import pandas as pd

import numpy as np

from sklearn.linear\_model import LogisticRegression

# --- Generate hypothetical OHLCV data (for example purposes only) ---

np.random.seed(42)

n = 200

df = pd.DataFrame({

'close': np.cumsum(np.random.randn(n)) + 100,

'high': np.nan,

'low': np.nan,

'open': np.nan,

'volume': np.random.randint(100, 1000, n)

})

df['open'] = df['close'].shift(1).fillna(df['close'])

df['high'] = df[['open','close']].max(axis=1) + np.random.rand(n)

df['low'] = df[['open','close']].min(axis=1) - np.random.rand(n)

# --- Features (from previous logic) ---

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

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

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

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

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

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

atr\_val = atr(df)

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

# --- Target: 1 if next close is higher ---

df['y'] = (df['close'].shift(-1) > df['close']).astype(int)

df = df.dropna()

# --- Prepare data for logistic regression ---

X = df[['ema\_diff\_pct', 'volume\_ratio', 'macd\_hist', 'atr\_ratio']].fillna(0)

y = df['y']

# --- Fit logistic regression ---

model = LogisticRegression()

model.fit(X, y)

# --- Coefficients ---

intercept = model.intercept\_[0]

coeffs = model.coef\_[0]

print("Intercept:", intercept)

print("Coefficients:", coeffs)

# --- Predict P(Up) using logistic regression ---

df['p\_up'] = model.predict\_proba(X)[:, 1]

# --- Simple Decision Rule based on learned weights ---

df['decision'] = np.where(df['p\_up'] >= 0.75, "Long Full",

np.where(df['p\_up'] >= 0.65, "Long Half",

np.where(df['p\_up'] <= 0.25, "Short Full",

np.where(df['p\_up'] <= 0.35, "Short Half", "No Trade"))))

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