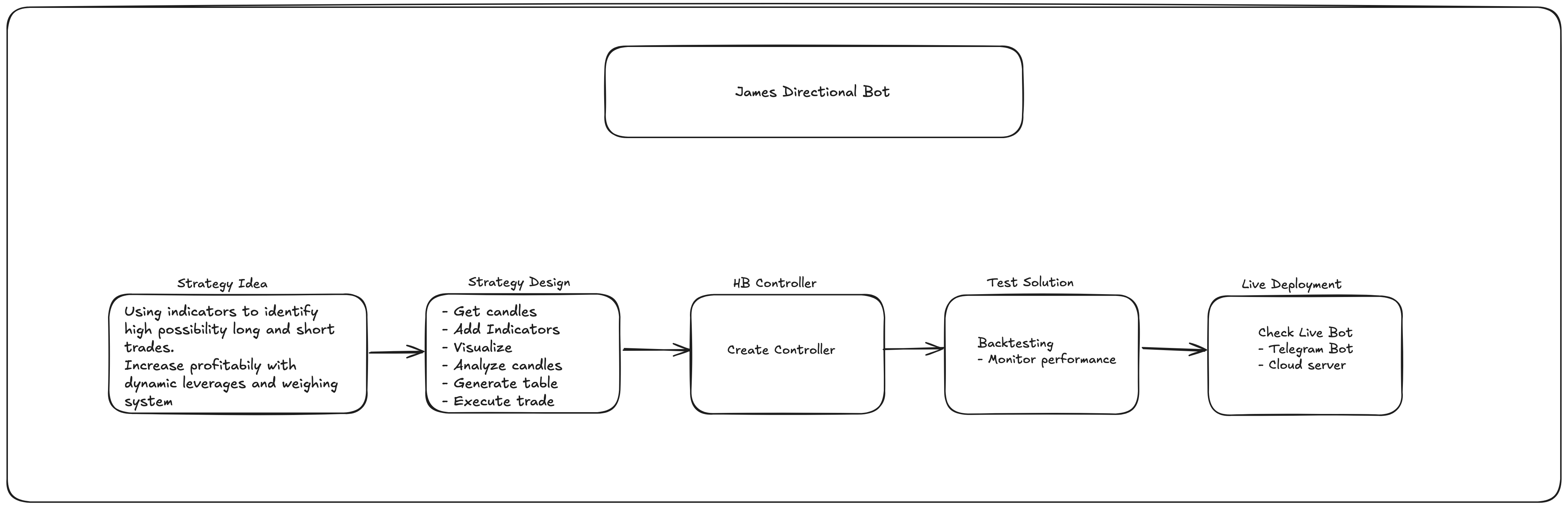
Directional Trading Bot:

Here is my detailed trade strategy in plain English.



Directional Bot Requirements:

[A] Screener: Continuously screen all tokens on an exchange (e.g. Binance\_perpetual), printout tokens that meet my trade condition in the order of priority and execute trade.

[B] Trade Interval: 15m, with a 15m cooldown time.

[C] Exchange: Binance\_perpetual

[D] Trading Capital: $100 | Market price.

[E] Indicators:

MACD: (Fast Length=12, Slow Length=26, Signal Smoothing=9, Oscillator & Signal Line MA Type=EMA),

RSI: (RSI Length=14, Source=close, MA type=SMA, MA Length=14, BB StdDev=2),

Ichimoku: default

Momentum: (Length=10)

Stochastic(%K Length = 5, %K Smoothing = 2, %D Smoothing = 3), Moving Averages (Simple and Exponential).

Volume: default

**[F] My Strategy for going “Long”, i.e. buying low and selling high is as stated below:**

**BUY: LONG**

1. Only buy an asset when all the following Simple Moving Averages (SMA or MA) and Exponential Moving Averages (EMA) are angling up, meaning that the current value is higher than previous value: 3MA, 3EMA, 6MA, 9MA, 13MA, 15EMA, 15MA, Volume (Vol), LSA (Ichimoku: Leading Span A), CL (Ichimoku: Conversion Line), and 20MA OR 33MA is angling up [i.e., 3MA + 3EMA + 6MA + 9MA + 13MA + 15EMA + 15MA + Vol + LSA + CL + 20MA or 33MA = Angling up] —> Condition 1A (Required) AND

Code sample:

# Condition 1A: All MAs angling up

condition\_1a = all([

indicators['ma3'][-1] > indicators['ma3'][-2], # 3MA up

indicators['ema3'][-1] > indicators['ema3'][-2], # 3EMA up

indicators['ma6'][-1] > indicators['ma6'][-2], # 6MA up

indicators['ma9'][-1] > indicators['ma9'][-2], # 9MA up

indicators['ma13'][-1] > indicators['ma13'][-2], # 13MA up

indicators['ema15'][-1] > indicators['ema15'][-2], # 15EMA up

indicators['ma15'][-1] > indicators['ma15'][-2], # 15MA up

indicators['volume'][-1] > indicators['volume'][-2],# Volume up

indicators['leading\_span\_a'][-1] > indicators['leading\_span\_a'][-2], # LSA up

indicators['conversion\_line'][-1] > indicators['conversion\_line'][-2] # CL up

]) and (

indicators['ma20'][-2] < indicators['ma20'][-1] or indicators['ma33'][-2] < indicators['ma33'][-1] # 20MA or 33MA up

)

2. Buy if Price is above Ichimoku Conversion Line (CL) AND CL is above Ichimoku Base Line (BL). —> Condition 1B (Required) AND

Code sample:

# Condition 1B: Price and CL conditions

condition\_1b = (

indicators['close'][-1] > indicators['conversion\_line'][-1] and # Price above CL

indicators['conversion\_line'][-1] > indicators['base\_line'][-1] # CL above BL

)

3. Buy if MACD has crossed above Signal. Also, MACD + %K (Stochastic) + RSI + Momentum (MoM) are all angling up. And RSI is above RSI-based MA. And %K > %D —> Condition 1C (Required) AND

Code sample:

# Condition 1C: MACD and other indicators

condition\_1c = (

indicators['macd'][-1] > indicators['macd\_signal'][-1] and # MACD crossed above Signal

all([

indicators['macd'][-1] > indicators['macd'][-2], # MACD up

indicators['stoch\_k'][-1] > indicators['stoch\_k'][-2], # %K up

indicators['rsi'][-1] > indicators['rsi'][-2], # RSI up

indicators['momentum'][-1] > indicators['momentum'][-2] # Momentum up

]) and

indicators['rsi'][-1] > indicators['rsi\_ma'][-1] # RSI above RSI-MA

)

4. Only Buy if 6MA is less than 3MA and 3EMA. --> Condition 2A (Required)

Code sample:

# Condition 2A: 6MA conditions

condition\_2a = (

indicators['ma6'][-1] < indicators['ma3'][-1] and # 6MA less than 3MA

indicators['ma6'][-1] < indicators['ema3'][-1] # 6MA less than 3EMA

)

if all([condition\_1a, condition\_1b, condition\_1c, condition\_2a]):

leverage = self.calculate\_long\_leverage(indicators)

return True, leverage

return False, 0

5. If the current asset price is higher than the value of 9MA, and 9MA is higher than 15EMA, and 15EMA is higher than 20MA, and 20MA is higher than 33MA. I call this “Partial Energy Alignment”. —> Condition 3A (Informative but required for calculating leverage)

Code sample:

def check\_partial\_energy\_alignment(self, indicators: Dict) -> bool:

"""Check for partial energy alignment (Condition 3A)"""

return (

indicators['close'][-1] > indicators['ma9'][-1] >

indicators['ema15'][-1] > indicators['ma20'][-1] >

indicators['ma33'][-1]

)

6. If the current asset price is higher than the value of 9MA, and 9MA is higher than 15EMA, and 15EMA is higher than 20MA, and 20MA is higher than 33MA, and 33MA is higher than 50MA. I call this “Complete Energy Alignment". —> Condition 3B (Informative but required for calculating leverage).

Code sample:

def check\_complete\_energy\_alignment(self, indicators: Dict) -> bool:

"""Check for complete energy alignment (Condition 3B)"""

return (

indicators['close'][-1] > indicators['ma9'][-1] >

indicators['ema15'][-1] > indicators['ma20'][-1] >

indicators['ma33'][-1] > indicators['ma50'][-1]

)

7. Long Leverage formula: My leverage formula suggests monitoring the different moving averages to ascertain that they are aligned according to their order of magnitude, from the least to the highest, e.g., 3MA is above or is higher than 9MA. While 9MA is higher than 20MA. 20MA is also higher than 33MA, etc.

1. 10x Leverage: Apply a 10x leverage when:

• All required conditions have been met.

• 33MA is angling up.

• CL is greater than 9MA

• 100MA is either flat or angled up

• Current volume (Vcurrent) > Previous volume (Vprevious) by up to 1.4x or more. For example, If Vcurrent is 1,000 and Vprevious is 700, apply a 10x leverage because 1,000 is greater than 700 by more than 1.4 times.

2. 20x Leverage: Apply a 20x leverage when:

• All required conditions have been met.

• 33MA and 100MA are angling up.

• CL is greater than 9MA

• 200MA is either flat or angled up

• Current volume (Vcurrent) > Previous volume (Vprevious) by up to 2.4x or more. For example, If Vcurrent is 2,000 and Vprevious is 700, apply a 20x leverage because 2,000 is greater than 700 by more than 1.4 times.

• There is “Partial Energy Alignment” (i.e., Condition 3A is met)

3. 30x Leverage: Apply a 30x leverage when:

• All required conditions have been met.

• 33MA and 100MA are angling up.

• CL is greater than 9MA

• 200MA is either flat or angled up

• Current volume (Vcurrent) > Previous volume (Vprevious) by up to 5x or more. For example, If Vcurrent is 4,000 and Vprevious is 700, apply a 30x leverage because 4,000 is greater than 700 by more than 5 times.

• There is “Complete Energy Alignment” (i.e., Condition 3B is met).

If the current price is higher than the value of 9MA, and 9MA is higher than 15EMA, and 15EMA is higher than 20MA, and 20MA is higher than 33MA, and 33MA is higher than 50MA. I call this “Complete Energy Alignment”. --> Condition 3B.

4. 5x Leverage: Apply a 5x leverage for all other trades

8. Please note that all conditions flagged as “Required”, must always be met.

Code sample:

def calculate\_long\_leverage(self, indicators: Dict) -> int:

"""Calculate leverage for long positions"""

volume\_ratio = indicators['volume'][-1] / indicators['volume'][-2]

# Check for 30x leverage conditions

if all([

indicators['ma33'][-1] > indicators['ma33'][-2], # 33MA up

indicators['ma100'][-1] > indicators['ma100'][-2], # 100MA up

indicators['conversion\_line'][-1] > indicators['ma9'][-1], # CL > 9MA

indicators['ma200'][-1] >= indicators['ma200'][-2], # 200MA flat or up

volume\_ratio >= 5, # Volume 5x

self.check\_complete\_energy\_alignment(indicators) # Complete alignment

]):

return 30

# Check for 20x leverage conditions

if all([

indicators['ma33'][-1] > indicators['ma33'][-2], # 33MA up

indicators['ma100'][-1] > indicators['ma100'][-2], # 100MA up

indicators['conversion\_line'][-1] > indicators['ma9'][-1], # CL > 9MA

indicators['ma200'][-1] >= indicators['ma200'][-2], # 200MA flat or up

volume\_ratio >= 2.4, # Volume 2.4x

self.check\_partial\_energy\_alignment(indicators) # Partial alignment

]):

return 20

# Check for 10x leverage conditions

if all([

indicators['ma33'][-1] > indicators['ma33'][-2], # 33MA up

indicators['conversion\_line'][-1] > indicators['ma9'][-1], # CL > 9MA

indicators['ma100'][-1] >= indicators['ma100'][-2], # 100MA flat or up

volume\_ratio >= 1.4 # Volume 1.4x

]):

return 10

return 5

**SELL: LONG**

1. Sell when 3MA or 3EMA + Stochastic (%K) + RSI + Momentum (MoM) + MACD are all angling down, with %D above or greater than %K. —> Condition 1 (Required). AND any of the following criteria below (i.e. from 2–8)

Code sample:

def check\_long\_exit\_conditions(self, indicators: Dict) -> bool:

"""Check conditions for exiting long positions"""

# Base condition (Required)

base\_condition = (

(indicators['ma3'][-1] < indicators['ma3'][-2] or

indicators['ema3'][-1] < indicators['ema3'][-2]) and

indicators['stoch\_k'][-1] < indicators['stoch\_k'][-2] and

indicators['rsi'][-1] < indicators['rsi'][-2] and

indicators['momentum'][-1] < indicators['momentum'][-2] and

indicators['macd'][-1] < indicators['macd'][-2] and

indicators['stoch\_d'][-1] > indicators['stoch\_k'][-1]

)

if not base\_condition:

return False

2. 6MA closes above either 3MA or 3EMA, and both 3MA & 3EMA close angling down (i.e. 6MA closes above 3MA or 3EMA and 3MA + 3EMA angles down).

3. Price closes below CL.

4. RSI dips (i.e., angles down and closes) below RSI-based MA + both 3MA & 3EMA angles down.

5. Stochastic (i.e., both %K + %D) + 3MA + 3EMA angles down. My Stochastic input parameters are: %K Length = 5, %K Smoothing = 2, %D Smoothing = 3

6. 5EMA OR CL OR RSI-based MA closes angling down. AND, 3MA + 3EMA + %K + %D = angling down.

7. When 3EMA closes lower at a lower price, 2 consecutive times in a row.

8. Sell when MACD, 3MA and 3EMA are all angling down.

Code sample:

# Additional exit conditions (2-8)

additional\_conditions = [

# Condition 2: 6MA crosses

((indicators['ma6'][-1] > indicators['ma3'][-1] and

(indicators['ma3'][-1] < indicators['ma3'][-2]) and indicators['ema3'][-1] < indicators['ema3'][-2]) or

(indicators['ma6'][-1] > indicators['ema3'][-1] and

(indicators['ma3'][-1] < indicators['ma3'][-2]) and indicators['ema3'][-1] < indicators['ema3'][-2]))),

# Condition 3: Price below CL

indicators['close'][-1] < indicators['conversion\_line'][-1],

# Condition 4: RSI conditions

(indicators['rsi'][-1] < indicators['rsi\_ma'][-1] and

indicators['ma3'][-1] < indicators['ma3'][-2] and

indicators['ema3'][-1] < indicators['ema3'][-2]),

# Condition 5: Stochastic and MA conditions

(indicators['stoch\_k'][-1] < indicators['stoch\_k'][-2] and

indicators['stoch\_d'][-1] < indicators['stoch\_d'][-2] and

indicators['ma3'][-1] < indicators['ma3'][-2] and

indicators['ema3'][-1] < indicators['ema3'][-2]),

# Condition 6: Combined conditions

((indicators['ema5'][-1] < indicators['ema5'][-2] or

indicators['conversion\_line'][-1] < indicators['conversion\_line'][-2] or

indicators['rsi\_ma'][-1] < indicators['rsi\_ma'][-2]) and

indicators['ma3'][-1] < indicators['ma3'][-2] and

indicators['ema3'][-1] < indicators['ema3'][-2] and

indicators['stoch\_k'][-1] < indicators['stoch\_k'][-2] and

indicators['stoch\_d'][-1] < indicators['stoch\_d'][-2]),

# Condition 7: Consecutive 3EMA downs

(indicators['ema3'][-1] < indicators['ema3'][-2] and

indicators['ema3'][-2] < indicators['ema3'][-3]),

# Condition 8: MACD and MA conditions

(indicators['macd'][-1] < indicators['macd'][-2] and

indicators['ma3'][-1] < indicators['ma3'][-2] and

indicators['ema3'][-1] < indicators['ema3'][-2])

]

# Return True if base condition AND any additional condition is met

return base\_condition and any(additional\_conditions)

**[G] My strategy for going “Short”, i.e., buying high and selling low:**

**ENTRY/SELL:**

1. Enter short (i.e., Sell) if :

a. the price for an asset closes below CL (i.e., P < CL). CL < BL. AND 6MA has closed above 3MA & 3 EMA, with both 3MA & 3EMA angling down AND

b. the following moving averages are angling down: 20MA + 15MA + 15EMA + 13MA + 9MA + 6MA + CL + LSA. AND

c. %K + %D + RSI + MACD + Momentum (MOM) = Angling down. AND RSI is less than (or below) RSI-based MA —> Condition X (All the above are required)

Code sample:

def check\_short\_conditions(self, indicators: Dict) -> Tuple[bool, int]:

"""Check conditions for short entry (Condition X)"""

# Part A: Price and MA conditions

condition\_x\_a = all([

indicators['close'][-1] < indicators['conversion\_line'][-1], # Price below CL

indicators['conversion\_line'][-1] < indicators['base\_line'][-1], # CL below BL

indicators['ma6'][-1] > indicators['ma3'][-1], # 6MA above 3MA

indicators['ma6'][-1] > indicators['ema3'][-1], # 6MA above 3EMA

indicators['ma3'][-1] < indicators['ma3'][-2], # 3MA angling down

indicators['ema3'][-1] < indicators['ema3'][-2] # 3EMA angling down

])

# Part B: Moving averages angling down

condition\_x\_b = all([

indicators[ma][-1] < indicators[ma][-2] for ma in [

'ma20', 'ma15', 'ema15', 'ma13', 'ma9', 'ma6', 'conversion\_line', 'leading\_span\_a'

]

])

# Part C: Technical indicators angling down

condition\_x\_c = all([

indicators['stoch\_k'][-1] < indicators['stoch\_k'][-2], # %K down

indicators['stoch\_d'][-1] < indicators['stoch\_d'][-2], # %D down

indicators['rsi'][-1] < indicators['rsi'][-2], # RSI down

indicators['rsi'][-1] < indicators['rsi\_ma'][-1] # RSI < RSI-based MA

indicators['macd'][-1] < indicators['macd'][-2], # MACD down

indicators['momentum'][-1] < indicators['momentum'][-2] # Momentum down

])

if all([condition\_x\_a, condition\_x\_b, condition\_x\_c]):

leverage = self.calculate\_short\_leverage(indicators)

return True, leverage

return False, 0

4. Short Leverage formula:

1. 10x Leverage: Apply a 10x leverage when:

• All required conditions have been met

• RSI-based MA is angling down.

• Current volume (Vcurrent) > Previous volume (Vprevious) by up to 1.4x or more. E.g., If Vcurrent is 1,000 and Vprevious is 700, apply a 10x leverage because 1,000 is greater than 700 by more than 1.4 times.

2. 20x Leverage: Apply a 20x leverage when:

• All required conditions have been met

• RSI-based MA is angling down.

• 33MA + 100MA + 200MA = angling down

• Price drops or closes below 20MA, and 20MA < 100MA, and 100MA < 200MA.

• Current volume (Vcurrent) > Previous volume (Vprevious) by up to 2.4x or more. For example, If Vcurrent is 2,000 and Vprevious is 700, apply a 10x leverage because 2,000 is greater than 700 by more than 2.4 times.

3. 30x Leverage: Apply a 30x leverage when:

• All required conditions have been met

• RSI-based MA is angling down

• 33MA + 100MA + 200MA = angling down

• Price drops or closes below 20MA, and 20MA < 100MA, and 100MA < 200MA.

• BL + LSB = angling down

• Current volume (Vcurrent) > Previous volume (Vprevious) by up to 5x or more. E.g., If Vcurrent is 4,000 and Vprevious is 700, apply a 30x leverage because 4,000 is greater than 700 by more than 5 times.

4. 5x Leverage: Apply a 5x leverage for all other trades

Code sample:

def calculate\_short\_leverage(self, indicators: Dict) -> int:

"""Calculate leverage for short positions"""

volume\_ratio = indicators['volume'][-1] / indicators['volume'][-2]

# 30x leverage conditions

if all([

indicators['conversion\_line'][-1] < indicators['conversion\_line'][-2], # CL down

indicators['stoch\_d'][-1] < indicators['stoch\_d'][-2], # %D down

indicators['ma100'][-1] < indicators['ma100'][-2], # 100MA down

indicators['ma200'][-1] < indicators['ma200'][-2], # 200MA down

indicators['close'][-1] < indicators['ma20'][-1], # Price below 20MA

indicators['ma20'][-1] < indicators['ma100'][-1], # 20MA < 100MA

indicators['ma100'][-1] < indicators['ma200'][-1], # 100MA < 200MA

indicators['base\_line'][-1] < indicators['base\_line'][-2], # BL down

indicators['leading\_span\_b'][-1] < indicators['leading\_span\_b'][-2], # LSB down

volume\_ratio >= 5 # Volume 5x

]):

return 30

# 20x leverage conditions

if all([

indicators['conversion\_line'][-1] < indicators['conversion\_line'][-2], # CL down

indicators['stoch\_d'][-1] < indicators['stoch\_d'][-2], # %D down

indicators['ma100'][-1] < indicators['ma100'][-2], # 100MA down

indicators['ma200'][-1] < indicators['ma200'][-2], # 200MA down

indicators['close'][-1] < indicators['ma20'][-1], # Price below 20MA

indicators['ma20'][-1] < indicators['ma100'][-1], # 20MA < 100MA

indicators['ma100'][-1] < indicators['ma200'][-1], # 100MA < 200MA

volume\_ratio >= 2.4 # Volume 2.4x

]):

return 20

# 10x leverage conditions

if all([

indicators['conversion\_line'][-1] < indicators['conversion\_line'][-2], # CL down

indicators['stoch\_d'][-1] < indicators['stoch\_d'][-2], # %D down

volume\_ratio >= 1.4 # Volume 1.4x

]):

return 10

return 5

**EXIT/BUY:**

In the order below:

1. Exit or close a Short when 5EMA + 3MA + 3EMA + RSI + Momentum (MoM) all closed angling up. —> Condition A, OR

2. Exit when Price closes above CL, with 3MA + 3EMA angling up —> Condition B, OR

3. Exit when 6MA + 3MA + 3EMA closes angling up. —> Condition C OR

4. Exit when MACD + RSI closes angling up —> Condition D (New)

5. Price closes with Candle High < 3MA & 3EMA. With both %K & %D < 30. And, Momentum (Mom) angles up. (New)

Code sample:

def check\_short\_exit\_conditions(self, indicators: Dict) -> bool:

"""Check conditions for exiting short positions"""

# Condition A

condition\_a = all([

indicators['ema5'][-1] > indicators['ema5'][-2], # 5EMA up

indicators['ma3'][-1] > indicators['ma3'][-2], # 3MA up

indicators['ema3'][-1] > indicators['ema3'][-2], # 3EMA up

indicators['rsi'][-1] > indicators['rsi'][-2], # RSI up

indicators['momentum'][-1] > indicators['momentum'][-2] # Momentum up

])

# Condition B

condition\_b = (

indicators['close'][-1] > indicators['conversion\_line'][-1] and # Price above CL

indicators['ma3'][-1] > indicators['ma3'][-2] and # 3MA up

indicators['ema3'][-1] > indicators['ema3'][-2] # 3EMA up

)

# Condition C

condition\_c = all([

indicators['ma6'][-1] > indicators['ma6'][-2], # 6MA up

indicators['ma3'][-1] > indicators['ma3'][-2], # 3MA up

indicators['ema3'][-1] > indicators['ema3'][-2] # 3EMA up

])

# Exit if any condition is met

return condition\_a or condition\_b or condition\_c

[H] TRADE PRIORITY:

1. Highest Priority (P1): Trades that qualify for a 30x leverage should be assigned the highest priority

2. Medium Priority (P2): Trades that qualify for a 20x leverage should be assigned the medium priority.

3. Low Priority (P3): Trades that qualify for a 10x leverage should be assigned the medium priority.

4. Minimal Priority (P4): Trades that qualify for a 5x leverage should be assigned the lowest priority.

4. In other words, if only one trade can be taken, then that slot should be given to a P1 trade. If no trade or asset meets the P1 criteria, then the next $100 should be used for a P2 trade; otherwise, a P3 trade can be entered, or else a P4 trade is taken. If two $100 trades can be entered, two P1 trades would be the preference; otherwise, one P1 and one P2 trade is entered, as the case may be.

5. Long trades are prioritized over Short trades by their priorities. Meaning if there is no P1 Long trade, then I will take a P1 Short trade before considering P2 trades.

Code sample:

@dataclass

class TradeOpportunity:

connector: str

trading\_pair: str

direction: str

leverage: int

priority: int

@property

def priority\_score(self) -> int:

direction\_score = 0 if self.direction == "LONG" else 1

return self.priority \* 10 + direction\_score

def get\_leverage\_priority(self, leverage: int) -> int:

"""Convert leverage to priority level"""

if leverage >= 30:

return 1 # P1

elif leverage >= 20:

return 2 # P2

elif leverage >= 10:

return 3 # P3

return 4 # P4

Tasks:

1. Develop a reliable and stable directional bot based on above conditions i.e., continuously screen all tokens on an exchange (e.g., Binance\_perpetual), printout tokens that meet trade conditions (either long or short) in the order of priority and execute trade.

2. Review the code to ensure connectors and executors meet the acceptance criteria, reporting and other requirements.

3. Perform automated and manual testing.

4. Work with client to optimize and host bot to a cloud server. Monitor bot performance over a period of time.

5. Enable Telegram reporting and control.