# Momentum Indicators

Momentum Indicators.

**classta.momentum.AwesomeOscillatorIndicator(high: pandas.core.series.Series, low: pandas.core.series.Series, window1: int = 5, window2: int = 34, fillna: bool = False)**

## Awesome Oscillator(AO)

From: <https://www.tradingview.com/wiki/Awesome_Oscillator_(AO>)

The Awesome Oscillator is an indicator used to measure market momentum. AO calculates the difference of a 34 Period and 5 Period Simple Moving Averages. The Simple Moving Averages that are used are not calculated using closing price but rather each bar’s midpoints. AO is generally used to affirm trends or to anticipate possible reversals.

From: <https://www.ifcm.co.uk/ntx-indicators/awesome-oscillator>

Awesome Oscillator is a 34-period simple moving average, plotted through the central points of the bars (H+L)/2, and subtracted from the 5-period simple moving average, graphed across the central points of the bars (H+L)/2.

MEDIAN PRICE = (HIGH+LOW)/2

AO = SMA(MEDIAN PRICE, 5)-SMA(MEDIAN PRICE, 34)

where

SMA — Simple Moving Average.

**Parameters**

* **high** (pandas.Series) – dataset ‘High’ column.
* **low** (pandas.Series) – dataset ‘Low’ column.
* **window1** (int) – short period.
* **window2** (int) – long period.
* **fillna** (bool) – if True, fill nan values with -50.

**awesome\_oscillator() → pandas.core.series.Series**

Awesome Oscillator

**Returns**

New feature generated.

**Return type**

pandas.Series

**classta.momentum.KAMAIndicator(close: pandas.core.series.Series, window: int = 10, pow1: int = 2, pow2: int = 30, fillna: bool = False)**

## Kaufman’s Adaptive Moving Average (KAMA)

Moving average designed to account for market noise or volatility. KAMA will closely follow prices when the price swings are relatively small and the noise is low. KAMA will adjust when the price swings widen and follow prices from a greater distance. This trend-following indicator can be used to identify the overall trend, time turning points and filter price movements.

<https://www.tradingview.com/ideas/kama/>

**Parameters**

* **close** (pandas.Series) – dataset ‘Close’ column.
* **window** (int) – n period.
* **pow1** (int) – number of periods for the fastest EMA constant.
* **pow2** (int) – number of periods for the slowest EMA constant.
* **fillna** (bool) – if True, fill nan values.

**kama() → pandas.core.series.Series**

Kaufman’s Adaptive Moving Average (KAMA)

**Returns**

New feature generated.

**Return type**

pandas.Series

**classta.momentum.PercentagePriceOscillator(close: pandas.core.series.Series, window\_slow: int = 26, window\_fast: int = 12, window\_sign: int = 9, fillna: bool = False)**

## Percentage Price Oscillator (PPO)

The Percentage Price Oscillator (PPO) is a momentum oscillator that measures the difference between two moving averages as a percentage of the larger moving average.

<https://school.stockcharts.com/doku.php?id=technical_indicators:price_oscillators_ppo>

**Parameters**

* **close** (pandas.Series) – dataset ‘Price’ column.
* **window\_slow** (int) – n period long-term.
* **window\_fast** (int) – n period short-term.
* **window\_sign** (int) – n period to signal.
* **fillna** (bool) – if True, fill nan values.

**ppo()**

Percentage Price Oscillator Line

**Returns**

New feature generated.

**Return type**

pandas.Series

**ppo\_hist()**

Percentage Price Oscillator Histogram

**Returns**

New feature generated.

**Return type**

pandas.Series

**ppo\_signal()**

Percentage Price Oscillator Signal Line

**Returns**

New feature generated.

**Return type**

pandas.Series

**classta.momentum.PercentageVolumeOscillator(volume: pandas.core.series.Series, window\_slow: int = 26, window\_fast: int = 12, window\_sign: int = 9, fillna: bool = False)**

## Percentage Volume Oscillator (PVO)

The Percentage Volume Oscillator (PVO) is a momentum oscillator for volume. The PVO measures the difference between two volume-based moving averages as a percentage of the larger moving average.

<https://school.stockcharts.com/doku.php?id=technical_indicators:percentage_volume_oscillator_pvo>

**Parameters**

* **volume** (pandas.Series) – dataset ‘Volume’ column.
* **window\_slow** (int) – n period long-term.
* **window\_fast** (int) – n period short-term.
* **window\_sign** (int) – n period to signal.
* **fillna** (bool) – if True, fill nan values.

**pvo() → pandas.core.series.Series**

PVO Line

**Returns**

New feature generated.

**Return type**

pandas.Series

**pvo\_hist() → pandas.core.series.Series**

Histgram

**Returns**

New feature generated.

**Return type**

pandas.Series

**pvo\_signal() → pandas.core.series.Series**

Signal Line

**Returns**

New feature generated.

**Return type**

pandas.Series

**classta.momentum.ROCIndicator(close: pandas.core.series.Series, window: int = 12, fillna: bool = False)**

## Rate of Change (ROC)

The Rate-of-Change (ROC) indicator, which is also referred to as simply Momentum, is a pure momentum oscillator that measures the percent change in price from one period to the next. The ROC calculation compares the current price with the price “n” periods ago. The plot forms an oscillator that fluctuates above and below the zero line as the Rate-of-Change moves from positive to negative. As a momentum oscillator, ROC signals include centerline crossovers, divergences and overbought-oversold readings. Divergences fail to foreshadow reversals more often than not, so this article will forgo a detailed discussion on them. Even though centerline crossovers are prone to whipsaw, especially short-term, these crossovers can be used to identify the overall trend. Identifying overbought or oversold extremes comes naturally to the Rate-of-Change oscillator.

<https://school.stockcharts.com/doku.php?id=technical_indicators:rate_of_change_roc_and_momentum>

**Parameters**

* **close** (pandas.Series) – dataset ‘Close’ column.
* **window** (int) – n period.
* **fillna** (bool) – if True, fill nan values.

**roc() → pandas.core.series.Series**

Rate of Change (ROC)

**Returns**

New feature generated.

**Return type**

pandas.Series

**classta.momentum.RSIIndicator(close: pandas.core.series.Series, window: int = 14, fillna: bool = False)**

## Relative Strength Index (RSI)

Compares the magnitude of recent gains and losses over a specified time period to measure speed and change of price movements of a security. It is primarily used to attempt to identify overbought or oversold conditions in the trading of an asset.

<https://www.investopedia.com/terms/r/rsi.asp>

**Parameters**

* **close** (pandas.Series) – dataset ‘Close’ column.
* **window** (int) – n period.
* **fillna** (bool) – if True, fill nan values.

**rsi() → pandas.core.series.Series**

Relative Strength Index (RSI)

**Returns**

New feature generated.

**Return type**

pandas.Series

**classta.momentum.StochRSIIndicator(close: pandas.core.series.Series, window: int = 14, smooth1: int = 3, smooth2: int = 3, fillna: bool = False)**

## Stochastic RSI

The StochRSI oscillator was developed to take advantage of both momentum indicators in order to create a more sensitive indicator that is attuned to a specific security’s historical performance rather than a generalized analysis of price change.

<https://school.stockcharts.com/doku.php?id=technical_indicators:stochrsi> <https://www.investopedia.com/terms/s/stochrsi.asp>

**Parameters**

* **close** (pandas.Series) – dataset ‘Close’ column.
* **window** (int) – n period
* **smooth1** (int) – moving average of Stochastic RSI
* **smooth2** (int) – moving average of %K
* **fillna** (bool) – if True, fill nan values.

**stochrsi()**

Stochastic RSI

**Returns**

New feature generated.

**Return type**

pandas.Series

**stochrsi\_d()**

Stochastic RSI %d

**Returns**

New feature generated.

**Return type**

pandas.Series

**stochrsi\_k()**

Stochastic RSI %k

**Returns**

New feature generated.

**Return type**

pandas.Series

**classta.momentum.StochasticOscillator(high: pandas.core.series.Series, low: pandas.core.series.Series, close: pandas.core.series.Series, window: int = 14, smooth\_window: int = 3, fillna: bool = False)**

## Stochastic Oscillator

Developed in the late 1950s by George Lane. The stochastic oscillator presents the location of the closing price of a stock in relation to the high and low range of the price of a stock over a period of time, typically a 14-day period.

<https://school.stockcharts.com/doku.php?id=technical_indicators:stochastic_oscillator_fast_slow_and_full>

**Parameters**

* **close** (pandas.Series) – dataset ‘Close’ column.
* **high** (pandas.Series) – dataset ‘High’ column.
* **low** (pandas.Series) – dataset ‘Low’ column.
* **window** (int) – n period.
* **smooth\_window** (int) – sma period over stoch\_k.
* **fillna** (bool) – if True, fill nan values.

**stoch() → pandas.core.series.Series**

Stochastic Oscillator

**Returns**

New feature generated.

**Return type**

pandas.Series

**stoch\_signal() → pandas.core.series.Series**

Signal Stochastic Oscillator

**Returns**

New feature generated.

**Return type:**pandas.Series

**classta.momentum.TSIIndicator(close: pandas.core.series.Series, window\_slow: int = 25, window\_fast: int = 13, fillna: bool = False)**

## True strength index (TSI)

Shows both trend direction and overbought/oversold conditions.

<https://school.stockcharts.com/doku.php?id=technical_indicators:true_strength_index>

**Parameters**

* **close** (pandas.Series) – dataset ‘Close’ column.
* **window\_slow** (int) – high period.
* **window\_fast** (int) – low period.
* **fillna** (bool) – if True, fill nan values.

**tsi() → pandas.core.series.Series**

True strength index (TSI)

**Returns**

New feature generated.

**Return type**

pandas.Series

**classta.momentum.UltimateOscillator(high: pandas.core.series.Series, low: pandas.core.series.Series, close: pandas.core.series.Series, window1: int = 7, window2: int = 14, window3: int = 28, weight1: float = 4.0, weight2: float = 2.0, weight3: float = 1.0, fillna: bool = False)**

## Ultimate Oscillator

Larry Williams’ (1976) signal, a momentum oscillator designed to capture momentum across three different timeframes.

<http://stockcharts.com/school/doku.php?id=chart_school:technical_indicators:ultimate_oscillator>

BP = Close - Minimum(Low or Prior Close). TR = Maximum(High or Prior Close) - Minimum(Low or Prior Close) Average7 = (7-period BP Sum) / (7-period TR Sum) Average14 = (14-period BP Sum) / (14-period TR Sum) Average28 = (28-period BP Sum) / (28-period TR Sum)

UO = 100 x [(4 x Average7)+(2 x Average14)+Average28]/(4+2+1)

**Parameters**

* **high** (pandas.Series) – dataset ‘High’ column.
* **low** (pandas.Series) – dataset ‘Low’ column.
* **close** (pandas.Series) – dataset ‘Close’ column.
* **window1** (int) – short period.
* **window2** (int) – medium period.
* **window3** (int) – long period.
* **weight1** (float) – weight of short BP average for UO.
* **weight2** (float) – weight of medium BP average for UO.
* **weight3** (float) – weight of long BP average for UO.
* **fillna** (bool) – if True, fill nan values with 50.

**ultimate\_oscillator() → pandas.core.series.Series**

Ultimate Oscillator

**Returns**

New feature generated.

**Return type**

pandas.Series

**classta.momentum.WilliamsRIndicator(high: pandas.core.series.Series, low: pandas.core.series.Series, close: pandas.core.series.Series, lbp: int = 14, fillna: bool = False)**

## Williams %R

Developed by Larry Williams, Williams %R is a momentum indicator that is the inverse of the Fast Stochastic Oscillator. Also referred to as %R, Williams %R reflects the level of the close relative to the highest high for the look-back period. In contrast, the Stochastic Oscillator reflects the level of the close relative to the lowest low. %R corrects for the inversion by multiplying the raw value by -100. As a result, the Fast Stochastic Oscillator and Williams %R produce the exact same lines, only the scaling is different. Williams %R oscillates from 0 to -100.

Readings from 0 to -20 are considered overbought. Readings from -80 to -100 are considered oversold.

Unsurprisingly, signals derived from the Stochastic Oscillator are also applicable to Williams %R.

%R = (Highest High - Close)/(Highest High - Lowest Low) \* -100

Lowest Low = lowest low for the look-back period Highest High = highest high for the look-back period %R is multiplied by -100 correct the inversion and move the decimal.

<https://school.stockcharts.com/doku.php?id=technical_indicators:williams_r>

The Williams %R oscillates from 0 to -100. When the indicator produces readings from 0 to -20, this indicates overbought market conditions. When readings are -80 to -100, it indicates oversold market conditions.

**Parameters**

* **high** (pandas.Series) – dataset ‘High’ column.
* **low** (pandas.Series) – dataset ‘Low’ column.
* **close** (pandas.Series) – dataset ‘Close’ column.
* **lbp** (int) – lookback period.
* **fillna** (bool) – if True, fill nan values with -50.

**williams\_r() → pandas.core.series.Series**

Williams %R

**Returns**

New feature generated.

**Return type**

pandas.Series

**ta.momentum.ppo\_hist(close: pandas.core.series.Series, window\_slow: int = 26, window\_fast: int = 12, window\_sign: int = 9, fillna: bool = False) → pandas.core.series.Series**

## Percentage Price Oscillator hist (PPO\_hist)

The Percentage Price Oscillator (PPO) is a momentum oscillator that measures the difference between two moving averages as a percentage of the larger moving average.

<https://school.stockcharts.com/doku.php?id=technical_indicators:price_oscillators_ppo>

**Parameters**

* **close** (pandas.Series) – dataset ‘Price’ column.
* **window\_slow** (int) – n period long-term.
* **window\_fast** (int) – n period short-term.
* **window\_sign** (int) – n period to signal.
* **fillna** (bool) – if True, fill nan values.

**Returns**

New feature generated.

**Return type**

pandas.Series

**ta.momentum.ppo\_signal(close: pandas.core.series.Series, window\_slow=26, window\_fast=12, window\_sign=9, fillna=False) → pandas.core.series.Series**

## Percentage Price Oscillator signal(PPO\_signal)

The Percentage Price Oscillator (PPO) is a momentum oscillator that measures the difference between two moving averages as a percentage of the larger moving average.

<https://school.stockcharts.com/doku.php?id=technical_indicators:price_oscillators_ppo>

**Parameters**

* **close** (pandas.Series) – dataset ‘Price’ column.
* **window\_slow** (int) – n period long-term.
* **window\_fast** (int) – n period short-term.
* **window\_sign** (int) – n period to signal.
* **fillna** (bool) – if True, fill nan values.

**Returns**

New feature generated.

**Return type**

pandas.Series

**ta.momentum.pvo\_hist(volume: pandas.core.series.Series, window\_slow: int = 26, window\_fast: int = 12, window\_sign: int = 9, fillna: bool = False) → pandas.core.series.Series**

## Percentage Volume Oscillator hist (PVO\_hist)

The Percentage Volume Oscillator (PVO) is a momentum oscillator for volume. The PVO measures the difference between two volume-based moving averages as a percentage of the larger moving average.

<https://school.stockcharts.com/doku.php?id=technical_indicators:percentage_volume_oscillator_pvo>

**Parameters**

* **volume** (pandas.Series) – dataset ‘Volume’ column.
* **window\_slow** (int) – n period long-term.
* **window\_fast** (int) – n period short-term.
* **window\_sign** (int) – n period to signal.
* **fillna** (bool) – if True, fill nan values.

**Returns**

New feature generated.

**Return type**

pandas.Series

**ta.momentum.pvo\_signal(volume: pandas.core.series.Series, window\_slow: int = 26, window\_fast: int = 12, window\_sign: int = 9, fillna: bool = False) → pandas.core.series.Series**

## Percentage Volume Oscillator signal(PVO\_signal)

The Percentage Volume Oscillator (PVO) is a momentum oscillator for volume. The PVO measures the difference between two volume-based moving averages as a percentage of the larger moving average.

<https://school.stockcharts.com/doku.php?id=technical_indicators:percentage_volume_oscillator_pvo>

**Parameters**

* **volume** (pandas.Series) – dataset ‘Volume’ column.
* **window\_slow** (int) – n period long-term.
* **window\_fast** (int) – n period short-term.
* **window\_sign** (int) – n period to signal.
* **fillna** (bool) – if True, fill nan values.

**Returns**

New feature generated.

**Return type**

pandas.Series

**ta.momentum.stoch\_signal(high, low, close, window=14, smooth\_window=3, fillna=False) → pandas.core.series.Series**

## Stochastic Oscillator Signal

Shows SMA of Stochastic Oscillator. Typically a 3 day SMA.

<https://www.investopedia.com/terms/s/stochasticoscillator.asp>

**Parameters**

* **high** (pandas.Series) – dataset ‘High’ column.
* **low** (pandas.Series) – dataset ‘Low’ column.
* **close** (pandas.Series) – dataset ‘Close’ column.
* **window** (int) – n period.
* **smooth\_window** (int) – sma period over stoch\_k
* **fillna** (bool) – if True, fill nan values.

**Returns**

New feature generated.

**Return type**

pandas.Series

**ta.momentum.stochrsi\_d(close: pandas.core.series.Series, window: int = 14, smooth1: int = 3, smooth2: int = 3, fillna: bool = False) → pandas.core.series.Series**

## Stochastic RSI %d

The StochRSI oscillator was developed to take advantage of both momentum indicators in order to create a more sensitive indicator that is attuned to a specific security’s historical performance rather than a generalized analysis of price change.

<https://www.investopedia.com/terms/s/stochrsi.asp>

**Parameters**

* **close** (pandas.Series) – dataset ‘Close’ column.
* **window** (int) – n period
* **smooth1** (int) – moving average of Stochastic RSI
* **smooth2** (int) – moving average of %K
* **fillna** (bool) – if True, fill nan values.

**Returns**

New feature generated.

**Return type**

pandas.Series

**ta.momentum.stochrsi\_k(close: pandas.core.series.Series, window: int = 14, smooth1: int = 3, smooth2: int = 3, fillna: bool = False) → pandas.core.series.Series**

## Stochastic RSI %k

The StochRSI oscillator was developed to take advantage of both momentum indicators in order to create a more sensitive indicator that is attuned to a specific security’s historical performance rather than a generalized analysis of price change.

<https://www.investopedia.com/terms/s/stochrsi.asp>

**Parameters**

* **close** (pandas.Series) – dataset ‘Close’ column.
* **window** (int) – n period
* **smooth1** (int) – moving average of Stochastic RSI
* **smooth2** (int) – moving average of %K
* **fillna** (bool) – if True, fill nan values.

**Returns**

New feature generated.

**Return type**

pandas.Series