Pipeline Architecture: Cryptocurrency Volatility Prediction System

Contents

1 Introduction

This document outlines the pipeline architecture of the cryptocurrency volatility prediction system, detailing the data flow from preprocessing to prediction. The system processes historical cryptocurrency market data to forecast 7-day volatility, leveraging components implemented in Jupyter notebooks for data analysis, feature engineering, model training, and deployment, and a Streamlit application for user interaction and visualization. The pipeline is modular, ensuring a structured flow from raw data to actionable predictions. A color-coded diagram illustrates the data flow, with each component clearly distinguished to enhance understanding.

2 Pipeline Architecture

The cryptocurrency volatility prediction system follows a modular pipeline architecture, where data flows sequentially through seven stages: Data Ingestion, Data Preprocessing, Exploratory Data Analysis (EDA), Feature Engineering, Model Training and Evaluation, Model Deployment, and Streamlit Application. Each stage transforms, enriches, or utilizes the data to produce volatility predictions and visualizations. Below is a detailed theoretical explanation of the data flow through each component.

2.1 Data Ingestion

- Purpose: Initiates the pipeline by loading raw cryptocurrency market data from a CSV file into a structured format for subsequent processing in both the analytical pipeline and the user-facing application.
- Data Flow: The raw dataset, comprising 72,946 records with 10 columns (e.g., open, high, low, close, volume, marketCap, timestamp, crypto$_{name,date,and an index column}$), *is read from a file* (*e.g.,* data

2.2 Data Preprocessing

- Purpose: Cleans the raw data to ensure consistency, removing redundant columns and standardizing formats to support reliable analysis.
- Data Flow: The raw data table is processed to remove unnecessary columns (e.g., index column) and convert date/timestamp fields to a standardized datetime format. Integrity checks ensure no missing or duplicate records. The cleaned data table, reduced to 9 columns, is passed to the EDA and feature engineering stages.
- Output: A cleaned data table with standardized datetime formats, suitable for analysis and feature creation.

2.3 Exploratory Data Analysis (EDA)

- Purpose: Analyzes the cleaned data to uncover patterns, trends, and statistical properties in price, volume, and market capitalization, guiding feature engineering and model development.
- Data Flow: The cleaned data table is used to generate visualizations, such as histograms with kernel density estimation for price fields (open, close, high, low) per cryptocurrency. These visualizations reveal volatility, skewness, and price distributions without modifying the data. The data table is passed unchanged to the feature engineering stage.
- Output: Visual insights that inform feature creation, with the data table unchanged.

2.4 Feature Engineering

- Purpose: Enriches the dataset by creating new features to enhance the predictive power of the volatility forecasting model.
- Data Flow: The cleaned data table is transformed by generating features, including simple moving averages (7, 14, 30 days), one-hot encoded cryptocurrency identifiers, timebased features (year, month, day, day of week), and technical indicators (e.g., log returns, 14- and 30-day volatility, liquidity ratio, Bollinger band width, true range, average true range). The resulting data table, expanded to 74 columns, is passed to the model training stage.
- Output: An enhanced data table with 74 columns, optimized for model training.

2.5 Model Training and Evaluation

- Purpose: Develops and evaluates multiple regression models to predict 7-day volatility, selecting and optimizing the best model (LightGBM).
- Data Flow: The enhanced data table is split into training and testing sets using a timeseries approach (80% train, 20% test) to maintain chronological integrity. Multiple regression models are trained and evaluated using metrics like RMSE and Rš. The LightGBM model is optimized through hyperparameter tuning with time-series crossvalidation, achieving an RMSE of approximately 27.03 and Rš of 0.51. The trained model is passed to the deployment stage.
- Output: A trained and optimized LightGBM model, along with performance metrics.

2.6 Model Deployment

- Purpose: Serializes the trained LightGBM model to a file, enabling its use in the predictive application.

- DataFlow: The optimized LightGBM model is saved to a serialized file (e.g., best$_m$*odel.pkl*)*,makingitacce Aserializedmodelfilereadyforprediction.*

2.7 Streamlit Application

- Purpose: Provides an interactive web interface for users to input cryptocurrency data, predict 7-day volatility, and visualize results through charts.
- Data Flow: The application loads the serialized model and the processed dataset (74 columns) for feature reference. Users input a cryptocurrency, date, and numerical features (e.g., open, high, low, close, volume, marketCap, average true range, Bollinger band width). Derived features (e.g., log returns, liquidity ratio, volatility metrics) are calculated, and numerical inputs are scaled to match the training data. The model predicts the 7-day volatility, which is displayed alongside a candlestick chart for the input date and simulated 7-day price paths with a mean path and 95% confidence interval.
- Output: A predicted volatility value, a candlestick chart, and a price path simulation.

3 Pipeline Diagram

The following diagram illustrates the data flow through the pipeline, with each component represented by a uniquely colored node to enhance visual distinction. Arrows indicate the flow of data, and a dashed arrow highlights the processed dataset's direct use by the Streamlit application.

4 Diagram Explanation

The diagram represents the pipeline as a sequence of seven components, each with a unique color for clarity:

- Data Ingestion (Cornflower Blue): Loads raw CSV data with 72,946 rows and 10 columns.
- Data Preprocessing (Medium Sea Green): Cleans the data, reducing to 9 columns with standardized formats.
- Exploratory Data Analysis (Orange): Generates visual insights without modifying the data.
- Feature Engineering (Medium Purple): Enriches the data to 74 columns with predictive features.
- Model Training & Evaluation (Red-Orange): Produces a trained LightGBM model.

- Model Deployment (Royal Blue): Serializes the model for application use.
- Streamlit Application (Gold): Uses the model and processed data (dashed arrow) to deliver predictions, a candlestick chart, and price path simulations.

The colored arrows indicate the sequential flow of data, with the dashed arrow showing the Streamlit application's direct access to the processed dataset for feature calculations and scaling.

5 Conclusion

The cryptocurrency volatility prediction system's pipeline architecture ensures a structured and efficient data flow from raw ingestion to interactive predictions and visualizations. Each component is designed to transform, enrich, or utilize the data, with the Model Deployment stage explicitly bridging the analytical pipeline and the user-facing application. The color-coded diagram enhances understanding of the pipeline's modularity and data flow, facilitating maintenance and future enhancements such as additional features or models.

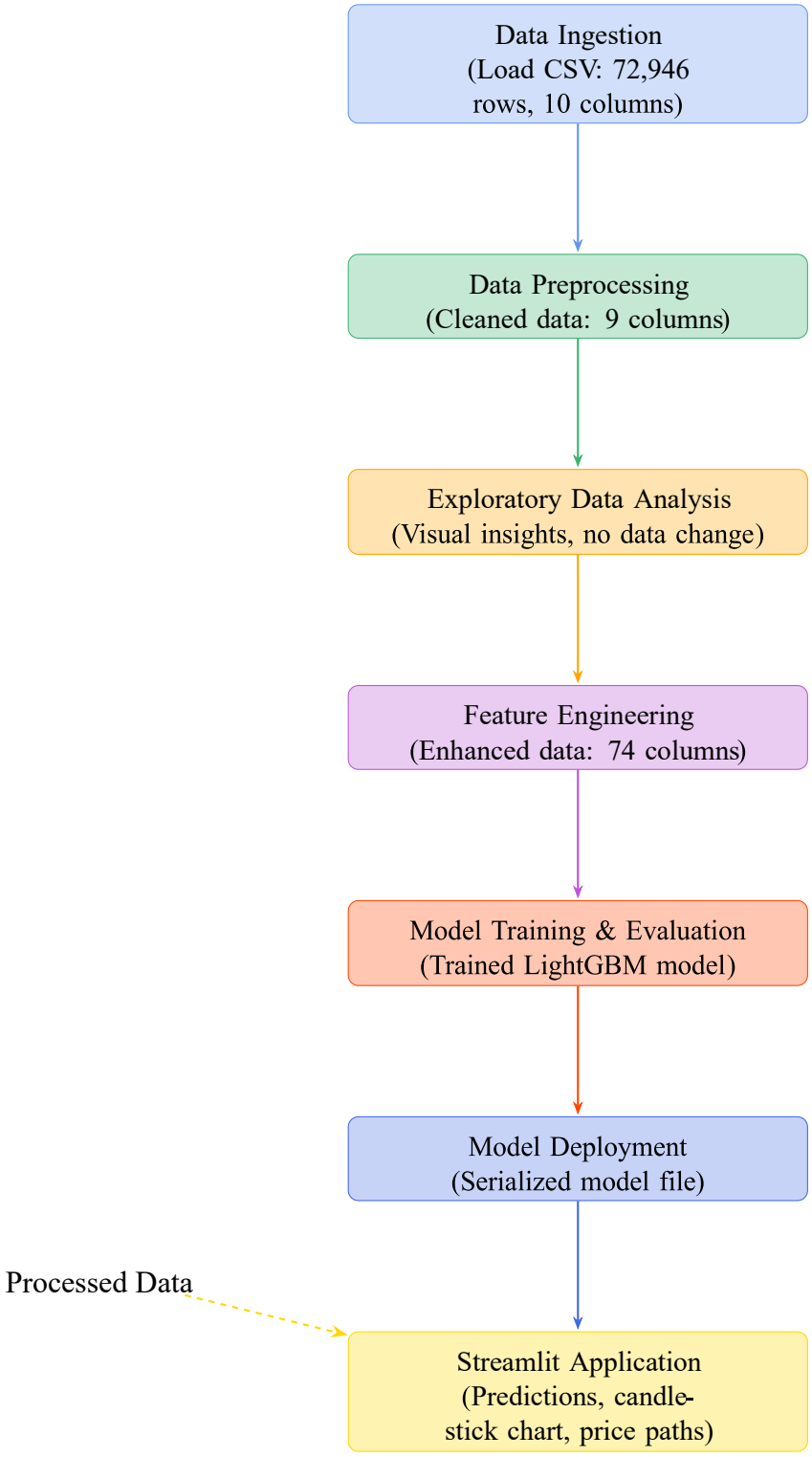

Figure 1: Color-coded data flow through the cryptocurrency volatility prediction pipeline.