



NATIONAL TECHNICAL UNIVERSITY OF ATHENS

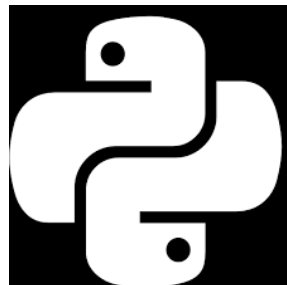
School of Electrical and Computer Engineering

MSc in Data Science and Machine Learning

Προγραμματιστικά Εργαλεία και Τεχνολογίες για Επιστήμη Δεδομένων

Εργασία στην Python:

“Time-Travel Problem”



Instructor:

Athanasios Voulodimos

Student: Theofili Stavroula, 03400166

1. Εισαγωγή

Ο στόχος της παρούσας εργασίας είναι να χρησιμοποιήσουμε ένα σύνολο δεδομένων με την αξία των μετοχών του χρηματιστηρίου της Νέας Υόρκης, προκειμένου να βγάλουμε υψηλό κέρδος μέσα από τις κατάλληλες ακολουθίες αγοραπωλησιών σε ημερομηνίες του παρελθόντος. Τα αρχικά δεδομένα της ανάλυσης αποτέλεσαν οι χρονοσειρές για τις τιμές Open, Low, High και Close, καθώς και ο ημερήσιος όγκος Volume, για 8395 μετοχές του χρηματιστηρίου της Νέας Υόρκης (για τις ημερομηνίες από 1/1/1960 έως και 10/11/2017). Στόχος ήταν η κατασκευή αλγορίθμων που να αποφέρουν υψηλό (όχι απαραίτητα μέγιστο) κέρδος σε βάθος έως και N κινήσεων, με το N να ισούται το πολύ με 1.000 στη μία περίπτωση (Small) και 1.000.000 στην άλλη (Large). Πέραν του άνω φράγματος στο πλήθος κινήσεων, υπήρχαν τρεις επιπλέον βασικοί περιορισμοί:

- i. Το απαραίτητο κεφάλαιο για τις αγορές κάθε ημέρας έπρεπε να είναι διαθέσιμο αμέσως πριν την αρχή της ημέρας και να μην αποκτάται κατά τη διάρκειά της.
- ii. Το μέγιστο πλήθος μετοχών που μπορούσαν να αγοραστούν ή να πουληθούν κάθε ημέρα δε μπορούσε να ξεπερνά το 10% του όγκου της μετοχής για την ημέρα αυτή.
- iii. Εάν στην αρχή μιας ημέρας υπήρχαν n μετοχές κάποιου είδους, τότε την ημέρα αυτή μπορούσαν να αγοραστούν μέχρι και $n + 1$ μετοχές του ίδιου είδους.

Η μεθοδολογία που ακολουθείται στην παρούσα εργασία είναι η εξής:

- A. Καθάρισμα των αρχικών δεδομένων, μείωση του μεγέθους τους και ενοποίηση τους σε έναν πίνακα προκειμένου να μπορούμε να διαχειριστούμε την διαθέσιμη πληροφορία με τον βέλτιστο τρόπο.
- B. Αναλόγως της ζητούμενης ακολουθίας (για $N < 1000$ ή $N < 1.000.000$), γίνονται περαιτέρω μετασχηματισμοί των πινάκων με τα δεδομένα μας. Να σημειωθεί πως ο κώδικας διαφοροποιείται για την περίπτωση Small και Large, καθώς απαιτούνται διαφορετικοί μετασχηματισμοί για την κάθε περίπτωση.
- C. Στη συνέχεια, ακολουθεί ο αλγόριθμος που εκτελεί μια σειρά ελέγχων προκειμένου να ορίσει ποιες είναι οι «βέλτιστες» μετοχές που αναζητούμε. Πρακτικά, ο αλγόριθμος αυτός υπολογίζει τα κέρδη όλων δυνατών επενδύσεων, επιλέγει μεταξύ αυτών ποια επένδυση είναι βέλτιστη και έτσι προκύπτει η ακολουθία των ζητούμενων συναλλαγών N .
- D. Ολοκληρώνουμε με οπτική αξιολόγηση των ακολουθιών που δημιουργήσαμε και των κερδών που προέκυψαν από αυτές, με τη χρήση διαγραμμάτων που απεικονίζουν τη σχέση των κερδών και των αγορών μετοχών ανά το πέρασ του χρόνου.

2. Προεπεξεργασία των δεδομένων (Cleaning - Reducing - Merging)

Ξεκινώντας την ανάλυσή μας, έχοντας συνενώσει τους φακέλους 'ETFs' και 'Stocks', δημιουργήσαμε ένα σετ δεδομένων 'Data1' που ουσιαστικά πρόκειται για έναν φάκελο με 8539 αρχεία μορφής txt. Κάθε αρχείο αφορά μια συγκεκριμένη μετοχή του χρηματιστηρίου της Αμερικής για ημερομηνίες από το 1960 έως και το 2017, συνεπώς το αρχικό σετ δεδομένων μας αφορά 8539 διαφορετικές μετοχές που μπορούμε να αξιοποιήσουμε στις συναλλαγές μας για να πετύχουμε το μέγιστο κέρδος σε αυτό το χρονικό διάστημα.

Cleaning: Για να διασφαλίσουμε ότι ένα αρχείο κειμένου (.txt) δεν προκαλεί προβλήματα κατά την περαιτέρω επεξεργασία των δεδομένων μας, προβαίνουμε αρχικά σε έλεγχο για την απουσία γραμμών ή στηλών με κενά πεδία (με χρήση της μεθόδου `.dropna()`). Επιπλέον, ελέγχουμε τυχόν αρχεία με διαφορετικό format ή που βρίσκονται σε διαφορετικό κατάλογο προέλευσης και δεν μπορούν να διαβαστούν από τον αλγόριθμό μας. Για την επίτευξη αυτού, διαβάζουμε τα αρχεία μετοχών .txt σειριακά, και χρησιμοποιούμε δοκιμές (try/except) για να αντιμετωπίσουμε πιθανά προβλήματα. Η μεταβλητή 's' καταγράφει πόσα αρχεία από το αρχικό σύνολο δεδομένων κατάφεραν να διαβαστούν με επιτυχία (8507 από τα 8539). Αντίστοιχα, η μεταβλητή 'd' καταγράφει τον αριθμό των αρχείων που δεν κατάφεραν να διαβαστούν (32 από τα 8539). Τα αποτυχημένα αρχεία απορρίπτονται και δεν λαμβάνονται υπόψη για την περαιτέρω βελτιστοποίηση του συνόλου δεδομένων 'BestStocks'.

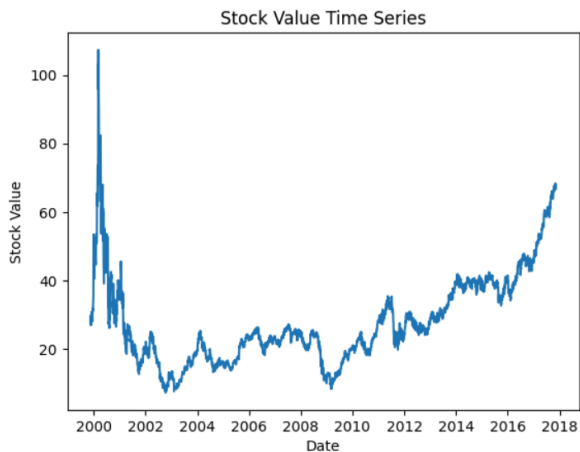
Reducing: Συνεχίζοντας, προχωράμε στη μεθοδο ελαχιστοποίησης των δεδομένων μας. Υπάρχουν πολλοί τρόποι αξιολόγησης της απόδοσης μιας μετοχής στο χρηματιστήριο αλλά για τους σκοπούς της συγκεκριμένης εργασίας αξιοποιούμε την λογική της "συσσωρευτικής απόδοσης" ή αλλιώς του "αθροιστικού γινομένου" της τιμής μιας μετοχής ανά το χρόνο. Η "συσσωρευτική απόδοση" (cumulative return) μιας μετοχής ουσιαστικά είναι ένας δείκτης που δείχνει πώς μεταβάλλεται η αρχική αξία μιας μετοχής με την πάροδο του χρόνου. Εδώ ως αρχική αξία εννοούμε την τιμή που είχε στο κλείσιμο κάθε μέρας, για την πρώτη χρονολογικά ημερομηνία που υπάρχει εγγραφή στο αρχείο .txt της μετοχής. Αν η τιμή του cumulative return είναι μεγαλύτερη του 1 κατά την τελευταία ημερομηνία εγγραφής, σημαίνει ότι με την πάροδο του χρόνου η αξία της μετοχής μεγάλωσε, ενώ αντίστροφα αν η τιμή είναι μικρότερη του 1, η αξία της μετοχής μειώθηκε. Πιο αναλυτικά, χρησιμοποιώντας τις εντολές της βιβλιοθήκης Pandas, για κάθε μετοχή:

1. Υπολογίζουμε τον ημερήσιο ρυθμό μεταβολής της αξίας της στο κλείσιμο κάθε μέρας,
2. Στη συνέχεια υπολογίζουμε το αθροιστικό γινόμενο όλων των διαδοχικών ημερήσιων μεταβολών σύνολο,
3. Κρατάμε την τελευταία τιμή αυτής της ακολουθίας, καθώς αυτή μας δείχνει τη συνολική μεταβολή της αξίας της μετοχής.

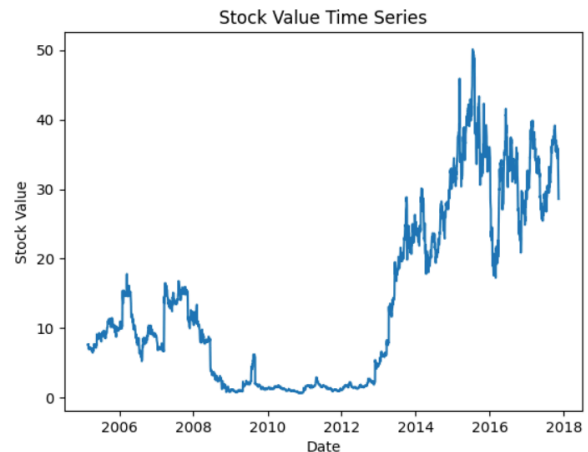
Για παράδειγμα, αν οι ημερήσιοι ρυθμοί μεταβολής της αξίας της μετοχής για τις τρεις πρώτες μέρες είναι 1%, 2% και -3%, τότε αντίστοιχα η ακολουθία αριθμών που θα προκύψει είναι [1.01, 1.02, 0.97] και το αθροιστικό γινόμενο θα είναι η τιμή 0.9993, που δείχνει τη συνολική αξία επιστροφής της μετοχής για αυτές τις τρεις ημέρες. Σημειώνεται ότι ο συγκεκριμένος δείκτης χρησιμοποιείται για να αναδείξει την γενική τάση μιας μετοχής (είτε αυξητική είτε μειωτική) σε ένα μεγάλο χρονικό διάστημα. Ωστόσο, δεν παρέχει πολλές πληροφορίες σχετικά με πιθανές απότομες διακυμάνσεις της μετοχής σε συγκεκριμένα χρονικά διαστήματα, τα οποία ενδέχεται να αποτελούν ευκαιρίες για επενδύσεις.

Προκειμένου να γίνει πιο πλήρης αντιληπτός ο δείκτης αξιολόγησης που χρησιμοποιούμε, παρακάτω παρατίθενται ενδεικτικά οι γραφικές παραστάσεις ορισμένων μετοχών. Απεικονίζονται οι τιμές των μετοχών ανά το πέρασ του χρόνου, ενώ παράλληλα παρουσιάζεται ο δείκτης "συσσωρευτικής απόδοσης" που τους αντιστοιχεί.

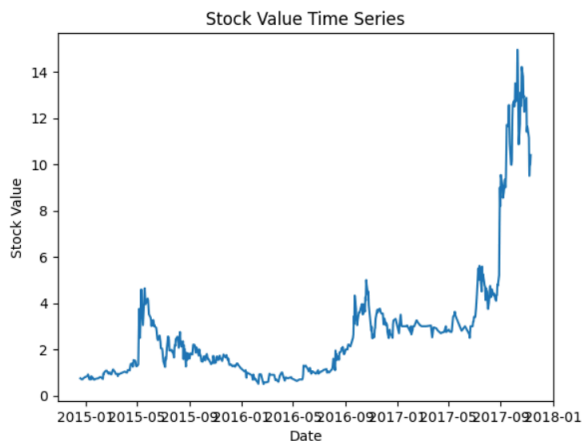
Stock = a.us.txt
Cumulative Product = 2.2493434785536333



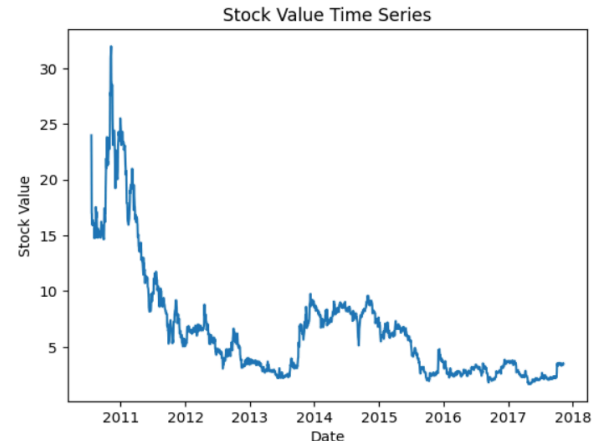
Stock = acad.us.txt
Cumulative Product = 3.7421465968586363



Stock = abeow.us.txt
Cumulative Product = 13.866666666666667



Stock = abac.us.txt
Cumulative Product = 0.14500000000000027



Στο πλαίσιο της συγκεκριμένης εργασίας, θεωρούμε μια μετοχή ως επικερδή, εάν η αξία της με τον χρόνο δεκαπλασιάζεται, δηλαδή αυξάνεται δέκα φορές περισσότερο από τον αρχικό όγκο δεδομένων. Συγκεκριμένα, διατηρούμε μόνο αυτές τις μετοχές για τις οποίες ισχύει ότι η συσσωρευτική απόδοση είναι μεγαλύτερη από 10, κατά τη διάρκεια της χρονικής περιόδου παρουσίας τους στο χρηματιστήριο. Κατά συνέπεια, εξετάσαμε ότι από τον αρχικό αριθμό των 8539 μετοχών, διατηρήσαμε μόνο τις 394 που θεωρούνται επενδυτικά πιο ελκυστικές. Έτσι, κρατήσαμε περίπου το 5% των αρχικών δεδομένων μας.

Ένα ακόμη βήμα που εφαρμόστηκε για τη μείωση των αρχικών δεδομένων ήταν να κρατήσουμε από τον πλήρη σετ δεδομένων 'BestStocks' μόνο εκείνες τις μετοχές για τις οποίες χρονικά ισχύει ότι η χαμηλότερη ετήσια αξία τους προηγείται χρονικά της υψηλότερης ετήσιας αξίας τους. Αυτό γίνεται με σκοπό να έχουμε τη δυνατότητα να εκμεταλλευτούμε την τάση να αγοράζουμε αυτές τις μετοχές σε χαμηλή τιμή και να τις πωλούμε σε υψηλότερη τιμή, δημιουργώντας έτσι ευκαιρίες επένδυσης.

Merging: Μέχρι στιγμής, η επεξεργασία των δεδομένων γίνεται διαβάζοντας ξεχωριστά txt αρχεία για κάθε μετοχή. Με σκοπό την ελαχιστοποίηση του συνόλου των δεδομένων, μπορούμε τώρα να ενοποιήσουμε την πληροφορία από όλα αυτά τα ξεχωριστά αρχεία σε ένα ενιαίο data frame (df). Στη συνέχεια, τροποποιούμε τα ονόματα των μετοχών έτσι ώστε να είναι στη μορφή κεφαλαίων γραμμάτων που ζητείται, αφαιρούμε τη στήλη 'Open Int' και προσθέτουμε μια νέα στήλη 'Year', περιορίζοντας τα δεδομένα μόνο στο έτος των

συναλλαγών της κάθε μετοχής. Στη συνέχεια, ταξινομούμε το data frame βάσει της στήλης των ημερομηνιών ('Dates') και ορίζουμε ένα νέο σύνολο δεικτών αρίθμησης με αύξουσα σειρά, ξεκινώντας από την τιμή 0. Από εδώ και πέρα, η προσπάθεια περαιτέρω επεξεργασίας των δεδομένων γίνεται σε αυτό το ενοποιημένο data frame, που περιέχει πλέον 2.484.942 γραμμές και 7 στήλες.

3. Αλγόριθμος Small

Η λογική του αλγορίθμου που αναπτύχθηκε για την εύρεση των ακολουθιών είναι ιδιαίτερα απλή και ακολουθεί μια λογική χρονικής διακριτοποίησης των δεδομένων, προκειμένου η προσέγγιση τους να είναι αποτελεσματική μέσω καλά οργανωμένων πινάκων δεδομένων. Στην περίπτωση της μικρής ακολουθίας, υποθέτουμε ότι κάθε έτος θα γίνεται επένδυση σε μια μόνο μετοχή (αυτή που συγκριτικά με όλες τις άλλες μας δίνει το μέγιστο κέρδος). Ουσιαστικά, για κάθε ξεχωριστό έτος του συνόλου δεδομένων βρίσκουμε σε ποιά μετοχή μας συμφερει να επενδύσουμε και συνεπώς καθ' όλη τη διάρκεια του έτους βρίσκουμε την ημερομηνία αγοράς της μετοχής (όταν έχει την χαμηλότερη τιμή) και την ημερομηνία πώλησης της (όταν έχει τη μέγιστη τιμή μέσα στο έτος). Να σημειωθεί πως ανά έτος θα γίνει μια αγορά και μια πώληση (της ίδιας μετοχής S, για την ίδια ποσότητα μεριδίων x της εν λόγω μετοχής).

3a. Data Transformation

Επιδιώκοντας μια ετήσια στρατηγική επένδυσης σε μετοχές, ο στόχος μας είναι να εντοπίζουμε για κάθε μετοχή τη χαμηλότερη και υψηλότερη τιμή κατά τη διάρκεια κάθε έτους. Για να επιτύχουμε αυτό, χρησιμοποιούμε ένα μοντέλο συγκέντρωσης (aggregation) όπου κατηγοριοποιούμε τα δεδομένα μας (groupby(['S', 'Year'])) για να ανακαλύψουμε τις μέγιστες και ελάχιστες τιμές των χρονοσειρών Open, High, Low και Close για κάθε έτος, καθώς και τις αντίστοιχες ημερομηνίες. Επομένως, προστίθενται νέες στήλες στο dataframe μας, συγκεκριμένα οι ['Open_Min', 'Low_Min', 'Close_Min', 'Open_Max', 'High_Max', 'Close_Max']. Ύστερα, μέσω ενός λεξικού, αντιστοιχίζουμε τις σχετικές ημερομηνίες σε αυτές τις τιμές, δημιουργώντας τις αντίστοιχες στήλες ['Date_O_Min', 'Date_Min', 'Date_C_Min', 'Date_O_Max', 'Date_Max', 'Date_C_Max']. Αυτό μας χρησιμεύει στο να εντοπίζουμε γρήγορα αν, κατά την ετήσια περίοδο, η χαμηλότερη τιμή της μετοχής προέκυψε κατά την έναρξη (Open_Min), κατά τη διάρκεια της ημέρας (Low_Min) ή κατά το κλείσιμο (Close_Min).

Στη συνέχεια, επιλέγουμε με βάση αυτό ποιο είδος συναλλαγής (m) μας συμφέρει να πραγματοποιήσουμε. Το είδος της συναλλαγής με τη μικρότερη αξία αποθηκεύεται στη στήλη ['Min_col'] = min('Open_Min', 'Low_Min', 'Close_Min') και, κατ' επέκταση, μπορούμε να διαγράψουμε αυτές τις τρεις στήλες. Επιπλέον, στη στήλη ['Min_val'], αποθηκεύουμε την αριθμητική αξία της μετοχής για τη συγκεκριμένη ημερομηνία ('Min_date') και το είδος της συναλλαγής (αγορά).

Ομοίως, ακολουθούμε την ίδια διαδικασία για τον εντοπισμό της υψηλότερης (ετήσια) τιμής της μετοχής. Συγκεκριμένα, δημιουργούμε τις στήλες ['Max_col'], όπου αποθηκεύεται το είδος της συναλλαγής, και ['Max_val'], όπου αποθηκεύεται η ακριβής αξία της μετοχής κατά τη διάρκεια της συγκεκριμένης ημερομηνίας ('Max_date'). Επιπλέον, καταγράφουμε το είδος της συναλλαγής, με την πώληση να αντιστοιχεί στην εν λόγω περίπτωση.

Καθώς διαχειριζόμαστε το data frame μας με αυτόν τον τρόπο, διατηρούμε μόνο την πληροφορία που είναι απαραίτητη για τον αλγόριθμό μας, με αισθητή μείωση του όγκου των δεδομένων μας. Το τελευταίο βήμα αποτελεί ο έλεγχος της συνθήκης ότι η ημερομηνία `Min_date` (κατά την οποία μας συμφέρει να αγοράσουμε τη μετοχή) προηγείται της ημερομηνίας `Max_date` (κατά την οποία μας συμφέρει να πουλήσουμε τη μετοχή). Μόνο οι μετοχές `S` που πληρούν αυτό το κριτήριο παραμένουν στο dataframe και προχωρούν στο επόμενο στάδιο. Αυτό υποδηλώνει ότι ο όγκος των δεδομένων που επεξεργάζεται ο αλγόριθμός μας μειώνεται ακόμα περισσότερο (χωρίς να χάνουμε την ουσιαστική πληροφορία που χρειαζόμαστε για την επίτευξη μεγάλου κέρδους).

Στο data frame διατηρούμε τις τιμές της στήλης `'Close'`, αφού αυτές έχουν ήδη κατανεμηθεί στις στήλες `'Close_Min'` και `'Close_Max'`. Αυτή η διατήρηση γίνεται γιατί αυτές οι τιμές θα χρησιμοποιηθούν σε μεταγενέστερο στάδιο για τον υπολογισμό της αξίας του Portfolio των μετοχών μας, βασισμένη στην τιμή της μετοχής στο κλείσιμο, δηλαδή στη στήλη `'Close'`.

3b. The Algorithm (Small.txt)

Μετά την επιτυχημένη επεξεργασία των δεδομένων μας με τον κατάλληλο τρόπο και τη φορμάρηση τους σε μια πιο "αξιοποιήσιμη" μορφή, προχωρούμε στο στάδιο όπου ο αλγόριθμός μας επιλέγει ανά έτος ποια μετοχή είναι βέλτιστη. Σύμφωνα με τον λογικό συλλογισμό του προβλήματος, υιοθετούμε ένα απλό μοτίβο όπου υποθέτουμε ότι για κάθε χρονική περίοδο βρίσκουμε ποια μετοχή παρουσιάζει το μεγαλύτερο κέρδος. Ο στόχος είναι να πραγματοποιηθεί μια αγορά ανά έτος (για μετοχή `S` και ποσότητα `x`), όταν η μετοχή βρίσκεται στη χαμηλότερη τιμή της, και έπειτα να πραγματοποιηθεί μια πώληση όλων των μεριδίων (`x`) της μετοχής, όταν η μετοχή φτάσει την υψηλότερη τιμή της στο ίδιο έτος. Κατά συνέπεια, για κάθε έτος, προκύπτει ένα ιδιαίτερο σενάριο συναλλαγής (πρώτα μια αγορά και στη συνέχεια μια πώληση) που αφορά την ίδια μετοχή `S` και την ίδια ποσότητα `x`.

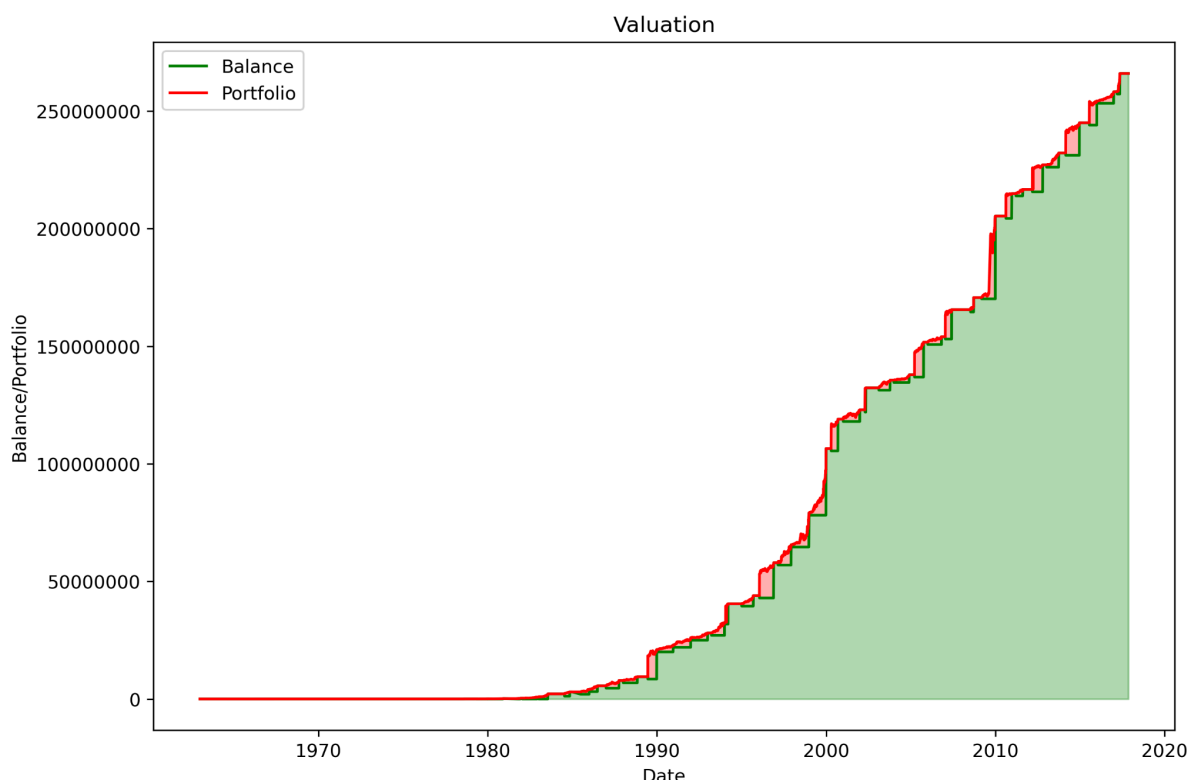
Βάσει της στρατηγικής μας για ετήσια επένδυση σε μετοχές, σκοπός μας είναι να ανακαλύψουμε ποια μετοχή προσφέρει το μέγιστο δυνατό κέρδος για κάθε χρονική περίοδο. Για να το πετύχουμε αυτό, υπολογίζουμε τα κέρδη από κάθε μετοχή στο χρονικό διάστημα, λαμβάνοντας υπόψη το ποσό που μπορούμε να επενδύσουμε και τον περιορισμό στον αριθμό των μετοχών που μπορούμε να αγοράσουμε. Στη συνέχεια, επιλέγουμε τη μετοχή που παρέχει το μέγιστο συνολικό κέρδος και ενημερώνουμε το υπόλοιπο του λογαριασμού μας με τα κέρδη που έχουμε λάβει, επαναλαμβάνοντας τη διαδικασία για το επόμενο χρονικό διάστημα. Ο αλγόριθμός επαναλαμβάνει αυτήν τη διαδικασία σειριακά για κάθε χρονική περίοδο του τελικού μας dataset. Για κάθε έτος, δημιουργούμε ένα λεξικό που περιλαμβάνει το όνομα της μετοχής και το αντίστοιχο κέρδος. Ανανεώνοντας το υπόλοιπο του λογαριασμού μας, επαναλαμβάνουμε τη διαδικασία για την επόμενη περίοδο, και έτσι συνεχίζουμε.

Προτού ξεκινήσει ο επαναληπτικός βρόγχος του αλγορίθμου μας, αρχικοποιούμε ένα ξεχωριστό data frame, το οποίο ονομάζουμε `N`. Σε κάθε επανάληψη της λούπας, προσθέτουμε σε αυτό το data frame τις απαραίτητες πληροφορίες για κάθε συναλλαγή (`d`, `m`, `S`, `x`), αφού έχουμε ήδη εντοπίσει ποια μετοχή θα επιλέξουμε για κάθε χρονική περίοδο. Έτσι, κατά τη διάρκεια κάθε επανάληψης, αποθηκεύουμε το εν λόγω data frame `N` σε ένα αρχείο κειμένου (txt) με το όνομα `N.Small`. Είναι σημαντικό να σημειωθεί ότι η σειρά των εγγραφών στο `N.Small` αντιστοιχεί στην σειρά των επαναλήψεων της λούπας. Πρέπει να αναφέρουμε εκ των προτέρων ότι η μέγιστη δυνατή τιμή για το μέγεθος της ακολουθίας `N.Small` είναι 114. Αυτό προκύπτει από το γεγονός ότι έχουμε 57 χρονιές δεδομένων και για κάθε χρονιά έχουμε δύο συναλλαγές

(αγορά και πώληση), κάνοντας το συνολικό πλήθος των εγγραφών ίσο με 114. Στο data frame N, επιλέγουμε να κρατήσουμε ορισμένες άλλες στήλες που περιέχουν πληροφορίες, όπως η στήλη με την τιμή κλεισίματος ('Close'), προκειμένου να μπορούν να χρησιμοποιηθούν για τη δημιουργία του διαγράμματος σύνοψης της ακολουθίας συναλλαγών μας.

Κάνοντας έλεγχο στην ακολουθία N.Small (αποθηκευμένη ως small.txt) στην πλατφόρμα "The Time-Travel Project", παρατηρούμε ότι για **106 συναλλαγές** στο διάστημα των ετών 1960-2017 προκύπτει κέρδος ύψους **265.694.416** δολαρίων.

3c. Diagram (Small.txt)



Διάγραμμα.1: Διάγραμμα Αποτίμησης της Ακολουθίας N.Small

Στο διάγραμμα που παρουσιάζει την ακολουθία αγοραπωλησιών N.Small, μπορούμε να παρατηρήσουμε διάφορα φαινόμενα σχετικά με την συγκεκριμένη ακολουθία επενδύσεων. Καταρχάς, το διάγραμμα παρουσιάζει μια γραφική αναπαράσταση που φαίνεται να κατατμείται σε "βήματα" λόγω του γεγονότος ότι ο αλγόριθμός μας υπολογίζει και εκτελεί ένα συγκεκριμένο αριθμό συναλλαγών (μια αγορά και μια πώληση) ανά έτος. Έτσι, αντικατοπτρίζεται η "διακριτοποίηση" του χρόνου, δηλαδή πώς η λογική του αλγορίθμου μας έχει σχεδιαστεί να προκαλεί την εμφάνιση αυτών των "βημάτων", τα οποία αντιπροσωπεύουν τις συναλλαγές και δίνουν μια πρώτη εικόνα του πώς αυξάνονται τα συνολικά κέρδη μας ανά διάστημα. Παρότι τα βήματα φαίνονται χοντρικά, παρουσιάζουν πώς εξελίσσονται τα συνολικά μας κέρδη κατά τη διάρκεια του χρόνου, προσδίδοντας έτσι μια εικόνα της απόδοσης του αλγορίθμου μας.

Το διάγραμμα προκύπτει από την στήλη τιμών "Date" οπότε απεικονίζεται ακριβώς η ημερομηνία κατά την οποία πραγματοποιήθηκε η εκάστοτε αγορά ή πώληση μιας μετοχής.

Κάθε φορά που πραγματοποιείται ένα ανοδικό "σκαλοπάτι" (κίνηση προς τα πάνω), δηλώνει ότι αγοράζουμε μια νέα μετοχή, και ως αποτέλεσμα το συνολικό υπόλοιπο ('Balance') μειώνεται. Αυτό συμβαίνει επειδή αγοράζουμε μετοχές με την εκάστοτε δυνατή τιμή, και το υπόλοιπο μας πρέπει να προσαρμοστεί ανάλογα. Στην αρχή αυτής της ήπιας πτώσης του 'Balance' παρατηρούμε ότι η τιμή του 'Portofolio' ξεκινά να αυξάνεται. Αυτό αναμένεται, καθώς η τιμή του 'Portofolio' προκύπτει ως ο πολλαπλασιασμός του αριθμού των μετοχών που έχουμε στη διάθεσή μας με την τιμή της μετοχής κατά το κλείσιμο του χρηματιστηρίου για κάθε επιμέρους χρονική στιγμή. Δεδομένου ότι επιλέγουμε να επενδύουμε σε μετοχές με ανοδική τάση, παρατηρούμε ότι η αξία του 'Portofolio' μας αυξάνεται ως συνάρτηση της ανοδικής πορείας της κάθε μετοχής. Κάθε φορά που προσθέτουμε μια νέα μετοχή, το υπόλοιπο του χαρτοφυλακίου μειώνεται αλλά συνολικά παρατηρούμε αύξηση. Ωστόσο, λόγω της λογικής του αλγορίθμου μας (που απαιτεί την πώληση όλων των μετοχών που αγοράζουμε στο ίδιο έτος), η αξία του 'Portofolio' δεν προλαβαίνει να αυξηθεί σημαντικά, καθώς "εξαργυρώνεται" γρήγορα λόγω των αναγκαίων πωλήσεων όλων των μετοχών.

Τέλος, μπορεί κάποιος να παρατηρήσει ότι, λόγω του ότι επενδύουμε σε μια μετοχή μια φορά το χρόνο, ο συγκεκριμένος αλγόριθμος προσφέρει αρχικά χαμηλά κέρδη. Αυτό οφείλεται στο γεγονός ότι ξεκινούμε με ένα χαμηλό ποσό στον λογαριασμό μας (1\$), και χρειάζονται αρκετά χρόνια προτού αυτά τα χαμηλά ποσά αυξηθούν σημαντικά.

4. Αλγόριθμος Large

Στην περίπτωση της μεγάλης ακολουθίας δεδομένων, η ίδια απλή λογική εφαρμόζεται, αλλά εδώ αξιοποιούμε τη λογική του **"intra-day trading"**. Αντί να εξετάζουμε τη βέλτιστη αγοραπωλησία μιας μετοχής σε ένα χρονικό παράθυρο ενός έτους, περιορίζουμε το χρονικό διάστημα σε μία ημέρα. Συνεπώς, κάθε φορά που γίνεται μια συναλλαγή, αγοράζουμε ή πουλάμε μόνο ένα είδος μετοχής (εκείνη που υπολογίζουμε ότι συγκριτικά με τις άλλες αποφέρει το μέγιστο κέρδος). Η αγορά πραγματοποιείται όταν η μετοχή φτάνει την ελάχιστη τιμή της μια συγκεκριμένη ημέρα, ενώ η πώληση γίνεται όταν η μετοχή φτάνει τη μέγιστη δυνατή αξία της σε μια μέρα. Σε κάθε χρονική στιγμή του συνόλου δεδομένων, πραγματοποιούνται διαδοχικές συναλλαγές που αφορούν την ίδια μετοχή. Το κριτήριο για την επιλογή της μετοχής είναι το κέρδος που προσφέρει σε κάθε συγκεκριμένη μέρα, λαμβάνοντας υπόψη το διαθέσιμο κεφάλαιο για επένδυση (δηλαδή πόσες μετοχές μπορούμε να αγοράσουμε).

4a. Data Transformation

Σε αντίθεση με την περίπτωση Small, σε αυτήν την ειδική περίπτωση δεν υπάρχει ανάγκη για συγκεκριμένο μετασχηματισμό των δεδομένων, καθώς διατηρούμε καταγραφές για κάθε ημερήσια διακύμανση όλων των μετοχών και δεν απαιτείται χρονική ομαδοποίηση τους, όπως συνέβαινε προηγουμένως. Επίσης, δεν προστίθενται νέες βοηθητικές στήλες, αλλά απευθείας μπορεί κάποιος να βρει την ελάχιστη και μέγιστη τιμή για κάθε μετοχή σε κάθε ημέρα. Έτσι προκύπτουν οι στήλες ['Min_col'] = min: ['Open', 'Low',

'Close'] και ['Max_col'] = max: ['Open', 'High', 'Close'] με τις αντίστοιχες τιμές τους να αποθηκεύονται στις μεταβλητές 'Min_val', 'Max_val'.

Και εδώ είναι το ίδιο σημαντικό ο έλεγχος της συνθήκης ότι η χαμηλότερη τιμή της μετοχής μέσα στη μέρα (στιγμή κατά την οποία μας συμφέρει να αγοράσουμε τη μετοχή) προκύπτει νωρίτερα από όταν προκύπτει η υψηλότερη τιμή της μετοχής μέσα στη μέρα (στιγμή κατά την οποία μας συμφέρει να πουλήσουμε τη μετοχή), προκειμένου να έχει νόημα το intraday trading. Προκειμένου να κρατήσουμε στο dataframe μας μονάχα τις μετοχές S που πληρούν αυτό το κριτήριο, ελεγχουμε τις τιμές των μεταβλητών 'Min_col' και 'Max_col', προκειμένου να κρατήσουμε μόνο τις συναλλαγές που έχουν νόημα σε αυτό που θέλουμε να πετύχουμε.

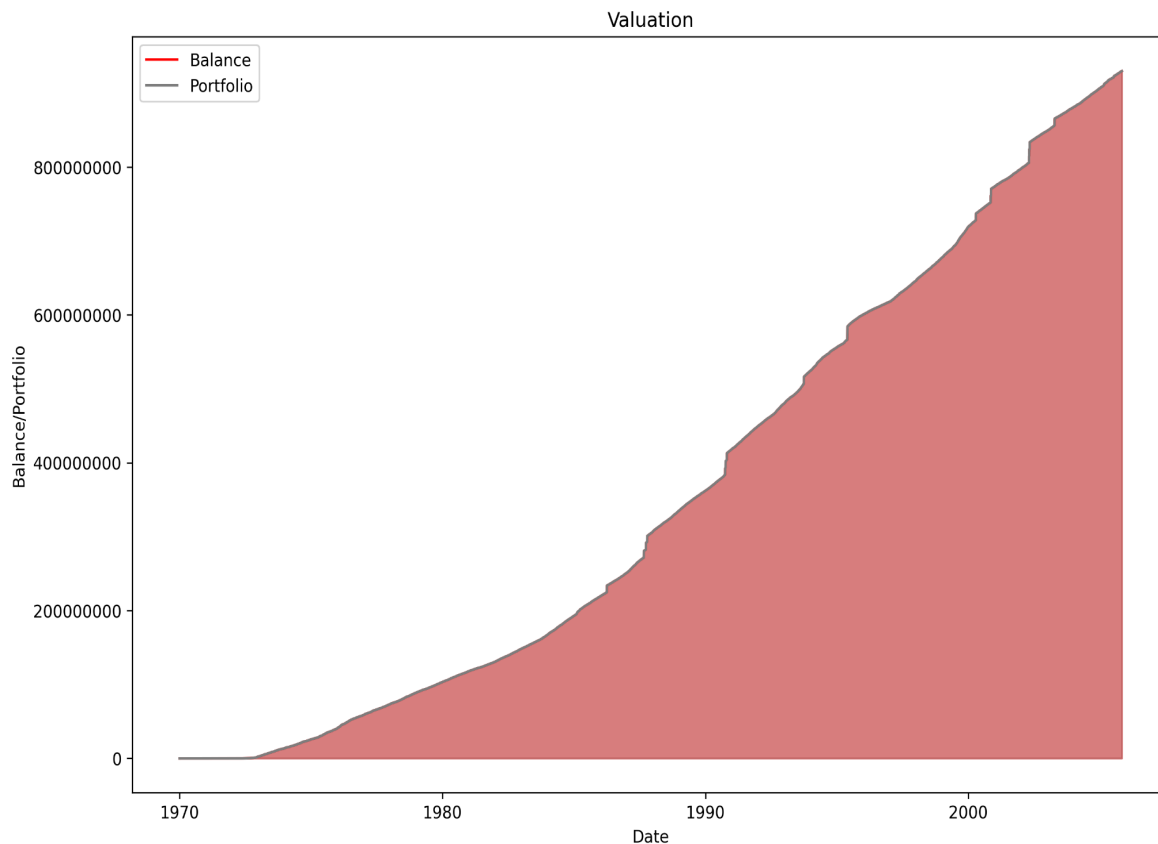
Εφόσον μόνο συγκεκριμένες μετοχές προχωρούν στο επόμενο στάδιο, ο όγκος των δεδομένων που επεξεργάζεται ο αλγόριθμός μας μειώνεται. Παρολα αυτά, διατηρούμε παράλληλα τις μετοχές και τις πληροφορίες που είναι ουσιώδεις για την επίτευξη σημαντικού κέρδους.

4b. The Algorithm (Large.txt)

Ο αλγόριθμος για την εύρεση της ακολουθίας N.Large είναι ουσιαστικά πανομοιότυπος με αυτόν που περιγράφεται παραπάνω για την ακολουθία N.Small. Δεν υπάρχει λόγος να επαναληφθεί η περιγραφή του. Η μόνη τροποποίηση και σημαντική διαφορά είναι ότι σε αυτή την περίπτωση, ο επαναληπτικός μας βρόγχος εκτελεί τη σειριακή προσπέλαση των δεδομένων μας όχι βάσει του έτους ('Year'), αλλά βάσει της ημερομηνίας ('Date'). Καθώς αναζητούμε το βέλτιστο ημερήσιο κέρδος συναλλαγών και όχι ετήσιο, το data frame μας έχει ταξινομηθεί χρονολογικά βάσει της στήλης 'Date', προκειμένου να εξελίσσεται χρονικά η ακολουθία. Έτσι, έχει δημιουργηθεί το αρχικό data frame N, στο οποίο προστίθενται οι μεταβλητές που μας ενδιαφέρουν για την εξαγωγή της τελικής ακολουθίας N.Large. Ωστόσο, εδώ υπάρχει ένας περιορισμός όσον αφορά τον αριθμό των σειρών που μπορεί να περιλαμβάνει η ακολουθία. Συγκεκριμένα, ο περιορισμός αυτός ορίζει ότι η ακολουθία δεν μπορεί να υπερβαίνει το μέγιστο όριο των 1.000.000 σειρών. Στα δεδομένα που αναφέρονται, παρατηρούμε ότι υπάρχουν συνολικά 57 έτη και έχουμε 2 συναλλαγές για κάθε μέρα του έτους. Συνεπώς, η ακολουθία για αυτά τα δεδομένα δεν μπορεί να περιλαμβάνει περισσότερες από $(57 * 365 * 2) = 41.610$ σειρές.

Ελέγχοντας την ακολουθία N.Large στην πλατφόρμα "The Time-Travel Project", παρατηρείται ότι για **25.982** συναλλαγές στο χρονικό διάστημα των ετών 1960-2017, το κέρδος ανέρχεται σε **925.536.853** δολάρια. Αυτό αποτελεί σχεδόν 3,5 φορές μεγαλύτερο κέρδος σε σύγκριση με τον αλγόριθμο των ετήσιων συναλλαγών. Παρατηρείται ότι η προσέγγιση του intra-day trading και η δυνατότητα πολλαπλών ημερήσιων συναλλαγών οδηγούν σε σημαντικά μεγαλύτερα κέρδη σε σύγκριση με τη λογική του αλγορίθμου Small, που επιτρέπει μόνο μια ετήσια συναλλαγή.

4c. Diagram (Large.txt)



Διάγραμμα.2: Διάγραμμα Αποτίμησης της Ακολουθίας N.Large

Σε αντίθεση με την περίπτωση της ακολουθίας N.Small, όπου μπορούμε να διακρίνουμε στο διάγραμμα αποτίμησης τα ετήσια "βήματα", η λογική της ακολουθίας N.Large διατηρεί τη λογική διακριτοποίησης του χρόνου, ωστόσο η μονάδα πλέον είναι η μέρα και όχι το ένα έτος και συνεπώς η απεικόνιση αποκτά πλέον χαρακτηρα χρονοσειράς του Balance ανά ημέρα. Βλέπουμε ότι η τάση εδώ είναι περισσότερο γραμμική (και προς το τέλος γίνεται πιο εκθετική), και αυτό οφείλεται στο ότι, επειδή έχουμε πολλές συχνές χρονικές συναλλαγές (μια αγορά-πώληση ανά ημέρα), αρχίζουμε από νωρίς να μαζεύουμε κέρδη. Παρατηρούμε βέβαια ότι ο ρυθμός αύξησης του κέρδους αυξάνεται με το χρόνο. Επιπλέον, μπορεί κανείς να παρατηρήσει κάποια μικρά "σκαλοπάτια" που προκύπτουν (σχεδόν κατακόρυφες αυξήσεις του κέρδους) που δείχνουν ότι εκείνη την ημέρα η επένδυσή μας ήταν κάπως πιο κερδοφόρα σε σχέση με τις ημέρες που προηγούνται και έπονται. Όσο πιο οριζόντια είναι η καμπύλη, τόσο λιγότερο κέρδος είχαμε εκείνη τη χρονική περίοδο, ενώ όσο πιο απότομη γίνεται η κλίση της τόσο μεγαλύτερο είναι ο ρυθμός αύξησης του κέρδους.

Πρέπει να σχολιαστεί πως, με βάση τη συγκεκριμένη αλγοριθμική λογική, το διάγραμμα που συνδυάζει την απεικόνιση του Balance (τα κέρδη που έχουμε στον τραπεζικό μας λογαριασμό) και του Portfolio (αριθμός των μετοχών που έχουμε στη διάθεσή μας, πολλαπλασιασμένος με την τιμή της μετοχής κατά το κλείσιμο του χρηματιστηρίου) δεν δίνει σημαντική πληροφορία. Αυτό οφείλεται στο γεγονός ότι, σύμφωνα με την

υπόθεση του αλγορίθμου, κάθε αγοραπωλησία μετοχών σε μια μέρα οδηγεί στο να μην έχουμε καμία μετοχή στην κατοχή μας στο τέλος της ημέρας. Αυτό ακριβώς επιβεβαιώνεται και στο διάγραμμα, καθώς οι δυο καμπυλες και η σκιαγραφημένη επιφάνεια τους ταυτίζονται (και έτσι το κόκκινο χρώμα της τιμης 'Balance' σε συνδυασμό με το γκρι χρώμα της τιμης 'Portfolio', δημιουργούν το ιδιαίτερο χρώμα του τελικού διαγράμματος).