Πρόβλημα “time-travel”

1. ΕΙΣΑΓΩΓΗ

Η εργασία αφορά την αγοραπωλησία μετοχών στο παρελθόν, με σκοπό τη μεγιστοποίηση του κέρδους μας. Έχοντας γνώση των τιμών των μετοχών μέσω τη βοήθειας της χρονομηχανής, αναζητούμε μια ακολουθία αγοραπωλησιών ικανή να μας αποφέρει το μεγαλύτερο δυνατό κέρδος σε χίλες και ένα εκατομμύριο κινήσεις αντίστοιχα.

2. ΕΠΕΞΗΓΗΣΗ ΚΩΔΙΚΑ

- read_stock_data: Διαβάζουμε όλα τα .txt αρχεία και αποθηκεύουμε τις πληροφορίες σε dictionary, όπου τα keys είναι τα ονόματα των μετοχών και values το dataframe κάθε μετοχής που δημιουργούμε μέσω της pandas.
- preprocessing: Επεξεργαζόμαστε το dictionary που δημιουργήθηκε στο προηγούμενο βήμα, ως εξής:
    - Αφαίρεση μετοχών, όπου το Highest προηγείται χρονικά του Lowest, καθώς δεν μας χρειάζονται για το πρόβλημά μας.
    - Αφαίρεση μετοχών, οι οποίες δεν έχουν τουλάχιστον ένα χρόνο παρουσίας στο χρηματιστήριο.
    - Αφαίρεση μετοχών, οι οποίες έχουν Highest τιμή μικρότερη από 10 δολάρια, καθώς δεν μας βοηθούν στο σκοπό μας.
    - Αφαίρεση ημερομηνιών στις οποίες υπάρχουν μη θετικές τιμές μετοχών, θεωρώντας ότι πρόκειται για συντακτικά λάθη.
    - Αφαίρεση ημερομηνιών όπου ο μέγιστος όγκος αγοραπωλησιών δεν ξεπερνά μια συγκεκριμένη τιμή (100), καθώς ο χαμηλός όγκος μας εμποδίζει να επιτύχουμε μεγάλη αύξηση κεφαλαίου.
    - Επιπλέον, αφαιρούμε τη στήλη OpenInt που δεν μας χρειάζεται, και μετατρέπουμε τη στήλη ημερομηνιών σε datetime format για περαιτέρω επεξεργασία.
- trading_strategy: Δημιουργούμε ένα dataframe που περιέχει τις γραμμές των κινήσεων που πρόκειται να γίνουν. Η στρατηγική αυτή καθ’ αυτή εξηγείται παρακάτω.

- create_moves: Λαμβάνει το dataframe από το trading_strategy και δημιουργεί μια εμφωλευμένη λίστα, όπου κάθε εσωτερική λίστα είναι μια κίνηση. Ο όγκος της κίνησης αρχικοποιείται με μηδέν και υπολογίζεται σε επόμενο βήμα.
- process_move: Λαμβάνει ως είσοδο μια κίνηση και την κατάσταση του λογαριασμού την συγκεκριμένη στιγμή. Υλοποιεί την κίνηση μέσω της κλάσης Move, υπολογίζει τον μέγιστο δυνατό όγκο της κίνησης με βάση το υπάρχον κεφάλαιο και ανανεώνει την κατάσταση του λογαριασμού. Τέλος, επιστρέφει τη νέα κατάσταση του λογαριασμού.
- execute_transactions: Για κάθε κίνηση από τη λίστα κινήσεων καλεί την process_move για να υλοποιήσει την κίνηση και τυπώνει την παρούσα κατάσταση του λογαριασμού. Επιπλέον δημιουργεί εμφωλευμένη λίστα με ημερομηνίες, κέρδος και αποτίμηση , την οποία επιστρέφει για τη δημιουργία διαγραμμάτων.
- create_graph: Λαμβάνει ως είσοδο τη λίστα που επιστρέφει η execute_transactions και δημιουργεί το ζητούμενο γράφημα κέρδους + αποτίμησης ως προς χρόνο.
- save_file: Αποθηκέυει τις κινήσεις σε txt αρχείο.

3. ΠΕΡΙΟΡΙΣΜΟΙ

Ο κατάλληλος αλγόριθμος έπρεπε να πληρεί τα ακόλουθα κριτήρια:

- Καμία κίνηση να μην υπερβαίνει το 10% του καθημερινού όγκου της μετοχής.
- Για κάθε κίνηση αγοράς, τα χρήματα πρέπει να υπάρχουν ήδη στο λογαριασμό πριν την εκτέλεσή της.
- Για κάθε κίνηση πώλησης, πρέπει οι μετοχές να υπάρχουν ήδη στο λογαριασμό.
- Εντός μιας ημέρας, η σειρά των κινήσεων πρέπει να είναι η εξής:
  Buy-open, sell-close » buy-low, sell-high » buy-close, sell-close
- Οι συναλλαγές να γίνονται με χρονολογική σειρά.
- Οι συναλλαγές να μην ξεπερνούν σε αριθμό τις 1000 και 1000000 αντίστοιχα.

4. ΑΛΓΟΡΙΘΜΟΣ ΕΠΙΛΟΓΗΣ ΚΙΝΗΣΕΩΝ

Για τον αλγόριθμο, χρησιμοποιήθηκε αποκλειστικά intraday trading. Μετά από πειραματισμούς με πολλές buy-low, sell-high αγοραπωλησίες, καταλήξαμε ότι είναι αρκετά δύσκολο να γίνει αλγόριθμος με βάση τα buy-low, sell-high, καθώς προϋποθέτει στην αρχή να αγοράζουμε συνεχώς νέες μετοχές και να τις κρατάμε στο λογαριασμό έως ότου αυτές φτάσουν στη μέγιστη χρηματιστηριακή του αξία. Κάτι τέτοιο είναι αρκετά δύσκολο με αρχικό κεφάλαιο 1 δολάριο.

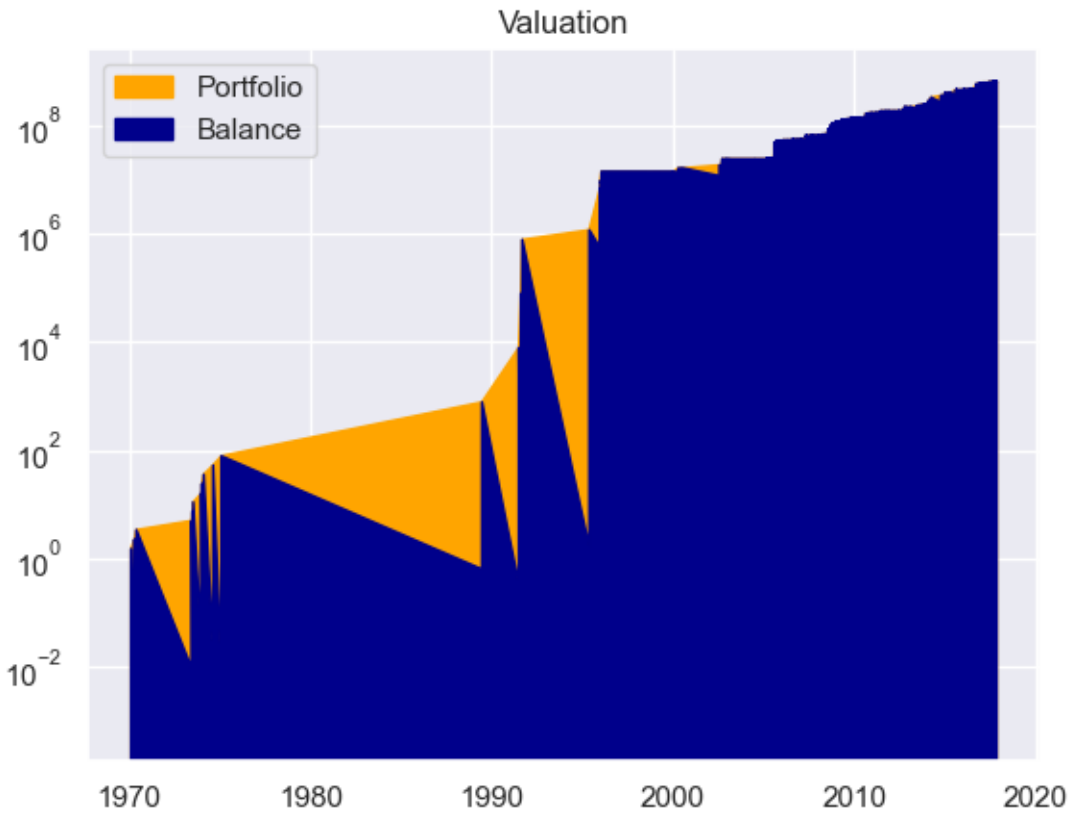
Για να βρούμε τις καλύτερες intraday trades (buy-open, sell-close) υπολογίζουμε για κάθε μετοχή, για κάθε μέρα, το ποσοστό ημερήσιας επιστροφής της μετοχής, δηλαδή: $\frac{\text{Τιμή κλεισίματος} - \text{Τιμή ανοίγματος}}{\text{Τιμή ανοίγματος}}$

Βάζουμε όλα τα ποσοστά ημερήσιας επιστροφής σε ένα dataframe, και ταξινομούμε με βάση το ποσοστό (φθίνουσα σειρά). Η συγκεκριμένη μετρική, χρησιμοποιήθηκε για να καταπολεμήσει το πρόβλημα του χαμηλού ποσού έναρξης της διαδικασίας. Στη συνέχεια, επιλέγουμε τις N πρώτες γραμμές (N=1000 στην πρώτη περίπτωση και N=1000000 στη δεύτερη περίπτωση). Ταξινομούμε εκ νέου τις N γραμμές με αύξουσα σειρά ημερομηνίας, και λαμβάνουμε την σειρά των κινήσεων μας, σε χρονολογική σειρά. Σε κάθε γραμμή, αντιστοιχούν δύο κινήσεις, μία buy-open και μία sell-close. Με αυτό τον τρόπο, διασφαλίζουμε ότι θα τηρηθεί η σειρά των συναλλαγών εντός της ημέρας.

5. Αποτελέσματα

N = 1000 κινήσεις

Για N=1000 κινήσεις, ο αλγόριθμος αποφέρει κέρδος 633910899 (633.9 εκατομμύρια) δολάρια. Το διά γραμμα είναι το εξής:



N = 1000000

Για N=999956 κινήσεις, ο αλγόριθμος αποφέρει κέρδος 163657198288 (163.65 δισεκατομμύρια) δολάρια. Το διάγραμμα είναι το εξής: