

AI Financial Portfolio Rebalancer: Implementation and Evaluation Report

Abstract

This project showcases the development of an AI-powered Agent that guide in rebalancing financial portfolio. I do this by integrating real-time stock market data and leveraging LangChain's agent framework to dynamically analyze and optimize investment portfolios. By combining three key tools - market trend analysis, individual stock price lookup, and portfolio rebalancing, the tool provides actionable recommendations to maintain a balanced asset distribution.

Skill Set Summary

Through this endeavor, I refined my skills in integrating real-time data, evaluating large language models (LLMs) for financial tasks, and crafting effective prompts to guide AI outputs. This project demonstrates my ability to develop sophisticated AI tools that enhance decision-making processes.

Introduction

This project showcases my expertise in developing an AI-powered financial portfolio rebalancer using LangChain's agent framework. Through this mini-project, I honed several key skills:

1. **Integration of Real-Time Data:** I successfully integrated real-time stock market data using APIs like Yahoo Finance (yfinance) to inform portfolio rebalancing decisions.
2. **LLM Selection and Evaluation:** I compared and evaluated different large language models (LLMs) from OpenAI (GPT-4) and Groq (LLaMA3-8B and LLaMA3-70B) for financial analysis tasks, assessing their strengths and weaknesses in terms of accuracy, speed, and cost.
3. **Agent Framework Development:** I designed and implemented a LangChain agent that dynamically selects tools based on user queries, ensuring efficient workflow and decision-making.
4. **Prompt Engineering:** I refined prompts to guide LLMs in generating actionable and accurate financial recommendations, improving output quality through iterative testing and refinement.
5. **Performance Analysis:** I analyzed the performance of different LLMs in terms of latency and accuracy, providing insights into their suitability for various financial analysis scenarios.
6. **Portfolio Rebalancing Logic:** I implemented a portfolio rebalancing strategy based on equal-weight distribution, ensuring that the AI assistant can provide actionable recommendations for maintaining optimal portfolio balance.

By combining these skills, I created a tool that automates portfolio analysis and rebalancing, offering insights for financial decision-making.

1. Implementation Details

The AI financial portfolio rebalancer was built using LangChain's agent framework, integrating three core tools for real-time data analysis and decision-making:

Tools

1. Stock Price Lookup (`get_stock_price`)

- **Functionality:** Fetches real-time stock prices, daily changes, and percentage changes using Yahoo Finance (`yfinance`).
- **Key Features:**
 - Fallback to historical data if metadata is unavailable.
 - Error handling for invalid symbols or delisted stocks.
- **Example Output:**

```
{'symbol': 'AAPL', 'price': 189.84, 'change': 1.23, 'change_pct': 0.65}
```

2. Portfolio Rebalancer (`rebalance_portfolio`)

- **Functionality:** Analyzes portfolio weights and suggests buy/sell actions to achieve equal-weight distribution.
- **Key Features:**
 - Validates portfolio weight sum (must total ~ 1.0).
 - Generates actionable recommendations (e.g., "Sell AAPL: Decrease weight by 0.1667").

3. Market Trend Analysis (`market_trend_analysis`)

- **Functionality:** Evaluates S&P 500 (via SPY ETF) trends over five days, including returns and volatility.
- **Key Metrics:**
 - 5-day return
 - Annualized volatility

Agent Framework

- **LLM Integration:** Tested with OpenAI's GPT-4, Groq's LLaMA3-8B, and LLaMA3-70B.
- **Workflow:**
 1. **Market Analysis:** Always invoked first to assess broader market conditions.
 2. **Individual Stock Price Checks:** Each symbol in the portfolio is queried separately.
 3. **Rebalancing:** Recommendations generated based on equal-weight strategy.

Code -

```
# Simplified agent initialization
```

```
agent = initialize_agent(  
    tools=[stock_price_tool, rebalance_tool, trend_tool],  
    llm=llm,  
    agent=AgentType.ZERO_SHOT_REACT_DESCRIPTION,  
)
```

2. Challenges and Solutions

Challenge 1: Tool Compatibility Errors

- **Issue:** `AttributeError: 'Tool' object has no attribute 'is_single_input'` arose due to LangChain's expectation of Pydantic schemas for multi-input tools.
- **Solution:**
 - Defined explicit `args_schema` using Pydantic models.
 - Switched to `AgentType.STRUCTURED_CHAT_ZERO_SHOT_REACT_DESCRIPTION` to support structured inputs.

Challenge 2: Incorrect Stock Data

- **Issue:** Frequent 404 errors from Yahoo Finance (e.g., `'GOOGL'`: possibly delisted).
- **Solution:**
 - Added error fallbacks using historical data.
 - Implemented input validation to reject invalid symbols.

Challenge 3: Unreliable LLM Outputs

- **Issue:** Groq LLaMA3-8B produced incomplete or hallucinated recommendations (e.g., *"Agent stopped due to iteration limit"*).
- **Solution:**
 - **Prompt Engineering:** Enforced strict workflow order and individual stock checks.
 - **Response Formatting:** Required outputs to include numerical adjustments (e.g., *"Decrease weight by 0.1667"*).

Challenge 4: Performance Variability

- **Issue:** GPT-4 produced accurate but slow responses (~22 seconds), while Groq LLaMA3-70B was faster (~6 seconds) but occasionally less precise.
- **Solution:**
 - Cached market trend data to reduce API calls.
 - Limited agent iterations to prevent timeouts.

3. Analysis of LLM Strengths and Weaknesses for Financial Tasks

Based on the test results, each LLM demonstrates unique strengths and weaknesses for financial portfolio analysis tasks.

OpenAI GPT-4

- **Strengths:** OpenAI's GPT-4 is basic in tool selection and quality of advice. It mostly accurately calculated portfolio rebalancing actions and identified balanced portfolios.
- **Weaknesses:** In some edge cases (e.g., a portfolio consisting solely of Bitcoin), GPT-4 struggled to understand the broader context, leading to less useful recommendations. The cost per query is also higher.

Groq LLaMA3-8B

- **Strengths:** The Groq LLaMA3-8B model offers low latency, making it suitable for rapid analysis and real-time applications. It was the only model that did best in context understanding (albeit in only one case) where it correctly used the other tools in combination with the primary rebalancing tool. It could correctly identify even the edge case.
- **Weaknesses:** The model stopped due to iteration or time limits. Seems it lacks the reliability to be deployed for critical financial decisions.

Groq LLaMA3-70B

- **Strengths:** This model strikes a balance between speed and accuracy, delivering responses faster than GPT-4 while maintaining a better level of correctness. It even identified the edge case correctly. The action plan is also relatively easy to interpret
- **Weaknesses:** In the almost balanced case, it failed to use other tools before making a recommendation.

LLaMA3-8B model is the best in understanding larger context and task and purpose, and using the right tools, though it failed due to iteration limits. If this issue could be resolved, I would choose LLaMA3-8B model. LLaMA3-70B offers a solid tradeoff of speed and cost for real-time analysis and could be ranked second.

4. Recommendations for Financial Analysis Scenarios

Scenario	Recommended LLM	Rationale
High-Stakes Rebalancing	Groq LLaMA3-8B	Accuracy and nuanced reasoning critical for large portfolios.
Real-Time Analysis	Groq LLaMA3-70B	Speed and reliability for time-sensitive decisions.
Simple Portfolios	Groq LLaMA3-8B	Cost-effective for basic checks (e.g., 2–3 assets).
Educational Use	Groq LLaMA3-70B	Balances speed and clarity for student use cases.

Conclusion

The AI portfolio rebalancer successfully integrates real-time data with LangChain's agent framework, but LLM selection depends on trade-offs between accuracy, speed, and cost. While GPT-4 remains the gold standard for reliability, Groq's LLaMA3-70B offers a compelling alternative for real-time applications. Future work should focus on improving error handling for stock data APIs and fine-tuning smaller models for financial tasks.

Citation -

Report Generated Using Perplexity: https://www.perplexity.ai/search/what-error-in-this-line-mean-b-7LHNRtq8Q8OpxF90PGw?124=d&125=d&126=d&132=d&103=d&106=d&118=d&120=d&121=d&utm_source=copy_output