UDACITY

‹ Back to AI for Trading

Smart Beta Portfolio and Portfolio Optimization

REVIEW

CODE REVIEW

HISTORY

Meets Specifications

Awesome! 😃 you have done a wonderful job. 🏆 Congratulations 👏👏

Part 1: Smart Beta Portfolio

The function `generate_dollar_volume_weights` computes dollar volume weights.

Great Work. Your function correctly computes dollar volume weights.

The function `calculate_dividend_weights` computes dividend weights.

Awesome. 😃 Your function correctly computes dividend weights. Nice use of `cumsum`

The function `generate_returns` computes returns.

Wonderful job, nice and succinct code. 👏

The function `generate_weighted_returns` computes weighted returns.

Good work. You correctly computed weighted returns.

The function `calculate_cumulative_returns` computes cumulative returns.

👍

The function `tracking_error` computes tracking error.

Good work. You correctly computed tracking error.

Part 2: Portfolio Optimization

The function `get_covariance_returns` computes covariance of the returns.

Well done, an alternative fast way:

```
return np.cov(returns.fillna(0), rowvar=False)
```

The function `get_optimal_weights` computes optimal weights.

Great Work. 😃 This was a tough function to implement, you successfully computed the optimal weights.

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

👍

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

Excellent work!

DOWNLOAD PROJECT

RETURN TO PATH

Rate this review