Data-driven portfolio management

Back-testing and analysis of investing strategies

Programming for Data Processing Final Project

Author: Guillermo Vigueras & Raúl Gutiérrez

Institute: Computer Science Faculty - UPM

Date: February, 2022

Contents

Disclaimer

It should be noted that concepts, investment methods and assets data described in this document are provided with education purposes only and are not intended to provide specific advice or recommendations for any individual or on any specific security or investment product. Thus, it is only intended to provide an academic example of data harvesting, manipulation and analysis.

Chapter 1 Investment strategies generation

Based on the assets data harvested, according to first part of assignment, Smallville Asset Management wants to generate different investment strategies in order to later analyse and characterise each strategy. The main aspect that will impact the performance obtained by an investment strategy is:

- **Investment portfolio**: consists of defining what type of assets and the weight assigned to each type in the portfolio.

1.1 Investment portfolio

An investment portfolio is defined by mainly two input arguments:

- **Type of assets**: assets selected to invest in, among all the assets considered by SAM (see first part of assignment).
- **Asset allocation**: the weight assigned to each asset selected for the portfolio. The weight is typically expressed as a percentage. Thus, the weights for all assets within a portfolio must sum 1.0 or 100%.

Following an example of portfolio definition is shown according to typical representation used in finance and also adopted in SAM .

Example 1.1 Portfolio definition

As mentioned before, Smallville Asset Management has decided to form portfolios, assigning weights to the following assets:

- Stocks (ST)
- Corporate bonds (CB)
- Public bonds (PB)
- Gold (GO)
- Cash (CA)

Thus, a portfolio is defined by a sequence of numbers separated by slash symbol (/). The order of each number in the sequence indicates the type of asset and each number in the sequence represents the weight for that type of asset in the portfolio. If an asset should not be included in the portfolio, the assigned weight is 0.0. Table 1.1 shows some examples of portfolio mix definitions:

Based on the definition of portfolios explained, it is assumed that the investment of the available funds in each portfolio is made on the first day, that is, on `01/01/2020`. Thus, invested money is divided among the assets according to the defined portfolio allocation. Here is an example explaining the investment for a specific portfolio:

Example 1.2 Purchase of assets in an investment portfolio

Table 1.1: Examples of portfolio definition

Asset allocation					Portfolio mix
ST	CB	PB	GO	CA	ST / CB / PB / GO / CA
50%	20%	20%	0%	10%	50 / 20 / 20 / 0 / 10
10%	20%	20%	40%	10%	10 / 20 / 20 / 40 / 10
50%	40%	10%	0%	0%	50 / 40 / 10 / 0 / 0
0%	20%	20%	60%	0%	0 / 20 / 20 / 60 / 0

Suppose an investor has 100$ to invest using the following portfolio allocation: 40 / 20 / 0 / 20 / 20. Such 100$ are fully invested on the first day according to the asset allocation in the portfolio. The following table shows how the number of shares of each purchased asset is calculated, assuming a hypothetical price of the assets for the day `01/01/2020`:

Table 1.2: Portfolio investment based on allocation 40 / 20 / 0 / 20 / 40

Asset	Allocation (%)	Amount ($)	Price	# Shares
ST	40	40	12.5	3.2
CB	20	20	4	5
PB	0	0	–	0
GO	20	20	10	2
CA	20	20	1	20

Note that the number of shares to be purchased of each asset will depend on the price of the asset, except for cash (CA). So, for example, the investor would buy 3.2 shares (ST) since the price is 12.5$. For simplicity, it is assumed that a decimal number of shares can be purchased. Regarding the use of share prices, in the case of cash (CA), the price is not obtained from the `US Dollar index` (described first part of assignment), but rather the purchase price (or cost) is always considered as 1$.

1.2 Investment strategy performance

Smallville Asset Management has defined some metrics to analyse and classify a given portfolio that follows some trading methodology. These metrics are the following:

- **Return**: This metric used by SAM is the return of profit obtained by a portfolio. Return is computed as a percentage taking into account the price paid for every *buy* transaction of shares and the current value of portfolio, calculated using share's price at the time return is computed. If the current portfolio value is higher than the amount paid for shares bought, then return would be positive, otherwise a negative return would be obtained. Additionally, portfolio return is computed for a 12 months period. Thus, assuming the set of buy operations performed within the portfolio is defined as:

$$B = \{b^j\} \quad with \quad j \in Assets = \{ST, CB, PB, GO, CA\}$$

where each b^j is the number of shares of the j-th asset adquired. Thus, return can be computed as:

$$buy\ amount = \sum_{j \in Assets} (b^j * p_b^j)$$

$$current\ value = \sum_{j \in Assets} (b^j * p_c^j)$$

$$portfolio\ return = \frac{(current\ value - buy\ amount)}{buy\ amount} * 100$$

where p_b^j is the share price paid and p_c^j is the current share price, both for the j-th asset. Since return is computed for 12 months period, p_c^j is the price of each asset at `12/31/2020`.

- **Volatility**: This metric refers to the amount of uncertainty or risk related to the size of changes in an asset value. A higher volatility means that an asset value can potentially be spread out over a larger range of values. This means that the price of the asset can change dramatically over a short time period in either direction. A lower volatility means that an asset value does not fluctuate dramatically, and tends to be more steady.
  Volatility can be measured with different metrics. For this matter is computed based on the standard deviation of the evolution of an asset price over time. So, suppose x_i is the different daily prices of an asset with $1 \leq i \leq N$ where N is the number of days considered in a sample. First the standard deviation (σ_X) of the sample (X) is computed, using the sample average (μ_X) as:

$$\sigma_X = \sqrt{\frac{1}{N} \sum_{i}^{N} (x_i - \mu_X)^2} \quad where\ \ X = \{x_0, x_1, ..., x_N\}$$

  Then volatility of the asset is computed as a percentage with respect to the sample average (μ_X):

$$Volatility(X) = \frac{\sigma_X}{\mu_X} * 100$$

  In order to characterise an investment strategy, Smallville Asset Management computes the yearly (12 months) volatility of its associated portfolio. In this way, the sample X would contain prices of a given asset in the range of dates from `01/01/2020` to `12/31/2020`. For that matter, a temporal sequence with the daily values of the portfolio

$$Values = \{value_i\} = \{value_0, value_1, ..., value_N\}$$

  Where each individual $value_i$ of the portfolio is computed as:

$$value_i = \sum_{j \in Assets} shares_i^j * price_i^j \quad where \;\; Assets = \{ST, CB, PB, GO, CA\}$$

Note *$shares_i^j$ represents the number of shares for the j-th asset of the portfolio on the i-th day. This value is obtained from the number of shares bought on the first day (i.e. `01/01/2020`). On the other hand, $price_i^j$ represents the share price for the j-th asset on the i-th day. This information corresponds to the daily prices obtained from investing.com through web scraping (see first part of assignment).*

Based on the sequence of portfolio values defined above (i.e. $Values$), the volatility of an investment strategy is computed using such value sequence and $Volatility()$ function as:

$$Volatility\ of\ portfolio = Volatility(Values)$$

Note *Since return and volatility are calculated as percentages, it does not really matter what initial amount is invested (i.e. 1,000$, 10,000$, ...), since return and volatility values should be the same.*

1.3 Tasks related to investment strategies generation

Smallville Asset Management requires the following tasks to automatise the generation and evaluation of investment strategies:

1. **Portfolio allocation**: automatic generation of portfolio allocations must be generated, where the Δ or increment/decrement in each asset weight is 0.2 (20%). Note that the sum of asset weights for a portfolio mix must be always equal to 1.0 (100%). Table 1.3 shows the schema for portfolio generation.

   It should be noted that the number of different portfolio allocations can be computed a priori, since the 100% is split in 5 pieces, representing each piece a 20% of the portfolio. Table 1.4 tries to represent that:

   Thus according to Table 1.4, a portfolio with weights 40/20/0/20/20, would have the following distribution for the different asset types:

   - ST: 2 'pieces' of 20% each, for a total weight of 40% in the portfolio.
   - CB: 1 'pieces' of 20% each, for a total weight of 20% in the portfolio.
   - PB: 0 'pieces' of 20% each, since the asset weight in the portfolio is 0%.
   - GO: 1 'pieces' of 20% each, for a total weight of 20% in the portfolio.
   - CA: 1 'pieces' of 20% each, for a total weight of 20% in the portfolio.

   Thus, by representing this portfolio with weights 40/20/0/20/20, according to the notation in Table 1.4, we would obtain an assignment like the one shown in Table 1.5.

Table 1.3: Portfolio automatic generation with different asset weights

Asset Alloc.	ST	CB	PB	GO	CA
1	100%	0%	0%	0%	0%
2	80%	20%	0%	0%	0%
3	80%	0%	20%	0%	0%
4	80%	0%	0%	20%	0%
5	80%	0%	0%	0%	20%
6	60%	20%	20%	0%	0%
7	60%	20%	0%	20%	0%
		...			
i	40%	60%	0%	0%	0%
i+1	40%	0%	60%	0%	0%
i+2	40%	0%	0%	60%	0%
i+3	40%	0%	0%	0%	60%
		...			
i+n	0%	100%	0%	0%	0%
		...			

Table 1.4: Description of portfolio generation with $\Delta = 20\%$

20%	20%	20%	20%	20%
ST	ST	ST	ST	ST
CB	CB	CB	CB	CB
PB	PB	PB	PB	PB
GO	GO	GO	GO	GO
CA	CA	CA	CA	CA
?	?	?	?	?

Table 1.5: Portfolio generation with $\Delta = 20\%$

20%	20%	20%	20%	20%
ST	ST	ST	ST	ST
CB	CB	CB	CB	CB
PB	PB	PB	PB	PB
GO	GO	GO	GO	GO
CA	CA	CA	CA	CA
ST	ST	CB	GO	CA

On the other hand, according to the previous formulation of the portfolio generation based on segments of 20% each, the number of different portfolios can be computed as n-combinations with repetition of m elements (element = asset), where n=5 and m=5. Thus, the number of different allocations to be computed can be calculated as:

$$CR_m^n = \binom{m+n-1}{n} = \frac{(m+n-1)!}{n! * (m-1)!} = 126\ portfolio\ allocations$$

Note *This number is just a reference for checking if your python code computes 126 portfolio allocations.*

As part of this subtask, portfolio allocations generated should be stored in the file `portfolio_allocations.csv` in order to be used in later tasks and also to allow individual validation of this subtask.

2. **Portforlio performance**: Compute the 12 month (12M) metrics from `01/01/2020` to `31/12/2020` of the 126 investment portfolio previously generated. Metrics of each portfolio must be computed according to description in Section 1.2.

   The information generated from the different portfolios and the annual return must be stored in a file named `portfolio_metrics.csv` to be used later in the analysis task and also to allow the individual validation of this subtask.

   Such file `portfolio_metrics.csv` must have 7 columns and 126 rows, one row for each generated portfolio. The columns must be in the order:

   ```
   ST,CB,PB,GO,CA,RETURN,VOLAT
   ```

   Thus, the first 5 columns represent the weight of each asset in each portfolio (i.e. row) and the sixth and seventh columns represent the annual return and volatility of each portfolio, respectively.

Note *It is important to note that, it might be case that prices data obtained through Web Scraping, might have missing values for the date ranges considered for this exercise. Missing values have to be properly accounted for in order to correctly compute portfolio metrics. Thus, how to deal with missing values and the best filling strategy is a decision of your team.*

Chapter 2 Investment strategies analysis

With all the portfolio information generated in previous tasks, Smallville Asset Management wants to develop some model for automatic financial advisoring of its customers. But before starting with predictive models, SAM Data Science (DS) manager wants your team to perform some data analysis.

2.1 Tasks related to investment strategies analysis

Smallville Asset Management requires the following tasks to perform the analysis of investment strategies:

1. **Return**: SAM Data Science (DS) manager has asked you to analyse the distribution of returns obtained from the different portfolios generated. The analysis must be performed for the time period mentioned previously, i.e. 12 months (12M), from `01/01/2020` to `31/12/2020`. For such analysis, generate the plots and/or data your team considers necessary. According to the analysis performed, try to answer the following quesiton:

   Taking into account **ALL** generated returns, **does your team think it is more probable to obtain a positive or negative return?**

   Use the previous generated plots and/or data to support your answer to the question.

2. **Return vs. risk**: Your team has been requested to analyse the relation among return and risk for the 12M period. Return vs. risk is typically analysed in finance displaying a 2D plot with a measure of risk in the 0X axis and return in the 0Y axis. In this case, risk will be quantified using volatility. According, to the analysis performed, try to answer the following question:

   Does your team think it is **ALWAYS** true that **the higher the risk, the higher the obtained return is?**

   Use previous generated plot and/or data to support your answer to the question.

Chapter 3 Grading criteria and submission instructions

3.1 Grading criteria

Grading weights and criteria are:

- **60%** - **Implementation**: regarding the implemented code it will be evaluated if the required functionality is fulfilled, if the code is well organized and well documented and wether the code adopts a data programming paradigm or not, making a proper use of all Python and Pandas data structures and methods reviewed in class.
- **40%** - **Data analysis**: regarding analysis it will be evaluated wether questions are answered with a thorough explanation based on supporting evidence (i.e. plots and/or data), also if the appropriate evidence (plots/data) is used depending on the question addressed. Note that the analysis part is independent of the implementation part. In this way, analysis part can be correct even if is based on wrong data due to some programming error during implementation.

3.2 Submission instructions

- One submission (and unique) by **each group of students**.
- Submission will be done by emailing a .zip file
  - Emailed to: gvigueras@fi.upm.es
  - Email subject: `[PDP] Final assignment Group-<groupNumber>`
  - Email body must state your student group (e.g. Group 7) and list group members
- **Due date for submission is Thursday 21/04/2022 until 23:59**
- The following content should be included in your .zip file submission:
  - The Python code developed by your group, providing solution to the final project
  - A report regarding data analysis questions in Chapter 2. This report must include a brief answer to each question, supported by plots, tables or any other data representation you consider relevant. Please, be concise in your answer, taking into account that a longer answer is not always a good answer. The report must be a `.pdf` file, and the format can be freely selected by each group, either slides or document.
  - In addition, each group must upload to the project corresponding, a plain text file (e.g., README.txt) in which the following is briefly described:
    - Description of the content (files and dirs.) of the root directory of your practice, it is not necessary to describe the content of each subdirectory.

- The final project statement is divided in 3 parts: web scraping, data generation and data analysis. Thus, a brief explanation of how to run your code for each part (either explaining the instructions for the steps to follow or a Makefile or equivalent file). Also indicate python packages and versions required by your code.
- A list of the Python modules/files included in your project, briefly indicating a description of them. There is no need to describe each file on a function basis, simply a general description of the file. For example: file `File1.py` computes the different asset allocations for portfolios.
- What functionality required have been addressed in the code, relating (if possible) project requirements with the code implemented. For example, indicate: functionality about web scraping has been addressed in files `File1.py`, `File2.py`, ...