**Trend Indicators**

Trend Indicators.

***class*ta.trend.ADXIndicator(*high: pandas.core.series.Series*, *low: pandas.core.series.Series*, *close: pandas.core.series.Series*, *window: int = 14*, *fillna: bool = False*)**

## Average Directional Movement Index (ADX)

The Plus Directional Indicator (+DI) and Minus Directional Indicator (-DI) are derived from smoothed averages of these differences, and measure trend direction over time. These two indicators are often referred to collectively as the Directional Movement Indicator (DMI).

The Average Directional Index (ADX) is in turn derived from the smoothed averages of the difference between +DI and -DI, and measures the strength of the trend (regardless of direction) over time.

Using these three indicators together, chartists can determine both the direction and strength of the trend.

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

**Parameters**

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

**adx() → pandas.core.series.Series**

Average Directional Index (ADX)

**Returns**

New feature generated.tr

**Return type**

pandas.Series

**adx\_neg() → pandas.core.series.Series**

### Minus Directional Indicator (-DI)

**Returns**

New feature generated.

**Return type**

pandas.Series

**adx\_pos() → pandas.core.series.Series**

### Plus Directional Indicator (+DI)

**Returns**

New feature generated.

**Return type**

pandas.Series

**ta.trend.adx\_neg(*high*, *low*, *close*, *window=14*, *fillna=False*)**

### Average Directional Movement Index Negative (ADX)

The Plus Directional Indicator (+DI) and Minus Directional Indicator (-DI) are derived from smoothed averages of these differences, and measure trend direction over time. These two indicators are often referred to collectively as the Directional Movement Indicator (DMI).

The Average Directional Index (ADX) is in turn derived from the smoothed averages of the difference between +DI and -DI, and measures the strength of the trend (regardless of direction) over time.

Using these three indicators together, chartists can determine both the direction and strength of the trend.

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

**Parameters**

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

**Returns**

New feature generated.

**Return type**

pandas.Series

**ta.trend.adx\_pos(*high*, *low*, *close*, *window=14*, *fillna=False*)**

### Average Directional Movement Index Positive (ADX)

The Plus Directional Indicator (+DI) and Minus Directional Indicator (-DI) are derived from smoothed averages of these differences, and measure trend direction over time. These two indicators are often referred to collectively as the Directional Movement Indicator (DMI).

The Average Directional Index (ADX) is in turn derived from the smoothed averages of the difference between +DI and -DI, and measures the strength of the trend (regardless of direction) over time.Using these three indicators together, chartists can determine both the direction and strength of the trend.

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

**Parameters**

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

**Returns**

New feature generated.

**Return type:**pandas.Series

***class*ta.trend.AroonIndicator(*close: pandas.core.series.Series*, *window: int = 25*, *fillna: bool = False*)**

## Aroon Indicator

Identify when trends are likely to change direction.

Aroon Up = ((N - Days Since N-day High) / N) x 100 Aroon Down = ((N - Days Since N-day Low) / N) x 100 Aroon Indicator = Aroon Up - Aroon Down

<https://www.investopedia.com/terms/a/aroon.asp>

**Parameters**

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

**aroon\_down() → pandas.core.series.Series**

Aroon Down Channel

**Returns**

New feature generated.

**Return type**

pandas.Series

**aroon\_indicator() → pandas.core.series.Series**

Aroon Indicator

**Returns**

New feature generated.

**Return type**

pandas.Series

**aroon\_up() → pandas.core.series.Series**

### Aroon Up Channel

**Returns**

New feature generated.

**Return type**

pandas.Series

**ta.trend.aroon\_up(*close*, *window=25*, *fillna=False*)**

### Aroon uptrend Indicator (AI)

Identify when trends are likely to change direction (uptrend).

Aroon Up - ((N - Days Since N-day High) / N) x 100

<https://www.investopedia.com/terms/a/aroon.asp>

**Parameters**

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

**Returns**

New feature generated.

**Return type**

pandas.Series

**ta.trend.aroon\_down(*close*, *window=25*, *fillna=False*)**

### Aroon down Indicator (AI)

Identify when trends are likely to change direction (downtrend).

Aroon Down - ((N - Days Since N-day Low) / N) x 100

<https://www.investopedia.com/terms/a/aroon.asp>

**Parameters**

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

**Returns**

New feature generated.

**Return type**

pandas.Series

***class*ta.trend.CCIIndicator(*high: pandas.core.series.Series*, *low: pandas.core.series.Series*, *close: pandas.core.series.Series*, *window: int = 20*, *constant: float = 0.015*, *fillna: bool = False*)**

## Commodity Channel Index (CCI)

CCI measures the difference between a security’s price change and its average price change. High positive readings indicate that prices are well above their average, which is a show of strength. Low negative readings indicate that prices are well below their average, which is a show of weakness.

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

**Parameters**

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

**cci() → pandas.core.series.Series**

Commodity Channel Index (CCI)

**Returns**

New feature generated.

**Return type**

pandas.Series

***class*ta.trend.DPOIndicator(*close: pandas.core.series.Series*, *window: int = 20*, *fillna: bool = False*)**

## Detrended Price Oscillator (DPO)

Is an indicator designed to remove trend from price and make it easier to identify cycles.

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

**Parameters**

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

**dpo() → pandas.core.series.Series**

Detrended Price Oscillator (DPO)

**Returns**

New feature generated.

**Return type**

pandas.Series

***class*ta.trend.IchimokuIndicator(*high: pandas.core.series.Series*, *low: pandas.core.series.Series*, *window1: int = 9*, *window2: int = 26*, *window3: int = 52*, *visual: bool = False*, *fillna: bool = False*)**

## Ichimoku Kinkō Hyō (Ichimoku)

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

**Parameters**

* **high** (*pandas.Series*) – dataset ‘High’ column.
* **low** (*pandas.Series*) – dataset ‘Low’ column.
* **window1** (*int*) – n1 low period.
* **window2** (*int*) – n2 medium period.
* **window3** (*int*) – n3 high period.
* **visual** (*bool*) – if True, shift n2 values.
* **fillna** (*bool*) – if True, fill nan values.

**ichimoku\_a() → pandas.core.series.Series**

### Senkou Span A (Leading Span A)

**Returns**

New feature generated.

**Return type**

pandas.Series

**ichimoku\_b() → pandas.core.series.Series**

### Senkou Span B (Leading Span B)

**Returns**

New feature generated.

**Return type**

pandas.Series

**ichimoku\_base\_line() → pandas.core.series.Series**

### Kijun-sen (Base Line)

**Returns**

New feature generated.

**Return type**

pandas.Series

**ichimoku\_conversion\_line() → pandas.core.series.Series**

### Tenkan-sen (Conversion Line)

**Returns**

New feature generated.

**Return type**

pandas.Series

**ta.trend.ichimoku\_b(*high*, *low*, *window2=26*, *window3=52*, *visual=False*, *fillna=False*)**

### Ichimoku Kinkō Hyō (Ichimoku)

It identifies the trend and look for potential signals within that trend.

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

**Parameters**

* **high** (*pandas.Series*) – dataset ‘High’ column.
* **low** (*pandas.Series*) – dataset ‘Low’ column.
* **window2** (*int*) – n2 medium period.
* **window3** (*int*) – n3 high period.
* **visual** (*bool*) – if True, shift n2 values.
* **fillna** (*bool*) – if True, fill nan values.

**Returns**

New feature generated.

**Return type**

pandas.Series

**ta.trend.ichimoku\_base\_line(*high*, *low*, *window1=9*, *window2=26*, *visual=False*, *fillna=False*) → pandas.core.series.Series**

### Kijun-sen (Base Line)

It identifies the trend and look for potential signals within that trend.

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

**Parameters**

* **high** (*pandas.Series*) – dataset ‘High’ column.
* **low** (*pandas.Series*) – dataset ‘Low’ column.
* **window1** (*int*) – n1 low period.
* **window2** (*int*) – n2 medium period.
* **visual** (*bool*) – if True, shift n2 values.
* **fillna** (*bool*) – if True, fill nan values.

**Returns**

New feature generated.

**Return type**

pandas.Series

**ta.trend.ichimoku\_conversion\_line(*high*, *low*, *window1=9*, *window2=26*, *visual=False*, *fillna=False*) → pandas.core.series.Series**

### Tenkan-sen (Conversion Line)

It identifies the trend and look for potential signals within that trend.

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

**Parameters**

* **high** (*pandas.Series*) – dataset ‘High’ column.
* **low** (*pandas.Series*) – dataset ‘Low’ column.
* **window1** (*int*) – n1 low period.
* **window2** (*int*) – n2 medium period.
* **visual** (*bool*) – if True, shift n2 values.
* **fillna** (*bool*) – if True, fill nan values.

**Returns**

New feature generated.

**Return type**

pandas.Series

***class*ta.trend.KSTIndicator(*close: pandas.core.series.Series*, *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*, *fillna: bool = False*)**

## KST Indicator Oscillator (KST Signal)

It is useful to identify major stock market cycle junctures because its formula is weighed to be more greatly influenced by the longer and more dominant time spans, in order to better reflect the primary swings of stock market cycle.

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

**Parameters**

* **close** (*pandas.Series*) – dataset ‘Close’ column.
* **roc1** (*int*) – roc1 period.
* **roc2** (*int*) – roc2 period.
* **roc3** (*int*) – roc3 period.
* **roc4** (*int*) – roc4 period.
* **window1** (*int*) – n1 smoothed period.
* **window2** (*int*) – n2 smoothed period.
* **window3** (*int*) – n3 smoothed period.
* **window4** (*int*) – n4 smoothed period.
* **nsig** (*int*) – n period to signal.
* **fillna** (*bool*) – if True, fill nan values.

**kst() → pandas.core.series.Series**

### Know Sure Thing (KST)

**Returns**

New feature generated.

**Return type**

pandas.Series

**kst\_diff() → pandas.core.series.Series**

### Diff Know Sure Thing (KST)

KST - Signal\_KST

**Returns**

New feature generated.

**Return type**

pandas.Series

**kst\_sig() → pandas.core.series.Series**

### Signal Line Know Sure Thing (KST)

nsig-period SMA of KST

**Returns**

New feature generated.

**Return type**

pandas.Series

**ta.trend.kst(*close*, *roc1=10*, *roc2=15*, *roc3=20*, *roc4=30*, *window1=10*, *window2=10*, *window3=10*, *window4=15*, *fillna=False*)**

### KST Oscillator (KST)

It is useful to identify major stock market cycle junctures because its formula is weighed to be more greatly influenced by the longer and more dominant time spans, in order to better reflect the primary swings of stock market cycle.

<https://en.wikipedia.org/wiki/KST_oscillator>

**Parameters**

* **close** (*pandas.Series*) – dataset ‘Close’ column.
* **roc1** (*int*) – r1 period.
* **roc2** (*int*) – r2 period.
* **roc3** (*int*) – r3 period.
* **roc4** (*int*) – r4 period.
* **window1** (*int*) – n1 smoothed period.
* **window2** (*int*) – n2 smoothed period.
* **window3** (*int*) – n3 smoothed period.
* **window4** (*int*) – n4 smoothed period.
* **fillna** (*bool*) – if True, fill nan values.

**Returns**

New feature generated.

**Return type**

pandas.Series

**ta.trend.kst\_sig(*close*, *roc1=10*, *roc2=15*, *roc3=20*, *roc4=30*, *window1=10*, *window2=10*, *window3=10*, *window4=15*, *nsig=9*, *fillna=False*)**

### KST Oscillator (KST Signal)

It is useful to identify major stock market cycle junctures because its formula is weighed to be more greatly influenced by the longer and more dominant time spans, in order to better reflect the primary swings of stock market cycle.

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

**Parameters**

* **close** (*pandas.Series*) – dataset ‘Close’ column.
* **roc1** (*int*) – roc1 period.
* **roc2** (*int*) – roc2 period.
* **roc3** (*int*) – roc3 period.
* **roc4** (*int*) – roc4 period.
* **window1** (*int*) – n1 smoothed period.
* **window2** (*int*) – n2 smoothed period.
* **window3** (*int*) – n3 smoothed period.
* **window4** (*int*) – n4 smoothed period.
* **nsig** (*int*) – n period to signal.
* **fillna** (*bool*) – if True, fill nan values.

**Returns**

New feature generated.

**Return type**

pandas.Series

***class*ta.trend.MACD(*close: pandas.core.series.Series*, *window\_slow: int = 26*, *window\_fast: int = 12*, *window\_sign: int = 9*, *fillna: bool = False*)**

## Moving Average Convergence Divergence (MACD)

Is a trend-following momentum indicator that shows the relationship between two moving averages of prices.

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

**Parameters**

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

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

MACD Line

**Returns**

New feature generated.

**Return type**

pandas.Series

**macd\_diff() → pandas.core.series.Series**

### MACD Histogram

**Returns**

New feature generated.

**Return type**

pandas.Series

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

### Signal Line

**Returns**

New feature generated.

**Return type**

pandas.Series

**ta.trend.macd\_diff(*close*, *window\_slow=26*, *window\_fast=12*, *window\_sign=9*, *fillna=False*)**

### Moving Average Convergence Divergence (MACD Diff)

Shows the relationship between MACD and MACD Signal.

<https://en.wikipedia.org/wiki/MACD>

**Parameters**

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

**Returns**

New feature generated.

**Return type**

pandas.Series

**ta.trend.macd\_signal(*close*, *window\_slow=26*, *window\_fast=12*, *window\_sign=9*, *fillna=False*)**

### Moving Average Convergence Divergence (MACD Signal)

Shows EMA of MACD.

<https://en.wikipedia.org/wiki/MACD>

**Parameters**

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

**Returns**

New feature generated.

**Return type**

pandas.Series

***class*ta.trend.MassIndex(*high: pandas.core.series.Series*, *low: pandas.core.series.Series*, *window\_fast: int = 9*, *window\_slow: int = 25*, *fillna: bool = False*)**

## Mass Index (MI)

It uses the high-low range to identify trend reversals based on range expansions. It identifies range bulges that can foreshadow a reversal of the current trend.

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

**Parameters**

* **high** (*pandas.Series*) – dataset ‘High’ column.
* **low** (*pandas.Series*) – dataset ‘Low’ column.
* **window\_fast** (*int*) – fast period value.
* **window\_slow** (*int*) – slow period value.
* **fillna** (*bool*) – if True, fill nan values.

**mass\_index() → pandas.core.series.Series**

Mass Index (MI)

**Returns**

New feature generated.

**Return type**

pandas.Series

***class*ta.trend.PSARIndicator(*high: pandas.core.series.Series*, *low: pandas.core.series.Series*, *close: pandas.core.series.Series*, *step: float = 0.02*, *max\_step: float = 0.2*, *fillna: bool = False*)**

## Parabolic Stop and Reverse (Parabolic SAR)

The Parabolic Stop and Reverse, more commonly known as the Parabolic SAR,is a trend-following indicator developed by J. Welles Wilder. The Parabolic SAR is displayed as a single parabolic line (or dots) underneath the price bars in an uptrend, and above the price bars in a downtrend.

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

**Parameters**

* **high** (*pandas.Series*) – dataset ‘High’ column.
* **low** (*pandas.Series*) – dataset ‘Low’ column.
* **close** (*pandas.Series*) – dataset ‘Close’ column.
* **step** (*float*) – the Acceleration Factor used to compute the SAR.
* **max\_step** (*float*) – the maximum value allowed for the Acceleration Factor.
* **fillna** (*bool*) – if True, fill nan values.

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

PSAR value

**Returns**

New feature generated.

**Return type**

pandas.Series

**psar\_down() → pandas.core.series.Series**

### PSAR down trend value

**Returns**

New feature generated.

**Return type**

pandas.Series

**psar\_down\_indicator() → pandas.core.series.Series**

### PSAR down trend value indicator

**Returns**

New feature generated.

**Return type**

pandas.Series

**psar\_up() → pandas.core.series.Series**

### PSAR up trend value

**Returns**

New feature generated.

**Return type**

pandas.Series

**psar\_up\_indicator() → pandas.core.series.Series**

### PSAR up trend value indicator

**Returns**

New feature generated.

**Return type**

pandas.Series

**ta.trend.psar\_down(*high*, *low*, *close*, *step=0.02*, *max\_step=0.2*, *fillna=False*)**

### Parabolic Stop and Reverse down (Parabolic SAR)

Returns the PSAR series with non-N/A values for downward trends

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

**Parameters**

* **high** (*pandas.Series*) – dataset ‘High’ column.
* **low** (*pandas.Series*) – dataset ‘Low’ column.
* **close** (*pandas.Series*) – dataset ‘Close’ column.
* **step** (*float*) – the Acceleration Factor used to compute the SAR.
* **max\_step** (*float*) – the maximum value allowed for the Acceleration Factor.
* **fillna** (*bool*) – if True, fill nan values.

**Returns**

New feature generated.

**Return type**

pandas.Series

**ta.trend.psar\_down\_indicator(*high*, *low*, *close*, *step=0.02*, *max\_step=0.2*, *fillna=False*)**

### Parabolic Stop and Reverse (Parabolic SAR) Downward Trend Indicator

Returns 1, if there is a reversal towards an downward trend. Else, returns 0.

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

**Parameters**

* **high** (*pandas.Series*) – dataset ‘High’ column.
* **low** (*pandas.Series*) – dataset ‘Low’ column.
* **close** (*pandas.Series*) – dataset ‘Close’ column.
* **step** (*float*) – the Acceleration Factor used to compute the SAR.
* **max\_step** (*float*) – the maximum value allowed for the Acceleration Factor.
* **fillna** (*bool*) – if True, fill nan values.

**Returns**

New feature generated.

**Return type**

pandas.Series

**ta.trend.psar\_up(*high*, *low*, *close*, *step=0.02*, *max\_step=0.2*, *fillna=False*)**

### Parabolic Stop and Reverse (Parabolic SAR)

Returns the PSAR series with non-N/A values for upward trends

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

**Parameters**

* **high** (*pandas.Series*) – dataset ‘High’ column.
* **low** (*pandas.Series*) – dataset ‘Low’ column.
* **close** (*pandas.Series*) – dataset ‘Close’ column.
* **step** (*float*) – the Acceleration Factor used to compute the SAR.
* **max\_step** (*float*) – the maximum value allowed for the Acceleration Factor.
* **fillna** (*bool*) – if True, fill nan values.

**Returns**

New feature generated.

**Return type**

pandas.Series

**ta.trend.psar\_up\_indicator(*high*, *low*, *close*, *step=0.02*, *max\_step=0.2*, *fillna=False*)**

### Parabolic Stop and Reverse (Parabolic SAR) Upward Trend Indicator

Returns 1, if there is a reversal towards an upward trend. Else, returns 0.

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

**Parameters**

* **high** (*pandas.Series*) – dataset ‘High’ column.
* **low** (*pandas.Series*) – dataset ‘Low’ column.
* **close** (*pandas.Series*) – dataset ‘Close’ column.
* **step** (*float*) – the Acceleration Factor used to compute the SAR.
* **max\_step** (*float*) – the maximum value allowed for the Acceleration Factor.
* **fillna** (*bool*) – if True, fill nan values.

**Returns**

New feature generated.

**Return type**

pandas.Series

***class*ta.trend.SMAIndicator(*close: pandas.core.series.Series*, *window: int*, *fillna: bool = False*)**

## SMA - Simple Moving Average

**Parameters**

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

**sma\_indicator() → pandas.core.series.Series**

Simple Moving Average (SMA)

**Returns**

New feature generated.

**Return type**

pandas.Series

***class*ta.trend.EMAIndicator(*close: pandas.core.series.Series*, *window: int = 14*, *fillna: bool = False*)**

## EMA - Exponential Moving Average

**Parameters**

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

**ema\_indicator() → pandas.core.series.Series**

Exponential Moving Average (EMA)

**Returns**

New feature generated.

**Return type**

pandas.Series

***class*ta.trend.WMAIndicator(*close: pandas.core.series.Series*, *window: int = 9*, *fillna: bool = False*)**

## WMA - Weighted Moving Average

**Parameters**

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

**wma() → pandas.core.series.Series**

Weighted Moving Average (WMA)

**Returns**

New feature generated.

**Return type**

pandas.Series

***class*ta.trend.STCIndicator(*close: pandas.core.series.Series*, *window\_slow: int = 50*, *window\_fast: int = 23*, *cycle: int = 10*, *smooth1: int = 3*, *smooth2: int = 3*, *fillna: bool = False*)**

## Schaff Trend Cycle (STC)

The Schaff Trend Cycle (STC) is a charting indicator that is commonly used to identify market trends and provide buy and sell signals to traders. Developed in 1999 by noted currency trader Doug Schaff, STC is a type of oscillator and is based on the assumption that, regardless of time frame, currency trends accelerate and decelerate in cyclical patterns.

<https://www.investopedia.com/articles/forex/10/schaff-trend-cycle-indicator.asp>

**Parameters**

* **close** (*pandas.Series*) – dataset ‘Close’ column.
* **window\_fast** (*int*) – n period short-term.
* **window\_slow** (*int*) – n period long-term.
* **cycle** (*int*) – cycle size
* **smooth1** (*int*) – ema period over stoch\_k
* **smooth2** (*int*) – ema period over stoch\_kd
* **fillna** (*bool*) – if True, fill nan values.

**stc()**

Schaff Trend Cycle

**Returns**

New feature generated.

**Return type**

pandas.Series

***class*ta.trend.TRIXIndicator(*close: pandas.core.series.Series*, *window: int = 15*, *fillna: bool = False*)**

## Trix (TRIX)

Shows the percent rate of change of a triple exponentially smoothed moving average.

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

**Parameters**

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

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

Trix (TRIX)

**Returns**

New feature generated.

**Return type**

pandas.Series

***class*ta.trend.VortexIndicator(*high: pandas.core.series.Series*, *low: pandas.core.series.Series*, *close: pandas.core.series.Series*, *window: int = 14*, *fillna: bool = False*)**

## Vortex Indicator (VI)

It consists of two oscillators that capture positive and negative trend movement. A bullish signal triggers when the positive trend indicator crosses above the negative trend indicator or a key level.

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

**Parameters**

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

**vortex\_indicator\_diff()**

### Diff VI

**Returns**

New feature generated.

**Return type**

pandas.Series

**vortex\_indicator\_neg()**

### -VI

**Returns**

New feature generated.

**Return type**

pandas.Series

**vortex\_indicator\_pos()**

### +VI

**Returns**

New feature generated.

**Return type**

pandas.Series

**ta.trend.vortex\_indicator\_neg(*high*, *low*, *close*, *window=14*, *fillna=False*)**

### Vortex Indicator neg (VI)

It consists of two oscillators that capture positive and negative trend movement. A bearish signal triggers when the negative trend indicator crosses above the positive trend indicator or a key level.

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

**Parameters**

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

**Returns**

New feature generated.

**Return type**

pandas.Series

**ta.trend.vortex\_indicator\_pos(*high*, *low*, *close*, *window=14*, *fillna=False*)**

### Vortex Indicator pos (VI)

It consists of two oscillators that capture positive and negative trend movement. A bullish signal triggers when the positive trend indicator crosses above the negative trend indicator or a key level.

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

**Parameters**

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

**Returns**

New feature generated.

**Return type**

pandas.Series

**ta.trend.dpo(*close*, *window=20*, *fillna=False*)**