

# Homework 1

Instruction: Write a report including the screenshots of your code and backtesting performance, and briefly introduce your strategy.

## Iteration 1: MACD + RSI in the Spot Market

Code:

<https://github.com/pointmekin/IntroToFintechAssignments-t11902203/blob/main/HW1/HW1.ipynb>

```
class Strategy(StrategyBase):
    def __init__(self):
        # strategy property
        self.subscribed_books = {}
        self.period = 60 * 60
        self.options = {}

        self.last_type = 'sell'
        self.fast_period = 10
        self.slow_period = 20
        self.signal_period = 9
        self.proportion = 0.9
        self.rsi_period = 10

    def on_order_state_change(self, order):
        pass

    # called every self.period
    def trade(self, candles):
        exchange, pair, base, quote = CA.get_exchange_pair()

        close_price_history = [candle['close'] for candle in candles[exchange][pair]]
        high_price_history = [candle['high'] for candle in candles[exchange][pair]]
        low_price_history = [candle['low'] for candle in candles[exchange][pair]]

        # convert to chronological order for talib
        close_price_history.reverse()
        high_price_history.reverse()
        low_price_history.reverse()

        # convert np.array
        close_price_history = np.array(close_price_history)
        high_price_history = np.array(high_price_history)
        low_price_history = np.array(low_price_history)

        if len(close_price_history) < 2:
            return []

        # get macd from talib
        macd, macdsignal, macdhist = talib.MACD(close_price_history, fastperiod=self.fast_period, slowperiod=self.slow_period,
        signalperiod=self.signal_period)

        curr_macd = macdhist[-1]
        prev_macd = macdhist[-2]

        macd_now = macd[-1]
        signal_now = macdsignal[-1]

        # get the simple moving average of the RSI value in the the specified period from talib
        all_rsi = talib.SMA(talib.RSI(close_price_history, self.rsi_period), self.rsi_period)

        curr_rsi = all_rsi[-1]
```

```
signal = 0
# MACD crosses the zero line from below - buy signal
# RSI falls into the oversold region
if (prev_macd < 0 and curr_macd > 0 and curr_rsi <= 30) :
    signal = 1

# MACD crosses the zero line from above - sell signal
# RSI rises into the overbought region
elif (prev_macd > 0 and curr_macd < 0 and curr_rsi >= 70):
    signal = -1

# get available balance
base_balance = CA.get_balance(exchange, base)
quote_balance = CA.get_balance(exchange, quote)
available_base_amount = base_balance.available
available_quote_amount = quote_balance.available

# place buy order
if self.last_type == 'sell' and signal == 1:
    amount = np.around((available_quote_amount / close_price_history[-1]) * self.proportion, 5)
    self.last_type = 'buy'
    CA.log("Buy " + str(amount) + " " + base)

    if (amount > 0):
        CA.buy(exchange, pair, amount, CA.OrderType.MARKET)

# place sell order
elif self.last_type == 'buy' and signal == -1:
    self.last_type = 'sell'

    if (available_base_amount > 0):
        CA.log("Sell " + str(available_base_amount) + " " + base)
        CA.sell(exchange, pair, available_base_amount, CA.OrderType.MARKET)

return
```

## Backtest Performance



## Strategy

### What is the strategy?

I have taken Crypto-Arena's MACD strategy template and made additional adjustments by utilizing the RSI values and their moving average in addition to the MACD to identify buying and selling signals.

### What's been updated?

Firstly, I have increased the period variable set in the Strategy class so that the closing price data will be pulled half as often, at 1 record per hour.

Next, I experimented with the `fast\_period` and `slow\_period` to find a reasonable rate for the fast MACD and slow MACD lines, which I found to be 10 and 20 respectively.

Next, I added the elements of RSI to the strategy. Using talib's RSI function, I collect the RSI values for the closing prices with the RSI period of 10 days, the same as the fast MACD line. Upon obtaining the RSI, I use talib's SMA function to find its simple moving average.

Finally, I edit the logic responsible for deciding whether the current tick should be a buy or a sell signal.

### **Justifications and Explanations for the Strategy**

Initially, when the default MACD strategy is selected, the ROI backtested from August 1 to August 31 of 2022 was negative. Due to the short investment period of 1 month, I figured that a shorter MACD period and RSI will be more reflective of the suitable strategy. This is achieved by reducing the fast and slow MACD period from 12 and 26 to 10 and 20 respectively.

While MACD is most helpful for determining the strength of the price movement rather than a trend direction, it is still respectable to help identify trading opportunities through the divergence signals it provides. The Moving average convergence divergence (MACD) tracks the moving averages of 2 different lengths, the shorter and the longer timeframe. Once the shorter timeframe MACD crosses the longer timeframe MACD, the scenario indicates a change in the momentum of the asset price. I have decided to utilize the MACD as the first indicator to decide when a trade should be considered.

Next, I use RSI, an indicator that provides signals about the bullish and bearish price movement momentum. RSI works by measuring how quickly and by how much an asset's price changes to identify whether the asset is overbought or oversold. An overbought condition, traditionally an RSI reading above 70, indicates that the asset price may be overvalued and might see a change in momentum in a downward direction soon. On the contrary, the oversold condition traditionally marked by RSI falling below 30, suggests that the asset is oversold. This suggests that it is perhaps a good buying opportunity as the price may soon change the momentum towards the upside in the near short term.

Moreover, once the RSI values are obtained, I put it into talib's SMA function to smooth out the RSI and find the moving average instead because individual RSI readings changes quite drastically. By finding the moving average, I can better identify the overall trend of the RSI rather than using the actual RSI value itself.

In my strategy, I used the MACD divergence to check whether a trade is worth considering. Once the MACD suggests a buying signal, I check the latest RSI SMA to identify whether the asset is still in the oversold region because it suggests that the prices may go up soon as more people will start buying the asset. Only when both conditions happen at the same time, do I proceed to buy the asset. On the other end, if the MACD suggests a selling signal, I verify with the RSI to check whether the asset is overbought, then proceed to sell the asset when both conditions were met.

Together MACD identifies a potential signal, and the RSI is used to confirm whether the signal might lead to a change in the price movements soon and make trades accordingly.

## The Results

The strategy is then backtested from August 1 to August 31 of 2022 to evaluate the performance. The strategy shows a reasonable positive return over 1 month at an ROI of 9.8% and a Sharpe ratio of 3.22.

## Iteration 2: MACD + RSI in the Futures Market

### Code

<https://github.com/pointmekin/IntroToFintechAssignments-t11902203/blob/main/HW1/HW1.ipynb>

```
class Strategy(StrategyBase):
    def __init__(self):
        # strategy property
        self.subscribed_books = {}
        self.period = 60 * 60
        self.options = {}

        self.last_type = 'sell'
        self.fast_period = 10
        self.slow_period = 20
        self.signal_period = 9
        self.proportion = 0.9
        self.rsi_period = 10

    def on_order_state_change(self, order):
        pass

    # called every self.period
    def trade(self, candles):
        exchange, pair, base, quote = CA.get_exchange_pair()

        close_price_history = [candle['close'] for candle in candles[exchange][pair]]
        high_price_history = [candle['high'] for candle in candles[exchange][pair]]
        low_price_history = [candle['low'] for candle in candles[exchange][pair]]

        # convert to chronological order for talib
        close_price_history.reverse()
        high_price_history.reverse()
        low_price_history.reverse()

        # convert np.array
        close_price_history = np.array(close_price_history)
        high_price_history = np.array(high_price_history)
        low_price_history = np.array(low_price_history)

        if len(close_price_history) < 2:
            return []

        # get macd from talib
        macd, macdsignal, macdhist = talib.MACD(close_price_history, fastperiod=self.fast_period, slowperiod=self.slow_period,
        signalperiod=self.signal_period)

        curr_macd = macdhist[-1]
        prev_macd = macdhist[-2]

        macd_now = macd[-1]
        signal_now = macdsignal[-1]

        # get the simple moving average of the RSI value in the specified period from talib
        all_rsi = talib.SMA(talib.RSI(close_price_history, self.rsi_period), self.rsi_period)

        # curr_rsi = all_rsi[-1]
        curr_rsi = all_rsi[-1]

        signal = 0
        # MACD crosses the zero line from below - buy signal
        # RSI falls into the oversold region
        if (prev_macd < 0 and curr_macd > 0 and curr_rsi <= 30) :
            signal = 1

        # MACD crosses the zero line from above - sell signal
        # RSI rises into the overbought region
        elif (prev_macd > 0 and curr_macd < 0 and curr_rsi >= 70):
            signal = -1

        # get available balance
        base_balance = CA.get_balance(exchange, base)
        quote_balance = CA.get_balance(exchange, quote)
        available_base_amount = base_balance.available
        available_quote_amount = quote_balance.available
```

```
# On buy signal
if self.last_type == 'sell' and signal == 1:
    amount = np.around((available_quote_amount / close_price_history[-1]) * self.proportion, 5)
    self.last_type = 'buy'

# Close all short positions
short_position = CA.get_position(exchange, pair, CA.PositionSide.SHORT)
if short_position and short_position.available_size > 0:
    CA.log('Close all short positions for ' + base)
    CA.close_short(exchange, pair, short_position.available_size, CA.OrderType.MARKET)

# Long the asset if available
if (amount > 0):
    # Long
    CA.log('Long ' + str(amount) + ' ' + base)
    CA.open_long(exchange, pair, amount, CA.OrderType.MARKET)

# On sell signal
elif self.last_type == 'buy' and signal == -1:
    amount = np.around((available_quote_amount / close_price_history[-1]) * self.proportion, 5)
    self.last_type = 'sell'

# Close all long positions
long_position = CA.get_position(exchange, pair, CA.PositionSide.LONG)
if long_position and long_position.available_size > 0:
    CA.log('Close all long positions for ' + base)
    CA.close_long(exchange, pair, long_position.available_size, CA.OrderType.MARKET)

# Short the asset if available
if (amount > 0):
    CA.log('Short ' + str(amount) + ' ' + base)
    CA.open_short(exchange, pair, amount, CA.OrderType.MARKET)

return
```

## Backtest Performance



## Strategy

### What is the strategy?

Upon competing in the backtesting process of my first iteration of the strategy in the spot market, I noticed that in August of 2022, there are both upsides and downsides to the ETH price. As a result, I want to try to make profits both during the upward price movement and the downwards one.

### What's been updated?

To do this, I moved to the USD<sup>Ⓢ</sup>-M Futures (Perp.) futures market. I further modify my code from buying and selling to opening long and short positions instead.

Using the same MACD and RSI simple moving average as the indicators, each time my strategy suggests a buying signal in the oversold region, I will close all my current short positions and open a long position at the market price if there are available asset remaining. Vice versa, each time the strategy suggests a selling signal in the overbought region, I will close all my long positions and open a short position at the market price.

### **Justifications and Explanations for the Strategy**

In the futures market, I am now theoretically able to make profits regardless of whether the price of ETH is moving upward (through long positions) or downward (through short positions).

### **The Results**

Finally, with the strategy applied in the futures market, I was able to obtain an ROI of 12.72%, which is a 2.92% increase from applying the same strategy to the spot market.