High-Level Design Document: Cryptocurrency Volatility Prediction System

Contents

Introduction

This High-Level Design (HLD) document outlines the architecture and components of a machine learning system designed to predict cryptocurrency volatility. The system processes historical cryptocurrency market data to forecast the 7-day volatility of various cryptocurrencies, leveraging exploratory data analysis (EDA), feature engineering, and predictive modeling. The primary goal is to provide accurate volatility predictions to inform trading or investment strategies.

1 System Scope

The system focuses on:

- Collecting and cleaning historical cryptocurrency data, including price, volume, and market capitalization.
- Performing exploratory data analysis to identify patterns and trends.
- Engineering features to enhance predictive power for volatility forecasting.
- Training and evaluating multiple machine learning models to predict 7-day volatility.
- Selecting and optimizing the best-performing model for deployment.

2 System Architecture

The system follows a modular pipeline architecture, with the following high-level components:

- Data Ingestion Module: Loads raw cryptocurrency data from a CSV file containing historical market data (open, high, low, close, volume, market cap, timestamp, crypto name, and date).
- Data Preprocessing Module: Cleans the dataset by removing unnecessary columns and converting date/timestamp fields to appropriate formats.
- Exploratory Data Analysis (EDA) Module: Analyzes data distributions, visualizes price trends, and identifies patterns in volume and market capitalization.
- Feature Engineering Module: Creates new features such as moving averages, volatility metrics, and time-based features to enhance model performance.

- Model Training and Evaluation Module: Trains multiple machine learning models, evaluates them using time-series cross-validation, and selects the best model based on performance metrics (RMSE, Rš).
- Model Deployment Module: Saves the trained model for future use in predictions.

3 Architecture Diagram

The system follows a sequential pipeline where raw data flows through preprocessing, EDA, feature engineering, model training, and deployment. Due to LaTeX limitations, a textual description is provided instead of a diagram:

- Input: Raw CSV data → Preprocessing: Cleaned data → EDA: Insights and visualizations → Feature Engineering: Enhanced dataset → Model Training: Trained models → Deployment: Saved model file.

4 System Components

4.1 Data Ingestion

- Input: CSV file with 72,946 records and 9 columns (open, high, low, close, volume, marketCap, timestamp, crypto_name, date).
- Function: Reads data using pandas and ensures proper loading into memory.

4.2 Data Preprocessing

- Tasks: Drops unnecessary columns (e.g., ’Unnamed: 0’), converts ’date’ and ’timestamp’ to datetime formats, and verifies no missing or duplicate records.
- Output: Cleaned dataset ready for analysis.

4.3 Exploratory Data Analysis (EDA)

- Tasks: Visualizes price distributions (open, high, low, close) using histograms with KDE curves, analyzes volume and market cap, and identifies patterns such as volatility and price clusters.
- Output: Insights into data distributions and market behaviors.

4.4 Feature Engineering

- Tasks: Creates features like Simple Moving Averages (SMA), volatility metrics, log returns, liquidity ratios, Bollinger Bands, and time-based features (year, month, day, day of week).

- Output: Enhanced dataset with 74 columns, including one-hot encoded cryptocurrency names and the target variable (Volatility_7_target).

4.5 Model Training and Evaluation

- Tasks: Trains multiple regression models (e.g., Linear Regression, Random Forest, LightGBM) using a time-series split (80% train, 20% test). Evaluates models using RMSE and Rš metrics.
- Output: Performance report and selection of the best model (LightGBM with RMSE 27.03 and Rš 0.51).

4.6 Model Deployment

- Tasks: Saves the best model (LightGBM) to a pickle file for future predictions.
- Output: Deployable model file (best_model.pkl).

5 Data Flow

1. Raw data is ingested from a CSV file.
2. Data is cleaned and preprocessed to remove inconsistencies.
3. EDA generates visualizations and insights to guide feature engineering.
4. Features are engineered to create a robust dataset for modeling.
5. Models are trained and evaluated using time-series splits to ensure chronological integrity.
6. The best model is optimized using hyperparameter tuning and saved for deployment.

6 Non-Functional Requirements

- Scalability: The system can handle large datasets (72,946 records) and can be extended to include more cryptocurrencies or features.
- Performance: Models are evaluated for low RMSE and high Rš, with LightGBM achieving the best performance.
- Reliability: Time-series splits ensure robust evaluation, and data cleaning ensures no missing or duplicate data.
- Maintainability: Modular code structure allows easy updates to preprocessing, feature engineering, or model selection.

7 Conclusion

The cryptocurrency volatility prediction system is designed as a modular pipeline that processes historical market data to predict 7-day volatility. The architecture ensures scalability, reliability, and maintainability, with LightGBM identified as the best-performing model. This HLD provides a foundation for implementing and extending the system.