسی میکنیم.

- نمونه کد رسم نمودار:

```
plot_trade_signals(df, 'BTC-USD')
```

- توضیح:  
خطوط باندهای بولینگر باعث میشوند تا مشاهده نقاط سیگنال به شکل واضح‌تری انجام شود.  
فکرهای خرید (مثل مثلث‌های سبز) و فروش (مثلث‌های قرمز) نشان‌دهنده نقاط ورود و خروج هستند.  
**تغییر پارامترها و ضریب‌ها**

دوره محاسبه باندهای بولینگر (window): ما میتوانیم مقدار این پارامتر را تغییر میدهیم (مثلاً از ۲۰ به ۱۵ یا ۲۵) تا تاثیر آن بر سیگنال‌ها را مشاهده کنیم.

ضریب ضرب (multiplier):  
بسته به نیاز، این مقدار را تغییر میدهیم (مثلاً از ۲ به ۲.۵ یا ۱.۵) تا حساسیت سیگنال‌ها را تنظیم می‌دهیم.

مدیریت ریسک:  
در این نسخه از Backtest، تمام سرمایه در هر معامله سرمایه‌گذاری می‌شود. میتوانیم درصدی از سرمایه را برای هر معامله اختصاص دهیم (مثلاً ۴۰٪) تا ریسک معاملات را مدیریت کنیم. برای این کار، نیاز است که تابع Backtest را به گونه‌ای اصلاح کنیم که تنها درصدی از سرمایه را هر بار سرمایه‌گذاری کند.

بررسی نتایج:

پس از اجرای Backtest، خروجی شامل معیارهای ارزیابی (Total Return, Net Profit, Sharpe Ratio, Sortino Ratio, Max Drawdown) نمایش داده می‌شود.

نسبت شارپ و سورتینو: مقادیر مثبت نشان‌دهنده عملکرد بهتر است. هر چقدر این نسبت‌ها بالاتر باشند، استراتژی عملکرد بهتری دارد.

بیشینه افت (Max Drawdown): کمتر بودن این مقدار نشان‌دهنده ریسک کمتر است.  
سود خالص (Net Profit): مقدار مثبت نشان‌دهنده سود بوده و منفی نشان‌دهنده زیان است.

با تغییر این پارامترها می‌فهمیم که

با کاهش window، تعداد buy ها کاهش می‌یابد و روی هر دو رمز راز اثر منفی می‌گذارد.

بر روی بیت کوین فقط پوزیشن short باز میکند.

با ثابت ماندن windows و کاهش ضریب از دو به یک عملکرد eth ضعیف می‌شود (net profit, sharpe ratio)

sortino ratio و total return آن کاهش می‌یابد.)

اما بالعکس برای بیت‌کوین عملکرد بهتری به دست می‌آورد و پوزیشن های short بیشتری باز میکند که منجر به سوددهی در این تنظیمات پارامترهای خاص می‌شود.

با افزایش windows از 20 به 25 و همچنین کاهش multiplier برای keltner تغییر محسوسی در شاخص های ارزیابی سود و ضرر نمی‌بینیم

با افزایش windows به 25 و همچنین افزایش 25 درصدی ضرایب برای bollinger و keltner (از 2 به 2.5)

عملکرد بیت‌کوین افت میکند و درواقع هیچگاه وارد پوزیشن نمی‌شود اما عملکرد اتریوم بهبود بسیار چشمگیری می‌کند به

صورتی که sortino آن به 4.67 میرسد و sharpe آن به عدد 3.10 میرسد و همچنین سود خالص آن به 427 دلار یعنی 42.7 درصد میرسد.

اگر از روی شکل هم نگاه کنیم این پارامترها به خوبی ترند را پیش‌بینی کرده و در زمان مناسبی پوزیشن های buy و sell را وارد شده اند.

کاهش windows و window\_atr برای keltner موجب کاهش عملکرد هر دو رمزارز و شکل نگرفتن هیچ پوزیشنی در آن ها شود (شاید به دلیل ریسک پایین تر)

با رست کردن window به مقدار 15 و کاهش multiplier برای bollinger به 0.5 و برای keltner به 1.5 مشاهده

می‌کنیم که عملکرد bitcoin بسیار بهتر می‌شود و در سوی دیگر عملکرد اتریوم با کاهش چشمگیری همراه می‌شود.

با تغییر کانال های overbought و oversold برای RSI به سمت ریسک کمتر (20 و 80) متوجه می‌شویم که بر روی

عملکرد bitcoin تاثیری ندارد اما موجب بهبود عملکرد اتریوم می‌شود زیرا موجب افزایش سود (دو برابر) و افزایش چشمگیر

sortino و sharpe (از 0.77 به 1.36 برای sharpe و از 0.98 به 1.77 برای sortino) می‌شود اما

maxdrawdown را تغییر نمی‌دهد.

در الگوریتم دوم نیز مشخص می‌شود اگر با درصدی از سرمایه ی مان به عنوان حاشیه امنیت، وارد پوزیشن شویم با اینکه به

سود کمتری می‌رسیم اما maxdrawdown ما به صورت قابل توجهی کاهش می‌یابد و کمتر از نصف و حتی برای بیت‌کوین

یک سوم می‌شود، باقی شاخص های نیز بدون تغییر قابل توجهی باقی می‌مانند.

با دستکاری و تنظیم این پارامترها و مشاهده ی نمودار مربوط به buy و sell آن ها متوجه می‌شویم که میتوان پیش‌بینی این

الگوریتم درباره ی ترند بازار را بهبود داد تا جایی که با تنظیم برخی پارامترهای خاص به طور کامل روند نزولی و صعودی

بازار پیش‌بینی شده و وارد پوزیشن های مربوط به ترند پیش رو می‌شود و چیزی که حائز اهمیت است این است که رفتار این دو

رمز ارز در اکثر اوقات برعکس یک دیگر است پس نتیجه می‌گیریم برای هر بازاری نباید یک نسخه پیچید و با یک الگوریتم

هرچند که برای یک بازار جواب می‌دهد اما نباید در بقیه ی بازارها نیز تردید کرد.

خلاصه ی گزارش

در این گزارش، مفاهیم کلیدی مربوط به باندهای بولینگر، کلنتر و شاخص RSI توضیح داده شد و نحوه پیاده‌سازی الگوریتم معاملاتی بر اساس این اندیکاتورها با استفاده از تابع Backtest، بررسی شد. همچنین، معیارهای ارزیابی مانند Sharpe Ratio، Sortino Ratio، Max Drawdown و Net Profit به صورت کامل و دقیق توضیح داده شد و نحوه پیاده‌سازی آن‌ها و همچنین خود الگوریتم نیز مورد بررسی قرار گرفت.

با تنظیم پارامترهای مناسب برای داده‌های ۵ روزه و اعمال مدیریت ریسک، عملکرد ترید ما می‌تواند بهینه‌سازی شود. در نهایت، اجرای دقیق Backtest، تغییر پارامترها و بررسی نتایج کمک کرد تا از صحت عملکرد استراتژی معاملاتی خود اطمینان حاصل کنیم و در صورت نیاز، تنظیمات بیشتری برای بهبود آن اعمال نماییم.