Simple Trading Strategies Backtester

Arsen Rayev Jeremy Brosnan

MATH 209

Initial Problem: Analyzing Stock Data with Trading Strategies:

- Load historical time series data of a stock or option into Python. Test five trading strategies based on past data, such as "sell if daily change exceeds 3%" or "buy on dips." Choose one strategy and evaluate its performance on historical data. Your code should rely solely on Python libraries (excluding external software).
- Suggested Skills: Data analysis, algorithmic trading strategies, handling time series data in Python.

Requirements: Python 3.9+ recommended, pip install numpy pandas matplotlib yfinance.

Project Structure:

1. Imports & settings
2. Data loading helpers
3. Strategy definitions
4. Backtesting logic
5. Visualization

Market Data Used:

Default ticker: SPY (S&P 500 ETF)

Data source: Yahoo Finance

Price used: Adjusted Close (accounts for dividends & splits)

Daily Returns Formula: (today's price / yesterday's price) - 1

Implemented Strategies:

1. Buy & Hold

- Always invested
- Baseline comparison

2. Moving Average Crossover

- Buy when short-term trend > long-term trend
- Exit when trend weakens

Uses:

- 20-day moving average
- 50-day moving average

3. Sell Big Up Days

- Exit after unusually strong daily gains
- Re-enter after a pullback

4. Buy the Dip

- Buy after sharp daily drops
- Hold for a fixed number of days

5. Rolling Regression (Advanced)

- Fits a trend line to price history
- Buys when price is far below trend
- Sells when price is far above trend
- Cash earns **2.5%** APR

This strategy introduces:

- Linear Regression
- Volatility bands
- Opportunity cost of holding cash