Assessment 4: Uniswap v3 LP Rebalancing

Overview

You are tasked with building a Python-based backtesting framework to simulate Uniswap v3 liquidity provision on the ETH/USDC pool. The simulation should start with an initial balance of **100 ETH** on January 1st, 2024, and use historical price data from CoinMarketCap (CMC) for the previous year.

Key Components

1. **LP Position Setup:**
    - **Liquidity Range:** For a given spot price P, Equal amounts of ETH and USDC should be provided in a fixed range of
      [P/(1+x%),P*(1+x%)]
      where x is a configurable parameter.
    - **Example:** If x=0.5 (or 50%), and the current price P is $1,000, 50 ETH will be swapped for 50,000 USDC and deployed in [1000/1.5,1000*1.5] or [666,1500]
2. **Rebalancing Logic:**
    - **Monitoring:** Continuously monitor the market.
    - **Trigger:** When the market price exits the active liquidity range, you must “remove” the current LP position.
    - **Action:** Record the impermanent loss (or gain) from that position.
    - **Re-entry:** Immediately swap to maintain equal amounts of ETH and USDC and open a new LP position centered at the current price using the same logic.
3. **Alternative Ranges:**
    - **Testing:** The framework should allow you to experiment with different values of x. Specifically, calculate for round figures of **5%, 10%, 15%, and 20%**.

- **Optimization:** Additionally, try to determine the optimal x value (i.e., the range that provides the best risk-adjusted returns) by analyzing performance across these scenarios.

Performance Evaluation

Your strategy's performance should be compared against a simple **HODL** (buy-and-hold) strategy using the following metrics:

- **Total Returns:** Compare the final portfolio value of the LP strategy with that of HODL.
- **Sharpe Ratio:** Calculate the Sharpe ratio based on daily returns to understand the risk-adjusted performance.
- **Maximum Drawdown:** Identify the maximum drawdown during the simulation.
- **Impermanent Loss:** Quantify the impermanent loss (or gain) incurred each time a position is rebalanced.
- **Time-Based Returns:** Compute returns on a daily, weekly, and monthly basis.
- **Position Value Over Time:** Track how the value of the LP strategy evolves over the simulation period.

Visualization Requirements

Include visualizations to help interpret the results and support your analysis:

- **Equity Curves:** Plot the cumulative returns of both the LP strategy and the HODL strategy over time.
- **Price & Range Visualization:** Overlay the historical price data with the active LP ranges. Mark the points at which rebalancing occurs.
- **Risk Metrics:** Visualize trends for risk metrics such as drawdown and the Sharpe ratio.

Submission Requirements

- **Jupyter Notebook:** Your complete analysis should be delivered as a Jupyter Notebook. This notebook should contain:
    - Your backtesting framework code.

- Visualizations.
- Detailed commentary explaining your approach, assumptions, and any challenges you encountered.

- **requirements.txt:** Include a file listing all the Python packages used.

Conclusion

This assignment is designed to evaluate your ability to integrate market data, implement a trading strategy simulation, and perform a thorough quantitative analysis. We encourage you to be creative in your analysis and to think critically about the risk-return trade-offs of providing liquidity on Uniswap v3.

Good luck, and we look forward to seeing your solution!