High - Level Design (HLD) Document

Cryptocurrency Volatility Prediction Using Machine Learning

1. Introduction

Cryptocurrency markets are highly volatile due to rapid price fluctuations, speculative trading, and sensitivity to global events. Accurate volatility prediction is essential for risk management, portfolio optimization, and informed trading decisions.

This project focuses on predicting next-day volatility of Bitcoin using historical market data and machine learning techniques. The system processes historical price and volume data, computes volatility using statistical methods, engineers relevant features, and applies a regression-based machine learning model to generate volatility predictions.

2. Project Obbjective

The primary objectives of this project are:

- To analyze historical Bitcoin price data
- To compute and model market volatility
- To predict next-day volatility using engineered features
- To evaluate model performance using standard regression metrics

The project is designed as time-series prediction system with a clear end-to-end pipeline.

3. System Overview

The system follows a data-driven machine learning architecture. It transforms raw historical cryptocurrency data into meaningful featuresn and uses a trained model to predict future volatility.

Key Characteristics

- Focused on Bitcoin only for clarity and stability
- Uses rolling statistical measures for volatility
- Applying Rabdom Forest Regressor to capture non-linear patterns
- Ensures time=based train-test split to avoid data leakage

High-Level System Architecture

The system consists of the following major components:

1. Data Ingestion Module
2. Data Preprocessing Module

3. Volatility Computation Module
4. Faeture Engineering Module
5. Exploratory Data Analysis (EDA)
6. Machine Learning Model
7. Model Evaluation and Visualization

Each module ooperates sequentially, forming a raw structure pipeline from raw data to final prediction.

5. Component Description (High Level)

(5.1) Data Ingestion Module

- Loads historical cryptocurrency data from a compressed CSV file
- Reads data into a structured DataFrame
- Supports scalable ingestion for multiple cryptocurrencies

(5.2) 5.2 Data Preprocessing Module

- Filters the dataset to include Bitcoin only
- Converts date fields into datetime format
- Sorts data chronologically for time-series consistency
- Removes invalid records such as zero trading volume

This ensures clean and reliable input data for modeling.

(5.3) Volatility Computation Module

- Computes daily log returns from closing prices
- Calculates rolling volatility using a 7-day window
- Defines next-day volatility as the prediction target

Volatility is treated as a derived statistical feature rather than a raw input.

(5.4) 5.4 Feature Engineering Module

To enhance predictive power, additional features are generated:

- Price movement indicators (absolute returns, high–low range)
- Liquidity-based measures (volume to market cap ratio)
- Lag-based volatility features (1, 3, 7, 14 days)

These features help capture temporal dependency and market behavior.

(5.5) 5.5 Exploratory Data Analysis (EDA)

EDA is performed to understand:

- Long-term Bitcoin price trends
- Volatility behavior over time
- Relationships between engineered features

Visual tools such as line plots and correlation heatmaps are used to validate feature relevance.

(5.6) 5.6 Machine Learning Model

- A Random Forest Regressor is used for volatility prediction
- The model learns non-linear relationships between historical features and future volatility
- Hyperparameters are selected to balance bias and variance

The model is trained on historical data and tested on unseen future data.

(5.7) 5.7 Model Evaluation & Visualization

Model performance is evaluated using:

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

Prediction results are visualized using:

- Feature importance plots
- Actual vs predicted volatility curves

6. Assumptions and Constraints

Assumptions

- Historical volatility patterns contain predictive information
- Bitcoin data is sufficient to validate the modeling approach
- Market conditions are stationary within short time windows

Constraints

- Model is trained on historical data only
- External factors such as news or macroeconomic events are not included

- Predictions are limited to short-term volatility

7. Conclusion

The High-Level Design outlines a structured and modular system for Bitcoin volatility prediction. The architecture ensures clarity, scalability, and alignment with machine learning best practices. Detailed implementation details are intentionally abstracted and will be covered in subsequent design documents.