**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: pandas.core.series.Series*, *volume: pandas.core.series.Series*, *window: 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.
* **window** (*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: pandas.core.series.Series*, *window: int = 14*, *fillna: bool = False*)**

## Ease of movement (EoM, EMV)

It relates 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.
* **window** (*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*, *window: 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**

* **close** (*pandas.Series*) – dataset ‘Close’ column.
* **volume** (*pandas.Series*) – dataset ‘Volume’ column.
* **window** (*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*, *window: 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.
* **window** (*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.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*, *window: 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.
* **window** (*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

***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>

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.

**negative\_volume\_index() → pandas.core.series.Series**

Negative Volume Index (NVI)

**Returns**

New feature generated.

**Return type:** pandas.Series

# 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.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