Smart Beta Portfolio and Portfolio Optimization

REVIEW

CODE REVIEW

HISTORY

Meets Specifications

Your hard work shows in this project, good job completing this project and the all the best for the remaining program 👍

Part 1: Smart Beta Portfolio

✓

The function `generate_dollar_volume_weights` computes dollar volume weights.

The method `generate_dollar_volume_weights` accurately computes the dollar volume weights 👍

✓

The function `calculate_dividend_weights` computes dividend weights.

Using cumsum() did the trick! The method `calculate_dividend_weights` accurately calculates the dividend weights ✅

✓

The function `generate_returns` computes returns.

✓

✓

The function `generate_weighted_returns` computes weighted returns.

Perfectly computed, well done!

✓

The function `calculate_cumulative_returns` computes cumulative returns.

The `calculate_cumulative_returns` was a bit hard to implement compared to the earlier functions but you pass the test case successfully!

✓

The function `tracking_error` computes tracking error.

Tracking error is absolutely correct according to the test case 👍

Part 2: Portfolio Optimization

✓

The function `get_covariance_returns` computes covariance of the returns.

Covariance returns are absolutely correct!

✓

The function `get_optimal_weights` computes optimal weights.

A lot of functions had to be used in this method and you have used them to produce the required results.

✓

The function `rebalance_portfolio` computes weights for each rebalancing of the portfolio.

The ETF weights are rebalanced correctly, `get_optimal_weights` and `get_covariance_returns` are used correctly to compute rebalanced weights 👍

✓

The function `get_portfolio_turnover` computes cost of all the rebalancing.

Perfectly done, the formulas have been correctly translated into code 👍

DOWNLOAD PROJECT

RETURN TO PATH