Technical Analysis Library in Python Documentation

Release 0.1.4

Dario Lopez Padial (Bukosabino)

Apr 21, 2020

CONTENTS

It is a Technical Analysis library to financial time series datasets (open, close, high, low, volume). You can use it to do feature engineering from financial datasets. It is builded on Python Pandas library.

INSTALLATION (PYTHON >= V3.6)

```
> virtualenv -p python3 virtualenvironment
> source virtualenvironment/bin/activate
> pip install ta
```

TWO

EXAMPLES

Example adding all features:

```python
import pandas as pd
import ta

# Load datas
df = pd.read_csv('ta/tests/data/datas.csv', sep=',')

# Clean NaN values
df = ta.utils.dropna(df)

# Add ta features filling NaN values
df = ta.add_all_ta_features(
    df, open="Open", high="High", low="Low", close="Close", volume="Volume_BTC",␣
↪fillna=True)
```

Example adding a particular feature:

```python
import pandas as pd
import ta

# Load datas
df = pd.read_csv('ta/tests/data/datas.csv', sep=',')

# Clean NaN values
df = ta.utils.dropna(df)

# Initialize Bollinger Bands Indicator
indicator_bb = ta.volatility.BollingerBands(close=df["Close"], n=20, ndev=2)

# Add Bollinger Bands features
df['bb_bbm'] = indicator_bb.bollinger_mavg()
df['bb_bbh'] = indicator_bb.bollinger_hband()
df['bb_bbl'] = indicator_bb.bollinger_lband()

# Add Bollinger Band high indicator
df['bb_bbhi'] = indicator_bb.bollinger_hband_indicator()

# Add Bollinger Band low indicator
df['bb_bbli'] = indicator_bb.bollinger_lband_indicator()
```

THREE

MOTIVATION

- English: https://towardsdatascience.com/technical-analysis-library-to-financial-datasets-with-pandas-python-4b2b390d3543

- Spanish: https://medium.com/datos-y-ciencia/biblioteca-de-an%C3%A1lisis-t%C3%A9cnico-sobre-series-temporales-financieras-para-machine-learning-con-cb28f9427d0

CONTENTS

4.1 Documentation

It is a technical analysis library to financial time series datasets. You can use it to do feature engineering from financial datasets. It is builded on pandas python library.

4.1.1 Momentum Indicators

Momentum Indicators.

class ta.momentum.**AwesomeOscillatorIndicator**(*high: pandas.core.series.Series*, *low: pandas.core.series.Series*, *s: int = 5*, *len: int = 34*, *fillna: bool = False*)

> Awesome Oscillator
>
> 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.
> > - **s** (*int*) – short period.
> > - **len** (*int*) – long period.
> > - **fillna** (*bool*) – if True, fill nan values with -50.

ao() → pandas.core.series.Series
    Awesome Oscillator

        Returns New feature generated.

        Return type pandas.Series

class ta.momentum.**KAMAIndicator**(*close: pandas.core.series.Series, n: 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.

                • **n** (*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

class ta.momentum.**ROCIndicator**(*close: pandas.core.series.Series, n: 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.

                • **n** (*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

class ta.momentum.**RSIIndicator**(*close: pandas.core.series.Series, n: 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.
> - **n** (*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

class ta.momentum.**StochasticOscillator**(*high: pandas.core.series.Series, low: pandas.core.series.Series, close: pandas.core.series.Series, n: int = 14, d_n: 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.
> - **n** (*int*) – n period.
> - **d_n** (*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

class ta.momentum.**TSIIndicator**(*close: pandas.core.series.Series*, *r: int = 25*, *s: 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.
>
> - **r** (*int*) – high period.
>
> - **s** (*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

class ta.momentum.**UltimateOscillator**(*high: pandas.core.series.Series*, *low: pandas.core.series.Series*, *close: pandas.core.series.Series*, *s: int = 7*, *m: int = 14*, *len: int = 28*, *ws: float = 4.0*, *wm: float = 2.0*, *wl: 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.
>
> - **s** (*int*) – short period.
>
> - **m** (*int*) – medium period.
>
> - **len** (*int*) – long period.
>
> - **ws** (*float*) – weight of short BP average for UO.
>
> - **wm** (*float*) – weight of medium BP average for UO.
>
> - **wl** (*float*) – weight of long BP average for UO.
>
> - **fillna** (*bool*) – if True, fill nan values with 50.

uo() → pandas.core.series.Series
> Ultimate Oscillator
>
> > **Returns** New feature generated.

Return type pandas.Series

class ta.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.

wr() → pandas.core.series.Series

Williams %R

> **Returns** New feature generated.
>
> **Return type** pandas.Series

ta.momentum.**ao**(*high*, *low*, *s=5*, *len=34*, *fillna=False*)

Awesome Oscillator

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.
>
> - **s** (*int*) – short period.
>
> - **len** (*int*) – long period.
>
> - **fillna** (*bool*) – if True, fill nan values with -50.
>
> **Returns** New feature generated.
>
> **Return type** pandas.Series

ta.momentum.**kama** (*close*, *n=10*, *pow1=2*, *pow2=30*, *fillna=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.
>
> - **n** (*int*) – n number of periods for the efficiency ratio.
>
> - **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.
>
> **Returns** New feature generated.
>
> **Return type** pandas.Series

ta.momentum.**roc** (*close*, *n=12*, *fillna=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.
>
> - **n** (*int*) – n periods.
>
> - **fillna** (*bool*) – if True, fill nan values.

> **Returns** New feature generated.
>
> **Return type** pandas.Series

ta.momentum.**rsi**(*close*, *n=14*, *fillna=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.
> >
> > - **n** (*int*) – n period.
> >
> > - **fillna** (*bool*) – if True, fill nan values.
>
> **Returns** New feature generated.
>
> **Return type** pandas.Series

ta.momentum.**stoch**(*high*, *low*, *close*, *n=14*, *d_n=3*, *fillna=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://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.
> >
> > - **n** (*int*) – n period.
> >
> > - **d_n** (*int*) – sma period over stoch_k
> >
> > - **fillna** (*bool*) – if True, fill nan values.
>
> **Returns** New feature generated.
>
> **Return type** pandas.Series

ta.momentum.**stoch_signal**(*high*, *low*, *close*, *n=14*, *d_n=3*, *fillna=False*)
> 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.
> >
> > - **n** (*int*) – n period.

- **d_n** (*int*) – sma period over stoch_k

- **fillna** (*bool*) – if True, fill nan values.

Returns New feature generated.

Return type pandas.Series

ta.momentum.**tsi**(*close*, *r=25*, *s=13*, *fillna=False*)

True strength index (TSI)

Shows both trend direction and overbought/oversold conditions.

https://en.wikipedia.org/wiki/True_strength_index

Parameters

- **close** (*pandas.Series*) – dataset 'Close' column.

- **r** (*int*) – high period.

- **s** (*int*) – low period.

- **fillna** (*bool*) – if True, fill nan values.

Returns New feature generated.

Return type pandas.Series

ta.momentum.**uo**(*high*, *low*, *close*, *s=7*, *m=14*, *len=28*, *ws=4.0*, *wm=2.0*, *wl=1.0*, *fillna=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.

- **s** (*int*) – short period.

- **m** (*int*) – medium period.

- **len** (*int*) – long period.

- **ws** (*float*) – weight of short BP average for UO.

- **wm** (*float*) – weight of medium BP average for UO.

- **wl** (*float*) – weight of long BP average for UO.

- **fillna** (*bool*) – if True, fill nan values with 50.

Returns New feature generated.

Return type pandas.Series

`ta.momentum.`**`wr`**(*high*, *low*, *close*, *lbp=14*, *fillna=False*)

    Williams %R

    From: http://stockcharts.com/school/doku.php?id=chart_school:technical_indicators: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.

    From: https://www.investopedia.com/terms/w/williamsr.asp 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.

        Returns New feature generated.

        Return type pandas.Series

4.1.2 Volume Indicators

Volume Indicators.

class `ta.volume.`**`AccDistIndexIndicator`**(*high: pandas.core.series.Series*, *low: pandas.core.series.Series*, *close: pandas.core.series.Series*, *volume: pandas.core.series.Series*, *fillna: bool = False*)

    Accumulation/Distribution Index (ADI)

    Acting as leading indicator of price movements.

    https://school.stockcharts.com/doku.php?id=technical_indicators:accumulation_distribution_line

        Parameters

- **high** (*pandas.Series*) – dataset 'High' column.
- **low** (*pandas.Series*) – dataset 'Low' column.
- **close** (*pandas.Series*) – dataset 'Close' column.
- **volume** (*pandas.Series*) – dataset 'Volume' column.

- **fillna** (*bool*) – if True, fill nan values.

acc_dist_index() → pandas.core.series.Series
    Accumulation/Distribution Index (ADI)

    Returns New feature generated.

    Return type pandas.Series

class ta.volume.**ChaikinMoneyFlowIndicator**(*high: pandas.core.series.Series,   low:
                                        pandas.core.series.Series,   close: pan-
                                        das.core.series.Series,   volume: pan-
                                        das.core.series.Series, n: int = 20, fillna:
                                        bool = False*)

Chaikin Money Flow (CMF)

It measures the amount of Money Flow Volume over a specific period.

http://stockcharts.com/school/doku.php?id=chart_school:technical_indicators:chaikin_money_flow_cmf

    Parameters

        - **high** (*pandas.Series*) – dataset 'High' column.

        - **low** (*pandas.Series*) – dataset 'Low' column.

        - **close** (*pandas.Series*) – dataset 'Close' column.

        - **volume** (*pandas.Series*) – dataset 'Volume' column.

        - **n** (*int*) – n period.

        - **fillna** (*bool*) – if True, fill nan values.

chaikin_money_flow() → pandas.core.series.Series
    Chaikin Money Flow (CMF)

    Returns New feature generated.

    Return type pandas.Series

class ta.volume.**EaseOfMovementIndicator**(*high: pandas.core.series.Series,   low:
                                        pandas.core.series.Series,   volume: pan-
                                        das.core.series.Series, n: int = 14, fillna: bool =
                                        False*)

Ease of movement (EoM, EMV)

It relate an asset's price change to its volume and is particularly useful for assessing the strength of a trend.

https://en.wikipedia.org/wiki/Ease_of_movement

    Parameters

        - **high** (*pandas.Series*) – dataset 'High' column.

        - **low** (*pandas.Series*) – dataset 'Low' column.

        - **volume** (*pandas.Series*) – dataset 'Volume' column.

        - **n** (*int*) – n period.

        - **fillna** (*bool*) – if True, fill nan values.

ease_of_movement() → pandas.core.series.Series
    Ease of movement (EoM, EMV)

    Returns New feature generated.

> **Return type** pandas.Series

sma_ease_of_movement() → pandas.core.series.Series
> Signal Ease of movement (EoM, EMV)

>> **Returns** New feature generated.

>> **Return type** pandas.Series

class ta.volume.**ForceIndexIndicator**(*close:* *pandas.core.series.Series*, *volume:* *pandas.core.series.Series*, *n: int = 13*, *fillna: bool = False*)

Force Index (FI)

It illustrates how strong the actual buying or selling pressure is. High positive values mean there is a strong rising trend, and low values signify a strong downward trend.

http://stockcharts.com/school/doku.php?id=chart_school:technical_indicators:force_index

> **Parameters**

>> • **high** (*pandas.Series*) – dataset 'High' column.

>> • **low** (*pandas.Series*) – dataset 'Low' column.

>> • **close** (*pandas.Series*) – dataset 'Close' column.

>> • **volume** (*pandas.Series*) – dataset 'Volume' column.

>> • **n** (*int*) – n period.

>> • **fillna** (*bool*) – if True, fill nan values.

force_index() → pandas.core.series.Series
> Force Index (FI)

>> **Returns** New feature generated.

>> **Return type** pandas.Series

class ta.volume.**MFIIndicator**(*high:* *pandas.core.series.Series*, *low:* *pandas.core.series.Series*, *close:* *pandas.core.series.Series*, *volume:* *pandas.core.series.Series*, *n: int = 14*, *fillna: bool = False*)

Money Flow Index (MFI)

Uses both price and volume to measure buying and selling pressure. It is positive when the typical price rises (buying pressure) and negative when the typical price declines (selling pressure). A ratio of positive and negative money flow is then plugged into an RSI formula to create an oscillator that moves between zero and one hundred.

http://stockcharts.com/school/doku.php?id=chart_school:technical_indicators:money_flow_index_mfi

> **Parameters**

>> • **high** (*pandas.Series*) – dataset 'High' column.

>> • **low** (*pandas.Series*) – dataset 'Low' column.

>> • **close** (*pandas.Series*) – dataset 'Close' column.

>> • **volume** (*pandas.Series*) – dataset 'Volume' column.

>> • **n** (*int*) – n period.

>> • **fillna** (*bool*) – if True, fill nan values.

money_flow_index() → pandas.core.series.Series
> Money Flow Index (MFI)

> **Returns** New feature generated.
>
> **Return type** pandas.Series

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

Negative Volume Index (NVI)

http://stockcharts.com/school/doku.php?id=chart_school:technical_indicators:negative_volume_inde

> **Parameters**
>
> - **close** (*pandas.Series*) – dataset 'Close' column.
>
> - **volume** (*pandas.Series*) – dataset 'Volume' column.
>
> - **fillna** (*bool*) – if True, fill nan values with 1000.

negative_volume_index() → pandas.core.series.Series

Negative Volume Index (NVI)

> **Returns** New feature generated.
>
> **Return type** pandas.Series

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

On-balance volume (OBV)

It relates price and volume in the stock market. OBV is based on a cumulative total volume.

https://en.wikipedia.org/wiki/On-balance_volume

> **Parameters**
>
> - **close** (*pandas.Series*) – dataset 'Close' column.
>
> - **volume** (*pandas.Series*) – dataset 'Volume' column.
>
> - **fillna** (*bool*) – if True, fill nan values.

on_balance_volume() → pandas.core.series.Series

On-balance volume (OBV)

> **Returns** New feature generated.
>
> **Return type** pandas.Series

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

Volume-price trend (VPT)

Is based on a running cumulative volume that adds or substracts a multiple of the percentage change in share price trend and current volume, depending upon the investment's upward or downward movements.

https://en.wikipedia.org/wiki/Volume%E2%80%93price_trend

> **Parameters**
>
> - **close** (*pandas.Series*) – dataset 'Close' column.
>
> - **volume** (*pandas.Series*) – dataset 'Volume' column.
>
> - **fillna** (*bool*) – if True, fill nan values.

volume_price_trend() → pandas.core.series.Series

Volume-price trend (VPT)

> **Returns** New feature generated.
>
> **Return type** pandas.Series

class ta.volume.**VolumeWeightedAveragePrice**(*high:* *pandas.core.series.Series*, *low:* *pandas.core.series.Series*, *close:* *pandas.core.series.Series*, *volume:* *pandas.core.series.Series*, *n:* *int = 14*, *fillna:* *bool = False*)

Volume Weighted Average Price (VWAP)

VWAP equals the dollar value of all trading periods divided by the total trading volume for the current day. The calculation starts when trading opens and ends when it closes. Because it is good for the current trading day only, intraday periods and data are used in the calculation.

https://school.stockcharts.com/doku.php?id=technical_indicators:vwap_intraday

> **Parameters**
>
> - **high** (*pandas.Series*) – dataset 'High' column.
> - **low** (*pandas.Series*) – dataset 'Low' column.
> - **close** (*pandas.Series*) – dataset 'Close' column.
> - **volume** (*pandas.Series*) – dataset 'Volume' column.
> - **n** (*int*) – n period.
> - **fillna** (*bool*) – if True, fill nan values.
>
> **Returns** New feature generated.
>
> **Return type** pandas.Series

volume_weighted_average_price() → pandas.core.series.Series
Volume Weighted Average Price (VWAP)

> **Returns** New feature generated.
>
> **Return type** pandas.Series

ta.volume.**acc_dist_index**(*high*, *low*, *close*, *volume*, *fillna=False*)
Accumulation/Distribution Index (ADI)

Acting as leading indicator of price movements.

https://en.wikipedia.org/wiki/Accumulation/distribution_index

> **Parameters**
>
> - **high** (*pandas.Series*) – dataset 'High' column.
> - **low** (*pandas.Series*) – dataset 'Low' column.
> - **close** (*pandas.Series*) – dataset 'Close' column.
> - **volume** (*pandas.Series*) – dataset 'Volume' column.
> - **fillna** (*bool*) – if True, fill nan values.
>
> **Returns** New feature generated.
>
> **Return type** pandas.Series

ta.volume.**chaikin_money_flow**(*high*, *low*, *close*, *volume*, *n=20*, *fillna=False*)
Chaikin Money Flow (CMF)

It measures the amount of Money Flow Volume over a specific period.

http://stockcharts.com/school/doku.php?id=chart_school:technical_indicators:chaikin_money_flow_cmf

> **Parameters**
>
> - **high** (*pandas.Series*) – dataset 'High' column.
> - **low** (*pandas.Series*) – dataset 'Low' column.
> - **close** (*pandas.Series*) – dataset 'Close' column.
> - **volume** (*pandas.Series*) – dataset 'Volume' column.
> - **n** (*int*) – n period.
> - **fillna** (*bool*) – if True, fill nan values.
>
> **Returns** New feature generated.
>
> **Return type** pandas.Series

`ta.volume.`**`ease_of_movement`**(*high*, *low*, *volume*, *n=14*, *fillna=False*)

> Ease of movement (EoM, EMV)

It relate an asset's price change to its volume and is particularly useful for assessing the strength of a trend.

https://en.wikipedia.org/wiki/Ease_of_movement

> **Parameters**
>
> - **high** (*pandas.Series*) – dataset 'High' column.
> - **low** (*pandas.Series*) – dataset 'Low' column.
> - **volume** (*pandas.Series*) – dataset 'Volume' column.
> - **n** (*int*) – n period.
> - **fillna** (*bool*) – if True, fill nan values.
>
> **Returns** New feature generated.
>
> **Return type** pandas.Series

`ta.volume.`**`force_index`**(*close*, *volume*, *n=13*, *fillna=False*)

> Force Index (FI)

It illustrates how strong the actual buying or selling pressure is. High positive values mean there is a strong rising trend, and low values signify a strong downward trend.

http://stockcharts.com/school/doku.php?id=chart_school:technical_indicators:force_index

> **Parameters**
>
> - **close** (*pandas.Series*) – dataset 'Close' column.
> - **volume** (*pandas.Series*) – dataset 'Volume' column.
> - **n** (*int*) – n period.
> - **fillna** (*bool*) – if True, fill nan values.
>
> **Returns** New feature generated.
>
> **Return type** pandas.Series

`ta.volume.`**`money_flow_index`**(*high*, *low*, *close*, *volume*, *n=14*, *fillna=False*)

> Money Flow Index (MFI)

Uses both price and volume to measure buying and selling pressure. It is positive when the typical price rises (buying pressure) and negative when the typical price declines (selling pressure). A ratio of positive and negative money flow is then plugged into an RSI formula to create an oscillator that moves between zero and one hundred.

http://stockcharts.com/school/doku.php?id=chart_school:technical_indicators:money_flow_index_mfi

> **Parameters**
>
> - **high** (*pandas.Series*) – dataset 'High' column.
>
> - **low** (*pandas.Series*) – dataset 'Low' column.
>
> - **close** (*pandas.Series*) – dataset 'Close' column.
>
> - **volume** (*pandas.Series*) – dataset 'Volume' column.
>
> - **n** (*int*) – n period.
>
> - **fillna** (*bool*) – if True, fill nan values.
>
> **Returns** New feature generated.
>
> **Return type** pandas.Series

ta.volume.**negative_volume_index**(*close*, *volume*, *fillna=False*)
Negative Volume Index (NVI)

http://stockcharts.com/school/doku.php?id=chart_school:technical_indicators:negative_volume_inde

The Negative Volume Index (NVI) is a cumulative indicator that uses the change in volume to decide when the smart money is active. Paul Dysart first developed this indicator in the 1930s. [. . .] Dysart's Negative Volume Index works under the assumption that the smart money is active on days when volume decreases and the not-so-smart money is active on days when volume increases.

The cumulative NVI line was unchanged when volume increased from one period to the other. In other words, nothing was done. Norman Fosback, of Stock Market Logic, adjusted the indicator by substituting the percentage price change for Net Advances.

This implementation is the Fosback version.

If today's volume is less than yesterday's volume then: nvi(t) = nvi(t-1) * (1 + (close(t) - close(t-1)) / close(t-1))

Else nvi(t) = nvi(t-1)

Please note: the "stockcharts.com" example calculation just adds the percentange change of price to previous NVI when volumes decline; other sources indicate that the same percentage of the previous NVI value should be added, which is what is implemented here.

> **Parameters**
>
> - **close** (*pandas.Series*) – dataset 'Close' column.
>
> - **volume** (*pandas.Series*) – dataset 'Volume' column.
>
> - **fillna** (*bool*) – if True, fill nan values with 1000.
>
> **Returns** New feature generated.
>
> **Return type** pandas.Series

See also:

https://en.wikipedia.org/wiki/Negative_volume_index

`ta.volume.`**`on_balance_volume`**(*close*, *volume*, *fillna=False*)

> On-balance volume (OBV)
>
> It relates price and volume in the stock market. OBV is based on a cumulative total volume.
>
> https://en.wikipedia.org/wiki/On-balance_volume
>
> > **Parameters**
> >
> > - **close** (*pandas.Series*) – dataset 'Close' column.
> >
> > - **volume** (*pandas.Series*) – dataset 'Volume' column.
> >
> > - **fillna** (*bool*) – if True, fill nan values.
> >
> > **Returns** New feature generated.
> >
> > **Return type** pandas.Series

`ta.volume.`**`put_call_ratio`**()

> Put/Call ratio (PCR) https://en.wikipedia.org/wiki/Put/call_ratio

`ta.volume.`**`sma_ease_of_movement`**(*high*, *low*, *volume*, *n=14*, *fillna=False*)

> Ease of movement (EoM, EMV)
>
> It relate an asset's price change to its volume and is particularly useful for assessing the strength of a trend.
>
> https://en.wikipedia.org/wiki/Ease_of_movement
>
> > **Parameters**
> >
> > - **high** (*pandas.Series*) – dataset 'High' column.
> >
> > - **low** (*pandas.Series*) – dataset 'Low' column.
> >
> > - **volume** (*pandas.Series*) – dataset 'Volume' column.
> >
> > - **n** (*int*) – n period.
> >
> > - **fillna** (*bool*) – if True, fill nan values.
> >
> > **Returns** New feature generated.
> >
> > **Return type** pandas.Series

`ta.volume.`**`volume_price_trend`**(*close*, *volume*, *fillna=False*)

> Volume-price trend (VPT)
>
> Is based on a running cumulative volume that adds or substracts a multiple of the percentage change in share price trend and current volume, depending upon the investment's upward or downward movements.
>
> https://en.wikipedia.org/wiki/Volume%E2%80%93price_trend
>
> > **Parameters**
> >
> > - **close** (*pandas.Series*) – dataset 'Close' column.
> >
> > - **volume** (*pandas.Series*) – dataset 'Volume' column.
> >
> > - **fillna** (*bool*) – if True, fill nan values.
> >
> > **Returns** New feature generated.
> >
> > **Return type** pandas.Series

`ta.volume.`**`volume_weighted_average_price`**(*high: pandas.core.series.Series, low: pandas.core.series.Series, close: pandas.core.series.Series, volume: pandas.core.series.Series, n: int = 14, fillna: bool = False*)

Volume Weighted Average Price (VWAP)

VWAP equals the dollar value of all trading periods divided by the total trading volume for the current day. The calculation starts when trading opens and ends when it closes. Because it is good for the current trading day only, intraday periods and data are used in the calculation.

https://school.stockcharts.com/doku.php?id=technical_indicators:vwap_intraday

>   **Parameters**
>
> * **high** (*pandas.Series*) – dataset 'High' column.
> * **low** (*pandas.Series*) – dataset 'Low' column.
> * **close** (*pandas.Series*) – dataset 'Close' column.
> * **volume** (*pandas.Series*) – dataset 'Volume' column.
> * **n** (*int*) – n period.
> * **fillna** (*bool*) – if True, fill nan values.
>
>   **Returns** New feature generated.
>
>   **Return type** pandas.Series

4.1.3 Volatility Indicators

Volatility Indicators.

class `ta.volatility.`**`AverageTrueRange`**(*high: pandas.core.series.Series, low: pandas.core.series.Series, close: pandas.core.series.Series, n: int = 14, fillna: bool = False*)

Average True Range (ATR)

The indicator provide an indication of the degree of price volatility. Strong moves, in either direction, are often accompanied by large ranges, or large True Ranges.

http://stockcharts.com/school/doku.php?id=chart_school:technical_indicators:average_true_range_atr

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

`average_true_range`() → pandas.core.series.Series

> Average True Range (ATR)
>
>   **Returns** New feature generated.
>
>   **Return type** pandas.Series

class ta.volatility.**BollingerBands**(*close: pandas.core.series.Series, n: int = 20, ndev: int = 2, fillna: bool = False*)

> Bollinger Bands
>
> https://school.stockcharts.com/doku.php?id=technical_indicators:bollinger_bands
>
> > **Parameters**
> >
> > - **close** (*pandas.Series*) – dataset 'Close' column.
> >
> > - **n** (*int*) – n period.
> >
> > - **ndev** (*int*) – n factor standard deviation
> >
> > - **fillna** (*bool*) – if True, fill nan values.
>
> **bollinger_hband**() → pandas.core.series.Series
>
> > Bollinger Channel High Band
> >
> > > **Returns** New feature generated.
> > >
> > > **Return type** pandas.Series
>
> **bollinger_hband_indicator**() → pandas.core.series.Series
>
> > Bollinger Channel Indicator Crossing High Band (binary).
> >
> > It returns 1, if close is higher than bollinger_hband. Else, it returns 0.
> >
> > > **Returns** New feature generated.
> > >
> > > **Return type** pandas.Series
>
> **bollinger_lband**() → pandas.core.series.Series
>
> > Bollinger Channel Low Band
> >
> > > **Returns** New feature generated.
> > >
> > > **Return type** pandas.Series
>
> **bollinger_lband_indicator**() → pandas.core.series.Series
>
> > Bollinger Channel Indicator Crossing Low Band (binary).
> >
> > It returns 1, if close is lower than bollinger_lband. Else, it returns 0.
> >
> > > **Returns** New feature generated.
> > >
> > > **Return type** pandas.Series
>
> **bollinger_mavg**() → pandas.core.series.Series
>
> > Bollinger Channel Middle Band
> >
> > > **Returns** New feature generated.
> > >
> > > **Return type** pandas.Series
>
> **bollinger_pband**() → pandas.core.series.Series
>
> > Bollinger Channel Percentage Band
> >
> > From: https://school.stockcharts.com/doku.php?id=technical_indicators:bollinger_band_perce
> >
> > > **Returns** New feature generated.
> > >
> > > **Return type** pandas.Series
>
> **bollinger_wband**() → pandas.core.series.Series
>
> > Bollinger Channel Band Width
> >
> > From: https://school.stockcharts.com/doku.php?id=technical_indicators:bollinger_band_width

> **Returns** New feature generated.
>
> **Return type** pandas.Series

class `ta.volatility.`**`DonchianChannel`**(*close: pandas.core.series.Series*, *n: int = 20*, *fillna: bool = False*)

Donchian Channel

https://www.investopedia.com/terms/d/donchianchannels.asp

> **Parameters**
>
> - **close** (*pandas.Series*) – dataset 'Close' column.
>
> - **n** (*int*) – n period.
>
> - **ndev** (*int*) – n factor standard deviation
>
> - **fillna** (*bool*) – if True, fill nan values.

`donchian_channel_hband`() → pandas.core.series.Series

Donchian Channel High Band

> **Returns** New feature generated.
>
> **Return type** pandas.Series

`donchian_channel_hband_indicator`() → pandas.core.series.Series

Donchian Channel Indicator Crossing High Band (binary)

It returns 1, if close is higher than donchian_channel_hband. Else, it returns 0.

> **Returns** New feature generated.
>
> **Return type** pandas.Series

`donchian_channel_lband`() → pandas.core.series.Series

Donchian Channel Low Band

> **Returns** New feature generated.
>
> **Return type** pandas.Series

`donchian_channel_lband_indicator`() → pandas.core.series.Series

Donchian Channel Indicator Crossing Low Band (binary)

It returns 1, if close is lower than donchian_channel_lband. Else, it returns 0.

> **Returns** New feature generated.
>
> **Return type** pandas.Series

class `ta.volatility.`**`KeltnerChannel`**(*high: pandas.core.series.Series*, *low: pandas.core.series.Series*, *close: pandas.core.series.Series*, *n: int = 14*, *fillna: bool = False*, *ov: bool = True*)

Keltner Channels are a trend following indicator used to identify reversals with channel breakouts and channel direction. Channels can also be used to identify overbought and oversold levels when the trend is flat.

https://school.stockcharts.com/doku.php?id=technical_indicators:keltner_channels

> **Parameters**
>
> - **high** (*pandas.Series*) – dataset 'High' column.
>
> - **low** (*pandas.Series*) – dataset 'Low' column.
>
> - **close** (*pandas.Series*) – dataset 'Close' column.
>
> - **n** (*int*) – n period.

- **fillna** (*bool*) – if True, fill nan values.

- **ov** (*bool*) – if True, use original version as the centerline (SMA of typical price) if False, use EMA of close as the centerline. More info: https://school.stockcharts.com/doku.php?id=technical_indicators:keltner_channels

keltner_channel_hband() → pandas.core.series.Series
 Keltner Channel High Band

  Returns New feature generated.

  Return type pandas.Series

keltner_channel_hband_indicator() → pandas.core.series.Series
 Keltner Channel Indicator Crossing High Band (binary)

 It returns 1, if close is higher than keltner_channel_hband. Else, it returns 0.

  Returns New feature generated.

  Return type pandas.Series

keltner_channel_lband() → pandas.core.series.Series
 Keltner Channel Low Band

  Returns New feature generated.

  Return type pandas.Series

keltner_channel_lband_indicator() → pandas.core.series.Series
 Keltner Channel Indicator Crossing Low Band (binary)

 It returns 1, if close is lower than keltner_channel_lband. Else, it returns 0.

  Returns New feature generated.

  Return type pandas.Series

keltner_channel_mband() → pandas.core.series.Series
 Keltner Channel Middle Band

  Returns New feature generated.

  Return type pandas.Series

keltner_channel_pband() → pandas.core.series.Series
 Keltner Channel Percentage Band

  Returns New feature generated.

  Return type pandas.Series

keltner_channel_wband() → pandas.core.series.Series
 Keltner Channel Band Width

  Returns New feature generated.

  Return type pandas.Series

ta.volatility.**average_true_range**(*high*, *low*, *close*, *n=14*, *fillna=False*)
 Average True Range (ATR)

The indicator provide an indication of the degree of price volatility. Strong moves, in either direction, are often accompanied by large ranges, or large True Ranges.

http://stockcharts.com/school/doku.php?id=chart_school:technical_indicators:average_true_range_atr

 Parameters

- **high** (*pandas.Series*) – dataset 'High' column.

- **low** (*pandas.Series*) – dataset 'Low' column.

- **close** (*pandas.Series*) – dataset 'Close' column.

- **n** (*int*) – n period.

- **fillna** (*bool*) – if True, fill nan values.

> **Returns** New feature generated.

> **Return type** pandas.Series

`ta.volatility.`**`bollinger_hband`**(*close*, *n=20*, *ndev=2*, *fillna=False*)
> Bollinger Bands (BB)

> Upper band at K times an N-period standard deviation above the moving average (MA + Kdeviation).

> https://en.wikipedia.org/wiki/Bollinger_Bands

> **Parameters**

>> - **close** (*pandas.Series*) – dataset 'Close' column.

>> - **n** (*int*) – n period.

>> - **ndev** (*int*) – n factor standard deviation

>> - **fillna** (*bool*) – if True, fill nan values.

> **Returns** New feature generated.

> **Return type** pandas.Series

`ta.volatility.`**`bollinger_hband_indicator`**(*close*, *n=20*, *ndev=2*, *fillna=False*)
> Bollinger High Band Indicator

> Returns 1, if close is higher than bollinger high band. Else, return 0.

> https://en.wikipedia.org/wiki/Bollinger_Bands

> **Parameters**

>> - **close** (*pandas.Series*) – dataset 'Close' column.

>> - **n** (*int*) – n period.

>> - **ndev** (*int*) – n factor standard deviation

>> - **fillna** (*bool*) – if True, fill nan values.

> **Returns** New feature generated.

> **Return type** pandas.Series

`ta.volatility.`**`bollinger_lband`**(*close*, *n=20*, *ndev=2*, *fillna=False*)
> Bollinger Bands (BB)

> Lower band at K times an N-period standard deviation below the moving average (MA Kdeviation).

> https://en.wikipedia.org/wiki/Bollinger_Bands

> **Parameters**

>> - **close** (*pandas.Series*) – dataset 'Close' column.

>> - **n** (*int*) – n period.

>> - **ndev** (*int*) – n factor standard deviation

- **fillna** (*bool*) – if True, fill nan values.

> **Returns** New feature generated.

> **Return type** pandas.Series

`ta.volatility.`**`bollinger_lband_indicator`**(*close*, *n=20*, *ndev=2*, *fillna=False*)
> Bollinger Low Band Indicator

> Returns 1, if close is lower than bollinger low band. Else, return 0.

> https://en.wikipedia.org/wiki/Bollinger_Bands

> > **Parameters**

> > - **close** (*pandas.Series*) – dataset 'Close' column.

> > - **n** (*int*) – n period.

> > - **ndev** (*int*) – n factor standard deviation

> > - **fillna** (*bool*) – if True, fill nan values.

> > **Returns** New feature generated.

> > **Return type** pandas.Series

`ta.volatility.`**`bollinger_mavg`**(*close*, *n=20*, *fillna=False*)
> Bollinger Bands (BB)

> N-period simple moving average (MA).

> https://en.wikipedia.org/wiki/Bollinger_Bands

> > **Parameters**

> > - **close** (*pandas.Series*) – dataset 'Close' column.

> > - **n** (*int*) – n period.

> > - **fillna** (*bool*) – if True, fill nan values.

> > **Returns** New feature generated.

> > **Return type** pandas.Series

`ta.volatility.`**`bollinger_pband`**(*close*, *n=20*, *ndev=2*, *fillna=False*)
> Bollinger Channel Percentage Band

> From: https://school.stockcharts.com/doku.php?id=technical_indicators:bollinger_band_perce

> > **Parameters**

> > - **close** (*pandas.Series*) – dataset 'Close' column.

> > - **n** (*int*) – n period.

> > - **ndev** (*int*) – n factor standard deviation

> > - **fillna** (*bool*) – if True, fill nan values.

> > **Returns** New feature generated.

> > **Return type** pandas.Series

`ta.volatility.`**`bollinger_wband`**(*close*, *n=20*, *ndev=2*, *fillna=False*)
> Bollinger Channel Band Width

> From: https://school.stockcharts.com/doku.php?id=technical_indicators:bollinger_band_width

> **Parameters**
>
> > - **close** (*pandas.Series*) – dataset 'Close' column.
> >
> > - **n** (*int*) – n period.
> >
> > - **ndev** (*int*) – n factor standard deviation
> >
> > - **fillna** (*bool*) – if True, fill nan values.
>
> **Returns** New feature generated.
>
> **Return type** pandas.Series

ta.volatility.**donchian_channel_hband**(*close*, *n=20*, *fillna=False*)
> Donchian channel (DC)

> The upper band marks the highest price of an issue for n periods.

> https://www.investopedia.com/terms/d/donchianchannels.asp

> **Parameters**
>
> > - **close** (*pandas.Series*) – dataset 'Close' column.
> >
> > - **n** (*int*) – n period.
> >
> > - **fillna** (*bool*) – if True, fill nan values.
>
> **Returns** New feature generated.
>
> **Return type** pandas.Series

ta.volatility.**donchian_channel_hband_indicator**(*close*, *n=20*, *fillna=False*)
> Donchian High Band Indicator

> Returns 1, if close is higher than donchian high band channel. Else, return 0.

> https://www.investopedia.com/terms/d/donchianchannels.asp

> **Parameters**
>
> > - **close** (*pandas.Series*) – dataset 'Close' column.
> >
> > - **n** (*int*) – n period.
> >
> > - **fillna** (*bool*) – if True, fill nan values.
>
> **Returns** New feature generated.
>
> **Return type** pandas.Series

ta.volatility.**donchian_channel_lband**(*close*, *n=20*, *fillna=False*)
> Donchian channel (DC)

> The lower band marks the lowest price for n periods.

> https://www.investopedia.com/terms/d/donchianchannels.asp

> **Parameters**
>
> > - **close** (*pandas.Series*) – dataset 'Close' column.
> >
> > - **n** (*int*) – n period.
> >
> > - **fillna** (*bool*) – if True, fill nan values.
>
> **Returns** New feature generated.
>
> **Return type** pandas.Series

`ta.volatility.`**`donchian_channel_lband_indicator`**(*close*, *n=20*, *fillna=False*)
  Donchian Low Band Indicator

  Returns 1, if close is lower than donchian low band channel. Else, return 0.

  https://www.investopedia.com/terms/d/donchianchannels.asp

  > **Parameters**
  >
  > - **close** (*pandas.Series*) – dataset 'Close' column.
  >
  > - **n** (*int*) – n period.
  >
  > - **fillna** (*bool*) – if True, fill nan values.
  >
  > **Returns** New feature generated.
  >
  > **Return type** pandas.Series

`ta.volatility.`**`keltner_channel_hband`**(*high*, *low*, *close*, *n=10*, *fillna=False*, *ov=True*)
  Keltner channel (KC)

  Showing a simple moving average line (high) of typical price.

  https://en.wikipedia.org/wiki/Keltner_channel

  > **Parameters**
  >
  > - **high** (*pandas.Series*) – dataset 'High' column.
  >
  > - **low** (*pandas.Series*) – dataset 'Low' column.
  >
  > - **close** (*pandas.Series*) – dataset 'Close' column.
  >
  > - **n** (*int*) – n period.
  >
  > - **fillna** (*bool*) – if True, fill nan values.
  >
  > **Returns** New feature generated.
  >
  > **Return type** pandas.Series

`ta.volatility.`**`keltner_channel_hband_indicator`**(*high*, *low*, *close*, *n=10*, *fillna=False*, *ov=True*)
  Keltner Channel High Band Indicator (KC)

  Returns 1, if close is higher than keltner high band channel. Else, return 0.

  https://en.wikipedia.org/wiki/Keltner_channel

  > **Parameters**
  >
  > - **high** (*pandas.Series*) – dataset 'High' column.
  >
  > - **low** (*pandas.Series*) – dataset 'Low' column.
  >
  > - **close** (*pandas.Series*) – dataset 'Close' column.
  >
  > - **n** (*int*) – n period.
  >
  > - **fillna** (*bool*) – if True, fill nan values.
  >
  > **Returns** New feature generated.
  >
  > **Return type** pandas.Series

`ta.volatility.`**`keltner_channel_lband`**(*high*, *low*, *close*, *n=10*, *fillna=False*, *ov=True*)
  Keltner channel (KC)

  Showing a simple moving average line (low) of typical price.

---

https://en.wikipedia.org/wiki/Keltner_channel

> **Parameters**
>
> - **high** (*pandas.Series*) – dataset 'High' column.
> - **low** (*pandas.Series*) – dataset 'Low' column.
> - **close** (*pandas.Series*) – dataset 'Close' column.
> - **n** (*int*) – n period.
> - **fillna** (*bool*) – if True, fill nan values.
>
> **Returns** New feature generated.
>
> **Return type** pandas.Series

ta.volatility.**keltner_channel_lband_indicator**(*high*, *low*, *close*, *n=10*, *fillna=False*, *ov=True*)

> Keltner Channel Low Band Indicator (KC)

Returns 1, if close is lower than keltner low band channel. Else, return 0.

https://en.wikipedia.org/wiki/Keltner_channel

> **Parameters**
>
> - **high** (*pandas.Series*) – dataset 'High' column.
> - **low** (*pandas.Series*) – dataset 'Low' column.
> - **close** (*pandas.Series*) – dataset 'Close' column.
> - **n** (*int*) – n period.
> - **fillna** (*bool*) – if True, fill nan values.
>
> **Returns** New feature generated.
>
> **Return type** pandas.Series

ta.volatility.**keltner_channel_mband**(*high*, *low*, *close*, *n=10*, *fillna=False*, *ov=True*)

> Keltner channel (KC)

Showing a simple moving average line (central) of typical price.

https://en.wikipedia.org/wiki/Keltner_channel

> **Parameters**
>
> - **high** (*pandas.Series*) – dataset 'High' column.
> - **low** (*pandas.Series*) – dataset 'Low' column.
> - **close** (*pandas.Series*) – dataset 'Close' column.
> - **n** (*int*) – n period.
> - **fillna** (*bool*) – if True, fill nan values.
>
> **Returns** New feature generated.
>
> **Return type** pandas.Series

ta.volatility.**keltner_channel_pband**(*high*, *low*, *close*, *n=10*, *fillna=False*, *ov=True*)

> Keltner Channel Percentage Band

https://en.wikipedia.org/wiki/Keltner_channel

> **Parameters**

- **high** (*pandas.Series*) – dataset 'High' column.

- **low** (*pandas.Series*) – dataset 'Low' column.

- **close** (*pandas.Series*) – dataset 'Close' column.

- **n** (*int*) – n period.

- **fillna** (*bool*) – if True, fill nan values.

> **Returns** New feature generated.

> **Return type** pandas.Series

`ta.volatility.`**`keltner_channel_wband`**(*high*, *low*, *close*, *n=10*, *fillna=False*, *ov=True*)
Keltner Channel Band Width

https://en.wikipedia.org/wiki/Keltner_channel

> **Parameters**

- **high** (*pandas.Series*) – dataset 'High' column.

- **low** (*pandas.Series*) – dataset 'Low' column.

- **close** (*pandas.Series*) – dataset 'Close' column.

- **n** (*int*) – n period.

- **fillna** (*bool*) – if True, fill nan values.

> **Returns** New feature generated.

> **Return type** pandas.Series

4.1.4 Trend Indicators

Trend Indicators.

class `ta.trend.`**`ADXIndicator`**(*high: pandas.core.series.Series*, *low: pandas.core.series.Series*, *close: pandas.core.series.Series*, *n: 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.

- **n** (*int*) – n period.

- **fillna** (*bool*) – if True, fill nan values.

adx() → pandas.core.series.Series
> Average Directional Index (ADX)

>> **Returns** New feature generated.

>> **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

class ta.trend.**AroonIndicator**(*close: pandas.core.series.Series*, *n: 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.

>>> • **n** (*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

class ta.trend.**CCIIndicator**(*high: pandas.core.series.Series*, *low: pandas.core.series.Series*, *close: pandas.core.series.Series*, *n: int = 20*, *c: 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.
>
> - **n** (*int*) – n period.
>
> - **c** (*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*, *n: 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.
>
> - **n** (*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.**EMAIndicator**(*close: pandas.core.series.Series*, *n: int = 14*, *fillna: bool = False*)
EMA - Exponential Moving Average

> **Parameters**
>
> - **close** (*pandas.Series*) – dataset 'Close' column.
>
> - **n** (*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.**IchimokuIndicator**(*high: pandas.core.series.Series*, *low: pandas.core.series.Series*, *n1: int = 9*, *n2: int = 26*, *n3: 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.
>> - **n1** (*int*) – n1 low period.
>> - **n2** (*int*) – n2 medium period.
>> - **n3** (*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

class ta.trend.**KSTIndicator**(*close: pandas.core.series.Series, r1: int = 10, r2: int = 15, r3: int = 20, r4: int = 30, n1: int = 10, n2: int = 10, n3: int = 10, n4: int = 15, nsig: int = 9, fillna: bool = 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.
>> - **r1** (*int*) – r1 period.
>> - **r2** (*int*) – r2 period.
>> - **r3** (*int*) – r3 period.
>> - **r4** (*int*) – r4 period.
>> - **n1** (*int*) – n1 smoothed period.
>> - **n2** (*int*) – n2 smoothed period.
>> - **n3** (*int*) – n3 smoothed period.
>> - **n4** (*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

class ta.trend.**MACD**(*close: pandas.core.series.Series, n_slow: int = 26, n_fast: int = 12, n_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.

>>> • **n_fast** (*int*) – n period short-term.

>>> • **n_slow** (*int*) – n period long-term.

>>> • **n_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

class ta.trend.**MassIndex**(*high: pandas.core.series.Series, low: pandas.core.series.Series, n: int = 9, n2: 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.
>
> - **n** (*int*) – n low period.
>
> - **n2** (*int*) – n high period.
>
> - **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.

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
>
> > **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

> > **Returns** New feature generated.

> > **Return type** pandas.Series

class ta.trend.**SMAIndicator**(*close: pandas.core.series.Series*, *n: int*, *fillna: bool = False*)
> SMA - Simple Moving Average

> > **Parameters**

> > > • **close** (*pandas.Series*) – dataset 'Close' column.

> > > • **n** (*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.**TRIXIndicator**(*close: pandas.core.series.Series*, *n: 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.

> > > • **n** (*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*, *n: 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.

> > > • **n** (*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.**adx**(*high*, *low*, *close*, *n=14*, *fillna=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.
> - **n** (*int*) – n period.
> - **fillna** (*bool*) – if True, fill nan values.
>
> **Returns** New feature generated.
>
> **Return type** pandas.Series

ta.trend.**adx_neg**(*high*, *low*, *close*, *n=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.

- **n** (*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*, *n=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.

- **n** (*int*) – n period.

- **fillna** (*bool*) – if True, fill nan values.

Returns New feature generated.

Return type pandas.Series

ta.trend.**aroon_down**(*close*, *n=25*, *fillna=False*)

Aroon 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.

- **n** (*int*) – n period.

- **fillna** (*bool*) – if True, fill nan values.

Returns New feature generated.

Return type pandas.Series

ta.trend.**aroon_up**(*close*, *n=25*, *fillna=False*)

Aroon 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.
>
> - **n** (*int*) – n period.
>
> - **fillna** (*bool*) – if True, fill nan values.
>
> **Returns** New feature generated.
>
> **Return type** pandas.Series

ta.trend.**cci**(*high*, *low*, *close*, *n=20*, *c=0.015*, *fillna=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.
>
> - **n** (*int*) – n periods.
>
> - **c** (*int*) – constant.
>
> - **fillna** (*bool*) – if True, fill nan values.
>
> **Returns** New feature generated.
>
> **Return type** pandas.Series

ta.trend.**dpo**(*close*, *n=20*, *fillna=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.
>
> - **n** (*int*) – n period.
>
> - **fillna** (*bool*) – if True, fill nan values.
>
> **Returns** New feature generated.
>
> **Return type** pandas.Series

ta.trend.**ema_indicator**(*close*, *n=12*, *fillna=False*)
: Exponential Moving Average (EMA)

> **Returns** New feature generated.
>
> **Return type** pandas.Series

`ta.trend.`**`ichimoku_a`**(*high*, *low*, *n1=9*, *n2=26*, *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.
> - **n1** (*int*) – n1 low period.
> - **n2** (*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_b`**(*high*, *low*, *n2=26*, *n3=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.
> - **n2** (*int*) – n2 medium period.
> - **n3** (*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.`**`kst`**(*close*, *r1=10*, *r2=15*, *r3=20*, *r4=30*, *n1=10*, *n2=10*, *n3=10*, *n4=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.
> - **r1** (*int*) – r1 period.
> - **r2** (*int*) – r2 period.
> - **r3** (*int*) – r3 period.
> - **r4** (*int*) – r4 period.

- **n1** (*int*) – n1 smoothed period.

- **n2** (*int*) – n2 smoothed period.

- **n3** (*int*) – n3 smoothed period.

- **n4** (*int*) – n4 smoothed period.

- **fillna** (*bool*) – if True, fill nan values.

Returns New feature generated.

Return type pandas.Series

ta.trend.**kst_sig**(*close*, *r1=10*, *r2=15*, *r3=20*, *r4=30*, *n1=10*, *n2=10*, *n3=10*, *n4=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.

- **r1** (*int*) – r1 period.

- **r2** (*int*) – r2 period.

- **r3** (*int*) – r3 period.

- **r4** (*int*) – r4 period.

- **n1** (*int*) – n1 smoothed period.

- **n2** (*int*) – n2 smoothed period.

- **n3** (*int*) – n3 smoothed period.

- **n4** (*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

ta.trend.**macd**(*close*, *n_slow=26*, *n_fast=12*, *fillna=False*)
Moving Average Convergence Divergence (MACD)

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

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

Parameters

- **close** (*pandas.Series*) – dataset 'Close' column.

- **n_fast** (*int*) – n period short-term.

- **n_slow** (*int*) – n period long-term.

- **fillna** (*bool*) – if True, fill nan values.

Returns New feature generated.

> **Return type** pandas.Series

ta.trend.**macd_diff**(*close*, *n_slow=26*, *n_fast=12*, *n_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.
> > - **n_fast** (*int*) – n period short-term.
> > - **n_slow** (*int*) – n period long-term.
> > - **n_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*, *n_slow=26*, *n_fast=12*, *n_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.
> > - **n_fast** (*int*) – n period short-term.
> > - **n_slow** (*int*) – n period long-term.
> > - **n_sign** (*int*) – n period to signal.
> > - **fillna** (*bool*) – if True, fill nan values.
> >
> > **Returns** New feature generated.
> >
> > **Return type** pandas.Series

ta.trend.**mass_index**(*high*, *low*, *n=9*, *n2=25*, *fillna=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.
> > - **n** (*int*) – n low period.
> > - **n2** (*int*) – n high period.
> > - **fillna** (*bool*) – if True, fill nan values.
> >
> > **Returns** New feature generated.

Return type pandas.Series

ta.trend.**psar_down**(*high*, *low*, *close*, *step=0.02*, *max_step=0.2*)

Parabolic Stop and Reverse (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.

Returns New feature generated.

Return type pandas.Series

ta.trend.**psar_down_indicator**(*high*, *low*, *close*, *step=0.02*, *max_step=0.2*)

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.

Returns New feature generated.

Return type pandas.Series

ta.trend.**psar_up**(*high*, *low*, *close*, *step=0.02*, *max_step=0.2*)

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.

Returns New feature generated.

Return type pandas.Series

`ta.trend.`**`psar_up_indicator`**(*high*, *low*, *close*, *step=0.02*, *max_step=0.2*)
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.
>
> **Returns** New feature generated.
>
> **Return type** pandas.Series

`ta.trend.`**`sma_indicator`**(*close*, *n=12*, *fillna=False*)
Simple Moving Average (SMA)

> **Returns** New feature generated.
>
> **Return type** pandas.Series

`ta.trend.`**`trix`**(*close*, *n=15*, *fillna=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.
> - **n** (*int*) – n period.
> - **fillna** (*bool*) – if True, fill nan values.
>
> **Returns** New feature generated.
>
> **Return type** pandas.Series

`ta.trend.`**`vortex_indicator_neg`**(*high*, *low*, *close*, *n=14*, *fillna=False*)
Vortex Indicator (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.
> - **n** (*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*, *n=14*, *fillna=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.
> > - **n** (*int*) – n period.
> > - **fillna** (*bool*) – if True, fill nan values.
> >
> > **Returns** New feature generated.
> >
> > **Return type** pandas.Series

4.1.5 Others Indicators

Others Indicators.

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

> Cumulative Return (CR)

> > **Parameters**
> >
> > - **close** (*pandas.Series*) – dataset 'Close' column.
> > - **fillna** (*bool*) – if True, fill nan values.

> **cumulative_return**() → pandas.core.series.Series

> > Cumulative Return (CR)

> > > **Returns** New feature generated.
> > >
> > > **Return type** pandas.Series

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

> Daily Log Return (DLR)

> https://stackoverflow.com/questions/31287552/logarithmic-returns-in-pandas-dataframe

> > **Parameters**
> >
> > - **close** (*pandas.Series*) – dataset 'Close' column.
> > - **fillna** (*bool*) – if True, fill nan values.

> **daily_log_return**() → pandas.core.series.Series

> > Daily Log Return (DLR)

> > > **Returns** New feature generated.
> > >
> > > **Return type** pandas.Series

class `ta.others.``DailyReturnIndicator`(*close: pandas.core.series.Series*, *fillna: bool = False*)

    Daily Return (DR)

        Parameters

            • **close** (*pandas.Series*) – dataset 'Close' column.

            • **fillna** (*bool*) – if True, fill nan values.

    `daily_return`() → pandas.core.series.Series

        Daily Return (DR)

            Returns New feature generated.

            Return type pandas.Series

`ta.others.``cumulative_return`(*close*, *fillna=False*)

    Cumulative Return (CR)

        Parameters

            • **close** (*pandas.Series*) – dataset 'Close' column.

            • **fillna** (*bool*) – if True, fill nan values.

        Returns New feature generated.

        Return type pandas.Series

`ta.others.``daily_log_return`(*close*, *fillna=False*)

    Daily Log Return (DLR)

    https://stackoverflow.com/questions/31287552/logarithmic-returns-in-pandas-dataframe

        Parameters

            • **close** (*pandas.Series*) – dataset 'Close' column.

            • **fillna** (*bool*) – if True, fill nan values.

        Returns New feature generated.

        Return type pandas.Series

`ta.others.``daily_return`(*close*, *fillna=False*)

    Daily Return (DR)

        Parameters

            • **close** (*pandas.Series*) – dataset 'Close' column.

            • **fillna** (*bool*) – if True, fill nan values.

        Returns New feature generated.

        Return type pandas.Series

INDICES AND TABLES

- genindex
- modindex
- search

PYTHON MODULE INDEX

m
momentum, 9

o
others, 49

t
ta, 9
ta.momentum, 9
ta.others, 49
ta.trend, 34
ta.volatility, 25
ta.volume, 17
trend, 34

v
volatility, 25
volume, 17

W