## Contents

1  Installation (python 3) 3
2  Examples 5
3  Motivation 7
4  Contents 9
   4.1  Documentation ................................................................. 9
5  Indices and tables 29

Python Module Index 31
Index 33
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 built on Python Pandas library.
Installation (python 3)

> virtualenv -p python3 virtualenvironment
> source virtualenvironment/bin/activate
> pip install ta
Adding all features:

```python
import pandas as pd
from ta import *

# Load data
df = pd.read_csv('your-file.csv', sep=',')

# Clean nan values
df = utils.dropna(df)

# Add all TA features filling Nans values
df = add_all_ta_features(df, "Open", "High", "Low", "Close", "Volume_BTC", fillna=True)
```

Adding individual features:

```python
import pandas as pd
from ta import *

# Load data
df = pd.read_csv('your-file.csv', sep=',')

# Clean nan values
df = utils.dropna(df)

# Add bollinger band high indicator filling Nans values
df['bb_high_indicator'] = bollinger_hband_indicator(df['Close'], n=20, ndev=2, fillna=True)

# Add bollinger band low indicator filling Nans values
df['bb_low_indicator'] = bollinger_lband_indicator(df['Close'], n=20, ndev=2, fillna=True)
```
CHAPTER 3

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-temporales-financieras-para-machine-learning-con-cb28f9427d0
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.

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.
• \texttt{low (pandas.Series)} – dataset ‘Low’ column.
• \texttt{s (int)} – short period
• \texttt{len (int)} – long period
• \texttt{fillna (bool)} – if True, fill nan values with -50.

\textbf{Returns} \hspace{1em} New feature generated.
\textbf{Return type} \hspace{1em} \texttt{pandas.Series}

\texttt{ta.momentum.kama} \hspace{1em} \texttt{(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.

\url{https://www.tradingview.com/ideas/kama/}

\textbf{Parameters}
• \texttt{close (pandas.Series)} – dataset ‘Close’ column
• \texttt{n (int)} – n number of periods for the efficiency ratio
• \texttt{pow1 (int)} – number of periods for the fastest EMA constant
• \texttt{pow2 (int)} – number of periods for the slowest EMA constant

\textbf{Returns} \hspace{1em} New feature generated.
\textbf{Return type} \hspace{1em} \texttt{pandas.Series}

\texttt{ta.momentum.money_flow_index} \hspace{1em} \texttt{(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.

\url{http://stockcharts.com/school/doku.php?id=chart_school:technical_indicators:money_flow_index_mfi}

\textbf{Parameters}
• \texttt{high (pandas.Series)} – dataset ‘High’ column.
• \texttt{low (pandas.Series)} – dataset ‘Low’ column.
• \texttt{close (pandas.Series)} – dataset ‘Close’ column.
• \texttt{volume (pandas.Series)} – dataset ‘Volume’ column.
• \texttt{n (int)} – n period.
• \texttt{fillna (bool)} – if True, fill nan values.

\textbf{Returns} \hspace{1em} New feature generated.
\textbf{Return type} \hspace{1em} \texttt{pandas.Series}

\texttt{ta.momentum.roc} \hspace{1em} \texttt{(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.


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, 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/stochasticsoscillator.asp

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.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

Parameters
- high (pandas.Series) – dataset ‘High’ column.
- low (pandas.Series) – dataset ‘Low’ column.
- close (pandas.Series) – dataset ‘Close’ column.
- s (int) – short period
Technical Analysis Library in Python Documentation, Release 0.1.4

- \( m \) (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

```python
ta.momentum.wr(high, low, close, lbp=14, fillna=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.

\[
%R = \frac{\text{Highest High} - \text{Close}}{\text{Highest High} - \text{Lowest Low}} \times -100
\]

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.

```python
ta.volume.acc_dist_index(high, low, close, volume, fillna=False)
```

Accumulation/Distribution Index (ADI)

Acting as leading indicator of price movements.
Chaikin Money Flow (CMF)

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`

```python
import ta

chaikin_money_flow = ta.volume.chaikin_money_flow(high, low, close, volume, n=20, fillna=False)
```

Ease of movement (EoM, EMV)

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`

```python
ease_of_movement = ta.volume.ease_of_movement(high, low, close, volume, n=20, fillna=False)
```
The force_index function from the Ta library is defined as:

```python
ta.volume.force_index(close, volume, n=2, fillna=False)
```

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.

- **Parameters**
  - `n` (int): n period.
  - `fillna` (bool): if True, fill nan values.

- **Returns** New feature generated.

The negative_volume_index function is defined as:

```python
ta.volume.negative_volume_index(close, volume, fillna=False)
```

This function calculates the Negative Volume Index (NVI) which indicates when the smart money is active. 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.

- **Parameters**
  - `fillna` (bool): if True, fill nan values with 1000.

- **Returns** New feature generated.

The on_balance_volume function is defined as:

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

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

- **Parameters**
  - `fillna` (bool): if True, fill nan values with 1000.

- **Returns** New feature generated.

See also:
- https://en.wikipedia.org/wiki/Negative_volume_index

4.1. Documentation
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.volume_price_trend(close, volume, fillna=False)
```
Volume-price trend (VPT)
Is based on a running cumulative volume that adds or subtracts 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.
• 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.

```
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.


Parameters

• high (pandas.Series) – dataset ‘High’ column.
• low (pandas.Series) – dataset ‘Low’ column.
• close (pandas.Series) – dataset ‘Close’ column.
• n (int) – n period.

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).
Parameters
- close (pandas.Series) – dataset ‘Close’ column.
- n (int) – n period.
- ndev (int) – n factor standard deviation

Returns New feature generated.

Return type pandas.Series

```python
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

Returns New feature generated.

Return type pandas.Series

```python
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

Returns New feature generated.

Return type pandas.Series

```python
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

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.

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.

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.

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.

Returns New feature generated.

Return type pandas.Series
**Donchian Channel 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.

**Returns** New feature generated.

**Return type** pandas.Series

---

**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.

**Returns** New feature generated.

**Return type** pandas.Series

---

**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.

Returns New feature generated.

Return type pandas.Series
ta.volatility.keltner_channel_lband(high, low, close, n=10, fillna=False)
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.

Returns New feature generated.

Return type pandas.Series
ta.volatility.keltner_channel_lband_indicator(high, low, close, n=10, fillna=False)
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.

Returns New feature generated.

Return type pandas.Series

4.1.4 Trend Indicators

Trend Indicators.
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.
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.

```
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.
• 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.


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

```python
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.


**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

```python
ta.trend.ema_indicator(close, n=12,fillna=False)
```

EMA

Exponential Moving Average via Pandas

**Parameters**

- close (pandas.Series) – dataset ‘Close’ column.
- n_fast (int) – n period short-term.
- fillna (bool) – if True, fill nan values.

**Returns**  New feature generated.

**Return type**  pandas.Series

```python
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.


**Parameters**

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

**Returns**  New feature generated.

**Return type**  pandas.Series

```python
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.


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

**Returns** New feature generated.

**Return type** *pandas.Series*

```python
from ta.trend import kst
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*

```python
from ta.trend import kst_sig
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.


**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

```python
import ta
ta.trend.macd(close, n_fast=12, n_slow=26, 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

```python
import ta
ta.trend.macd_diff(close, n_fast=12, n_slow=26, 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

```python
import ta
ta.trend.macd_signal(close, n_fast=12, n_slow=26, 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.

---

4.1. Documentation
• **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

```python
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.


**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

```python
ta.trend.trix(close, n=15, fillna=False)
```

**Trix (TRIX)**

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


**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

```python
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.


**Parameters**

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

**Returns**
New feature generated.

**Return type**
pandas.Series

```python
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.


**Parameters**

• \( \text{high} \) (pandas.Series) – dataset ‘High’ column.
• \( \text{low} \) (pandas.Series) – dataset ‘Low’ column.
• \( \text{close} \) (pandas.Series) – dataset ‘Close’ column.
• \( n \) (int) – n period.
• \( \text{fillna} \) (bool) – if True, fill nan values.

**Returns**
New feature generated.

**Return type**
pandas.Series

### 4.1.5 Others Indicators

Others Indicators.

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

Cumulative Return (CR)

**Parameters**

• \( \text{close} \) (pandas.Series) – dataset ‘Close’ column.
• \( \text{fillna} \) (bool) – if True, fill nan values.

**Returns**
New feature generated.

**Return type**
pandas.Series

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

Daily Log Return (DLR)


**Parameters**

• \( \text{close} \) (pandas.Series) – dataset ‘Close’ column.
• \( \text{fillna} \) (bool) – if True, fill nan values.

**Returns**
New feature generated.

**Return type**
pandas.Series

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

Daily Return (DR)
• `close (pandas.Series)` – dataset ‘Close’ column.
• `fillna (bool)` – if True, fill nan values.

**Returns**  New feature generated.

**Return type**  pandas.Series
CHAPTER 5

Indices and tables

- genindex
- modindex
- search
Python Module Index

m
momentum, 9

o
others, 27

t
  ta, 9
  ta.momentum, 9
  ta.others, 27
  ta.trend, 20
  ta.volatility, 16
  ta.volume, 13
  trend, 20

v
  volatility, 16
  volume, 13
Index

A
acc_dist_index() (in module ta.volume), 13
adx() (in module ta.trend), 20
adx_neg() (in module ta.trend), 21
adx_pos() (in module ta.trend), 21
ao() (in module ta.momentum), 9
aroon_down() (in module ta.trend), 22
aroon_up() (in module ta.trend), 22
average_true_range() (in module ta.volatility), 16

B
bollinger_hband() (in module ta.volatility), 16
bollinger_hband_indicator() (in module ta.volatility), 17
bollinger_lband() (in module ta.volatility), 17
bollinger_lband_indicator() (in module ta.volatility), 17
bollinger_mavg() (in module ta.volatility), 17

ccci() (in module ta.trend), 22
chaikin_money_flow() (in module ta.volume), 14
cumulative_return() (in module ta.others), 27

D
daily_log_return() (in module ta.others), 27
daily_return() (in module ta.others), 27
donchian_channel_hband() (in module ta.volatility), 18
donchian_channel_hband_indicator() (in module ta.volatility), 18
donchian_channel_lband() (in module ta.volatility), 18
donchian_channel_lband_indicator() (in module ta.volatility), 18
dpo() (in module ta.trend), 23

E
ease_of_movement() (in module ta.volume), 14

ema_indicator() (in module ta.trend), 23

F
force_index() (in module ta.volume), 14

I
ichimoku_a() (in module ta.trend), 23
ichimoku_b() (in module ta.trend), 23

K
kama() (in module ta.momentum), 10
keltner_channel_central() (in module ta.volatility), 19
keltner_channel_hband() (in module ta.volatility), 19
keltner_channel_hband_indicator() (in module ta.volatility), 19
keltner_channel_lband() (in module ta.volatility), 20
keltner_channel_lband_indicator() (in module ta.volatility), 20
kst() (in module ta.trend), 24
kst_sig() (in module ta.trend), 24

M
macd() (in module ta.trend), 25
macd_diff() (in module ta.trend), 25
macd_signal() (in module ta.trend), 25
mass_index() (in module ta.trend), 26
momentum (module), 9
money_flow_index() (in module ta.momentum), 10

N
negative_volume_index() (in module ta.volume), 15

O
on_balance_volume() (in module ta.volume), 15
others (module), 27
put_call_ratio() (in module ta.volume), 16

roc() (in module ta.momentum), 10
rsi() (in module ta.momentum), 11

stoch() (in module ta.momentum), 11
stoch_signal() (in module ta.momentum), 11

ta (module), 9
ta.momentum (module), 9
ta.others (module), 27
ta.trend (module), 20
ta.volatility (module), 16

trix() (in module ta.trend), 26
tsi() (in module ta.momentum), 12

uo() (in module ta.momentum), 12

volatility (module), 16
volume (module), 13
volume_price_trend() (in module ta.volume), 16
vortex_indicator_neg() (in module ta.trend), 26
vortex_indicator_pos() (in module ta.trend), 27

wr() (in module ta.momentum), 13