PortOpt: Goals and Charter Document

Project Title: PortOpt

Description: PortOpt is a Python-based program that implements various portfolio optimization techniques. It is designed to handle historical stock price data, optimize portfolio weights using different mathematical models, and provide a robust framework for practical and theoretical portfolio management.

Goals

1. **Develop a Modular Framework:** Create a modular structure where each optimization method is implemented as an independent module.
2. **Implement Core Optimization Methods:** Include the following methods:
    - Mean-Variance Optimization
    - Risk Parity
    - Black-Litterman Model
    - Sharpe Ratio Maximization
    - Robust Optimization
    - Machine Learning-Based Optimization
3. **Ensure Practical Usability:** Allow users to input historical stock price data (e.g., in CSV format) and output optimal portfolio weights with key performance metrics such as return, risk, and Sharpe Ratio.
4. **Provide Visualization Tools:** Generate clear, informative visualizations for portfolio allocations, risk metrics, and performance comparisons.
5. **Enable Scalability:** Design the system to support additional optimization methods, alternative risk measures, and integration with real-time market data APIs.

Charter

Objectives

The primary objective of PortOpt is to serve as an educational and practical tool for portfolio optimization. It aims to:

- Showcase various optimization techniques and their mathematical foundations.
- Allow comparison of methods on historical data for performance analysis.
- Provide an extendable platform for advanced features like risk budgeting and dynamic rebalancing.

Scope

PortOpt will:

- Handle data preprocessing, including normalization and handling missing values.
- Support optimization over any asset class with time-series data.
- Focus on academic and professional applications in quant finance and fintech.
- Include documentation for each module, describing usage and methodology.

Features and Modules

1. **Data Loader:**
   - Reads and preprocesses historical stock price data (CSV format).
   - Handles missing data (-1 values) and normalizes inputs.
2. **Optimization Modules:** Implement the following methods:

   Mean-Variance Optimization: Minimizes portfolio variance for a given return.
   Risk Parity: Equalizes risk contribution across assets.
   Black-Litterman Model: Combines market equilibrium with investor views.
   Sharpe Ratio Maximization: Maximizes risk-adjusted returns.
   Robust Optimization: Accounts for parameter uncertainty in optimization.
   ML-Based Optimization: Uses machine learning to predict optimal weights or returns.
3. **Visualization Tools:**
   - Display portfolio allocations and risk metrics.
   - Compare performance across methods.
4. **Scalability:**
   - Support additional methods (e.g., multi-objective optimization).
   - Integrate real-time market data APIs.

Technical Stack

- **Programming Language:** Python
- **Libraries:**
  - Data Handling: `pandas`, `numpy`
  - Optimization: `scipy.optimize`, `cvxpy`
  - Machine Learning: `tensorflow`, `pytorch`
  - Visualization: `matplotlib`, `plotly`

Stretch Goals

1. Implement risk budgeting for custom risk tolerance.
2. Add dynamic allocation and rebalancing strategies.
3. Provide real-time data integration using APIs.
4. Explore alternative optimization techniques, such as genetic algorithms.

Timeline

- **Phase 1: Infrastructure (Weeks 1-2):** Implement data loader and preprocessing.
- **Phase 2: Core Methods (Weeks 3-6):** Develop optimization modules one by one.
- **Phase 3: Testing and Validation (Weeks 7-8):** Test with historical datasets and validate results.
- **Phase 4: Visualization and Scalability (Weeks 9-10):** Add visualization tools and integrate additional features.
- **Phase 5: Documentation and Finalization (Weeks 11-12):** Write user documentation and refine the system.

Conclusion

PortOpt will be a comprehensive and modular portfolio optimization platform. It will not only provide a hands-on understanding of optimization techniques but also demonstrate practical applications for academic and professional use in quantitative finance.