import pandas as pd

import numpy as np

# --- Indicators ---

def ema(series, length):

return series.ewm(span=length, adjust=False).mean()

def atr(df, length=14):

high\_low = df['high'] - df['low']

high\_close = np.abs(df['high'] - df['close'].shift())

low\_close = np.abs(df['low'] - df['close'].shift())

tr = pd.concat([high\_low, high\_close, low\_close], axis=1).max(axis=1)

return tr.rolling(length).mean()

def macd\_histogram(close, fast=12, slow=26, signal=9):

ema\_fast = ema(close, fast)

ema\_slow = ema(close, slow)

macd = ema\_fast - ema\_slow

signal\_line = ema(macd, signal)

return macd - signal\_line

# --- Load your OHLCV data ---

# df = pd.read\_csv('your\_data.csv') # should have columns: open,high,low,close,volume

# --- Indicators ---

df['ema5'] = ema(df['close'], 5)

df['ema20'] = ema(df['close'], 20)

df['atr'] = atr(df)

df['atr\_ratio'] = (df['atr'] / df['close']) / ((df['atr'] / df['close']).rolling(20).mean())

df['macd\_hist'] = macd\_histogram(df['close'])

df['volume\_ratio'] = df['volume'] / df['volume'].rolling(20).mean()

df['ema\_diff\_pct'] = (df['ema5'] - df['ema20']) / df['close'] \* 100

# --- Multipliers ---

def multiplier\_ema(diff):

if diff > 0.2: return 1.3

if diff < -0.2: return 0.7

return 1.0

def multiplier\_volume(vr):

if vr > 1.5: return 1.2

if vr < 0.8: return 0.8

return 1.0

def multiplier\_macd(hist):

if hist > 0: return 1.2

if hist < 0: return 0.8

return 1.0

def multiplier\_volatility(ratio):

if ratio < 0.8: return 1.1

if ratio > 1.2: return 0.9

return 1.0

# --- Compute Probability and Decision ---

results = []

for idx, row in df.iterrows():

emaMult = multiplier\_ema(row['ema\_diff\_pct'])

volMult = multiplier\_volume(row['volume\_ratio'])

macdMult = multiplier\_macd(row['macd\_hist'])

volatMult = multiplier\_volatility(row['atr\_ratio'])

odds = emaMult \* volMult \* macdMult \* volatMult

p\_up = odds / (1 + odds)

# Decision with sizing

if p\_up >= 0.75:

decision = "Long Full"

elif p\_up >= 0.65:

decision = "Long Half"

elif p\_up <= 0.25:

decision = "Short Full"

elif p\_up <= 0.35:

decision = "Short Half"

else:

decision = "No Trade"

results.append((p\_up, decision))

df['p\_up'], df['decision'] = zip(\*results)

# --- Show last rows ---

print(df[['close','ema5','ema20','p\_up','decision']].tail())