Wallet Risk Scoring on Compound Protocol

Your Name

July 27, 2025

Problem Statement

The objective of this project is to assign a **risk score between 0 to 1000** to a given list of wallet addresses based on their historical transaction behavior on the Compound V2/V3 lending protocol. The project involves fetching on-chain data, processing it into meaningful features, and applying a scoring logic using both heuristic and machine learning models.

Step-by-Step Approach

1. Data Collection

We begin by loading wallet addresses from a text file, `wallet.txt`, which contains 100 Ethereum wallet addresses. For each address, transaction data is fetched (or loaded from previously saved files) using Compound protocol APIs or subgraphs like The Graph. Due to time constraints or API limitations, static mock data in JSON format is also used in `data/raw_data/`.

2. Feature Engineering

The following key risk-indicating features are extracted for each wallet:

- **borrow_supply_ratio:** The ratio of total borrow amount to total supplied amount.
- **repay_ratio:** The proportion of total borrowed amount that has been repaid.
- **liquidated_count:** The number of times the wallet has been liquidated.

These features are saved in a file named `processed_features.json`.

3. Feature Normalization

The raw feature values are normalized to a [0, 1] range using `MinMaxScaler` from the `scikit-learn` library. This ensures that the feature values are on a common scale before applying scoring.

4. Risk Scoring Methods

We apply two risk scoring approaches:

(a) Heuristic Scoring

A weighted scoring formula is used:

$$\text{score} = 1000 \times [w_1(1 - \text{borrow\_supply\_ratio}) + w_2 \times \text{repay\_ratio} + w_3(1 - \text{liquidated\_count})]$$

where $w_1 = 0.3$, $w_2 = 0.3$, and $w_3 = 0.4$. This penalizes high borrowing, poor repayments, and frequent liquidations.

(b) Machine Learning Model

We also train a simple classification model (e.g., Logistic Regression or Random Forest) using synthetic or labeled training data. The model is trained to classify wallets into "high risk" and "low risk", then converted to a continuous score between 0 and 1000.

5. Output Generation

The final risk scores for each wallet are saved to:

```
output/risk_scores.csv
```

with the format:

wallet_id	**score**
0xfaa0768b...ef2	732
...	...

Justification of Risk Indicators

- **High borrow/supply ratio** increases protocol exposure to undercollateralized risk.
- **Low repay ratio** indicates poor repayment behavior and higher default chances.
- **Frequent liquidation** suggests unstable or risky collateralization behavior.

Scalability and Flexibility

The modular Python scripts allow seamless replacement of static data with real-time data pipelines using APIs or subgraphs. Feature engineering and model training scripts can be extended with additional DeFi protocols or features.

Tools Used

- Python, Pandas, Scikit-learn
- The Graph or DeFi SDKs (e.g., web3py for Ethereum)
- CSV and JSON for data storage

Future Work

- Real-time integration with on-chain APIs
- Deep learning models for anomaly detection
- Visualization dashboards for wallet behavior