

TechAnalysis

Cały zestaw modułów, klas i funkcji, które odpowiadają za analizę techniczną.

Wymagania systemowe :

- Zainstalowany Python 2.7.2
- Dołączona biblioteka pyNum w wersji 1.6.1

Moduł trendAnalysis

Moduł do analizy technicznej, zawiera funkcje wykrywające trend, oraz wszystkie formacje, które bazują na trendzie, czyli:

- formacja głowy i ramion,
- formacja odwróconej głowy i ramion
- formacja potrójnego szczytu/dna
- formacja rogu
- formacje geometryczne (trójkąty, prostokąty, kliny)
- flagi
- chorągiewki.

Ponadto wszystkie funkcje do znajdowania formacji zwracają moc sygnału w zależności od wielu czynników jak czas kształtowania się formacji, ilość cech charakterystycznych, zgodność z modelem książkowym czy odległość w czasie.

Atrybuty:

rectVul = 0.03 - o ile procent mogą odbiegać od siebie szczytowe wartości lewego i prawego ramienia w formacji głowy i ramion (HS)

hsVul = 0.1 - na ile podzbiorów dzielimy tablice przy wyznaczaniu linii kanału

div = 8 - na ile podzbiorów dzielimy tablice przy wyznaczaniu formacji HS

hsDiv = 12 - ile razy robimy przesunięcie przy wyznaczaniu formacji HS w celu uniknięcia pechowego podziału

dateVul = 6 - jaka ma być minimalna różnica procentowa między szczytem w głowie a ramionami, formacja HS

hsDiff = 0.03 - jaka ma być maksymalna różnica procentowa między kolejnymi szczytami w formacjach potrójnego szczytu i dna (triple)

tripleDiff = 0.03 - na ile podzbiorów dzielimy tablice przy wyznaczaniu formacji potrójnego szczytu

tripleDiv = 12 - jak długi może być okres formowania się formacji V w stosunku do badanego okresu

hornVul = 0.1 - jak bardzo lewy wierzchołek formacji rogu (horn) może odbiegać od prawego

hornDiv = 0.1 - ile razy wzrost procentowy na formacji przekracza średni wzrost

hornDiff = 5 - ile jednostek czasu trwa formowanie się podstawy

flagBaseTime = 3 – minimalny okres dla flag

formVul = 5 - Jest to coś jak trendVul, tylko dla formacji. Tzn. jest to wartość nachylenia prostej w stopniach, aby uznać linię za nie-poziomą.

convergenceVul = 15 - Miara zbieżności prostych, tzn różnica kątów nachylenia, powyżej której traktujemy linie jako nie-równoległe przy badaniu formacji.

Funkcje dla użytkownika:

getChannellines(array, a=3, b=4):

Zwraca linie wsparcia i oporu w postaci tablicy `[[x0, index_of_x0], [x1, index_of_x1], ...]`, argumenty:

a, b - jaką część tablicy wziąć, np. a = 1, b = 1 - cała tablica, a = 1, b = 2, druga połowa wejściowej tablicy

Output:

- tablice sup, res, które są zawierają punkty wyznaczające odpowiednio linie wsparcia i oporu

lookForHeadAndShoulders(values, volume, analyze=0):

Szukamy formacji głowy i ramion

Argumenty:

values – tablica wartości

volume – tablica wolumenu

analyze – czy jest używana w analizie, jeśli tak, zwraca tylko moc sygnału 0 – 1

Output: tablica: [moc sygnału, [punkty tworzące linie szyi]]

lookForReversedHeadAndShoulders(values, volume, analyze=0):

Szukamy odwróconej formacji głowy i ramion

Argumenty:

values – tablica wartości

volume – tablica wolumenu

analyze – czy jest używana w analizie, jeśli tak, zwraca tylko moc sygnału 0 - 1

Output: tablica: [moc sygnału, [punkty tworzące linie szyi]]

lookForTripleTop(values, volume, analyze=0):

Szukamy formacji potrójnego szczytu:

Argumenty:

values – tablica wartości

volume – tablica wolumenu

analyze – czy jest używana w analizie, jeśli tak, zwraca tylko moc sygnału 0 - 1

Output: tablica: [moc sygnału, [punkty tworzące linie szyi]]

lookForTripleBottom(values, volume, analyze=0):

Szukamy formacji potrójnego dna:

Argumenty:

values – tablica wartości

volume – tablica wolumenu

analyze – czy jest używana w analizie, jeśli tak, zwraca tylko moc sygnału 0 - 1

Output: tablica: [moc sygnału, [punkty tworzące linie szyi]]

hornTops(values, volume):

Szukamy formacji rogu wyższego:

Argumenty:

values – tablica wartości

volume – tablica wolumenu

Output: moc sygnału

hornBottoms(values, volume):

Szukamy formacji rogu niższego:

Argumenty:

values – tablica wartości

volume – tablica wolumenu

Output: moc sygnału

findGeometricFormations(values) - znajduje formacje geometryczne (trójkąty, kliny, prostokąty) na danej tablicy i przypisuje im wartości. Wewnątrz funkcji wywoływane jest *findGeometricFormationsOnFragment* dla całej tablicy, ostatnich 3/4, ostatniej połowy i ostatniej ćwiartki. Wartość formacji zależy od jej wielkości, tzn. formacja znaleziona na całej tablicy jest warta więcej niż formacja znaleziona na ćwiartce.

values – tablica, na której szukamy formacji (kursy zamknięć)

zwracana wartość: lista, której elementy to wartości zwracane przez *findGeometricFormationsOnArray*

findGeometricFormationsOnFragment(values,a,b) – znajduje formację geometryczną na podtablicy values określonej za pomocą parametrów a i b. Odbywa się to poprzez wywołanie *getChannelLines* z odpowiednimi parametrami, a następnie sprawdzeniu jakie kąty tworzą ze sobą proste wsparcia i oporu.

Prostokąt: linie wsparcia i oporu poziome

Trójkąt niższy: linia oporu malejąca, wsparcia pozioma

Trójkąt wyższy: linia oporu pozioma, wsparcia rosnąca

Trójkąt symetryczny: linia oporu malejąca, wsparcia rosnąca

Klin niższy: linia wsparcia malejąca, oporu jeszcze szybciej malejąca

Klin niższy: linia oporu rosnąca, wsparcia jeszcze szybciej rosnąca

values – j. w.

a, b – patrz *getChannelLines*

zwracana wartość: (nazwa,
(resx0,resy0,resx1,resy1),(supx0,supy0,supx1,supy1),wartość)
gdzie nazwa='rising_wedge', 'falling_wedge', 'rising_triangle', 'falling_triangle',
'symmetric_triangle', 'rect'
XXXx0,XXXx1,XXXy0,XXXy1 – współrzędne punktów wyznaczających proste
wsparcia/oporu na wykresie
wartość = 1 (modyfikowana w findGeometricFormations)

Moduł candles

Stałe

CANDLE_MAX_LEN = 20 - maksymalna ilość świec, które bierzemy pod uwagę szukając formacji. Generalnie ustawianie większych wartości nie ma sensu, bo formacje świecowe mają mały „zasięg rażenia”

STRONG_TREND=0.3 - wartość współczynnika kierunkowego prostej, powyżej którego traktujemy trend jako silny (wy tłumaczenie w dokumentacji luk)

STRAIGHT_TREND_VUL=0.05 - o ile wartości mogą odchyłać się od regresji przy szukaniu luk, wy tłumaczenie jw.

LONG_BODY=0.03 - parametr określający jaką różnicę między O a C traktujemy jako dużą (3%), tzn. jaką świecę uważamy za długą

SHORT_BODY=0.005 - parametr określający jaką różnicę między O a C traktujemy jako małą (0,5%), tzn. jaką świecę uważamy za krótką

LOW_PART=0.25 - poniżej tej części wykresu szukamy luki startowej

HIGH_PART=0.75 - powyżej tej części wykresu szukamy luki wyczerpania

Funkcje

findCandleFormations(O,H,L,C) - Szuka na ostatnich CANDLE_MAX_LEN pozycji na wykresie formacji świecowych, zwraca listę tych które znalazł z podanymi wartościami. Wartościowanie na podstawie tego jak dawno wystąpiła dana formacja (tzn. im dawniej, tym mniejsza wartość)

O – tablica kursów otwarcia

H – tablica maksymalnych kursów

L – tablica minimalnych kursów

C – tablica kursów zamknięcia

Zwracana wartość: lista krotek postaci: (nazwa formacji, indeks początku formacji w tablicy, indeks końca formacji w tablicy, wartość)

Zwracane nazwy formacji – w dokumentacjach do funkcji, które je znajdują

findDarkCloud(O,C): - Szuka na podanych tablicach formacji zasłony ciemnej chmury (nie ja wymyślałem te nazwy). W każdym razie chodzi o to, że

1. Mamy trend rosnący
2. Mamy świecę dużą białą
3. Następnie mamy świecę czarną, która zaczyna się ponad poprzednią świecą, a kończy się poniżej połowy tejże świecy

Jest to formacja odwrócenia trendu wzrostowego

O, C – j. w.

Zwraca ('dark_cloud', indeks początku, indeks końca)

findPiercing(O,C): - Szuka na podanych tablicach formacji przenikania. Jest to formacja dokładnie odwrotna do dark clouda, tzn. trend malejący, duża czarna świeca, biała świeca zaczynająca się poniżej poprzedniej i kończąca powyżej jej połowy. Formacja oznacza odwrócenie trendu spadkowego.

O, C – j. w.

Zwraca ('piercing', indeks początku, indeks końca)

findEveningStar(O,C) – znajduje na podanych tablicach formację gwiazdy wieczornej.

1. Mamy trend rosnący
2. Mamy długą białą świecę
3. Mamy po niej krótką świecę (białą lub czarną), która tworzy lukę lub jest na styk z poprzednią i następną
4. Mamy na końcu długą czarną świecę, która zamyka się poniżej połowy pierwszej świecy

Jest to formacja odwrócenia trendu wzrostowego

O, C – j. w.

Zwraca ('evening_star', indeks początku, indeks końca)

findMorningStar(O,C) – znajduje na podanych tablicach formację gwiazdy porannej. Jeśli ktoś się jeszcze nie domyślił, to jest to analogon formacji gwiazdy wieczornej: długa czarna świeca, krótka świeca, długa biała świeca

Jest to formacja odwrócenia trendu spadkowego

O, C – j.w.

Zwraca ('morning_star', indeks początku, indeks końca)

findBull3(O,H,L,C) – znajduje na podanych tablicach formację trójki hossy

1. Mamy trend rosnący

2. Długa biała świeca

3. 3 krótkie świece, z których co najmniej dwie są czarne i wszystkie są wewnątrz pasa wyznaczonego przez dolny i górny cień pierwszej świecy

4. Długa biała świeca, która ustanawia nowe maksimum

Jest to formacja potwierdzenia trendu wzrostowego

O, H, L, C – j.w.

Zwraca ('bull3', indeks początku, indeks końca)

findBear3(O,H,L,C) – znajduje na podanych tablicach formację trójki hossy. Jest to formacja odwrotna do bull3, czyli długa czarna świeca, 3 małe świece ograniczone przez górny i dolny cień pierwszej świecy, długa czarna świeca ustanawiająca nowe minimum. Jest to formacja potwierdzająca trend spadkowy.

O, H, L, C – j.w.

Zwraca ('bear3', indeks początku, indeks końca)

isStraightTrend(array) – funkcja sprawdza, czy tablica opisuje zdecydowany ruch. Na takim zdecydowanym ruchu jest sens szukać luk. Poprzez zdecydowany rozumiem: po pierwsze silny, po drugie bez latania nie wiadomo gdzie na boki. Poniżej przykłady.

Zwraca True lub False

findGaps(H,L,C) – znajduje luki (startowe, ucieczki i wyczerpania) i przypisuje im wartość H,L,C – j.w.

Zwraca listę par, których 1-szy element to lista luk zwrócona przez *findGapsOnFragment*, drugi to wartość. Sposób szukania i wartościowania podobny jak w formacjach geometrycznych – szukamy na całej tablicy, na 3/4, ..., im na większej podtablicy znajdziemy, tym większa wartość.

findGapsOnFragment(H,L,C,a,b) – znajduje luki (startowe, ucieczki, wyczerpania) na podtablicach H, L, C wyznaczonych przez a, b (patrz *getChannelLines*, *findGeometricFormationsOnFragment*). O co chodzi z tymi rodzajami luk (dla trendu rosnącego):

startowa – szukamy jej poniżej LOW_PART (w dolnej części wykresu), oznacza początek silnego ruchu w górę

ucieczki – szukamy jej pomiędzy HIGH_PART a LOW_PART, pojawia się zazwyczaj w połowie ruchu, czyli możemy zarobić jeszcze drugie tyle ile wynosi różnica między nią a luką startową

wyczerpania – szukamy jej powyżej HIGH_PART, oznacza że trend wkrótce się wyczerpie

H, L, C – j. w.

a, b – też j. w.

Zwraca:

1. listę pustą jeśli nic nie znalazł
2. listę 1,2, lub 3 elementową, gdzie elementy są postaci krotek 3-elementowych: (nazwa, indeks pozycji świeczki/słupka **przed** luką, wartość y w połowie wysokości luki – potrzebne do celów zaznaczania na wykresie). Nazwa = 'rising_breakaway_gap', 'rising_continuation_gap', 'rising_exhaustion_gap', analogicznie z 'falling_...'”

Pozostałe funkcje powinny być traktowane jako prywatne, bez możliwości wywoływania.

Moduł oscillators

simpleArithmeticAverage(array) :

Parametry :

array – tablica wartości liczbowych do obliczenia średniej artmetycznej

Działanie :

Funkcja oblicza najprostszą średnią armetyczną zgodnie ze wzorem $(p_0+p_1+\dots+p_n)/n$, gdzie p_i – kolejne wartosci tablicy, n-dlugosc tablicy

Przykład :

a = [4.5, 5.7, 3.4, 8.2]

simpleArithmeticAverage(a) = 5.4499999...

weightedAverage(array):

Parametry :

array – tablica wartości liczbowych do obliczenia średniej ważonej

Działanie :

Funkcja liczy średnią ważoną dla podanych wartości tablicy array. Najniższą wagę ma pierwsza liczba z tablicy i jest równa 1, każda następna jest większa o 1. Ostatnia wartość z tablicy ma wagę N gdzie N jest długością tablicy.

Wzór : $(n \cdot p_0 + (n-1)p_1 + \dots + p_{n-1}) / (n + (n-1) + \dots + 2 + 1)$

p_i – kolejne wartosci tablicy

n – dlugosc tablicy

Przykład :

a = [4.5, 5.7, 3.4, 8.2]

weightedAverage(a) = 5.8899999....

expotentialAverage(array):

Parametry :

array – tablica wartości liczbowych do obliczenia średniej ważonej

Działanie :

Funkcja oblicza srednią ekspotencjalną dla podanej tablicy. Kolejne wartości tablicy mają coraz większą wagę zgodnie ze wzorem $(1-\alpha)^{(N-i)}$ gdzie N- długość tablicy, i – indeks wartości w tablicy, $\alpha = 2/(1+N)$

Wzór :

$$\frac{(p_0 + (1-\alpha)p_1 + ((1-\alpha)^2)p_2 + \dots + ((1-\alpha)^N)p_N)}{(1 + (1-\alpha) + \dots + (1-\alpha)^N)}$$

Przykład :

a = [4.5, 5.7, 3.4, 8.2]

exponentialAverage(a) = 6.09558823...

movingAverage(array,duration,mode):

Parametry :

array – jednowymiarowa tablica wartości przekazywanych funkcji

duration – czas obliczanego wskaźnika, przykładowo miesięczny duration =

30

mode – tryb pracy funkcji : 1-SMA, 2-WMA, 3-EMA

Działanie :

Funkcja zwraca tablicę z wartościami średniej kroczącej odpowiednio w zależności od używanego trybu dla prostej średniej kroczącej, ważonej średniej kroczącej oraz eksponentowanej średniej kroczącej. Co ważne aby poprawnie obliczyć 10-dniową średnią kroczącą należy przekazać funkcji 20-elementową tablicę. Funkcja obliczy wtedy wartości średniej kroczącej dla ostatnich dziesięciu wartości.

Przykład :

a = [2.51,2.61,2.55,2.65,2.68,2.60]

duration = 3

mode = 2 (Ważona średnia krocząca)

movingAverage(a,duration,mode) = [2.61, 2.648333, 2.635]

highLowIndex(array):

Parametry :

array – tablica wartości przekazywanej funkcji

Działanie :

Funkcja oblicza wskaźnik giełdowy New High New Low Index. Zwraca wartość jest pojedynczą liczbą z zakresu 0-100 która mówi o nastrojach na rynku. Indeks >50 to pozytywny sygnał, <50 to negatywny sygnał.

Przykład :

a = [2.51,2.61,2.55,2.65,2.68,2.60]

highLowIndex(a) = 80

standardDeviation(array):

Parametry :

array – tablica wartości przekazywanych funkcji

Działanie :

Funkcja liczy standardowe odchylenie dla podanych wartości, korzystając ze średniej artmetycznej.

$$\text{Wzór : } \sqrt{((p_0 - \text{SMA})^2 + (p_1 - \text{SMA})^2 + \dots + (p_n - \text{SMA})^2) / n}$$

Przykład :

a = [2.51,2.61,2.55,2.65,2.68,2.60]

standardDeviation(a) = 0.0571547....

bollingerBands(array,duration,mode,D):

Parametry :

array – tablica przekazywanych wartości

duration – długość obliczanego wskaźnika, dla tygodniowego wykresu

duration = 7

mode – tryb pracy funkcji, 1: Górna wstęga bollingera, 2: Dolna wstęga Bollingera

D – stała używana w obliczeniach wstęg, w większości przypadków przyjmujemy że D=2

Działanie :

Funkcja zwraca tablicę wartości wstęg Bollingera dla podanej tablicy array.

Ważne aby przekazywana tablica była dwukrotnie większa od duration. Dla 14-elementowej tablicy wartości możemy obliczyć wstęgi bollingera z ostatnich 7 dni, tzn. ostatnich siedmiu wartości tablicy. Funkcja korzysta ze prostej średniej kroczącej SMA.

Wzór : $BBUpper(i) = SMA + (D * standardDeviation(i))$

$BBLower(i) = SMA - (D * standardDeviation(i))$

Przykład :

a = [2.51,2.61,2.55,2.65,2.68,2.60]

duration = 3

mode = 1

D = 2

bollingerBands(a,duration,mode,D) = [2.6855, 2.7378, 2.7093]

momentum(array,duration):

Parametry :

array – tablica przekazywanych wartościami

duration – długość tablicy wynikowej z wartościami wskaźnika impetu

Działanie :

Funkcja oblicza oscylator impetu na podstawie wzoru :

C = (Today – Cx) gdzie Today jest dzisiejszą wartością, Cx – wartością przed x-dniami

Zwraca tablicę wielkości tablicy wejściowej – duration

Przykład :

a = [2.51,2.61,2.55,2.65,2.68,2.60]

duration = 3

momentum(a,duration) = [0.14, 0.07, 0.05]

ROC(array,duration):

Parametry :

array – tablica przekazywanych wartościami

duration – długość tablicy wynikowej z wartościami wskaźnika ROC (Rate of change)

Działanie :

Funkcja zwraca tablicę z wartościami oscylatora ROC zgodnie ze wzorem :

wartość $[(\text{Close} - C_x)/C_x] * 100$ gdzie Close jest ostatnią wartością zamknięcia, C_x –
zamknięcia przed x-dniami
Zwraca tablicę wielkości tablicy wejściowej – duration

Przykład :

$a = [2.51, 2.61, 2.55, 2.65, 2.68, 2.60]$

duration = 3

$\text{ROC}(a, \text{duration}) = [5.57768924, 2.68199234, 1.96078431]$

meanDeviation(array):

Parametry :

array – tablica wejściowych wartości

Działanie :

Funkcja zwraca pojedynczą wartość średniego odchylenia dla podanej
tablicy. Różni się od standardDeviation tym że zamiast sumować kwadrat różnicy
wartości, sumuje ich wartość bezwzględną. Korzysta z
SimpleArithmeticAverage.

Przykład :

$a = [2.51, 2.61, 2.55, 2.65, 2.68, 2.60]$

$\text{meanDeviation}(a) = 0.0466666...$

CCI(closeTable,lowTable,highTable,duration):

Parametry :

closeTable – tablica wartości zamknięć sesji giełdowych

lowTable – tablica najniższych wartości z danych sesji

highTable – tablica najwyższych wartości z danych sesji

duration – wartość długości oscylatora CCI

Działanie :

Funkcja zwraca tablicę wartości oscylatora Comodity Channel Index, ważne
jest aby przekazane tablice były tych samych rozmiarów oraz ich
poszczególne indeksy odzwierciedlały kolejno zamknięcie sesji,
najwyższą i najniższą wartość na sesji.

Funkcja zwróci tablicę wielkości tablic wejściowych – duration. W praktyce
stosuję się najczęściej 20-dniowy CCI, tzn. duration = 20.

Przykład :

high=[24.2,24.07,24.04,23.87,23.67,23.59,23.8,23.8,24.3,24.15,24.05,24.06,23.88,25.14,25
.2,25.07,25.22,25.37,25.36,25.26,24.82,24.44,24.65,24.84,24.75,24.51,24.68,24.67,23.84,2
4.3])

low =

[23.85,23.72,23.64,23.37,23.46,23.18,23.4,23.57,24.05,23.77,23.6,23.84,23.64,23.94,24.74
,24.77,24.9,24.93,24.96,24.93,24.21,24.21,24.43,24.44,24.2,24.25,24.21,24.15,23.63,23.76
])

close =

[23.89,23.95,23.67,23.78,23.5,23.32,23.75,23.79,24.14,23.81,23.78,23.86,23.7,24.96,24.88
,24.96,25.18,25.07,25.27,25.0,24.46,24.28,24.62,24.58,24.53,24.35,24.34,24.23,23.76,24.2
]

```

duration = 20
CCI(close,low,high,duration) = [ 102.19852633, 30.77013938,
6.49897701, 33.16030534, 34.93862134, 13.99232679, -10.73054136, -11.52818783,
-29.31511456, -129.55641482, -73.17724562]

```

sumUnderCondition(array,mode):

Parametry :

- array – tablica wartości wejściowych
- mode – tryb pracy funkcji : 1 – sumuje wszystkie wartości dodatnie, 2 – sumuje wszystkie wartości ujemne i zmienia znak wynikowej wartości

Działanie :

Sumowanie podanej tablicy z warunkiem.

Przykład :

```

a = [-1,-2,-3,-4]
mode = 2
sumUnderCondition(a,mode) = 10

```

RSI(array,duration):

Parametry :

- array – tablica wartości sesji (najczęściej zamknięcia poszczególnych sesji)
- duration – czas odchylenia oscylatora RSI

Działanie :

Funkcja oblicza oscylator RSI (Relative Strength Index) na podstawie wejściowej tablicy. Wynikowa tablica jest wielkości tablicy wejściowej – duration a jej wartości odpowiadają wartościom oscylatora RSI dla indeksów wejściowej tablicy [duration,array.size]. Bardzo często duration = 14 (dwutygodniowy wskaźnik)

Przykład :

```

a =
[44.34,44.09,44.15,43.61,44.33,44.83,45.10,45.42,45.84,46.08,45.89,46.03,45.61,46.28,46.
28,46.0,46.03,46.41,46.22,45.64,46.21,46.25,45.71,46.45,45.78,45.35,44.03,44.18,44.22,4
4.57,43.42,42.66,43.13]
duration = 14
RSI(a,duration) = [ 70.46413502, 66.24961855, 66.48094183,
69.34685316,
66.29471266, 57.91502067, 62.88071831, 63.20878872, 56.01158479,
62.33992931, 54.67097138, 50.3868152 , 40.01942379, 41.4926354 ,
41.90242968, 45.49949724,
37.32277831, 33.09048257, 37.78877198]

```

highest(array):

Parametry :

- array – tablica wartości wejściowych

Działanie :

Funkcja zwraca największą wartości w tablicy.

Przykład :
a = [3,4,5]
highest(a) = 5

lowest(array):

Parametry :
array – tablica wartości wejściowych
Działanie :
Funkcja zwraca najmniejszą wartość w tablicy.

Przykład :
a = [3,4,5]
lowest(a) = 3

williamsOscillator(highTable,lowTable,closeTable,duration):

Parametry :
highTable – tablica najwyższych wartości danych sesji giełdowych
lowTable – tablica najniższych wartości sesji giełdowych
closeTable – tablica wartości zamknięć danych sesji giełdowych
duration – czas odchylenia oscylatora Williamsa

Działanie :
Funkcja zwraca tablicę wartości oscylatora Williamsa bardziej znanego jako

%R.

Ważne aby przekazane tablice były tych samych rozmiarów i
odzwierciedlały kolejno odpowiednio wartości najwyższe, najniższe i
zamknięć z tych samych sesji.

W wyniku otrzymujemy tablice wielkości tablic wejściowych – duration.

Moduł Strategy

Moduł odpowiada za wygenerowanie strategii inwestycyjnej. Użytkownik może z pośród listy wybrać, które formacje, wskaźniki oscylatory mają być brane pod uwagę, jak również może przypisać im priorytety – 0 – 100. Strategia jest obliczana jako suma iloczynów postaci wskaźnik * moc wskaźnika * priorytet wskaźnika.

Konstruktor:

`__init__(data):`

- data – obiekt z danymi klasy ChartData

Atrybuty:

positiveSignal = 50 - ile co najmniej punktów musi mieć sumaryczna analiza aby uznać, że istnieją pozytywne przesłanki do zakupu instrumentu finansowego
negativeSignal = -50 – analogicznie dla sprzedaży.

Wszystkie poniższe atrybuty odpowiadają za priorytet zdefiniowany przez użytkownika. Mają swoje odpowiedniki z przedrostkiem def – są to domyślne wartości twórców aplikacji. Do analizy bierzemy tylko te formacje, wskaźniki itd., które mają niezerową wartość atrybutów. Wartości dodatnie sugerują sygnał pozytywny – predykcja wzrostu wartości, wartości ujemne – negatywny, predykcja spadku.

1. trendVal = 100

Formacje

- Odwrócenie trendu wzrostowego

2. headAndShouldersVal = -100
3. tripleTopVal = -100
4. risingWedgeVal = -80
5. fallingTriangleVal = -80

- Odwrócenie trendu spadkowego

6. reversedHeadAndShouldersVal = 100
7. tripleBottomVal = 100
8. fallingWedgeVal = 80
9. risingTriangleVal = 80

- Kontynuacja trendu

10. symetricTriangleVal = 50
11. rectangleVal = 30
12. flagPennantVal = 20

- Wskaźniki i oscylatory

13. oscillatorsVal = 50
14. newHighNewLowVal = 50
15. bollingerVal = 50
16. momentumVal = 50
17. rocVal = 50
18. cciVal = 50
19. rsiVal = 50
20. williamsVal = 50

Luki

- Wzrostowe

21. risingBreakawayGapVal = 50
22. risingContinuationGapVal = 30
23. fallingExhaustionGapVal = 10

- Spadkowe

24. fallingBreakawayGapVal = -50
25. risingExhaustionGapVal = -50

26. fallingContinuationGapVal = -30

Formacje świecowe

- *Sygnał kupna*

27. bull3Val = 15

28. mornigStarVal = 10

29. piercingVal = 5

- *Sygnał sprzedaży*

30. bear3Val = -15

31. eveningStarVal = -10

32. darkCloudVal = -5

Funkcje:

Każdy atrybut ma odpowiadające funkcje:

setAtrybut(wartość) – ustawia atrybut na określoną wartość (0 – 100)

disableAtrybut() – wyzerowuje atrybut

enableAtrybut() – przywraca atrybut do wartości domyślnej.

Przykład:

setHeadAndShouldersVal(headAndShouldersUserVal)

disableHeadAndShouldersVal():

enableHeadAndShouldersVal():

resetCoefficients() – włączamy analizę wszystkich możliwych narzędzi z ich domyślną wartością atrybutów.

analize() – analizuje strategie i generuje raport, który zwraca w postaci stringa.