Pipeline Architecture Document

1. Architectural Diagram

The pipeline is structured into five distinct stages: Data Ingestion, Data Preprocessing & Feature Engineering, Model Training, Model Evaluation, and Model Persistence & Output.

---

2. Pipeline Stages and Data Flow

Stage	Input	Process	Output	Artifacts
1. Data Ingestion	Raw CSV files (coin_gecko_*.csv)	Loads files, merges, sorts, and converts the data into a single DataFrame.	Combined Raw DataFrame	$\text{df\_raw}$
2. Data Preprocessing & Feature Engineering	Combined Raw DataFrame	1. **Preprocessing:** Drop non-numeric columns (coin, symbol, date), Impute missing values (median). 2. **Feature Engineering:** Calculate the **$\text{Liquidity\_Index}$** target and $\text{Vol\_Price\_Ratio}$ feature. 3. **Scaling & Split:** Scale all features using **$\text{MinMaxScaler}$**, then split into Train/Test sets.	Scaled Train/Test Sets ($\text{X}_{\text{train}}, \text{X}_{\text{test}}, \text{y}_{\text{train}}, \text{y}_{\text{test}}$)	$\text{df\_featured}$, $\text{data\_scaler.pkl}$
3. Model Training	Scaled Train/Test Sets ($\text{X}_{\text{train}}, \text{y}_{\text{train}}$)	Initializes and trains the **$\text{RandomForestRegressor}$** model using the training features and the target $\text{Liquidity\_Index}$.	Trained ML Model	$\text{model}$ object

Stage	Input	Process	Output	Artifacts
4. Model Evaluation	Trained Model, Test Sets ($\text{X}_{\text{test}}, \text{y}_{\text{test}}$)	Uses the trained model to predict values for $\text{X}_{\text{test}}$, then compares predictions against $\text{y}_{\text{test}}$.	Performance Metrics (RMSE, MAE, $R^2$)	$\text{metrics\_report}$
5. Model Persistence & Output	Trained Model, Scaler, Metrics	Saves the trained model and scaler to disk for future deployment/inference. Generates the final project documentation.	Persistent Model Files	$\text{liquidity\_predictor.pkl}$, $\text{data\_scaler.pkl}$

---

3. Data Integrity and Control

Aspect	Description	Justification
Data Quality Check	Initial check for missing values and data types upon ingestion.	Ensures robustness; handled by **median imputation** for numerical stability.
Feature Transformation	**MinMaxScaler** applied across all numerical features (excluding the target during the split).	Standardizes the wide range of values (e.g., price vs. market cap) for effective model training.
Data Split Integrity	**Shuffle is set to $\text{False}$** during $\text{train\_test\_split}$ (or a time-series split is implied).	Crucial for time-series data to prevent future information from "leaking" into the training set, ensuring realistic model evaluation.

---

4. Output Documentation

The final deliverable package (e.g., GitHub repository or zipped folder) contains the outputs from the pipeline stages, fulfilling all project requirements:

1. **Machine Learning Model:** $\text{liquidity\_predictor.pkl}$ (The trained $\text{RandomForestRegressor}$).
2. **Data Processing & Feature Engineering:** $\text{data\_scaler.pkl}$ (The saved $\text{MinMaxScaler}$) and a conceptual definition of all engineered features.
3. **Exploratory Data Analysis (EDA) Report:** Summary of descriptive statistics and correlation analysis.
4. **Project Documentation:**
    - **High-Level Design (HLD) Document:** Architectural overview and component breakdown.
    - **Low-Level Design (LLD) Document:** Detailed function specifications and algorithms.
    - **Pipeline Architecture Document (This Document):** Description of the data flow and stages.
    - **Final Report (Conceptual):** Summary of findings, model performance (RMSE, MAE, $R^2$), and key insights into liquidity drivers.