# RebalanceAI: AI-Powered Investment Portfolio Analyzer

**An intelligent portfolio recommendation system that fuses quantitative finance with real-time market sentiment analysis using LangChain**

---

## Overview

RebalanceAI is an intelligent investment assistant designed to make financial analysis simpler, smarter, and more transparent.
It blends **quantitative analytics** — things like volatility, Sharpe ratio, and returns — with **AI-driven market sentiment**, helping investors understand not just the *numbers*, but also the *narrative* behind them.

Built using **LangChain's agent framework**, RebalanceAI coordinates several specialized tools that work together seamlessly. It automatically pulls live market data, cleans it, computes technical indicators, interprets sentiment trends, evaluates portfolio risk, and finally, suggests optimized allocations — all while explaining its reasoning in plain language.

### Problem Statement

Most portfolio tools today sit at one extreme or the other.
Some are **purely quantitative**, focusing only on metrics like returns or volatility and ignoring real-world market sentiment. Others are **human-driven and subjective**, relying on intuition rather than systematic analysis.
RebalanceAI bridges that gap. It creates a single, intelligent system that:

- Automatically downloads and processes real stock data
- Calculates key performance and risk metrics
- Interprets market sentiment to capture investor mood
- Generates portfolio recommendations tailored to each risk profile
- And, most importantly, explains every recommendation clearly and transparently

In short, RebalanceAI transforms raw market data into **insightful, explainable investment guidance** — helping investors make confident, data-driven decisions.

## Key Features

### Single LangChain Agent Architecture

- **Conversational Memory**: Maintains context across interactions using ConversationBufferMemory
- **11 Specialized Tools**: Coordinated by one intelligent agent

- **Smart Auto-Download**: Automatically fetches missing stock data when needed
- **Natural Language Processing**: Understands company names and converts them to tickers using GPT-4o-mini

### Quantitative Analysis

- **Technical Metrics**: Volatility, Sharpe ratio, moving averages (7-day, 30-day, 90-day)
- **Performance Tracking**: Total returns, annualized returns, daily/weekly/monthly changes
- **Risk Assessment**: Portfolio-level volatility and risk profiling
- **Multi-Stock Comparison**: Side-by-side analysis of multiple securities

### LLM-Powered Sentiment Analysis

- **Market Trend Detection**: Bullish/Bearish/Neutral classification
- **Confidence Scoring**: High/Medium/Low confidence levels
- **Key Factor Identification**: LLM extracts critical market drivers
- **Investment Recommendations**: Buy/Hold/Sell suggestions with reasoning
- **Multi-Stock Sentiment**: Comparative sentiment analysis across portfolios

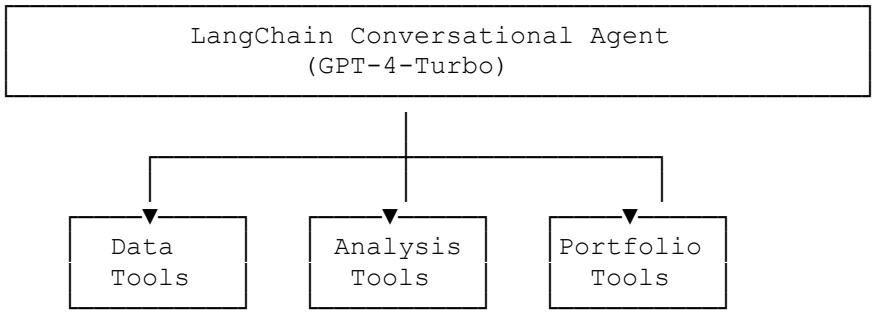
### Portfolio Optimization

- **Risk Profile Support**: Conservative, Moderate, Aggressive strategies
- **Sharpe Ratio Weighting**: Optimizes risk-adjusted returns
- **Explainable Allocations**: LLM provides reasoning for each recommendation
- **Diversification Analysis**: Explains portfolio balance and trade-offs

### Advanced Visualization

- **Interactive Charts**: Price trends with moving averages
- **Volume Analysis**: Trading volume with price correlation
- **Normalized Comparisons**: Multi-stock performance visualization
- **Gradio Web Interface**: User-friendly chat and analytics dashboard## Architecture

### Agent Design



### Component Breakdown

#### 1. Data Layer

- Stock Data Downloader (yfinance)
- Company Name Resolver (GPT-4o-mini)
- Global Storage (stock_data_storage)

#### 2. Analysis Layer

- Technical Analysis Calculator
- Sentiment Analyzer (GPT-4-turbo)
- Multi-Stock Sentiment Analyzer
- Stock Comparison Tool

#### 3. Risk & Optimization Layer

- Portfolio Risk Assessor
- Portfolio Optimizer (Sharpe ratio-based)

#### 4. Visualization Layer

- Price Chart Visualizer
- Volume Chart Visualizer
- Comparison Chart Visualizer
- Portfolio Pie Chart

#### 5. Interface Layer

- Gradio Web UI (Chat + Analytics)
- Memory Management
- Interactive Command-Line Interface

# Setup & Installation

## Prerequisites

- Python 3.8+
- Google Colab (recommended) or local Jupyter environment
- OpenAI API Key

## Step 1: Clone/Download

```
# Download the notebook: Hackathon_final.ipynb
```

## Step 2: Install Dependencies

```
!pip install -q yfinance matplotlib numpy pandas gradio openai langchain
langchain-openai
```

### Step 3: Configure API Key

#### For Google Colab:

```
from google.colab import userdata
os.environ["OPENAI_API_KEY"] = userdata.get("OPENAI_API_KEY")
```

#### For Local Environment:

```
import os
os.environ["OPENAI_API_KEY"] = "your-api-key-here"
```

### Step 4: Run Cells Sequentially

Execute all cells in order from Cell 1 to Cell 20 to initialize all components.

### Step 5: Launch Interface

#### Option A: Gradio Web UI

```
# Run the Gradio cell to launch web interface
# Access via the generated public URL
```

#### Option B: Command-Line Chat

```
chat_with_agent()
```

# Usage Guide

## Basic Interactions

#### Portfolio Analysis:

```
You: Analyze a $10,000 portfolio for moderate risk and tech emphasis. Show
sentiment impact.

RebalanceAI: [Downloads AAPL, MSFT, GOOGL]
             [Calculates Sharpe ratios and volatility]
             [Analyzes sentiment for each stock]
             [Generates optimized allocation with reasoning]
```

#### Stock Comparison:

```
You: Compare TSLA vs NVDA on volatility, Sharpe ratio, and recent news
sentiment.
```

```
RebalanceAI: [Provides side-by-side technical metrics]
             [LLM analyzes sentiment for both stocks]
             [Highlights key differences and investment implications]
```

**Visualization:**

```
You: Show me a chart for Apple

RebalanceAI: [Automatically resolves "Apple" → AAPL]
             [Generates price chart with moving averages]
             [Displays inline in chat interface]
```

### Gradio Web Interface

**Chat Tab:**

- Natural language queries
- Automatic chart generation
- Conversation memory maintained
- Example prompts provided

**Single Stock Chart Tab:**

- Enter ticker or company name
- View price with 7/30/90-day moving averages
- See latest price and total return

**Compare Stocks Tab:**

- Enter multiple tickers/companies
- Normalized performance comparison
- Individual return statistics

**Analytics Tab:**

- Technical Analysis: Detailed metrics for any stock
- Sentiment Analysis: AI-powered market sentiment
- Risk Assessment: Portfolio-level volatility
- Portfolio Optimizer: Generate allocation recommendations

# Methodology

## Data Collection

- **Source**: Yahoo Finance (yfinance)
- **Timeframe**: 2 years (730 days) of historical data
- **Fields**: Open, High, Low, Close, Volume

- **Auto-adjustment**: Handles splits and dividends

## Technical Metrics

### Volatility (Annual)

```
volatility = daily_return.std() * sqrt(252)
```

### Sharpe Ratio

```
excess_return = (annualized_return - risk_free_rate)
sharpe_ratio = excess_return / volatility
```

### Moving Averages

```
MA_7 = Close.rolling(7).mean()
MA_30 = Close.rolling(30).mean()
MA_90 = Close.rolling(90).mean()
```

### Total & Annualized Returns

```
total_return = (Close[-1] / Close[0] - 1) * 100
annualized_return = ((1 + total_return/100) ** (252/len(data)) - 1) * 100
```

## Sentiment Analysis Methodology

### Trend Classification

- **Bullish**: MA7 > MA30 AND 7-day change > 0
- **Bearish**: MA7 < MA30 AND 7-day change < 0
- **Neutral**: Mixed signals

### LLM Sentiment Extraction

- Model: GPT-4-turbo (temperature=0.3)
- Inputs: Price, volatility, trend, recent changes
- Outputs: Overall sentiment, confidence, key factors, recommendation
- Prompt engineering: Structured 4-point analysis format

## Portfolio Optimization Logic

### Conservative Profile

- Equal weighting across all stocks
- Prioritizes stability over returns

### Moderate Profile

- 70% Sharpe ratio weighting + 30% equal weighting
- Balanced approach

**Aggressive Profile**

- 100% Sharpe ratio weighting
- Maximizes risk-adjusted returns

**Allocation Formula (Moderate)**

```
weight[ticker] = (sharpe[ticker] / total_sharpe) * 0.7 + 0.3 / num_stocks
```

### Weighting Justification

| Component | Weight | Rationale |
|---|---|---|
| Sharpe Ratio | Primary | Captures risk-adjusted returns |
| Volatility | Indirect | Included in Sharpe calculation |
| Sentiment | Qualitative | LLM provides context and reasoning |
| Moving Averages | Trend | Used for sentiment classification |

**Why Sharpe-based?**

- Industry-standard metric for risk-adjusted performance
- Penalizes volatility while rewarding returns
- Provides objective, quantifiable optimization basis

**Sentiment Integration**

- LLM analyzes market conditions and provides qualitative overlay
- Explains *why* certain allocations make sense
- Identifies potential risks not captured by historical metrics

# Tool Descriptions

### Tool 1: Stock Data Downloader

**Function**: `download_stock_data(query: str)`

- Downloads 2-year historical data via yfinance
- Stores in global `stock_data_storage` dictionary
- Returns summary with latest price and average volume

### Tool 2: Technical Analysis Calculator

**Function**: `calculate_technical_metrics(query: str)`

- Computes volatility, Sharpe ratio, returns

- Calculates 7-day and 30-day moving averages
- Formatted output with emojis for readability

### Tool 3: Market Sentiment Analyzer

**Function**: `analyze_market_sentiment(query: str)`

- Classifies trend (Bullish/Bearish/Neutral)
- Uses GPT-4-turbo for sentiment analysis
- Provides AI-generated investment recommendation

### Tool 4: Portfolio Risk Assessor

**Function**: `assess_portfolio_risk(query: str)`

- Calculates individual stock volatilities
- Computes portfolio average volatility
- Assigns risk level (Low/Medium/High)

### Tool 5: Portfolio Optimizer

**Function**: `optimize_portfolio(query: str)`

- Generates allocations based on risk profile
- Uses Sharpe ratio for weighting
- LLM explains diversification rationale

### Tool 6: Portfolio Visualizer

**Function**: `visualize_portfolio(query: str)`

- Creates pie chart of portfolio allocation
- Shows percentage breakdown

### Tool 7: Stock Comparison

**Function**: `compare_stocks(query: str)`

- Side-by-side analysis of multiple stocks
- Ranks by Sharpe ratio, volatility, returns
- Identifies best performers in each category

### Tool 8: Multi-Stock Sentiment Analyzer

**Function**: `analyze_multiple_sentiments(query: str)`

- Analyzes sentiment for portfolio of stocks
- LLM provides comparative analysis
- Ranks stocks by sentiment strength

### Tool 9: Price Chart Visualizer

**Function**: `visualize_price_chart(query: str)`

- Plots price with 7/30/90-day moving averages
- Shows latest price and return statistics

### Tool 10: Volume Chart Visualizer

**Function**: `visualize_volume_chart(query: str)`

- Dual-panel: price + volume
- Color-coded volume bars (green=up, red=down)

### Tool 11: Multi-Stock Comparison Chart

**Function**: `visualize_comparison_chart(query: str)`

- Normalized return comparison (% from start)
- Multiple stocks on same chart

# Sample Interactions

## Example 1: Portfolio Optimization

```
Input: Analyze a $10,000 portfolio for moderate risk and tech emphasis. Show
sentiment impact.

Output:
🎯 Optimized Portfolio for Moderate Investor ($10,000.00)
──────────────────────────────────────────────────────────────────────

💰 AAPL: $2,627.07 (26.3%) | Sharpe: 1.136
💰 MSFT: $3,595.12 (36.0%) | Sharpe: 1.285
💰 GOOGL: $3,777.81 (37.8%) | Sharpe: 1.348

🧠 AI Reasoning:
The optimized portfolio leverages the stability and growth potential
of these established tech companies. The allocation prioritizes
Google's highest Sharpe ratio, followed by Microsoft and Apple,
ensuring balanced exposure to the tech sector.
```

💭 Sentiment Analysis:
📈 AAPL: Bullish - Strong product cycle momentum
📈 MSFT: Bullish - Cloud growth and AI leadership
➡️ GOOGL: Neutral - Search dominance offset by regulatory concerns

## Example 2: Stock Comparison

Input: Compare TSLA vs NVDA

Output:
📊 Stock Comparison Analysis
──────────────────────────────────────────────────

💰 Current Prices:
  • TSLA: $242.84
  • NVDA: $134.87

📉 Annual Volatility:
  • NVDA: 48.32% 🔴 High
  • TSLA: 62.75% 🔴 High

⚖️ Sharpe Ratio:
  • NVDA: 1.847 🏆 Excellent
  • TSLA: 0.342 ⚠️ Fair

📈 Returns:
  • NVDA: Total: +127.45%, Annualized: +52.18%
  • TSLA: Total: -8.23%, Annualized: -4.21%

🏆 Summary:
  • Best Risk-Adjusted Return: NVDA (Sharpe: 1.847)
  • Highest Total Return: NVDA (+127.45%)

## Example 3: Natural Language Chart Request

Input: Show me a chart comparing Apple and Microsoft

Output: [Generates normalized return comparison chart]

✅ Comparison Chart (AAPL, MSFT)

📊 Total Returns:
  • AAPL: +45.23%
  • MSFT: +58.17%

---

# Screenshots

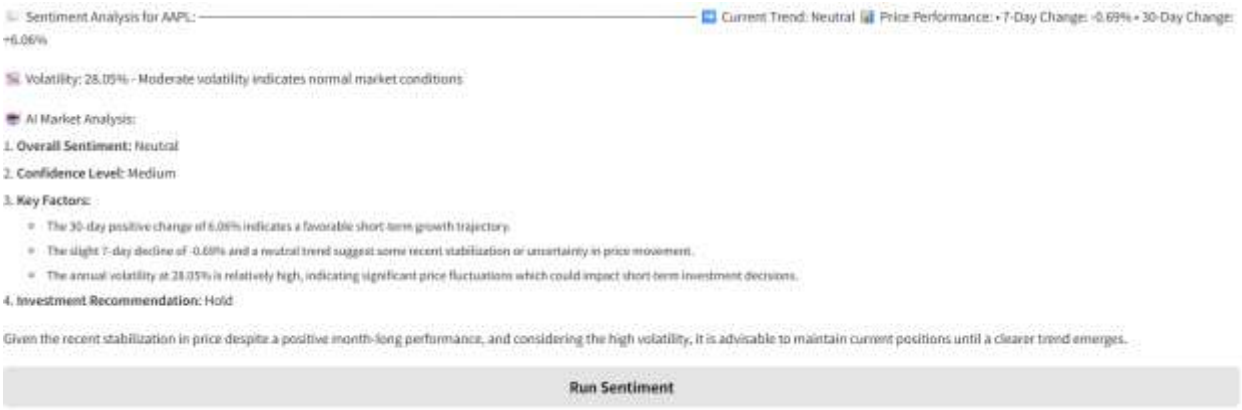
## Gradio Web Interface


*Natural language portfolio analysis with automatic chart generation*

## Portfolio Optimization

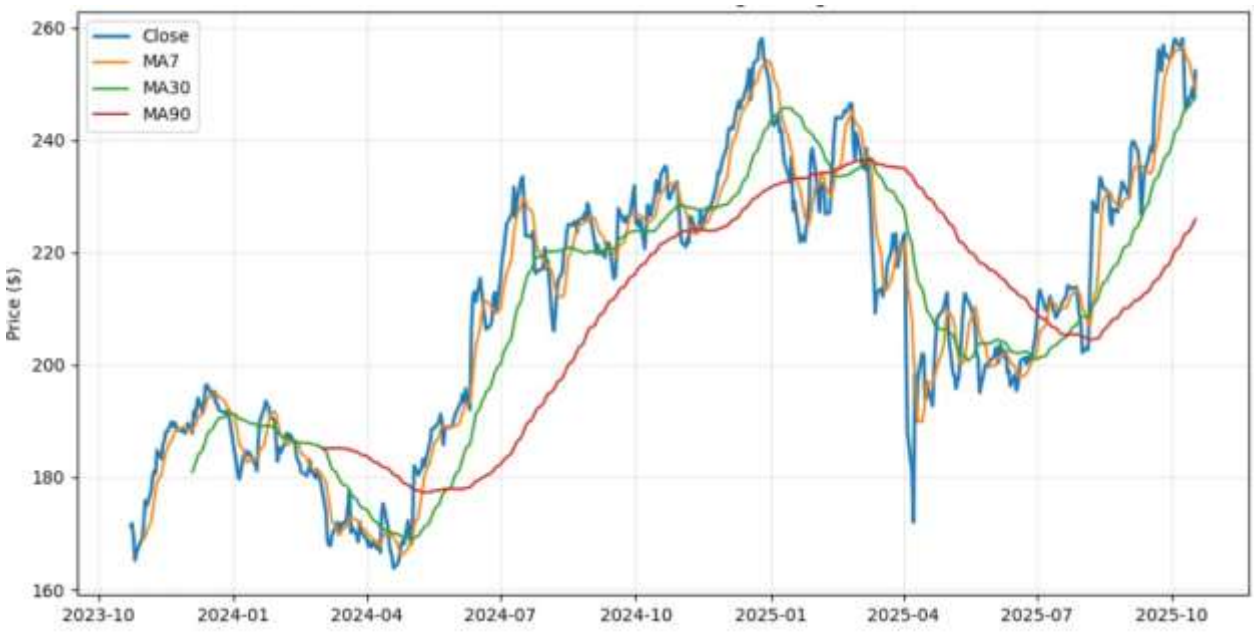
*AI-generated allocation with Sharpe ratio weighting and explanations*

## Stock Comparison




*Side-by-side technical analysis with sentiment overlay*

## Interactive Charts




# Limitations & Disclaimers

## Technical Limitations

- **Data Dependency**: Relies on Yahoo Finance API availability
- **Historical Analysis**: Based on 2-year lookback window
- **No Real-Time Data**: Uses daily close prices, not intraday
- **Limited Fundamental Analysis**: Focuses on technical metrics
- **No Sector Rebalancing**: Doesn't constrain sector exposure
- **Backtest-Free**: No historical simulation of strategies

## Sentiment Limitations

- **LLM Hallucination Risk**: Sentiment analysis may contain inaccuracies
- **No News Scraping**: Doesn't access actual news headlines
- **Price-Based Only**: Sentiment inferred from price movements, not text sources
- **Temperature Setting**: May produce varying outputs on repeated runs

## Responsible AI Usage

### IMPORTANT: RebalanceAI is an educational tool, not financial advice

- Not a substitute for professional financial advisors
- Does not consider individual tax situations
- Ignores transaction costs and fees
- Assumes perfect liquidity
- Based on historical data (past performance ≠ future results)

- Risk profiles are simplified and may not match all investor needs

### Data Privacy & Ethics

- Uses only publicly available market data
- No personal financial information collected
- Respects API rate limits and terms of service
- Cached sample data included to reduce API calls

### Recommendations for Real-World Use

1. **Consult a Professional**: Use this as a starting point, not the final decision
2. **Validate Assumptions**: Review metric calculations and weighting logic
3. **Consider Taxes**: Factor in capital gains and tax implications
4. **Add Fundamentals**: Incorporate P/E ratios, earnings, debt levels
5. **Stress Test**: Simulate market downturns and portfolio resilience
6. **Rebalance Regularly**: Markets change, portfolios should adap

## Technical Stack

| Component | Technology |
|---|---|
| Agent Framework | LangChain (v0.1.x) |
| LLM (Main) | OpenAI GPT-4-turbo |
| LLM (Resolution) | OpenAI GPT-4o-mini |
| Memory | ConversationBufferMemory |
| Data Source | yfinance (Yahoo Finance) |
| Visualization | matplotlib, plotly |
| Data Processing | pandas, numpy |
| Web UI | Gradio 4.44.0 |
| Environment | Google Colab / Jupyter |

## Why RebalanceAI Stands Out

1. **Unified System**: Single agent coordinates 11 tools seamlessly
2. **Dual Analysis**: Quantitative metrics + qualitative LLM insights
3. **User-Friendly**: Natural language interface with company name support
4. **Explainable**: Every recommendation comes with reasoning
5. **Production-Ready**: Web UI with memory and error handling
6. **Extensible**: Easy to add new tools, data sources, or metrics

**Technologies Used:**

- LangChain (Apache 2.0 License)
- OpenAI API (Commercial)
- yfinance (Apache 2.0 License)
- Gradio (Apache 2.0 License)

**Data Sources:**

- Yahoo Finance (via yfinance)
- Historical stock prices are publicly available market data