

# Low-Level Design (LLD)

## Script-Level Flow

### 1. `data_preparation_ml_project_sid.py`

- Reads two raw CSV files
- Adds a `date` column to each
- Merges, drops irrelevant columns, handles missing values
- Outputs: `merged_cleaned_crypto_data.csv`

### 2. `feature_engineering_ml_project_sid.py`

- Loads cleaned data
- Computes new features:
  - `liquidity_ratio = total_volume / market_cap`
  - `price_change_ratio = price_change_percentage_24h / current_price`
  - `cap_per_supply = market_cap / circulating_supply`
- Categorizes `liquidity_level` into 3 bins using `pd.qcut`
- Outputs: `engineered_crypto_data.csv`

### 3. `model_training_ml_project_sid.py`

- Loads engineered dataset
- Splits into `X` and `y`

- Scales `X` using `StandardScaler`
- Trains `RandomForestClassifier`
- Evaluates using `classification_report` and `confusion_matrix`
- Saves model and scaler to `.pkl` files

#### 4. `app_ml_project_sid.py`

- Streamlit web app
- Accepts user inputs for key features
- Recalculates derived features on the fly
- Scales inputs and predicts with trained model
- Displays the predicted `liquidity_level`

### Data Flow Diagram

- Raw CSVs → Data Preparation → Cleaned CSV
- ↓
- Feature Engineering → Engineered CSV
- ↓
- Model Training → Model + Scaler
- ↓
- Streamlit App