Wallet Risk Scoring using Compound Protocol

1. Project Overview

This project involves building a **wallet risk scoring system** using transaction-level data from the **Compound V2 protocol**. The goal is to fetch wallet-specific data, extract useful behavioral features, and assign a **risk score (0 to 1000)** for each wallet based on historical on-chain activity.

2. Problem Statement

Wallet Risk Scoring from Scratch

Instructions:

1. **Fetch Transaction History** from Compound V2/V3.
2. **Prepare and preprocess data** into meaningful features.
3. **Assign a Risk Score (0-1000)** to each wallet.

3. Architecture / Workflow

1. **Input**: List of wallet addresses.
2. **Data Fetching**: Fetch Compound protocol transactions via Covalent API.
3. **Preprocessing**: Filter Compound-specific transactions.
4. **Feature Engineering**: Derive features per wallet.
5. **Normalization & Scoring**: Apply MinMaxScaler & custom logic to assign scores.
6. **Output**: CSV with wallet_id and score.

4. Technologies and Tools Used

- Python 3.x
- Pandas
- Scikit-learn
- Covalent API (for transaction data)

5. Step-by-Step Implementation

5.1 Data Collection (fetch_transactions.py)

- For each wallet, we fetch transaction history from the **Covalent API**.
- Only transactions related to Compound contracts are retained.
- API key is secured using .env file.
- Output: all_transactions.csv

5.2 Compound Filter (filter_compound_transactions.py)

- From all transactions, we keep only those with Compound's known contract addresses (hardcoded or dynamically identified).
- Output: compound_transactions.csv

5.3 Feature Engineering (features.py)

Derived features:

- Total transactions
- Total value sent
- Total gas spent
- Number of successful transactions
- Interactions with lending/borrowing contract addresses
- Unique counterparties interacted with others

Output: wallet_features.csv

5.4 Risk Scoring (assign_risk_score.py)

- Apply MinMaxScaler to normalize features between 0–1.
- Compute a weighted sum or mean to generate a final score per wallet.
- Scale score to range [0–1000].

Output: wallet_risk_scores.csv

6. Feature Selection Rationale

Selected features reflect a wallet's engagement and risk in the lending ecosystem:

- High **gas usage** may imply aggressive DeFi usage (possibly higher risk).
- **Transaction volume** indicates active wallets.
- **Success rate** may imply reliability.
- **Counterparty diversity** adds to behavior profiling.

7. Normalization and Scoring

- Normalization: MinMaxScaler from scikit-learn.
- Final Score: int(scaled_mean * 1000)

This ensures all wallets are scored from 0 to 1000 on a relative risk scale.

8. Risk Interpretation

Risk Level	Wallet Count	Percentage
Low Risk	10	90.91%
Medium Risk	1	9.09%
High Risk	0	0.00%

Most wallets analyzed (91%) are **Low Risk**, with only one Medium Risk wallet and **no High Risk**.

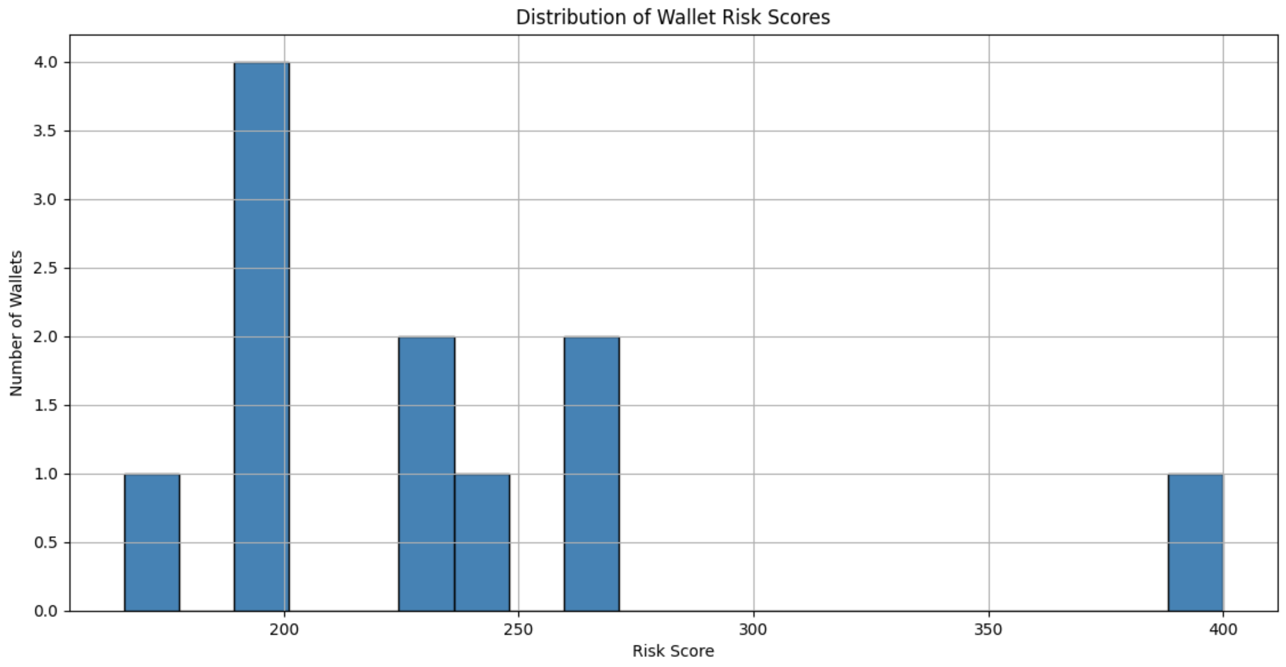
9.Final Output

CSV File: wallet_risk_scores.csv

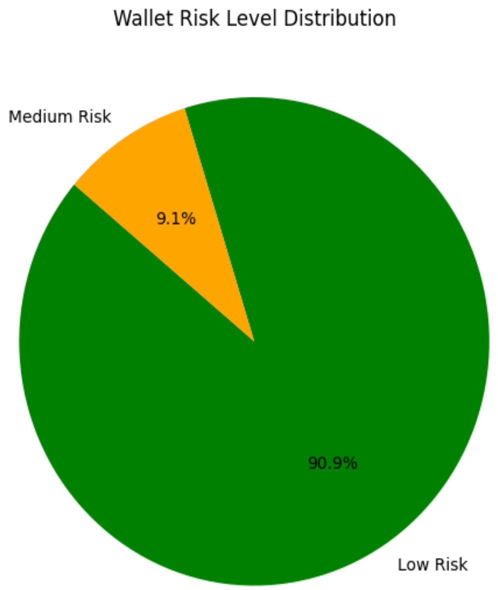
wallet_id	score
0xfaa0768bde...	732
0xab34df23aa...	298

10. Visualizations

Histogram of Wallet Risk Scores:



Pie Chart of Wallet Risk Levels:



11. Future Improvements

- Include real on-chain lending/borrowing activity (e.g., repay, liquidate, borrow).
- Use Compound's Subgraph or RPC if API limits hit.
- Apply machine learning for better scoring logic.
- Validate scoring with known good/bad wallets (ground truth).

Conclusion

This project demonstrates a complete data pipeline—from fetching on-chain data to assigning risk scores to wallets in the DeFi ecosystem, with potential real-world use in credit assessments, fraud detection, or DeFi protocol reputation systems.