CF969-7-SP/ZU: Machine Learning for Finance
Assignment 1

Preamble

This is Assignment 1 for CF969-7-SP and CF969-7-ZU. This is NOT for students in CF969-7-PT.

Familiarise yourself with the academic integrity and academic offences policy. More information can be found at the CSEE PGT Handbook and the Academic Offence section of the University's webpage

Objective

In this assignment, you will use the Capital Asset Pricing Model (CAPM) (*don't worry; details follow below*) to estimate expected returns and risks for a portfolio of assets. You are then requested to apply quadratic programming to optimise the portfolio based on these estimates.

Task: Portfolio Optimisation with Linear Regression

You are expected to perform the following subtasks.

1. Data collection (5%)

You are asked to download historical stock price data for at least 10 assets (e.g., from Yahoo Finance) over a 5-year period. Based on these, calculate daily returns for each asset.
Furthermore, download market index data (e.g., S&P 500 with ticker symbol `^GSPC`) for the same period.

2. Linear regression for expected returns and risk (15%)

Use the Capital Asset Pricing Model (CAPM) to estimate the expected returns and risks for each asset. The CAPM is given by:

$$R_i - R_f = \alpha_i + \beta_i(R_m - R_f) + \epsilon_i,$$

where:

- R_i is the return of the asset,

- R_f is the risk-free rate. For the purposes of the assignment, set it to 0.02.

- R_m is the market return,
- α_i and β_i are the regression coefficients,
- ϵ_i is the error term.

Perform linear regression to come up with values for α_i's and β_i's (not the ϵ_i's).
Use the estimated β_i to calculate the expected return for each asset:

$$\mu_i = R_f + \beta_i(E[R_m] - R_f),$$

where $E[R_m]$ is the average market return over the period.
Use the residuals (the ϵ_i's) from the regression to estimate the idiosyncratic risk (variance of ϵ_i) for each asset. The idiosyncratic risk is calculated as

$$\text{Idiosyncratic Variance} = \text{Var}(\epsilon_i) = \text{Var}(Y_i - (\alpha_i + \beta_i X))$$

where

$$Y_i = R_i - R_f.$$

3. Portfolio optimisation (20%)

Construct the covariance matrix $\mathbf{\Sigma}$ for the assets:

- Use the market variance σ_m^2 and the estimated β_i's to compute the systematic risk component:
$$\sigma_{ij} = \beta_i \beta_j \sigma_m^2 \quad (\text{for } i \neq j).$$
- Add the idiosyncratic risk to the diagonal elements:
$$\sigma_{ii} = \beta_i^2 \sigma_m^2 + \text{Var}(\epsilon_i).$$

Formulate the portfolio optimisation problem as a quadratic programming (QP) problem:

$$\text{Minimize } \frac{1}{2}\mathbf{w}^T \mathbf{\Sigma} \mathbf{w}$$

subject to:

$$\mathbf{w}^T \mu = \mu_p, \quad \mathbf{w}^T \mathbf{1} = 1, \quad \mathbf{w} \geq 0,$$

where:

- $\mathbf{w}$ is the vector of portfolio weights,
- μ is the vector of expected returns (from Step 2),
- μ_p is the target portfolio return.

Use a QP solver (e.g., `gurobipy`) in Python to find the optimal portfolio weights for a given target return μ_p.

4. Efficient frontier (20%)

Vary the target return μ_p and compute the corresponding minimum portfolio risk (standard deviation). Use at least 10 equally spaced different values for μ_p.

Plot the efficient frontier, showing the trade-off between risk and return.

5. Discussion (40%)

This is connected to Deliverable 2 below.
Interpret the results of the linear regression (e.g., significance of α_i and β_i).
Analyse the optimal portfolio weights and their sensitivity to changes in expected returns and risks.
Discuss the results and the shape of the efficient frontier and its implications for portfolio diversification.
Explain any coding choices you made, including the choice of μ_p values.

Deliverables

1. Code

Submit a well-documented script (.py or .ipynb) that implements the above tasks

2. Report

A PDF report (max 5 pages excluding title page) detailing your methodology, results, and interpretations. Include visualisations (e.g., efficient frontier, regression plots) and tables (e.g., portfolio weights, regression coefficients).

Evaluation Criteria

- **Correctness**: Accuracy of the mathematical formulations, code implementation, and results
- **Insightfulness**: Depth of analysis and interpretation of the results
- **Clarity**: Quality of the report and presentation, including visualisations and explanations

Suggested Libraries

`yfinance`, `numpy`, `pandas`, `scipy`, `gurobipy` or `cvxopt`, `statsmodels` or `sklearn`, `matplotlib`