

Προγραμματιστικά Εργαλεία και Τεχνολογίες για Επιστήμη
Δεδομένων

The Time-Travel Project

Ευάγγελος Τσόγκας

AM: 03400120

Email: evangelostsogkas@mail.ntua.gr

ΔΠΜΣ Επιστήμη Δεδομένων και Μηχανική Μάθηση

Εθνικό Μετσόβιο Πολυτεχνείο

Ιανουάριος 2021

1 Εισαγωγή

Έχοντας στη διάθεσή μας έναν τραπεζικό λογαριασμό με 1 δολάριο την 1/1/1960, είναι μια καλή ιδέα να προσπαθήσουμε να βγάλουμε λεφτά με αγοραπωλησίες μετοχών στο χρηματιστήριο της Νέας Υόρκης. Σε αυτή την εργασία θα προσπαθήσω να βρω τρόπο να έχω όσο το δυνατόν μεγαλύτερο κέρδος τηρώντας τους απαραίτητους περιορισμούς, αλλά ταυτόχρονα απλοποιώντας σε κάποια σημεία τον αλγόριθμο με χρήση απλών μετρικών, μιας και τα δεδομένα είναι πολλά και το πρόβλημα αλγοριθμικά δύσκολο.

2 Προετοιμασία Δεδομένων

Ένα από τα πρώτα πράγματα που θα βοηθήσουν πολύ στην ταχύτητα εκτέλεσης του αλγορίθμου είναι η μείωση των δεδομένων. Αρχικά υπολογίζω το ολικό ελάχιστο και ολικό μέγιστο κάθε μετοχής διαχρονικά, δηλαδή την ελάχιστη Low τιμή και τη μέγιστη High τιμή. Κρατάω, λοιπόν, μετοχές για τις οποίες η ημερομηνία του ελαχίστου προηγείται αυτής του μεγίστου και η διαφορά των τιμών είναι μεγαλύτερη από ένα δολάριο. Οι μετοχές έτσι μειώνονται στις 4473.

Επίσης, δημιουργώ ένα καινούριο date-frame, το **stocks_min_max**, με τις προαναφερθέντες τιμές και ημερομηνίες για κάθε μετοχή και index τους κωδικούς των μετοχών. Όπως θα εξηγήσω στη συνέχεια, αυτό το data-frame παίζει αρκετά σημαντικό ρόλο στην απλοποίηση του αλγορίθμου, αλλά δίνοντας ικανοποιητικά αποτελέσματα.

3 Ο αλγόριθμος

Όλη η διαδικασία γίνεται στο script main.py. Αρχικά φορτώνουμε τα δεδομένα και τα καθαρίζουμε χρησιμοποιώντας τη συνάρτηση `read_data()` του `utils.py`. Αυτή η συνάρτηση, επιστρέφει όλες τις μετοχές, το data-frame **stocks_min_max**, αλλά και ένα `date_range` αντικείμενο με όλες τις μέρες από την πρώτη έως και την τελευταία μέρα που εμφανίζεται στο data-set. Οι υπόλοιπες λειτουργίες υλοποιούνται μέσω της κλάσης `Account`, η οποία περιέχει όλες τις μεταβλητές, πίνακες και μεθόδους για τις αγοραπωλησίες μετοχών.

Η κύρια μέθοδος της κλάσης αυτής και την οποία καλούμε για να γίνουν όλα, είναι η `make_money()`, η οποία επιστρέφει τελικά την αντίστοιχη ακολουθία κινήσεων. Στη `make_money()` υλοποιούνται δύο διαφορετικού τύπου συναλλαγές:

1. **buy-low -> sell-high** (σε διαφορετικές μέρες)
2. **buy-low -> sell-close** (την ίδια μέρα)

Η συναλλαγή buy-low -> sell-high, βασίζεται στο data-frame **stocks_min_max** με πολύ απλό τρόπο. Για κάθε μέρα, αν υπάρχει μετοχή με ελάχιστο τη συγκριμένη μέρα την αγοράζουμε και από τις μετοχές που έχουμε στη διάθεσή μας, αν σε αυτή τη μέρα έχουν μέγιστη τιμή τις πουλάμε. Προφανώς, ελέγχουμε αν ικανοποιούνται όλοι οι περιορισμοί πριν γίνουν οι

συναλλαγές, αλλά θα μείνω σε ένα σημείο που έχει λίγο ενδιαφέρον. Όταν αγοράζουμε μια μετοχή ελέγχουμε ότι το πλήθος που θα πάρουμε δεν ξεπερνάει ούτε το 10% του volume όταν θα την πουλήσουμε σε άλλη μέρα. Προγραμματίζουμε δηλαδή από πριν, ώστε να μην μείνουν απούλητες μετοχές στο τέλος. Αν περισσότερες μετοχές μπορούν να αγοραστούν την ίδια μέρα αγοράζουμε μόνο αυτή με το μεγαλύτερο κέρδος, σε αυτού του είδους τις συναλλαγές.

Αφού, ελέγξουμε για συναλλαγές buy-low -> sell-high, προχωράμε σε **intra-day συναλλαγές**, μόνο αν δεν έχει γίνει κάποια από τις προηγούμενες. Αυτό για να αποφύγουμε πολλούς ελέγχους σχετικά με το volume. Οι επιλογές που έχουν νόημα για συναλλαγές την ίδια μέρα είναι είτε **buy-low -> sell-close**, είτε buy-open -> sell-high, προκειμένου να έχουμε μέγιστο κέρδος. Για απλότητα εξετάζω μόνο την πρώτη. Σε αυτήν την περίπτωση δεν περιορίζομαι στο να αγοράζω μόνο από μια μετοχή, όπως στην περίπτωση buy-low -> sell-high. Αγοράζω και πουλάω όσες μπορώ και μάλιστα τις καλύτερες, αφού τις ταξινομώ ανάλογα με το κέρδος σε φθίνουσα σειρά. Μια τελευταία λεπτομέρεια σε αυτού του είδους τις συναλλαγές είναι ότι τις αγνοώ αν το balance δεν είναι μεγαλύτερο από 100 δολάρια. Ο λόγος είναι ότι ειδικά στην αρχή το balance είναι πολύ μικρό και έτσι γίνονται πολλές πράξεις που καθυστερούν την εκτέλεση του αλγορίθμου, χωρίς σημαντική επίδραση στο τελικό κέρδος.

Ο αλγόριθμος που περιέγραψα είναι κοινός για την παραγωγή τόσο της μικρής ακολουθίας κινήσεων όσο και της μεγάλης, με μια μικρή, αλλά πολύ σημαντική διαφορά. Χρησιμοποιώ δύο βασικές μετρικές που είναι διαφορετικές ανάλογα με την ακολουθία που θέλω να παράγω και είναι οι εξής:

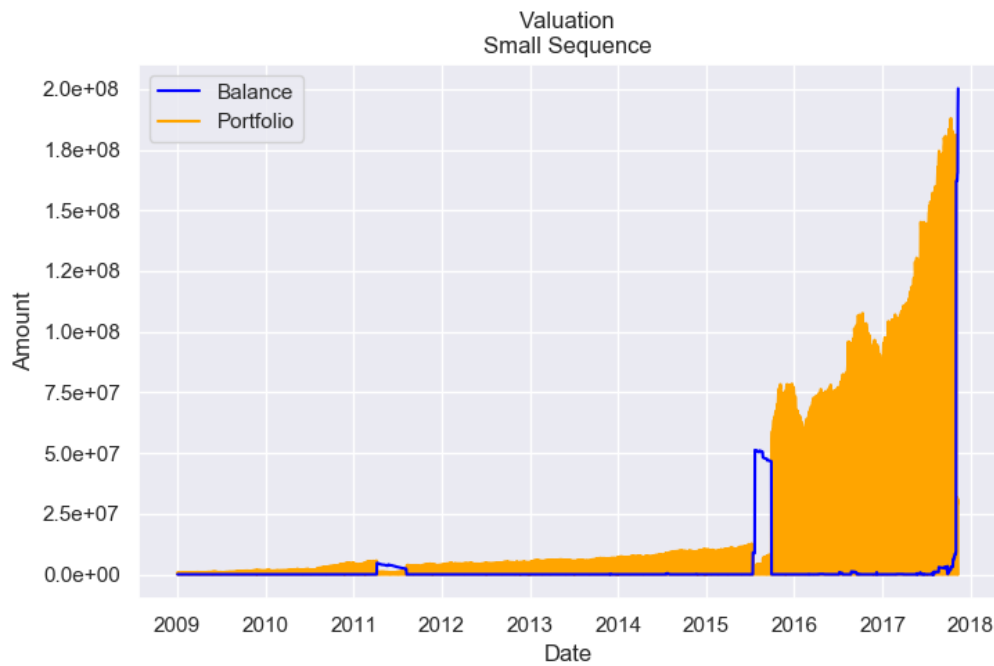
- **min_intraday_profit**: ελάχιστο κέρδος μετοχής για να κάνω intra-day trade
- **min_balance_fraction**: ελάχιστο ποσοστό του balance που θα μείνει μετά από συναλλαγές buy-low με σκοπό sell-high

Το τι σκοπό έχουν αυτές οι μετρικές και τι τιμές παίρνουν αναλόγως την ακολουθία θα το εξηγήσω στη συνέχεια.

4 Ακολουθία $N \leq 1000$ κινήσεων

Για την ακολουθία αυτή η παράμετρος min_intraday_profit παίρνει την τιμή 100. Ο λόγος είναι ότι θέλω να περιορίσω τις συναλλαγές επιτρέποντας μόνο αυτές που πραγματικά «αξίζουν». Επίσης, το min_balance_fraction είναι μηδέν. Στη μικρή ακολουθία βασίζομαι περισσότερο στις συναλλαγές buy-low -> sell-high για να βγάλω κέρδος οπότε δεν με ενδιαφέρει να έχω αρκετά λεφτά για intra-day trades. Αυτή η παράμετρος παίζει σημαντικό ρόλο στη μεγάλη ακολουθία.

Στην Εικόνα 1, φαίνεται το διάγραμμα αποτίμησης για αυτή την ακολουθία. Είναι εμφανές πως στη μικρή ακολουθία δεν βασίζομαι ιδιαίτερα σε intra-day trades για κέρδος. Επίσης, στην Εικόνα 2 φαίνεται η επιτυχής επικύρωση της λύσης, και στην Εικόνα 3 ένα screenshot από την εκτέλεση του αλγορίθμου. Μετά από 6 λεπτά αναμονής, ξέρω πώς να κερδίσω 200 εκατομμύρια δολάρια κάνοντας 606 κινήσεις.



Εικόνα 1: Διάγραμμα αποτίμησης για την ακολουθία $N \leq 1000$.

The Time-Travel Project

2020b edition

Validate your solution:

Don't try this for large solutions. There's a limit to the size of the file you can submit and you will probably exceed it!

Επιλογή αρχείου Δεν επιλέχθηκε κανένα αρχείο. Upload

Success:

After 606 transaction(s), your profit is 200204638.7568747 dollars.

Εικόνα 2: Επικύρωση λύσης για την ακολουθία $N \leq 1000$.

```

597 . Transaction: ['2017-11-09', 'sell-high', 'KRO', 25518.0] | Balance: 166211559.48397473
598 . Transaction: ['2017-11-09', 'sell-high', 'SBPH', 2.0] | Balance: 166211596.88617474
599 . Transaction: ['2017-11-09', 'sell-high', 'UBRT', 180.0] | Balance: 166239216.10417473
600 . Transaction: ['2017-11-09', 'sell-high', 'WB', 232868.0] | Balance: 191709200.17777473
601 . Transaction: ['2017-11-10', 'sell-high', 'AYX', 5860.0] | Balance: 191865199.82377473
602 . Transaction: ['2017-11-10', 'sell-high', 'CIFS', 5366.0] | Balance: 192128107.53037474
603 . Transaction: ['2017-11-10', 'sell-high', 'GDS', 10.0] | Balance: 192128299.78837472
604 . Transaction: ['2017-11-10', 'sell-high', 'ROKU', 210736.0] | Balance: 199774539.4443747
605 . Transaction: ['2017-11-10', 'sell-high', 'SAGE', 4397.0] | Balance: 200204618.8083747
606 . Transaction: ['2017-11-10', 'sell-high', 'SSTI', 1.0] | Balance: 200204638.7568747

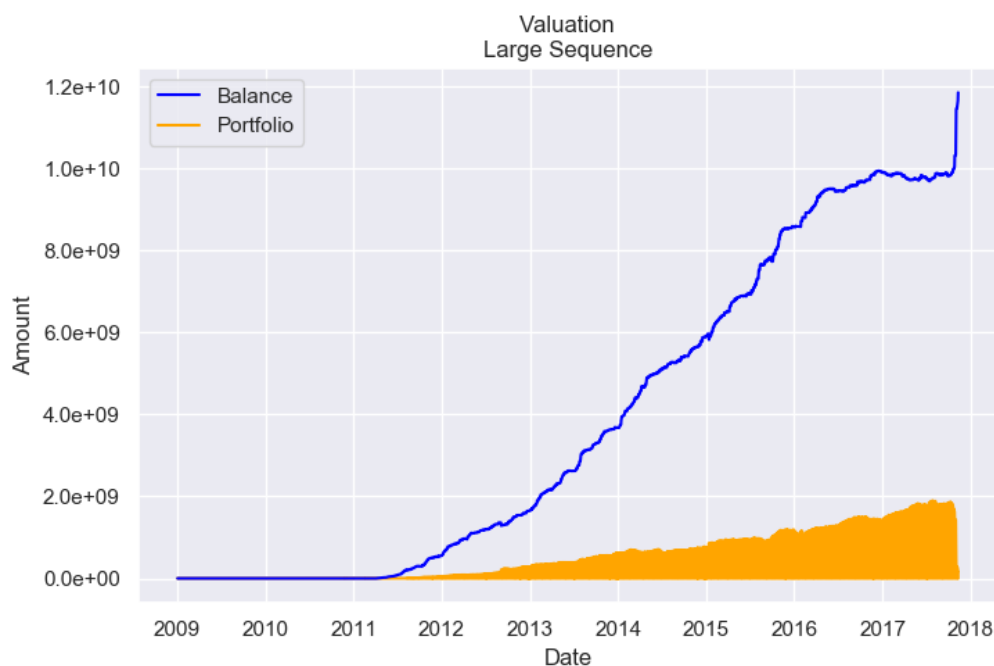
After 606 transaction(s), your profit is 200204638.7568747 dollars.
Total time to run script: 00:06:12

```

Εικόνα 3: Αποτέλεσμα εκτέλεσης αλγορίθμου για την ακολουθία $N \leq 1000$.

5 Ακολουθία $N \leq 1.000.000$ κινήσεων

Σε αντίθεση με την μικρή ακολουθία, σε αυτή την περίπτωση δεν έχει τόσο πολύ σημασία η παράμετρος `min_intraday_profit` που είναι απλά ένα δολάριο, αλλά πολύ σημαντικό ρόλο παίζει η παράμετρος `min_balance_fraction` που παίρνει την τιμή 0.1, δηλαδή, μετά από αγορές με σκοπό την πώληση σε διαφορετική μέρα θα πρέπει να μένει τουλάχιστον το 10% του προηγούμενου `balance` στο λογαριασμό. Ο λόγος είναι πως σε αυτή την ακολουθία βασίζομαι πολύ σε intra-day trades για κέρδος, επομένως με ενδιαφέρει κάθε μέρα να έχω αρκετά λεφτά για συναλλαγές. Παρακάτω φαίνονται τα αντίστοιχα σχήματα όπως και πριν. Σε αυτήν την περίπτωση, ο αλγόριθμος έκανε 24 λεπτά να τελειώσει, αλλά έχω και μεγαλύτερο κέρδος που πλησιάζει τα 12 δισεκατομμύρια δολάρια με περίπου 340 χιλιάδες συναλλαγές.



Εικόνα 4: Διάγραμμα αποτίμησης για την ακολουθία $N \leq 1.000.000$.

The Time-Travel Project 2020b edition

Validate your solution:

Don't try this for large solutions. There's a limit to the size of the file you can submit and you will probably exceed it!

Δεν επιλέχθηκε κανένα αρχείο.

Success:

After 348352 transaction(s), your profit is **11859079586.54261** dollars.

Εικόνα 5: Επικύρωση λύσης για την ακολουθία $N \leq 1.000.000$.

```
348343 . Transaction: ['2017-11-10', 'sell-high', 'KURA', 10316.0] | Balance: 11694637977.501604
348344 . Transaction: ['2017-11-10', 'sell-high', 'MNST', 458899.0] | Balance: 11722503081.964954
348345 . Transaction: ['2017-11-10', 'sell-high', 'OSTK', 47229.0] | Balance: 11725126133.395954
348346 . Transaction: ['2017-11-10', 'sell-high', 'ROKU', 3447376.0] | Balance: 11850209000.491955
348347 . Transaction: ['2017-11-10', 'sell-high', 'SAGE', 4397.0] | Balance: 11850639079.855955
348348 . Transaction: ['2017-11-10', 'sell-high', 'SITE', 62362.0] | Balance: 11854772773.088856
348349 . Transaction: ['2017-11-10', 'sell-high', 'SSTI', 6365.0] | Balance: 11854899745.291355
348350 . Transaction: ['2017-11-10', 'sell-high', 'TLND', 2539.0] | Balance: 11855014566.996155
348351 . Transaction: ['2017-11-10', 'sell-high', 'TNET', 79995.0] | Balance: 11858461927.522655
348352 . Transaction: ['2017-11-10', 'sell-high', 'URGN', 15070.0] | Balance: 11859079586.542656

After 348352 transaction(s), your profit is 11859079586.542656 dollars.
Total time to run script: 00:24:21
```

Εικόνα 6: Αποτέλεσμα εκτέλεσης αλγορίθμου για την ακολουθία $N \leq 1.000.000$.