Data Science and Machine Learning
National Technical University of Athens

Programming Tools and Technologies for Data Science

Time Travel Project

Nanos Georgios
03400144
nanosgiwrgos1997@gmail.com

"The truth is, time travel is hard, and people are lazy."
– Margaret Peterson Haddix, Redeemed, 2015.

Introduction

The report summarizes the strategy followed for preparation of the small and large sequence respectively, quoting the respective valuation charts. The stages of pre-processing and selection of data are briefly mentioned and it becomes a small one reference to how the diagrams are drawn.

The program contains the following files which are analyzed below:

- create_stocks.py
- buy_update.py
- write_plot.py
- time_travel.py
- grid_search.py

Pre-processing analysis

The primary goal is to select a subset of shares to create of the 2 sequences and of course the preparation of the appropriate modifications to the data, so that they are satisfied the restrictions given on pronunciation. The pre-processing of the data and the creation of a subset of the 7196 shares[1] is performed in the create _stocks.py function. The main reasons for using a subset of shares are the effort to maximize profits and the lack of computing power. The basic functions of the function performed are:

- Read the files
- Calculation of the stock growth approximation from the first day to the last
- Selection of N shares with the largest growth approximation
- Remove the "Open Int" column, as it does not provide information
- Sort data by date in ascending order
- Definition as Volume = 0.1 * Volume, as in each transaction there should not be more than 10
- Save the Dataframe generated as mystock.txt which contains the "good" shares used below

Methodology for finding sequences

The grid file _search.py calls the make _money function of the time _travel.py file, which accepts the parameters n, T and N _max. The parameters concern:

- N_max: Maximum number of moves implemented
- n: The number of shares that will be used to create mystocks.txt from the create _stocks.py function
- T: For how many extra days ahead is the control from the last move to make the decision for the next move

The time _travel.py file reads the dataframe $df = |Date|Open|High|Low|Close|Volume|Stock|$. In addition, dataframes are created and used:

- $profit = |date|profit|$, which saves after each transaction the date made and the currently available amount
- $bal = |date|balance|$, which saves the date and after each move the current valuation which is profit (profit) + value of holding shares * Volume profit) + value of our holding shares * Volume at the end of the day of the last move
- $moves\ _history = |date|move|stock|Xvolume|$, which stores the date, type of transaction, stock, and volume of stock purchased.
- $stocks\ _bought = |stocks|volume|pricepaid|$ which contains the shares we hold and their volume. It is used to update the amount purchased the share in order to sell it at a favorable price but mainly to update the volume in our possession since up to $n+1$ shares can be purchased every day if in the previous day in our possession there were n.

The logic that is followed is that there is a main repetition of control for the number of movements and the days we are. In order to decide whether the most appropriate move is the sale or the purchase, we check in the specific range of T days the shares that can be bought. If no shares can be bought in the specific range of T days then a check is made for the shares that can be sold, where a check is made if there is an advantageous option in relation to the initial price at which the shares were purchased. If there is an advantageous option then the sale is done differently, a check is made for the next period of T days. The logic that is followed is first to make as many purchases as the available amount allows and then to make the sales on the most advantageous dates of the T period. days where the datetrack is located and the balance (bal) valuation is renewed. The write_to_file.py function is used to create the .txt files and the plot_profit_balance function is used to design the valuation and profit charts, which accepts the profit and valuation dataframes respectively as an argument. profit and bal.

Due to lack of computing power, the grid_file_search.py was developed, which essentially calls the main program make_money with the parameters n (number of different shares), T (interval of days for the decision to choose the next move) and N_max (maximum movement limit). An attempt was made for exhaustive control and tests to find the most appropriate value of parameter T. Empirically we can say that the shares are about 50 years, therefore about 1800 days, so to achieve up to about 1000 moves need about $T = 100$ days per period, while for the large sequence the T must be between 5 days , a price which is prohibitive for the computing power of our mood.

Results of the sequences and their diagrams

For the small sequence, after exhaustive testing and trials we end up for $T = 94$ for maximum profit, making use of 8 "good" shares succeeding **629924782.33994 dollars**.

Figure 1: Check in the validator

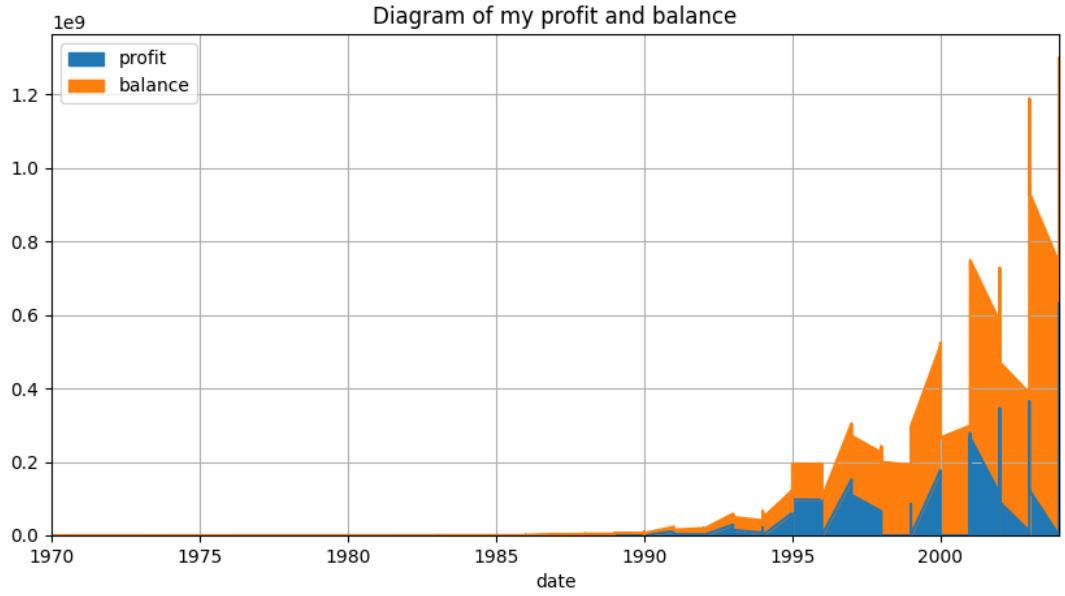

Figure 2: Valuation-profit chart

For the large sequence the exhaustive check for the suitability of the T parameter is prohibitive so after some values we end up indicatively for $T = 93$ for maximum profit, making use of 8 "good" shares succeeding **271828208.91104007 dollars**.

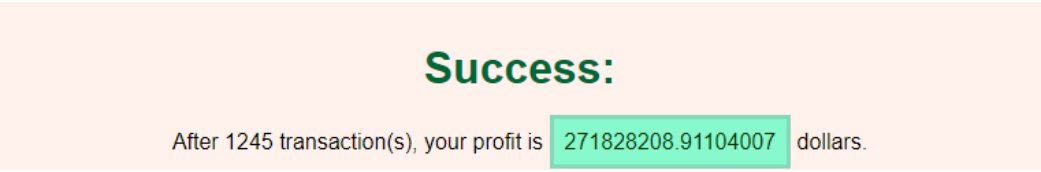

Figure 3: Check in the validator

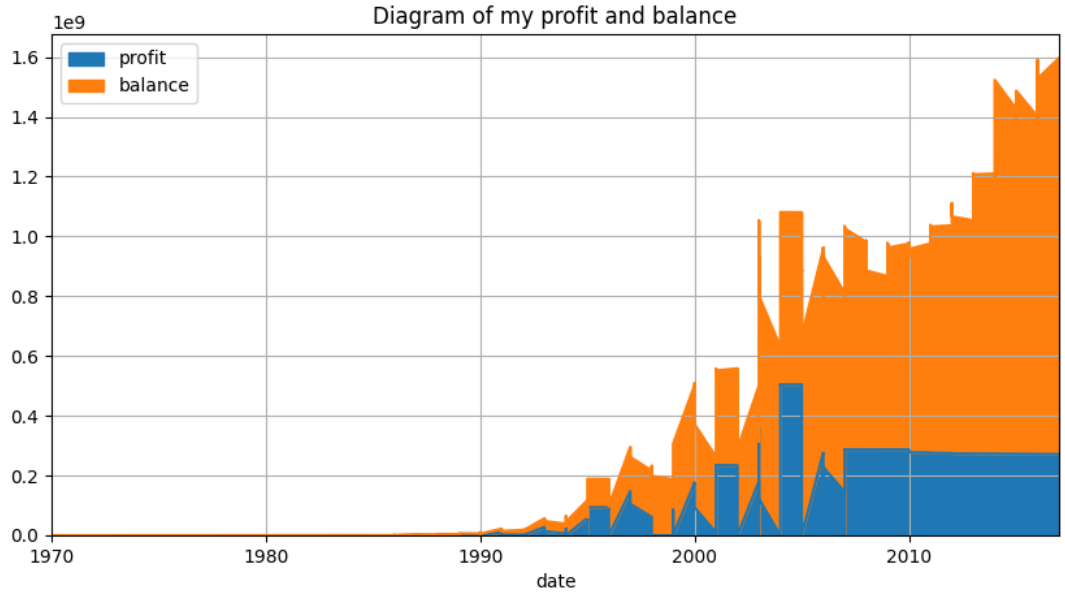

Figure 4: Valuation-profit chart

Possible improvements

Achieving maximum profit requires improving and increasing the available resources used. Also, an increase in profit can be achieved by using the entire dataset or more shares. Accounting regression in stocks could also be used to set a gradual increase in permitted transactions over the decades. Obviously, we observe that there is no linear dependence between movements and profit in the two sequences.

References

[1] Kaggle Boris Marjanovic. Huge stock market dataset, historical daily prices and volumes of all u.s. stocks and etfs. https://www.kaggle.com/borismarjanovic/price-volume-data-for-all-us-stocks-etfs/version/3.