

Low-Level Design (LLD) Document

1. Data Preprocessing

- Load CSV files from CoinGecko snapshots.
- Convert columns to proper datatypes (e.g., float, datetime).
- Handle missing values by filling or removing rows.
- Replace infinite values with NaN and normalize using StandardScaler.

2. Feature Engineering

- Create 'volume_mcap_ratio' = $24h_volume / market_cap$ to measure liquidity.
- Generate volatility metrics such as absolute percentage changes over 1h, 24h, and 7d.
- Label data for classification: 1 for low liquidity (bottom 10%), 0 for normal liquidity.

3. Machine Learning Model

- Split data into train and test sets (80/20).
- Use RandomForestClassifier for initial modeling.
- Evaluate using accuracy, precision, recall, and F1-score.
- Perform cross-validation and hyperparameter tuning for optimal performance.

4. Deployment

- Build a Streamlit app to provide a user-friendly interface for predictions.
- Deploy in Google Colab using pyngrok to generate a public URL.
- Input parameters: price, 24h trading volume, and market cap.
- Output: Predicted liquidity status (High or Low).

5. Error Handling

- Handle missing or corrupted data gracefully.
- Validate input types and ranges in the Streamlit UI.
- Catch and log exceptions during model inference and data processing.