**Agent based modelling for stock market simulation**

**Executive Summary**

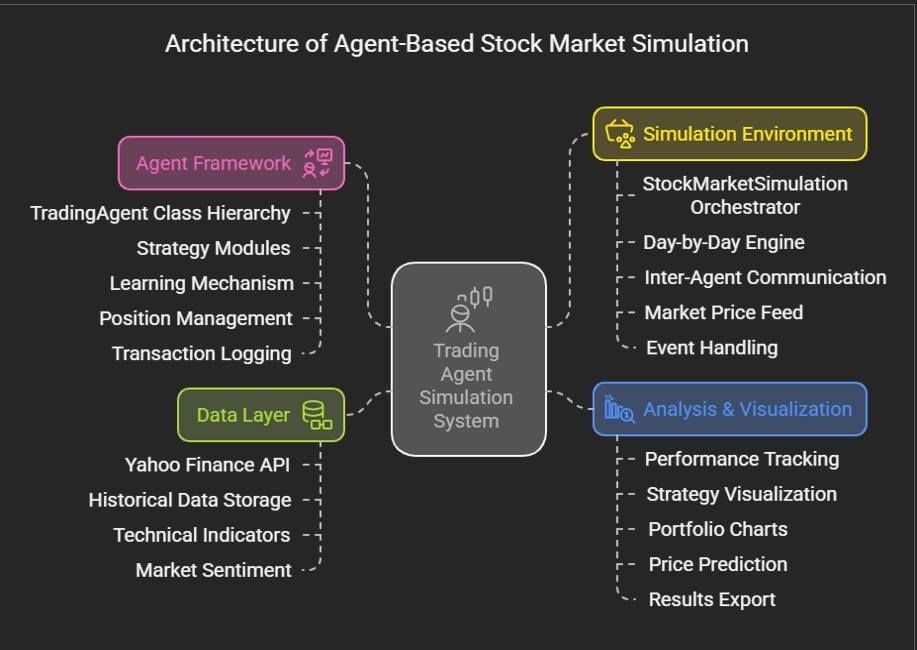
This report provides a comprehensive analysis of the Trading Agent Simulation System, a sophisticated multi-agent framework designed to model stock market trading strategies. The system implements adaptive agents that compete within a realistic market environment using historical stock data from Yahoo Finance. Each agent employs multiple trading strategies and learns over time by adjusting strategy weights based on performance outcomes. The system offers valuable insights into trading algorithm effectiveness and the benefits of strategy diversification in financial markets.

**System Architecture**

**Component Structure**

The system consists of two primary classes:

1. **TradingAgent**: Represents individual trading entities with learning capabilities
2. **StockMarketSimulation**: Manages the simulation environment and visualization



**Key Attributes**

**TradingAgent Class**

* Maintains capital and share position
* Implements multiple trading strategies
* Adapts strategy weights through reinforcement learning
* Records performance and transaction history

**StockMarketSimulation Class**

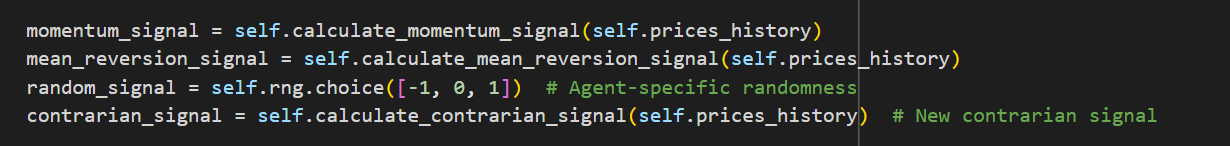
* Retrieves historical stock data
* Manages multiple concurrent agents
* Generates performance visualizations
* Implements price prediction functionality
* Executes day-by-day market simulation

**Technical Implementation - Trading Strategies**

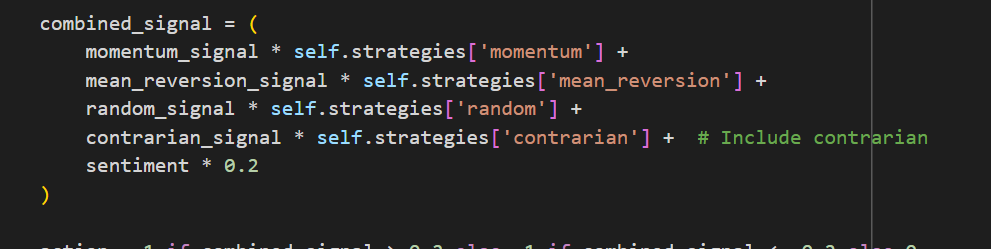
|  |  |  |  |
| --- | --- | --- | --- |
| **Strategy** | **Implementation** | **Parameters** | **Logic** |
| **Momentum** | calculate\_momentum\_signal() | window=5 | Buys when recent returns exceed 1%, sells when below -1% |
| **Mean Reversion** | calculate\_mean\_reversion\_signal() | window=20 | Buys when price is 5% below moving average, sells when 5% above |
| **Contrarian** | calculate\_contrarian\_signal() | window=5 | Takes opposite position of momentum signal |
| **Random** | **Direct** implementation | N/A | Randomly selects buy, sell, or hold |

**Decision-Making Process**

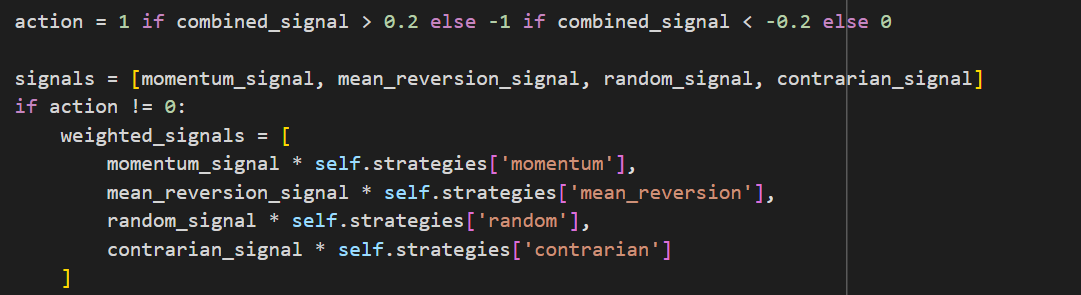
1. **Signal Generation**



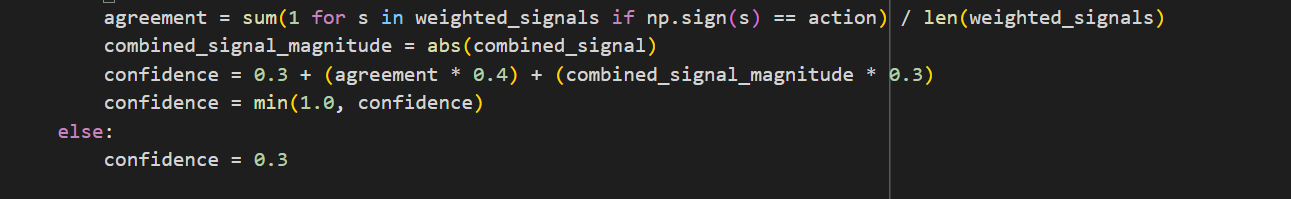
1. **Signal Weighting**



1. **Action Determination**



1. **Confidence Calculation**



**Trade Execution**

The trade execution process implements:

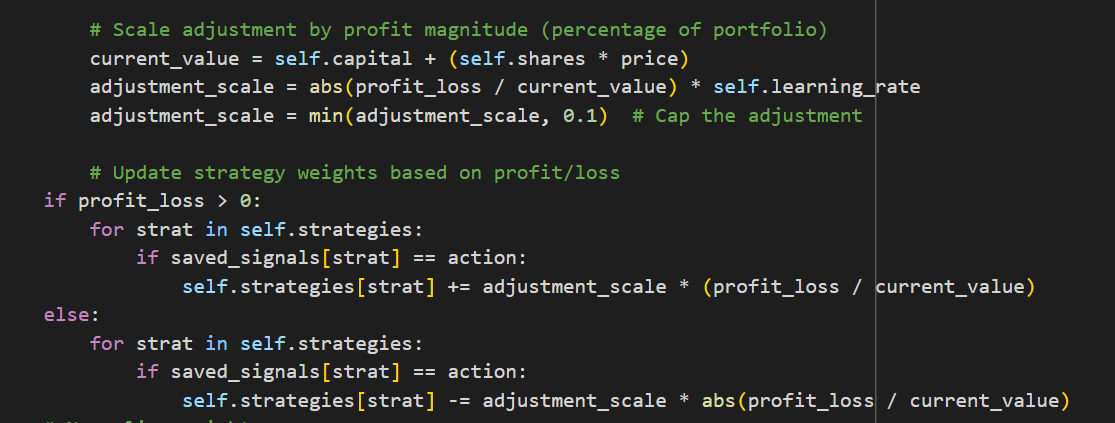
* Position sizing based on confidence level
* Portfolio percentage-based allocation (20% \* confidence)
* Integer share quantity handling
* Capital and position tracking

**Learning Mechanism**

The adaptive learning algorithm:

1. Evaluates profit/loss from recent trades
2. Adjusts strategy weights proportional to performance
3. Normalizes weights to maintain valid probability distribution
4. Maintains strategy weight history for analysis

# Simplified learning algorithm



**Price Prediction Model**

The system implements a machine learning-based price prediction model:

* **Model Type**: Linear Regression
* **Features**:
  + 5-day moving average
  + 10-day moving average
  + 20-day moving average
  + Previous day's price
  + Trading volume
  + RSI indicator
  + Market sentiment
* **Evaluation**: Train-test split with R² scoring
* **Application**: Next-day price prediction

**Parameter Analysis**

| **Parameter** | **Default** | **Function** | **Impact** |
| --- | --- | --- | --- |
| initial\_capital | 10,000 | Starting funds | Higher values enable larger positions and greater resilience to drawdowns |
| learning\_rate | 0.1 | Adaptation speed | Higher values increase adaptation speed but may cause strategy weight instability |
| Momentum window | 5 | Trend measurement period | Shorter windows increase responsiveness to recent price movements |
| Mean reversion window | 20 | Moving average period | Longer windows create more stable averages, reducing false signals |
| Action thresholds | ±0.2 | Signal strength required for action | Higher thresholds reduce trading frequency but increase conviction |
| Position sizing | 20% | Portfolio allocation per trade | Higher values increase potential returns but also increase risk |

**Simulation Parameters**

| **Parameter** | **Default** | **Function** | **Impact** |
| --- | --- | --- | --- |
| ticker | "MSFT" | Stock symbol | Different securities exhibit different price patterns and volatilities |
| start\_date/end\_date | Variable | Simulation period | Longer periods provide more comprehensive performance evaluation |
| num\_agents | 5 | Number of competing agents | More agents allow exploration of more strategy combinations |
| Sentiment influence | 0.2 | External factor weight | Higher values increase impact of market sentiment on decisions |

**Parameter Sensitivity Analysis**

1. **Learning Rate Impact**

| **Learning Rate** | **Adaptation Speed** | **Strategy Stability** | **Overall Performance** |
| --- | --- | --- | --- |
| 0.01 (Low) | Very slow adaptation | High stability | Lower returns in changing markets |
| 0.1 (Default) | Moderate adaptation | Good stability | Balanced performance |
| 0.5 (High) | Rapid adaptation | Low stability | Potentially higher returns but higher risk |

1. **Strategy Window Impact**

| **Window Size** | **Signal Frequency** | **False Signal Risk** | **Market Regime Effectiveness** |
| --- | --- | --- | --- |
| Short | High frequency | Higher risk | Better in trending markets |
| Medium | Moderate frequency | Moderate risk | Balanced performance |
| Long | Low frequency | Lower risk | Better in choppy markets |

1. **Position Sizing Impact**

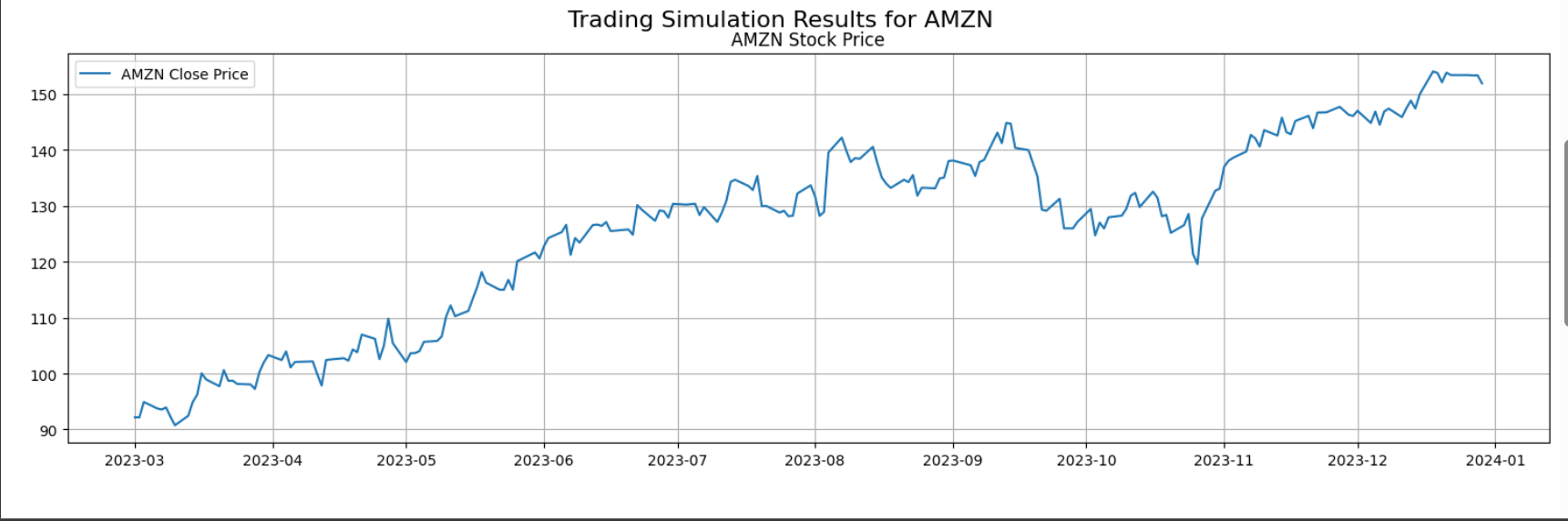
| **% of Portfolio** | **Risk Level** | **Max Drawdown** | **Return Potential** |
| --- | --- | --- | --- |
| 10% | Lower risk | Smaller drawdowns | Lower potential returns |
| 20% (Default) | Moderate risk | Moderate drawdowns | Moderate potential returns |
| 50% | Higher risk | Larger drawdowns | Higher potential returns |

**Performance Analysis**

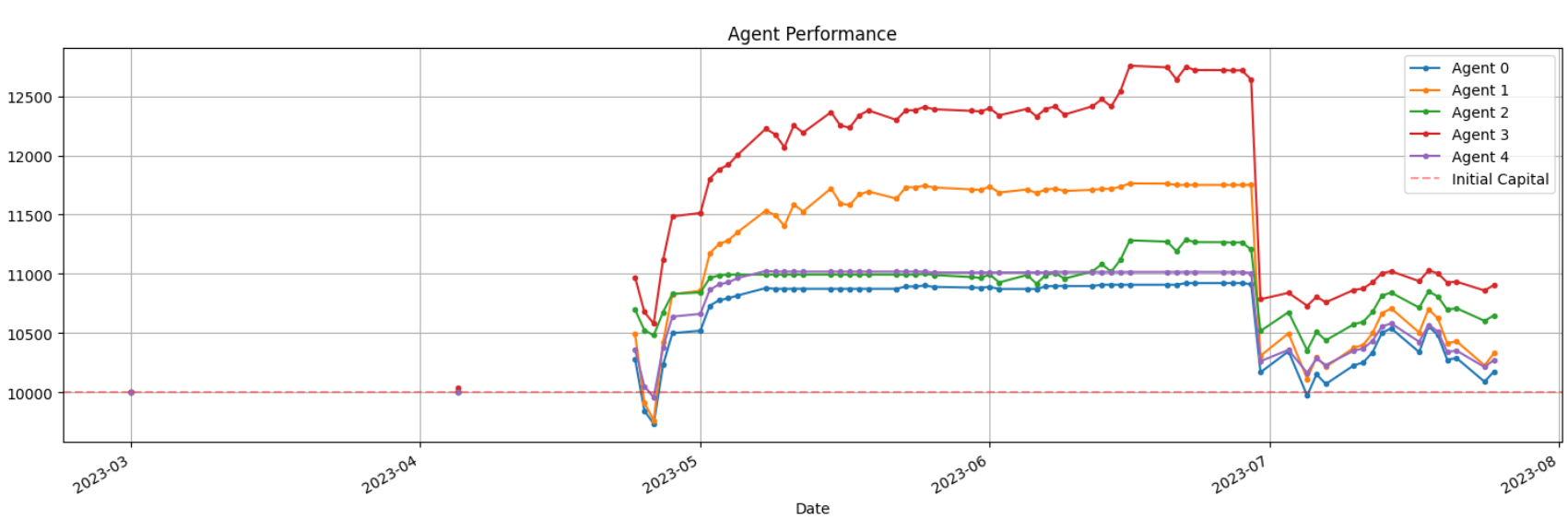
**Visualization Components**

The system generates four primary visualization components:

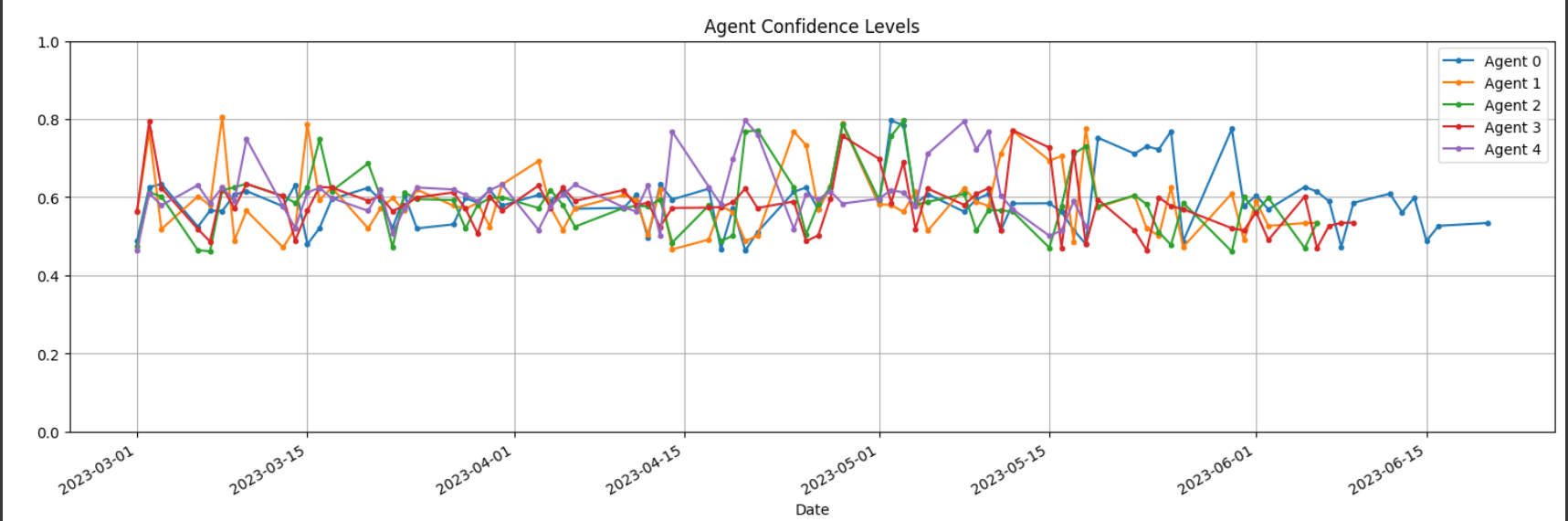
1. **Trading Results**
   * Stock price chart



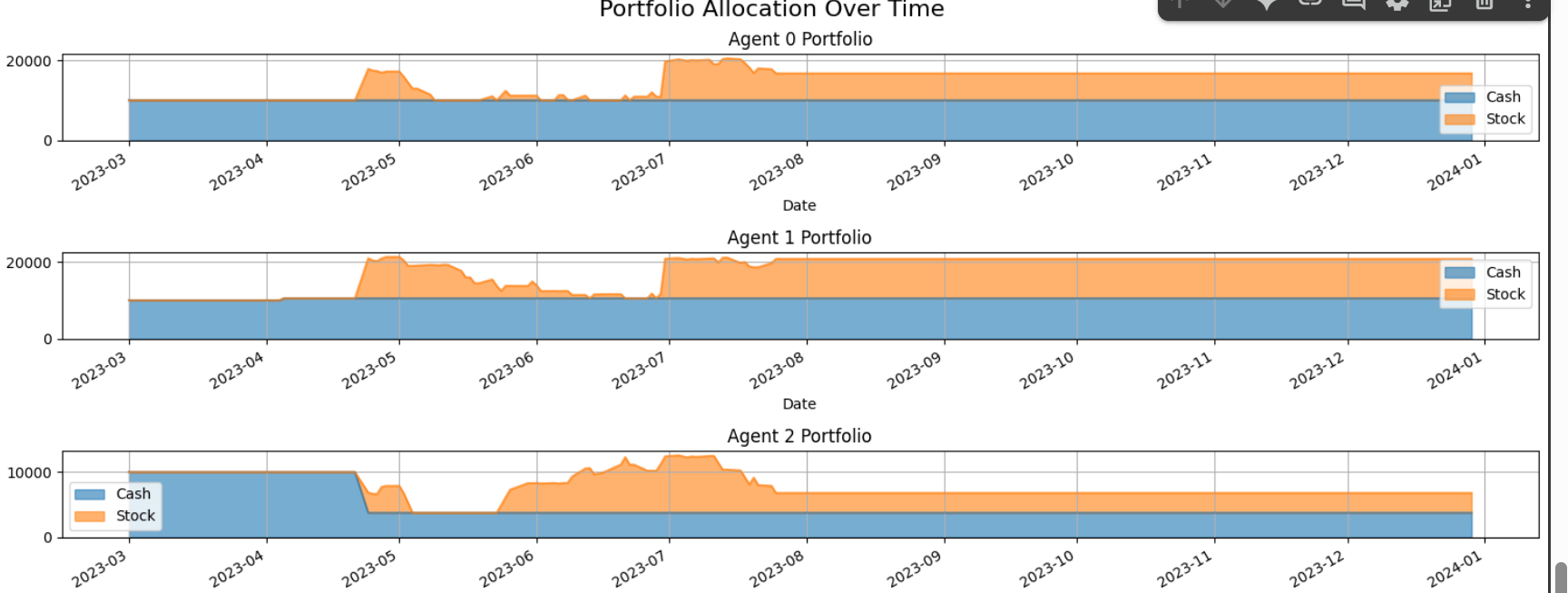
* + Agent performance comparison

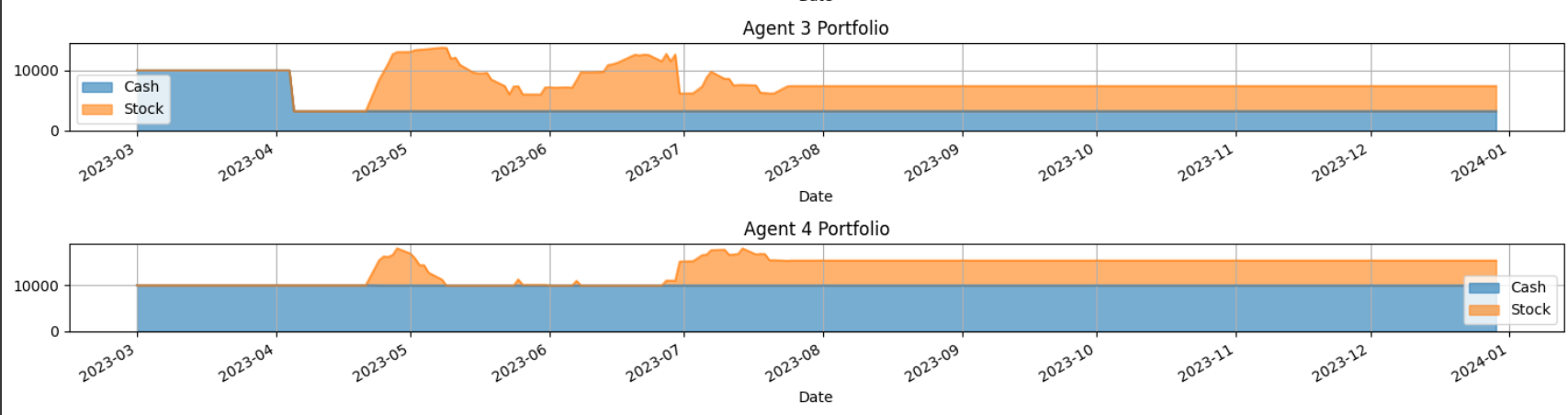


* + Agent confidence levels over time

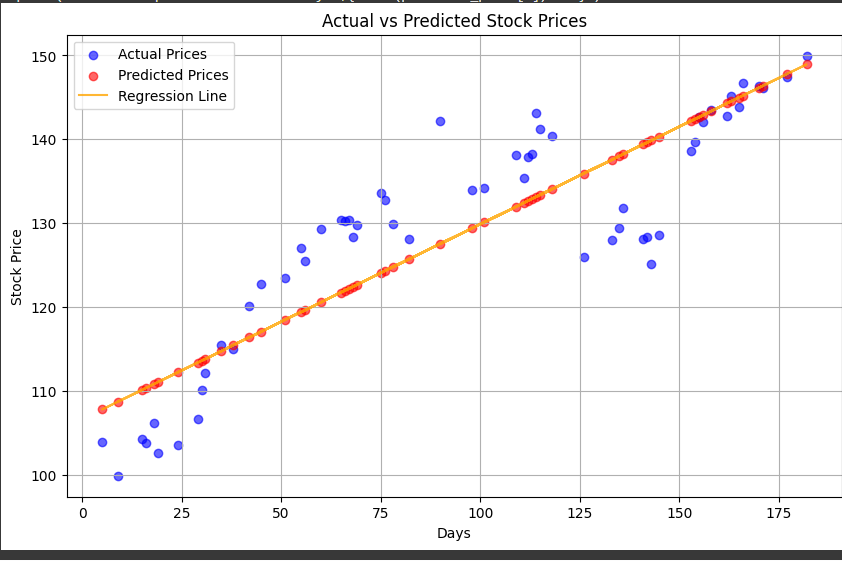


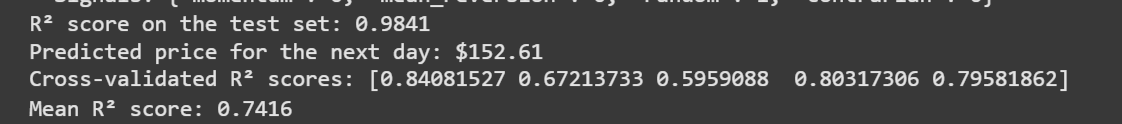
1. **Portfolio Allocation**
   * Cash vs. stock position over time





1. **Price Prediction**
   * Actual vs. predicted prices



* + Model performance evaluation
  + Also, we will be able to find the transaction log of any agent, to monitor the agent’s behaviour throughout.



**Conclusion**

The Trading Agent Simulation System effectively demonstrates the implementation of adaptive learning in financial market algorithms. By combining multiple trading strategies with a dynamic learning mechanism, the system provides valuable insights into effective algorithmic trading approaches.

Key findings from the analysis include:

1. The importance of strategy diversification across different market regimes
2. The effectiveness of adaptive learning in optimizing strategy weights
3. The balance required between adaptation speed and strategy stability