Pipeline Architecture Document

Project: Cryptocurrency Liquidity Prediction for Market Stability

Name: Shreya Patra

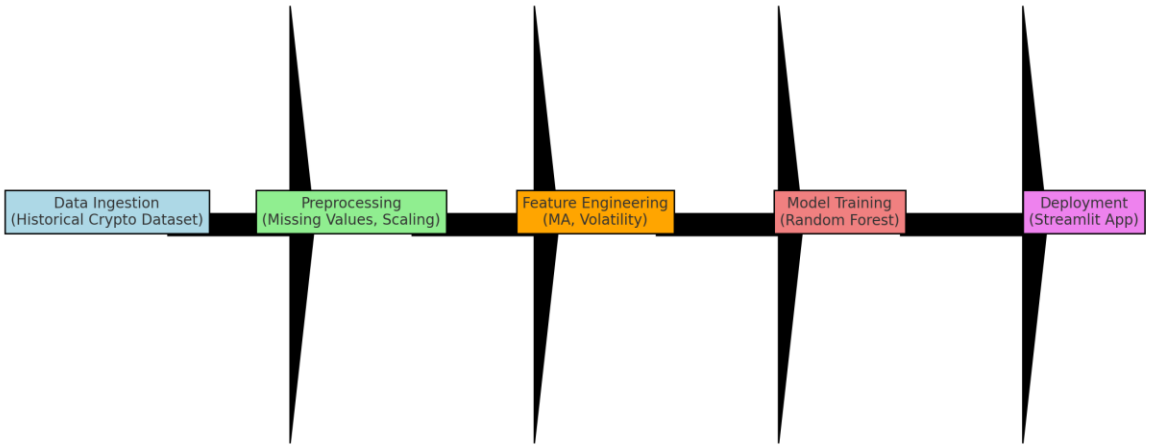
Date: 18 July 2025

1. Introduction

The pipeline architecture for this project outlines the entire process from collecting raw cryptocurrency data to making real-time predictions in a deployed web application. It helps ensure a systematic, modular, and repeatable approach to machine learning development and deployment.

2. Pipeline Flow Diagram

The diagram below provides a visual overview of how data flows through the system:



3. Component Descriptions

- Data Ingestion:

We begin by importing historical cryptocurrency data, including prices, trading volumes, and

market capitalization. This data forms the foundation of our predictive system.

- Preprocessing:

At this stage, missing values are handled using methods like forward-fill and mean replacement. All numerical features are normalized using MinMaxScaler to keep values within a standard range.

- Feature Engineering:

We enhance the dataset by adding useful indicators such as 7-day moving averages, daily volatility, and liquidity ratios. These features help the model capture important market dynamics.

- Model Training:

Using a Random Forest Regressor, we train the model on historical data to learn patterns related to market behavior and liquidity. The model is evaluated using metrics like RMSE, MAE, and R^2.

- Deployment:

Finally, we integrate the trained model into a Streamlit app. Users can interact with this app to input custom feature values and instantly get predictions for Bitcoin liquidity.

4. Technology Stack

The following tools and libraries were used to build and deploy the pipeline:
- Python
- Pandas
- NumPy
- Scikit-learn
- Streamlit
- Matplotlib & Seaborn
- Joblib (for model saving)

5. Summary

This pipeline ensures that the project remains scalable and easy to maintain. With each component separated into logical stages, it becomes easier to troubleshoot, improve, or expand the system in future iterations.