Cryptocurrency Volatility Prediction

Language: Python
Model Type: Regression
Includes: Deployment + Diagrams

A Machine Learning Project for Predicting Cryptocurrency Volatility Levels using Python

Abstract

This project focuses on predicting cryptocurrency volatility using regression-based machine learning models. It uses historical OHLC (Open, High, Low, Close) data along with trading volume and market capitalization. The system applies data preprocessing, feature engineering, exploratory analysis, and model training steps to estimate volatility patterns, assisting investors and analysts in making informed financial decisions.

Dataset Description

The dataset contains daily records of over 50 cryptocurrencies with the following fields: Date, Symbol, Open, High, Low, Close, Volume, and Market Cap. It represents historical performance and trading behavior across multiple assets.

Data Preprocessing

• Handle missing values using interpolation and removal of anomalies.
• Normalize numerical columns using MinMaxScaler.
• Encode categorical features (cryptocurrency symbols).
• Convert Date fields to datetime format.
• Ensure dataset consistency and alignment across multiple currencies.

Exploratory Data Analysis (EDA)

Exploratory analysis included trend visualization, correlation heatmaps, and volatility distributions. Key insights identified correlations between volume and market cap, and patterns of volatility clustering during high trading activity periods.

Feature Engineering

• Rolling Volatility (Standard deviation of returns over 7-day and 30-day windows).
• Liquidity Ratio (Volume / Market Cap).
• Moving Averages (7-day, 14-day, 30-day).
• Technical Indicators: Bollinger Bands, ATR (Average True Range).

Model Building & Evaluation

Regression models were explored, including Linear Regression, Random Forest Regressor, and XGBoost Regressor. The best-performing model was selected based on R^2 score, MAE, and RMSE. Cross-validation was used for generalization assessment.

Metric	Description
MAE	Mean Absolute Error – Measures average absolute deviation
RMSE	Root Mean Squared Error – Penalizes large errors
R^2 Score	Indicates how well model fits variance in volatility

High-Level Design (HLD)

The system follows a modular ML pipeline architecture consisting of Data Ingestion, Preprocessing, Feature Engineering, Model Training, Evaluation, and Deployment. Each stage communicates via serialized data objects and configuration files.

Low-Level Design (LLD)

1. DataLoader class for dataset ingestion.
2. Preprocessor module handling scaling, encoding, and imputation.
3. FeatureEngineer module for technical indicator computation.
4. ModelTrainer for regression model fitting and saving.
5. Evaluator for generating metrics and plots.
6. Flask-based Deployment script for real-time predictions.

Pipeline Architecture

Data → Preprocessing → Feature Engineering → Model Training → Evaluation → Deployment (Flask API)

Deployment (Flask)

The model is deployed using a Flask web application. Users can input latest OHLC and volume data, and receive predicted volatility values. The model file is serialized using joblib, and Flask endpoints serve predictions as JSON responses.

Results and Insights

The regression model achieved high accuracy with $R^2 \approx 0.89$, indicating strong predictability of volatility trends. Significant variables include trading volume and short-term price range (High - Low).

Conclusion

This project successfully demonstrates how machine learning regression techniques can predict cryptocurrency volatility. It establishes a framework for financial forecasting using structured and time-based data.

References

1. Pwskills ML Course Materials
2. CoinMarketCap Historical Data
3. Scikit-learn Documentation
4. Flask Official Docs