# 📊 Stock Portfolio Analysis and Optimization

This project provides tools to analyze and optimize a stock portfolio by calculating key metrics, preprocessing stock data, and visualizing portfolio allocations using Python. The project is implemented in a Jupyter Notebook and includes various modules for preprocessing, metrics calculation, and portfolio optimization.

## ✨ Features

1. **🔄 Data Preprocessing**:
   * Cleans and preprocesses historical stock data.
   * Calculates daily returns, log returns, rolling mean, and rolling standard deviation.
2. **📈 Portfolio Metrics Calculation**:
   * Computes portfolio return, portfolio risk, and Sharpe ratio.
   * Automatically assigns equal weights for assets if not explicitly provided.
3. **📊 Portfolio Optimization**:
   * Uses optimization techniques (e.g., Sharpe ratio maximization) to calculate the optimal allocation of assets.
   * Supports customizable constraints like risk-free rate and bounds for asset weights.
4. **🎨 Visualizations**:
   * Generates plots for adjusted closing prices of multiple stocks.
   * Creates a pie chart to visualize portfolio allocations.

## 🗂 Files

* **stock\_analysis.ipynb**: The main Jupyter Notebook containing the entire workflow.
* **portfolio\_data.csv**: Example dataset containing historical stock prices.
* **Additional Modules**:
  + Functions for preprocessing data.
  + Portfolio metrics calculation.
  + Portfolio optimization using SciPy.

## 🛠 Installation

1. Clone this repository:

git clone <repository\_url>

cd <repository\_name>

1. Install required libraries:

pip install -r requirements.txt

1. Ensure that portfolio\_data.csv is available in the project directory.

## 🚀 Usage

1. Open the Jupyter Notebook:

jupyter notebook stock\_analysis.ipynb

1. Follow the step-by-step workflow to:
   * Preprocess stock data.
   * Calculate portfolio metrics.
   * Optimize the portfolio allocation.
   * Visualize the results.
2. Customize the dataset (portfolio\_data.csv) with your stock symbols and historical prices.

## 🔍 Key Findings

* **Trend Analysis**: Visualized adjusted closing prices for AAPL, MSFT, GOOGL, and TSLA.
* **Optimized Portfolio Allocation**: Identified optimal asset weights to maximize the Sharpe ratio.
* **Risk Management**: Calculated portfolio risk and volatility for informed decision-making.

## 🧰 Requirements

* Python 3.7+
* Jupyter Notebook
* Required Python Libraries:
  + pandas
  + numpy
  + matplotlib
  + scipy

## 🤝 Contribution

Contributions are welcome! Please fork the repository and submit a pull request.

## 📜 License

This project is licensed under the MIT License.

## 👩‍💻 Author

Binita Roy