HLD & LLD Document

High-Level Design (HLD)

- **Project Overview:**
  Predict cryptocurrency closing prices and volatility using machine learning and deep learning methods.
- **Objectives:**
  - Build predictive models for crypto price/volatility forecasting.
  - Compare classical ML models (Random Forest) and deep learning (LSTM).
  - Feature engineering with technical indicators and time features.
- **Architecture Diagram:**
  Include a block diagram showing:
  Data collection → Data preprocessing → Feature engineering → Model training (RF & LSTM) → Model evaluation → Deployment (optional)
- **Data Flow:**
  - Input raw crypto historical data (prices, volume).
  - Apply preprocessing & feature extraction.
  - Train/test split.
  - Model training.
  - Prediction & evaluation.
- **Technology Stack:**
  - Python, Pandas, NumPy, Scikit-learn, TensorFlow/Keras, Matplotlib, Seaborn.

Low-Level Design (LLD)

- **Data Preprocessing:**
  - Handling missing values: drop rows <5% missing, impute others.
  - Encoding categorical variables (`crypto_name` via LabelEncoder).
  - Scaling numerical features using StandardScaler.
- **Feature Engineering:**
  - Rolling statistics: moving average, volatility, Bollinger Bands, ATR.
  - Date-based features: day, month, year.
- **Modeling:**
  - Random Forest Regressor: hyperparameters, feature importance selection.

    - LSTM: input sequences, network architecture (layers, neurons, activations, dropout), loss function, optimizer.

- **Evaluation Metrics:**
  MAE, RMSE, R^2 Score.
- **Pipeline:**
  Steps from data ingestion → preprocessing → training → validation → prediction.