Report on Trading Strategy and Analysis

Understanding the Strategy

The provided Python code implements a trading strategy using technical indicators such as Exponential Moving Averages (EMA) and Relative Strength Index (RSI). The strategy aims to generate buy and sell signals based on these indicators to profit from short-term price movements in the stock market.

Strategy Used

1. **Data Collection**: The code fetches intraday (15-minute interval) stock price data for Amazon (symbol: AMZN) from the Alpha Vantage API. It also retrieves EMA and RSI data to analyze price trends and momentum.
2. **Indicator Calculation**:
    - **EMA (Exponential Moving Average)**: Calculates short-term (12-period) and long-term (26-period) EMAs. These are used to identify trends and potential entry/exit points.
    - **RSI (Relative Strength Index)**: Computes RSI based on a 14-period calculation. RSI helps determine if a stock is overbought or oversold.
3. **Signal Generation**:
    - **Buy Signal**: Triggered when the short-term EMA crosses above the long-term EMA and RSI is below 70 (indicating potential buying opportunity).
    - **Sell Signal**: Triggered when the short-term EMA crosses below the long-term EMA and RSI is above 30 (indicating potential selling opportunity).

Trading Signals

- **Buy Signal**: Generated when the short-term EMA crosses above the long-term EMA and RSI is below 70.
- **Sell Signal**: Generated when the short-term EMA crosses below the long-term EMA and RSI is above 30.
- **Other Buying signal**: When RSI crosses below 30, it is likely to grow upwards that signals to buy.
- **Other Selling Signal**: When RSI crosses over 70, it is likely to fall downwards that indicates to short sell.

Backtesting Strategy

The backtesting process evaluates how well the trading strategy would have performed using historical data:

- **Position Management**: Tracks entry and exit points based on generated signals.
- **Profit and Loss Calculation**: Computes daily returns, strategy returns, and cumulative returns.
- **Risk Management**: Implements a 2% stop-loss and a 4% take-profit to manage risk and potential losses.

Key Performance Metrics

1. **Cumulative Return**: Measures the total profit or loss from the trading strategy over the backtested period.
2. **Annual Return**: Average annualized return percentage based on historical performance.
3. **Annual Volatility**: Standard deviation of returns, annualized, to assess the strategy's risk.
4. **Sharpe Ratio**: Evaluates risk-adjusted returns, indicating how well the strategy performed relative to its risk.

Backtesting results

```
Cumulative Return: $2865.06
Annual Return: 6107.70%
Annual Volatility: 32.86%
Sharpe Ratio: 185.89
```

Success Rate and Failure Analysis

- **Success Rate**: The strategy's success is assessed based on positive cumulative returns and a positive Sharpe ratio. A higher Sharpe ratio indicates better risk-adjusted performance.
- **Failure Analysis**: Potential failures include periods of negative cumulative returns, underperformance compared to benchmarks, or inadequacies in risk management (e.g., stop-loss not triggered effectively).

Conclusion

The strategy implemented in the code aims to capitalize on short-term price movements using technical indicators. By generating buy and sell signals based on EMA crossovers and RSI levels, it attempts to optimize trading decisions. The backtesting results provide insights into its historical performance, including key metrics like cumulative return and Sharpe ratio. Understanding both the successes and potential failures of the strategy is crucial for refining and improving its effectiveness in real-world trading scenarios.