Software verification and validation in portfolio optimization

Attila Rácz

Table of contents

1. Overview
2. Flow of the process
3. Verifying input data
4. Verifying optimization code
5. Check of classification

Overview

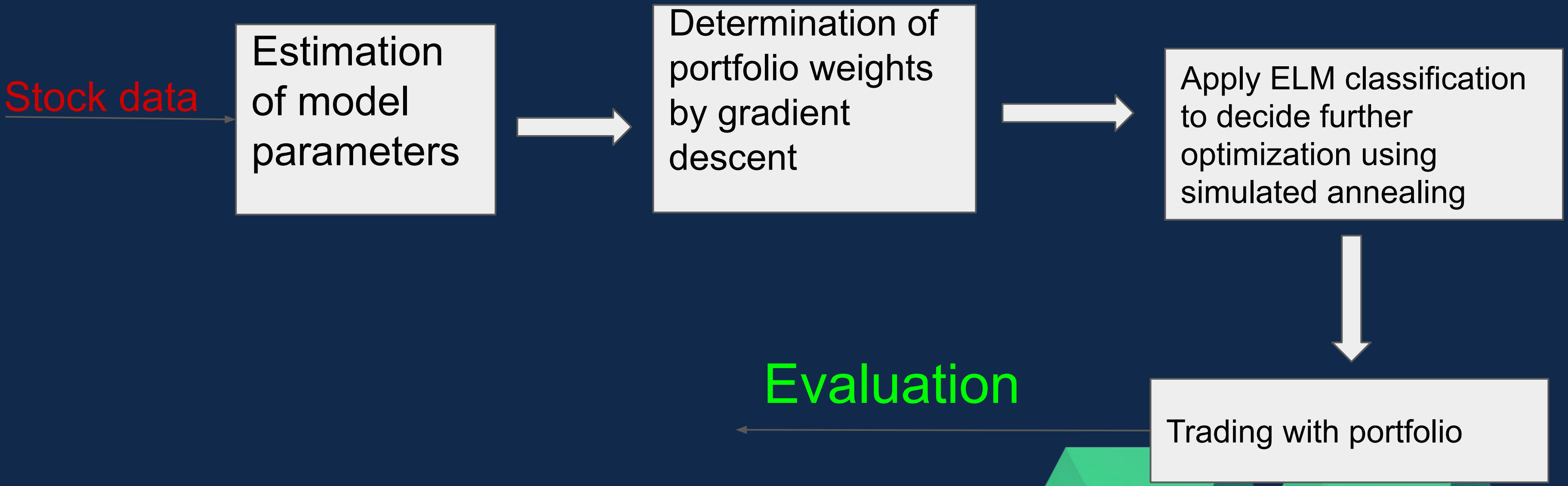
- **Portfolios**: linear combination of assets to minimize risk
  - Several aspects to create portfolio: i.e. Markovitz
  - today we see sparse mean reverting portfolios.

To make it we need:

- historical stock data
- a model to anticipate tendencies.
- optimization strategy

then trading...

Flow of the process



Stock data

Estimation of model parameters

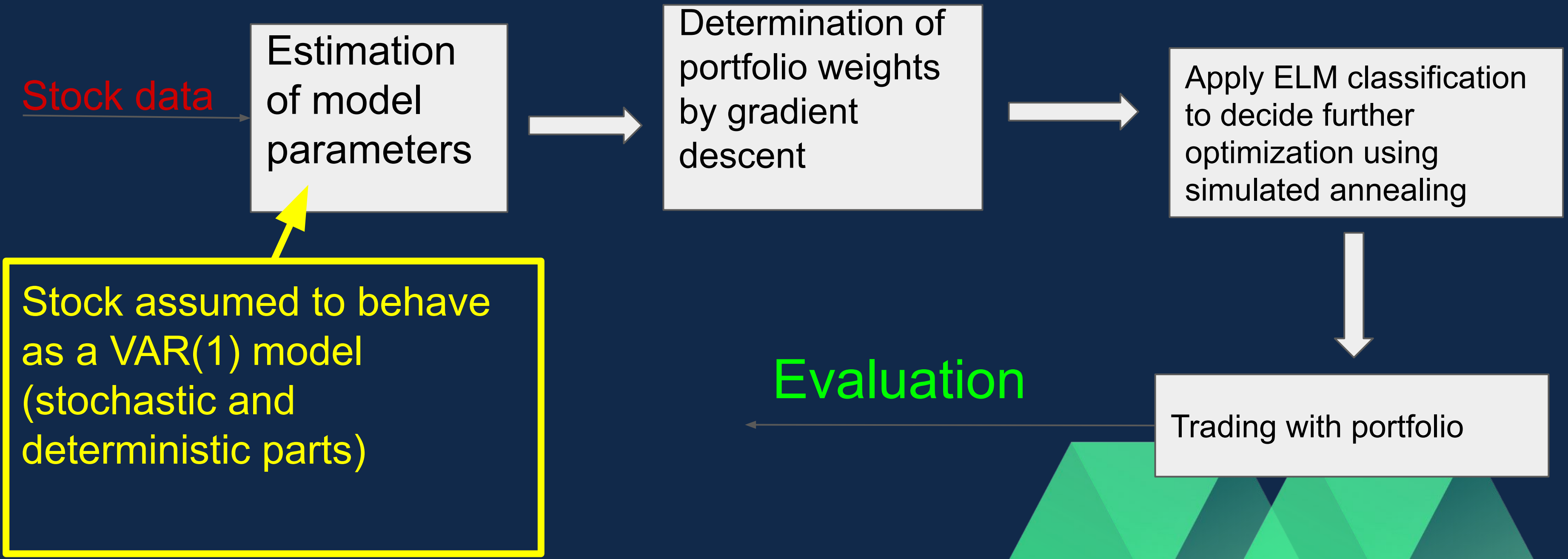
Determination of portfolio weights by gradient descent

Apply ELM classification to decide further optimization using simulated annealing

Trading with portfolio

Evaluation

Flow of the process



Stock data

Estimation of model parameters

Determination of portfolio weights by gradient descent

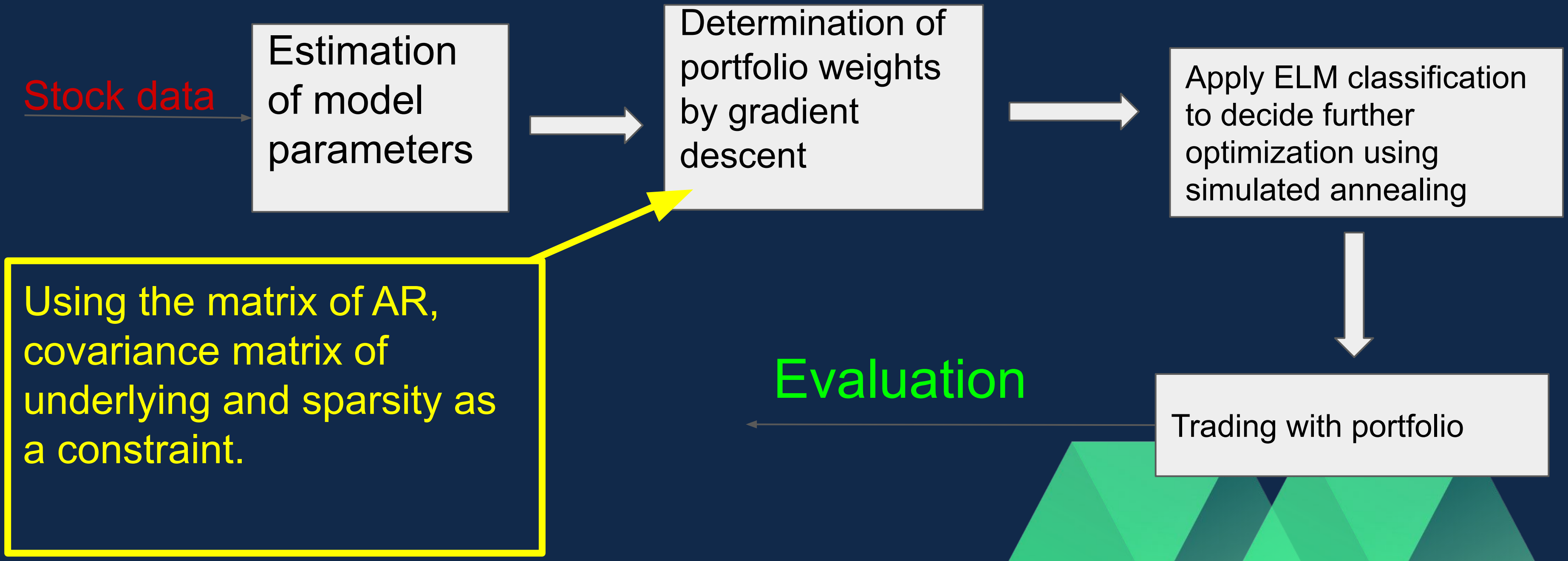
Apply ELM classification to decide further optimization using simulated annealing

Trading with portfolio

Evaluation

Stock assumed to behave as a VAR(1) model (stochastic and deterministic parts)

Flow of the process



Stock data

Estimation of model parameters

Determination of portfolio weights by gradient descent

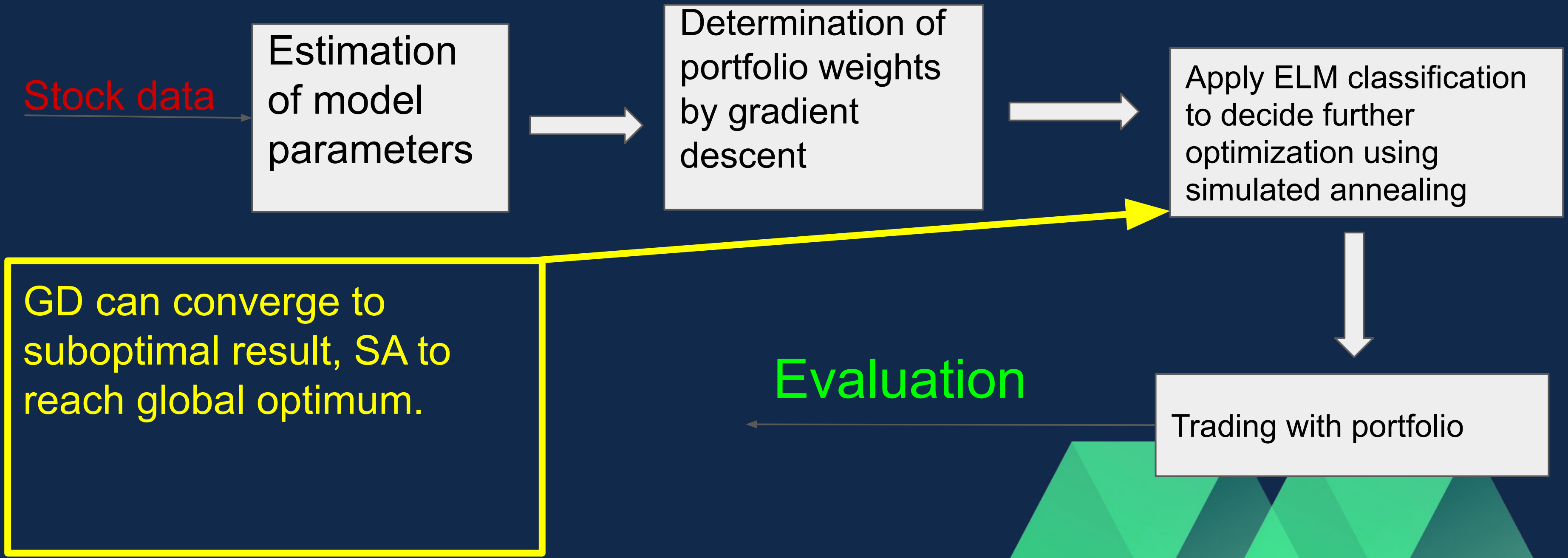
Apply ELM classification to decide further optimization using simulated annealing

Trading with portfolio

Evaluation

Using the matrix of AR, covariance matrix of underlying and sparsity as a constraint.

Flow of the process



Stock data

Estimation of model parameters

Determination of portfolio weights by gradient descent

Apply ELM classification to decide further optimization using simulated annealing

Trading with portfolio

Evaluation

GD can converge to suboptimal result, SA to reach global optimum.

Input data

- Gathering stock data from yahoo finance, using stocks in S&P500
- Check the stock data
  - verify data is bounded below by zero and some predefined max value above
  - retrieve the same data for the same date

Input data

- Stocks can move out or into the S&P500
  - different length of historical data
- Missing data due to close of stock exchanges (assets can be traded on multiple SE) should be identified and handled (ie interpolate with VAR(1) model)

Optimization code

- Verifying gradient descent method against brute force method
- brute force provide the exact global optimum, gradient descent should be always above it.
- Simulated annealing convergence

Optimization code

(Gradient descent and Simulated annealing)

- Unit test for Metropolis algorithm: check transition probability
    - Temperature parameter affects the prob of transition into state with higher energy, count the frequency of transitions to higher states.
- check optimization for exact analytic function with known minimum

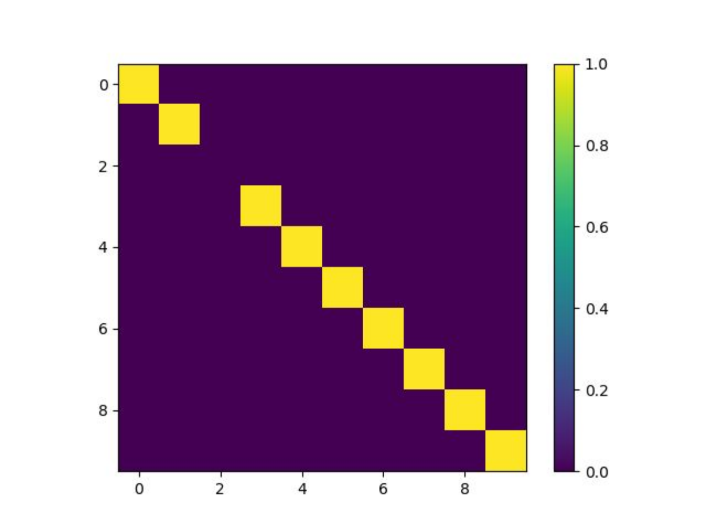
Optimization code

(Gradient descent and Simulated annealing)

- Regression test: give the same result for same input when code is modified
    - save portfolios with several combinations of assets
    - recalculate portfolios to compare saved values

Check of classification

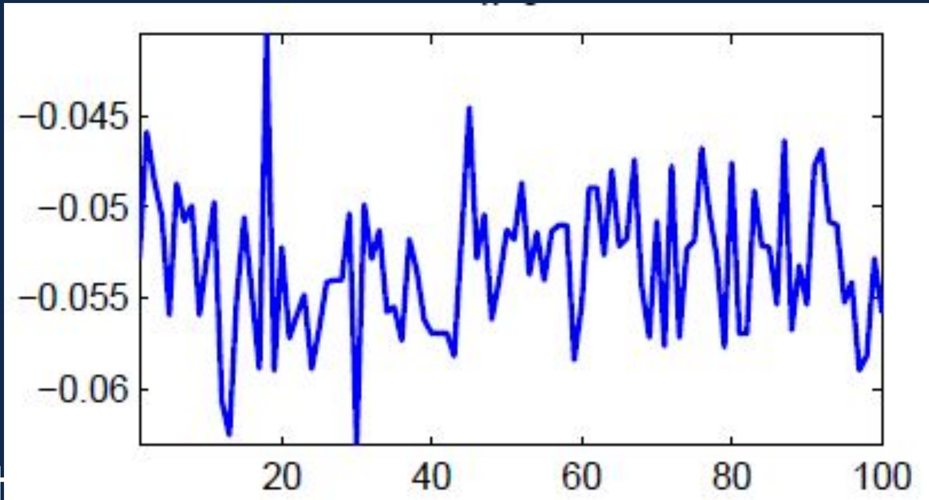
- ELM to decide which optimization should be used
- Algorithm contains the Moore-Penrose inverse calculation
  - Product of the matrix and its MP inverse should results the identity



Check of classification

- Check accuracy after learning
  - Necessity of classification is twofold
    - use Metropolis algorithm in case when gradient descent (time consuming compare to GD) is far from optimal
    - omit if GD is close to optimal
    - test: optimization result can be labeled as need4SA and noneed4SA by brute force method, then to compare against the decision of EML. Crucial to avoid case when BF labeled as need4SA while classification bucketed into noneed4SA.

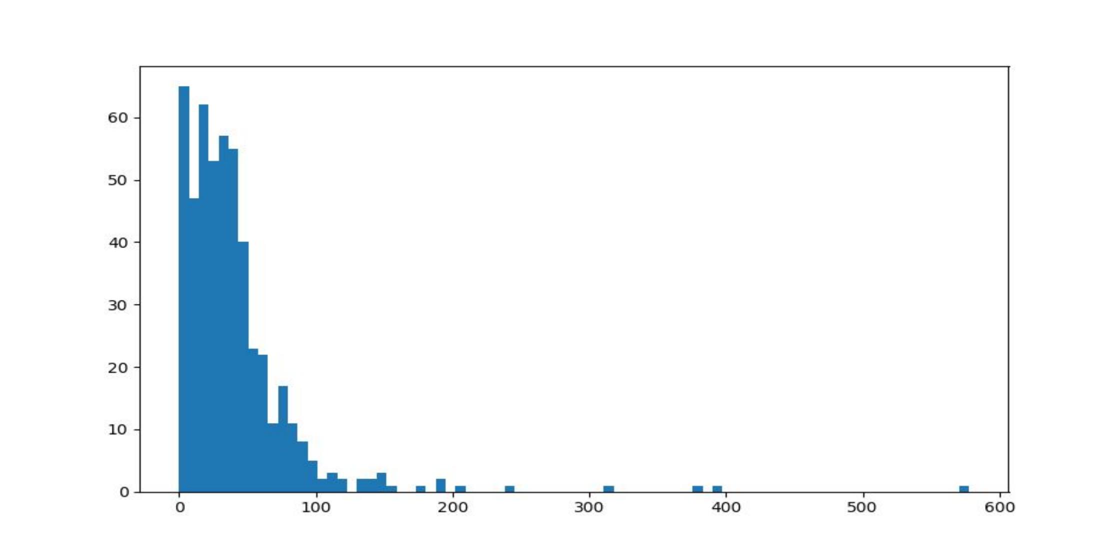
Stationarity test



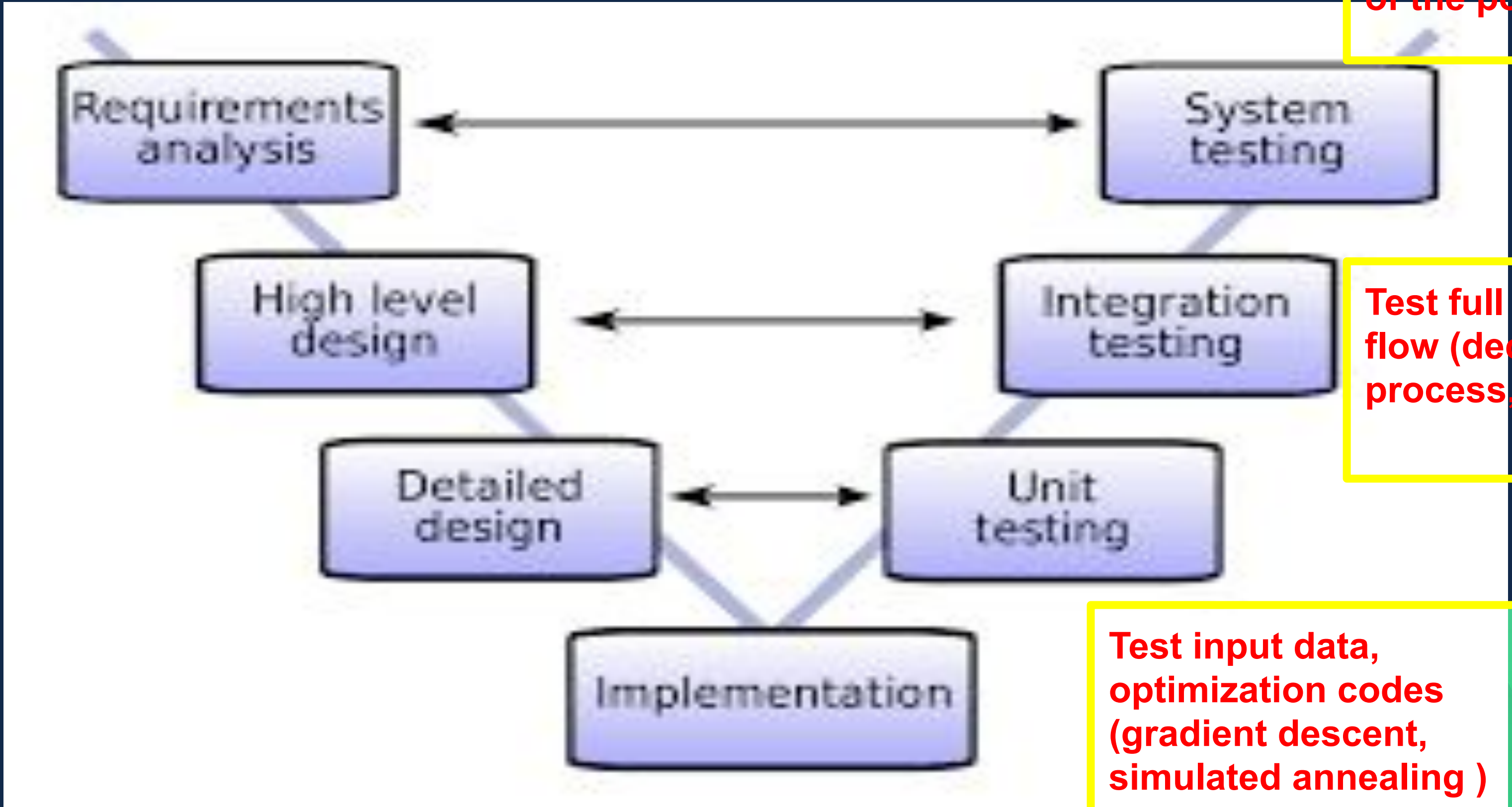
The portfolio has mean reverting property in order to have high predictability. Stationarity property checked by autoregression calculation. Handling loss or absence of stationarity is subject of current developement.

Validation

Method of validation is the evaluation distribution of return.



Summary



Distribution of return of the portfolio

Test full optimization flow (decision process, stationarity)

Test input data, optimization codes (gradient descent, simulated annealing)

Thanks for your attention!