

Low-Level Design (LLD)

Generated by [Avinesh Masih](#) – View on GitHub: [AVINESH MASIH](#)

Project Name: Cryptocurrency Liquidity Prediction for Market Stability

1. Data Ingestion

- **Files Used:**
 - coin_gecko_2022-03-16.csv
 - coin_gecko_2022-03-17.csv
- **Steps:**
 - Load both CSVs using `pandas.read_csv()`
 - Merge datasets into a single DataFrame using `pd.concat()`

2. Data Cleaning

- Drop missing values using `dropna()`
- Remove duplicate rows using `drop_duplicates()`
- Convert columns to appropriate types (e.g., date to datetime, numeric conversions)

3. Feature Engineering

- **Price Moving Average (2-period):**
 - `df['price_MA_2d'] = df['price'].rolling(window=2).mean()`
- **Market Cap Moving Average (2-period):**
 - `df['market_cap_MA_2d'] = df['mkt_cap'].rolling(window=2).mean()`
- **Volatility:**
 - `df['volatility'] = (df['24h'] - df['1h']).abs()`
- **Liquidity Ratio:**
 - `df['liquidity_ratio'] = df['24h_volume'] / df['mkt_cap']`

4. Exploratory Data Analysis (EDA)

- Plot **Ethereum price trends over time** using matplotlib

- Generate a **correlation heatmap** using `seaborn.heatmap()`
 - Summarize data using `df.describe()`, `info()`, etc.
-

5. Model Building

- **Train-Test Split:**
 - Use `train_test_split()` from `sklearn.model_selection`
 - **Models Used:**
 - Linear Regression (`LinearRegression`)
 - Random Forest Regressor (`RandomForestRegressor`)
 - **(Optional):** Hyperparameter tuning using `GridSearchCV` or `RandomizedSearchCV`
-

6. Model Evaluation

- **Metrics:**
 - Root Mean Squared Error (RMSE)
 - Mean Absolute Error (MAE)
 - R^2 Score

```
from sklearn.metrics import mean_squared_error, mean_absolute_error, r2_score
```

7. Model Saving

- Save the trained model using Joblib:

```
import joblib
```

```
joblib.dump(model, 'models/ crypto_liquidity_rf_model.pkl')
```

8. Local Deployment

- Build a **Streamlit** web application
 - Load the `.pkl` model
 - Accept user inputs (price, volume, market cap)
 - Display predicted liquidity ratio
-

LLD Flow Diagram

