System Design Document (HLD & LLD)

Project Title: Cryptocurrency Liquidity Prediction for Market Stability

1. High-Level Design (HLD)

1.1. Overview The system is a machine learning pipeline designed to predict the liquidity of cryptocurrencies, aiming to provide an early warning mechanism for potential market instability. It ingests historical market data, processes it, engineers predictive features, and trains a regression model to forecast a custom liquidity metric. The final output is accessible through a simple web-based interface.

1.2. System Architecture The architecture is composed of five core modules:

- **Data Collection:** Gathers raw market data from CSV files.
- **Data Preprocessing:** Cleans and prepares the data for analysis.
- **Feature Engineering:** Derives new, insightful features from the preprocessed data.
- **Model Training & Evaluation:** Trains and evaluates a predictive model.
- **Deployment:** Serves the trained model via a user-facing application.

1.3. Technology Stack

- **Programming Language:** Python
- **Data Manipulation:** Pandas
- **Machine Learning:** Scikit-learn
- **Web Framework:** Streamlit
- **Visualization:** Matplotlib, Seaborn

2. Low-Level Design (LLD)

2.1. Module: Data Preprocessing (src/preprocessing.py)

- **Function:** load_and_clean_data(filepaths)
  - **Description:** Loads one or more CSV files into a Pandas DataFrame, merges them, and handles missing (NaN) values to ensure data integrity.
  - **Input:** A list of file paths to the raw data CSVs.
  - **Output:** A single, cleaned Pandas DataFrame.
- **Function:** scale_data(df, columns_to_scale)
  - **Description:** Applies standard normalization to selected numerical columns to bring them to a common scale, which is essential for model performance.
  - **Input:** A Pandas DataFrame and a list of column names to scale.
  - **Output:** The DataFrame with scaled columns.
  -

2.2. Module: Feature Engineering (src/feature_engineering.py)

- **Function:** add_features(df)
    - **Description:** Engineers two new features: price_ma (a 3-day moving average of the price) and liquidity_ratio (a custom metric calculated from market cap and volume).
    - **Input:** The preprocessed DataFrame.
    - **Output:** The DataFrame with the two new feature columns.

2.3. Module: Model Training (src/model.py)

- **Function:** train_model(df, target_col)
    - **Description:** Trains a Random Forest Regressor model to predict the specified target column. After training, it evaluates the model's performance using R^2, RMSE, and MAE metrics and saves the trained model artifact for later use.
    - **Input:** The feature-engineered DataFrame and the name of the target column (liquidity_ratio).
    - **Output:** A trained model object.

2.4. Module: Deployment (app/app.py)

- **Description:** This script creates a Streamlit web application. It loads the pre-trained Random Forest model and provides a user interface where users can input cryptocurrency metrics to receive a real-time liquidity ratio prediction.