



Instrucciones de uso

Introducción

BackTest Wizard es una aplicación web capaz de evaluar estrategias de trading de criptomonedas, utilizando datos del pasado. Esta app simula años de trading spot en pocos segundos y elabora un informe con el resumen de los resultados.

Estrategia

En los renglones “**Estrategia de Compra**” y “**Estrategia de Venta**” deben ingresarse las condiciones de compra y venta respectivamente. Estas condiciones deben ser proposiciones booleanas, es decir, se le debe poder asignar un valor de verdad; **verdadero** o **falso**. Sólo se admiten caracteres proporcionados por el teclado del interfaz. Estos campos son obligatorios, por lo que no pueden quedar vacíos a la hora de ejecutar el backtesting.

Par de Criptomonedas

En este campo debe ingresarse el par de criptomonedas que se desea utilizar para la ejecución del backtesting. Sólo están disponibles aquellos pares presentes en Binance y cuyo historial está disponible en el API de Binance. Este campo es obligatorio, por lo que no puede quedar vacío a la hora de ejecutar el backtesting.

Temporalidad

Este campo corresponde a la temporalidad de las velas japonesas que se utilizarán para realizar el test. Mientras menor sea la temporalidad, mayor será el tiempo de ejecución. Algunas temporalidades como la de “1 minuto”, pueden demorar incluso varios minutos en ejecutarse. Este campo es obligatorio, por lo que no puede quedar vacío a la hora de ejecutar el backtesting.

Sistema de Precios

Cuando se realiza un backtesting, es importante recordar que sólo se cuenta con 4 datos por cada vela japonesa (High, Low, Open y Close). Por lo que se desconoce la evolución del precio durante el transcurso del intervalo de tiempo correspondiente a cada vela.

Por todo lo mencionado en el anterior párrafo, es que se necesita definir un precio de referencia con el que se evaluarán las estrategias y se ejecutarán las operaciones de compra y venta.

El sistema más utilizado en el mundo del backtesting es el “**Close**”, sin embargo la plataforma recomienda “**Random**” y “**Mean**” por ser más neutrales. Mientras que “**Worst**” y “**Best**” pueden servir para evaluar el peor y el mejor escenario de una estrategia.

A continuación se explicará con mayor detalle cada sistema:

- **Close:**
La evaluación de la estrategia y las operaciones de compraventa, usan como precio de referencia el de cierre de cada vela. Es el sistema de precios más utilizado en el mundo del backtesting.
- **Open:**
La evaluación de la estrategia y las operaciones de compraventa, usan como precio de referencia el de apertura de cada vela.
- **High:**
La evaluación de la estrategia y las operaciones de compraventa usan como precio de referencia el más alto de cada vela.
- **Low:**
La evaluación de la estrategia y las operaciones de compraventa usan como precio de referencia el más bajo de cada vela.
- **Random:**
La evaluación de la estrategia y las operaciones de compraventa usan como precio de referencia algún precio aleatorio dentro del rango de cada vela. Es aquella con menos sesgo. Cada ejecución puede producir resultados diferentes por la aleatoriedad introducida en la evaluación, por lo que permite observar varios escenarios.
- **Mean:**
La evaluación de la estrategia y las operaciones de compraventa usan como precio de referencia al promedio entre el High y el Low de cada vela.
- **Best:**
La evaluación de la estrategia de compra y las operaciones de compra usan como precio de referencia el más bajo de cada vela.
La evaluación de la estrategia de venta y las operaciones de venta usan como precio de referencia el más alto de cada vela.
Con este sistema se busca aproximar el mejor escenario posible para una estrategia dada. Si una estrategia es deficiente con este sistema, más lo será con el resto. Por lo que esta configuración nos permite filtrar pésimas estrategias.

- **Worst:**

La evaluación de la estrategia de compra y las operaciones de compra usan como precio de referencia el más alto de cada vela.

La evaluación de la estrategia de venta y las operaciones de venta usan como precio de referencia el más bajo de cada vela.

Con este sistema se busca aproximar el peor escenario posible para una estrategia dada. Si una estrategia es positiva con este sistema, más lo será con el resto. Por lo que esta configuración nos permite encontrar aquellas estrategias que son más robustas.

Se recomienda además, poner a prueba las estrategia usando diferentes sistemas de precios para tener varias perspectivas de la misma.

Este campo es obligatorio, por lo que no puede quedar vacío a la hora de ejecutar el backtesting.

Stop Loss

Es un número porcentual (entre 0 y 100) que representa la tolerancia a las caídas del precio por debajo del precio de compra. Si a este porcentaje lo llamamos "N", entonces, si el precio cae un N% por debajo del precio de compra, se dispara una señal de venta. Los valores más utilizados para este campo rondan entre el 0% y el 10%. En caso de dejar vacío este campo, se interpreta que la estrategia no utiliza stop loss.

Trailing Stop

Es un número porcentual (entre 0 y 100) que representa un seguimiento por debajo del precio actual. Si a este porcentaje lo llamamos "N", entonces, si el precio sube por encima del precio de compra, y luego cae un N% por debajo del precio máximo alcanzado, se dispara una señal de venta. Esta orden de venta sólo se ejecuta si con esta operación se obtiene ganancia. Los valores más utilizados para este campo rondan entre el 0% y el 10%. En caso de dejar vacío este campo, se interpreta que la estrategia no utiliza Trailing Stop.

Funciones Max y Min

Max y Min son funciones, por lo que reciben 1 argumento entre paréntesis.

En su argumento pueden recibir el **Precio** o algún indicador, de la siguiente manera:

- Max(Precio)
- Min(Precio)
- Max(Indicador(args))
- Min(Indicador(args))

Esta función se utiliza para declarar que el precio o algún indicador alcanza un máximo o un mínimo.

Indicadores

Los indicadores también se comportan como funciones, por lo que dentro de sus paréntesis reciben argumentos separados por “,” (coma).

A continuación se detalla cada indicador.

Para mayor detalle, revisar la siguiente [documentación](#).

SqueezeMomentumIndicator (SM)

- **length = 20**
- **length_KC = 20**

AwesomeOscillatorIndicator (AO)

- **window1: int = 5**
- **window2: int = 34**

KAMAIndicator (KAMA)

- **window: int = 10**
- **pow1: int = 2**
- **pow2: int = 30**

PercentagePriceOscillator (PPO)

- **window_slow: int = 26**
- **window_fast: int = 12**
- **window_sign: int = 9**

PercentageVolumeOscillator (PVO)

- **window_slow: int = 26**
- **window_fast: int = 12**
- **window_sign: int = 9**

ROCIIndicator (ROC)

- **window: int = 12**

RSIIndicator (RSI)

- **window: int = 14**

StochRSIIndicator (STOCHRSI)

- **window: int = 14**
- **smooth1: int = 3**
- **smooth2: int = 3**

StochasticOscillator (STOCH)

- **window: int = 14**
- **smooth_window: int = 3**

TSIIndicator (TSI)

- window_slow: int = 25
- window_fast: int = 13

UltimateOscillator (UO)

- window1: int = 7
- window2: int = 14
- window3: int = 28
- weight1: float = 4.0
- weight2: float = 2.0
- weight3: float = 1.0

WilliamsRIndicator (WI)

- lbp: int = 14

AccDistIndexIndicator (ADI)

- No recibe parámetros.

ChaikinMoneyFlowIndicator (CMF)

- window: int = 20

EaseOfMovementIndicator (EOM)

- window: int = 14

ForceIndexIndicator (FI)

- window: int = 13

MFIIndicator (MFI)

- window: int = 14

NegativeVolumeIndexIndicator (NVI)

- No recibe parámetros.

OnBalanceVolumeIndicator (OBV)

- No recibe parámetros.

VolumePriceTrendIndicator (VPT)

- No recibe parámetros.

VolumeWeightedAveragePrice (VWAP)

- window: int = 14

AverageTrueRange (ATR)

- window: int = 14

BollingerBands High (BBH)

- window: int = 20
- window_dev: int = 2

BollingerBands Low (BBL)

- window: int = 20
- window_dev: int = 2

BollingerBands Medium(BBM)

- window: int = 20
- window_dev: int = 2

DonchianChannel High (DCH)

- window: int = 20
- offset: int = 0

DonchianChannel Low (DCL)

- window: int = 20
- offset: int = 0

DonchianChannel Medium (DCM)

- window: int = 20
- offset: int = 0

UlcerIndex (UI)

- window: int = 14

ADXIndicator (ADX)

- window: int = 14

AroonIndicator (AI)

- window: int = 25

CCIIndicator (CCI)

- window: int = 20
- constant: float = 0.015
-

DPOIndicator (DPO)

- window: int = 20

EMAIndicator (EMA)

- window: int = 14

IchimokuIndicator A (IIA)

- window1: int = 9
- window2: int = 26
- window3: int = 52

IchimokuIndicator B (IIB)

- window1: int = 9
- window2: int = 26
- window3: int = 52

IchimokuIndicator Base Line (IICS)

- window1: int = 9
- window2: int = 26
- window3: int = 52

IchimokuIndicator Conversion Line (IITS)

- window1: int = 9
- window2: int = 26
- window3: int = 52

KSTIndicator (KST)

- roc1: int = 10
- roc2: int = 15
- roc3: int = 20
- roc4: int = 30
- window1: int = 10
- window2: int = 10
- window3: int = 10
- window4: int = 15
- nsig: int = 9

MACD (MACD)

- window_slow: int = 26
- window_fast: int = 12
- window_sign: int = 9

MassIndex (MI)

- window_fast: int = 9
- window_slow: int = 25

PSARIndicator (PSAR)

- step: float = 0.02
- max_step: float = 0.2

SMAIndicator (SMA)

- window: int = 14

STCIndicator (STC)

- window_slow: int = 50
- window_fast: int = 23
- cycle: int = 10
- smooth1: int = 3
- smooth2: int = 3

TRIXIndicator (TRIX)

- window: int = 15

VortexIndicator (VIP)

- window: int = 14

WMAIndicator (WMA)

- window: int = 9

CumulativeReturnIndicator (CR)

- No recibe parámetros.

DailyLogReturnIndicator (DLR)

- No recibe parámetros.

DailyReturnIndicator (DR)

- No recibe parámetros.