Pipeline Architecture Document

Project Title: Cryptocurrency Liquidity Prediction for Market Stability

---

Overview

This document describes the end-to-end data pipeline architecture used for building, training, and deploying the machine learning model that predicts liquidity ratios in cryptocurrencies.

Pipeline Stages

1. Data Ingestion

- **Source**: Two CSV files from CoinGecko (2022-03-16 & 2022-03-17)
- **Tool Used**: pandas
- **Action**: Loaded and merged into a single dataframe

2. Data Cleaning

- Handled missing values using forward-fill and backward-fill
- Removed unnecessary columns like `symbol`
- Ensured numeric format for key features

3. Feature Engineering

- Computed new features:
  - `liquidity_ratio` = volume / market_cap
  - `price_change_pct`
  - `price_ma_3`, `volume_ma_3`
  - `price_volatility_3`
- Refilled missing values after rolling operations

4. Feature Scaling

- Applied `MinMaxScaler` on numeric features for normalization

5. Model Development

- Model: RandomForestRegressor
- Target: `liquidity_ratio`
- Train-Test Split: 80:20 ratio
- Evaluation Metrics: R^2 Score, MAE, RMSE

6. Model Persistence

- Serialized model using `joblib.dump()` to `liquidity_model.pkl`

7. Deployment (Flask App)

- **Route `/`**: Displays input form (HTML UI)
- **Route `/predict`**: Accepts input, loads model, predicts liquidity
- Flask serves model as a backend for predictions

🔄 Logical Flow Diagram (Textual Representation)

```
Raw CSV Files
    ↓
Data Preprocessing (Clean, Normalize)
    ↓
Feature Engineering (MA, Volatility, Ratio)
    ↓
Model Training (Random Forest)
    ↓
Model Evaluation (R2, MAE, RMSE)
    ↓
Model Saved (.pkl)
    ↓
Flask App (Input Form → Prediction Output)
```

Technologies Used

- **Python Libraries**: pandas, numpy, sklearn, joblib
- **Visualization**: seaborn, matplotlib
- **Deployment**: Flask Web Framework
