Project Workflow Report: Wallet Risk Scoring for Compound Protocol

Overview

This project focused on developing a **wallet risk scoring system** for addresses interacting with the Compound protocol on the Arbitrum chain. The goal was to create a **risk score (0-1000 scale)** for each wallet based on their on-chain activity, leveraging open-source data from The Graph protocol's subgraphs.

1. Data Acquisition and Schema Exploration

Subgraphs Used: Multiple Compound protocol subgraphs were queried — both **Compound V2 and V3** — to gather comprehensive transaction and position data.

Schema Introspection:

- Two **introspection scripts** (introspection_query.py for V2 and introspection_query_v3.py for V3) were executed to fetch and analyze the schemas (in JSON format) of the respective subgraphs.
- This provided the structural understanding necessary for crafting precise GraphQL queries.

Challenges:

- Out of four tested archival subgraphs, only **one V2 subgraph returned valid data for the 100 input wallets**.
- **No V3 subgraph provided usable data** for these wallets, so the project pivoted to rely exclusively on the V2 subgraph data for the risk scoring pipeline.

2. Raw Data Collection

- A **dedicated query script** (compound_query.py) was developed based on the schema insights.
- It queried the V2 subgraph's transaction and position data for all 100 wallets.
- The output was a consolidated **compund_wallets_raw.csv** file capturing all raw protocol interactions required for further processing.

3. Feature Engineering

Using the raw data, an engineering script (feature_engineering.py) was executed to derive meaningful **risk-relevant features** such as:

total_supplied_usd, total_borrowed_usd, collateralization_ratio, repayment_rate, liquidations_suffered, withdraw_to_supply_ratio, among others — a total of **17 core features**.

These features summarized each wallet's borrowing and lending behavior, position diversification, repayment consistency, liquidation history, and engagement intensity.

4. Data Cleaning and Processing (On Google Colab)

Exploratory Data Analysis (EDA) was performed to understand feature distributions, correlations, and outlier presence.

Based on these insights, a **data cleaning pipeline** was implemented to:

- Clip outlier values at the 5th and 95th percentiles.
- Perform log-transformations for skewed monetary features.
- Remove constant and highly correlated features while preserving critical ones like collateralization_ratio and repayment_rate.
- Handle zero-valued entries in sensitive fields by substituting small positive values for stability.

The cleaned dataset was saved as **engineered_features_cleaned.csv**.

5. Heuristic Risk Scoring

A **transparent heuristic function** was crafted to assign initial risk scores:

- It combined normalized feature values using domain-driven weights emphasizing core risk drivers such as collateralization, repayment, liquidation history, withdrawal behavior, diversification, protocol activity, and borrow size.
- Scores ranged **1 (safest) to 1000 (riskiest)**, reflecting increasing risk with higher scores.

These scores were added to the dataset resulting in **engineered_features_with_scores.csv**.

6. Machine Learning Model Training and Comparison (Google Colab)

Three models were trained and evaluated on the data using the heuristic scores as target labels:

- **XGBoost Regressor**
- **LightGBM Regressor**
- **Random Forest Regressor**

Performance metrics such as **RMSE, MAE, and R^2** were calculated on validation sets.

XGBoost exhibited superior predictive performance, balancing accuracy and interpretability.

7. Final Modeling and Prediction

- Using the entire dataset with heuristic scores, an **XGBoost model was trained** as the final step.
- The trained model generated predicted risk scores for all wallets, stored in the **final_predictions.csv** file.
- These ML-generated scores offer a refined and robust assessment of wallet risk, informed by data-driven learning from protocol activity and heuristic logic.