

Wstępny pomysł na testowanie algorytmów

inż. Paulina Brzęcka 184701 inż. Marek Borzyszkowski 184266
inż. Wojciech Baranowski 184574

24 lipca 2024

Spis treści

1	Wstęp	2
2	Dane	2
3	Algorytm	2
3.1	train	3
3.2	fit	3
3.3	history	3
3.4	save	3
3.5	load	3
4	Testowanie	3
4.1	AlgorithmTester	4
4.2	ResultPresenter	4

1 Wstęp

Podczas pracy nad algorytmami grającymi na giełdzie, kwantowymi lub klasycznymi, przydatne będą.

- standaryzacja uruchamiania algorytmów,
- standaryzacja procesu testowania algorytmów,
- standaryzacja danych udostępnianych przez algorytm.

Z powodu powyższych powstał ten dokument, dający wstępny zarys jak może (lub nawet będzie) wyglądać każdy z powyższych elementów.

2 Dane

Przed algorytmem warto zdefiniować na jakich danych będą testowane algorytmy. Pozwoli to uprościć definicję struktury klasy z algorytmem oraz wskazać kierunek przy decyzjach implementacyjnych. Z założenia dane wykorzystywane to będą:

1. Seria odczytów indeksów giełdowych, takich jak S&P500, WIG20 etc. Odczyty będą w odstępach dziennych.
2. Odczyty poszczególnych aktywów wchodzących w dany indeks (zazwyczaj akcje spółek). Odczyt jw. Dla danego indeksu zestaw odczytów poszczególnych aktywów będzie przechowywany w tablicy dwuwymiarowej.
3. Modele poszczególnych algorytmów, które wymagają trenowania przed ich użyciem. Jest to szczególnie ważne przy dalszej pracy, gdy potrzebne będzie wrócenie do modelu wcześniej przetrenowanego w celu porównania wyników.
4. Wyniki poszczególnych algorytmów w postaci pliku CSV zawierającego serię dat, liczby danych aktywów, łączna wartość aktywów.
5. Metadane o algorytmie w postaci pliku, takie jak czas działania na zadanej serii danych, czas trenowania modelu.

3 Algorytm

Podstawą algorytmów będzie `TradingAlgorytm`. `TradingAlgorytm` będzie klasą abstrakcyjną, po której dziedziczyć będą inne algorytmy, klasyczne i kwantowe. Metody abstrakcyjne jakie będzie posiadać klasa `TradingAlgorytm`:

- `train`
- `fit`
- `history`
- `save`
- `load`

3.1 train

Input: Tablica dwuwymiarowa, dla danego przedziału czasu (w dniach) jaka jest wartość danej akcji.

Output: algorytm/brak?

Metoda ta ma za zadanie wytrenować algorytm na wybranym zestawie akcji. Ważne aby dane użyte do trenowania danego algorytmu były danymi wcześniejszymi, niż dane później wykorzystywanymi do ewaluacji danego algorytmu. Najlepiej jakby to były dane 4-5 letnie przed datą wybraną jako rozpoczęcie predykcji. Dla algorytmów niewymagających uprzedniego uczenia (patrz MACD), powinno zwracać nic?

3.2 fit

Input: Tablica dwuwymiarowa, dla danego przedziału czasu (w dniach) jaka jest wartość danej akcji?

Output: Wyniki modelu w postaci tabeli, wyżej opisane w sekcji *Dane*

Metoda robiąca predykcje na zadanej serii krotek aktywów. Ważne, dane muszą być inne niż dane do trenowania.

3.3 history

Input: brak?

Output: historia procesu uczenia danego algorytmu?

3.4 save

Input: brak

Output: brak

Metoda zapisująca algorytm (i może historię?) do folderu.

3.5 load

Input: Ścieżka dojścia do folderu z danym algorytmem.

Output: Brak?

Metoda ładująca algorytm do danej instancji algorytmu?

4 Testowanie

Do testowania potrzebne będą dwie klasy:

1. **AlgorithmTester** - klasa wykorzystywana do wykonywania testów na jednym przy zadanym zbiorze danych.
2. **ResultPresenter** - klasa wykorzystywana do prezentowania wyników w postaci wykresów zawierających dane wynikowe jednego lub więcej algorytmu.

4.1 AlgorithmTester

Klasa `AlgorithmTester` udostępnia następujące metody:

- `perform_test`:
Input: Obiekt będący instancją danego algorytmu, tablica dwuwymiarowa, dla danego przedziału czasu (w dniach) jaka jest wartość danej akcji.
Output: Wyniki modelu w postaci tabeli, wyżej opisane w sekcji *Dane*

4.2 ResultPresenter

Klasa `ResultPresenter` udostępnia następujące metody:

- `print_results_single_chart`:
Input: Tablica trójwymiarowa, dla każdego wyniku algorytmu, dla danego przedziału czasu (w dniach) jaka jest wartość danej akcji.
Output: Obraz prezentujący wyniki na pojedynczym wykresie
- `print_results_separate_chart`:
Input: Tablica trójwymiarowa, dla każdego wyniku algorytmu, dla danego przedziału czasu (w dniach) jaka jest wartość danej akcji.
Output: Obrazy prezentujące wyniki na osobnych wykresach