Final Project Progress Report : Ethereum Weekly Volatility Prediction Using Market, Sentiment, and On-chain Data

Project scope update

This project aims to predict Ethereum weekly volatility in the short term using market and sentiment data. Currently, I have successfully set up a pipeline that can automatically retrieve ETH-USD historical market data from the Yahoo Finance API using the yfinance Python library. The pipeline is now able to perform daily-level preprocessing, create daily returns, reform weekly features, and generate visualizations which are stored in the result directory. The current stage primarily focuses on feature engineering pipeline and features of market data. Once the pipeline is finalized, we will focus on sentiment data by adding Crypto Fear & Greed Index. And Etherscan's on-chain features will also be incorporated.

Data sources

- Yahoo Finance API（ETH–USD）
  The code retrieves historical daily market data for ETH-USD from Yahoo Finance API by using the yfinance library. After we retrieve the data, we use fields such as Open, High, Low, Close, and Volume. The retrieved data is automatically saved to the data directory. Features like daily returns and weekly volatility are calculated by process.py in the src directory.
- Kaggle Crypto Fear & Greed Index
  We already include functions in load.py in the src directory that can retrieve data from Kaggle and Etherscan API. However, the pipeline is not finalized due to configuration issues. I will focus on creating a stable dependency to the pipeline. And once the pipeline is finalized, I will aggregate the sentiment and Ethereum on-chain features with the ETH market features.

Issues / difficulties

- Inconsistent data formats
  One difficulty is the inconsistent data formats of the Yahoo Finance CSV file that I retrieved from the API. It contains extra headers like Ticker and Date, which is different from the yfinance library’s format. It leads to issues when reading and parsing dates and closing prices. To address this, I added preprocessing in the process_eth_daily() function in process.py in the src directory. It skips invalid rows, converts date and price fields into numerical values, and discard values that cannot be parsed.
- Weekly aggregation complexities
  The Yahoo market data contains missing days in a few weeks. These weeks contain fewer days which can lead to distortion of weekly volatility and weekly averages. To address this, I dropped weeks with insufficient daily returns and I used resample(‘W’) to ensure consistency.