Cryptocurrency Liquidity Prediction - Project Report

1. High-Level Design (HLD)

This project aims to forecast cryptocurrency liquidity using a machine learning pipeline that includes:
- Data collection (CoinGecko)
- Data preprocessing (cleaning, normalization)
- Feature engineering (custom liquidity metrics)
- Model training (Linear Regression)
- Deployment (Streamlit)

Tech Stack: Python, pandas, scikit-learn, Streamlit.

2. Low-Level Design (LLD)

- Data Files: Two CSVs (March 16 & 17, 2022) from CoinGecko.
- Preprocessing: Dropping missing values, normalizing using MinMaxScaler.
- Engineered Features:
  - cap_to_volume = mkt_cap / volume
  - weighted_change = 24h * volume
  - liquidity_ratio = volume / price
- Model: LinearRegression trained on 8 features to predict liquidity_ratio.
- Streamlit UI: Sliders take normalized inputs and show prediction.

3. Model Performance

- Model: Linear Regression
- Features: price, percent changes, volume, mkt_cap, engineered ratios
- Target: liquidity_ratio
- Metrics:
  - RMSE (training): Low (exact value depends on small dataset)
  - R2 Score: Indicates decent fit, but overfit likely due to no split

4. EDA Insights

- Price is heavily skewed to the lower end of the scale.
- Liquidity ratio increases with volume.
- Strong positive correlations between liquidity_ratio and volume/mkt_cap.

5. Deployment

- Tool: Streamlit
- How to Run: streamlit run app.py
- Interface: User provides market inputs, app returns predicted liquidity.
- Currently trains model live from CSV (demo mode).

Cryptocurrency Liquidity Prediction - Project Report

6. Conclusion

This project shows a full ML pipeline for liquidity prediction:
- Preprocessing, feature engineering, regression model, deployment.
With more historical data and advanced models (XGBoost, LSTM), this tool could support real-time risk management in crypto trading platforms.