High-Level Design (HLD) Document

Project Title: Cryptocurrency Liquidity Prediction
Author: Sanyuktha Hansraj

1. Objective of the System

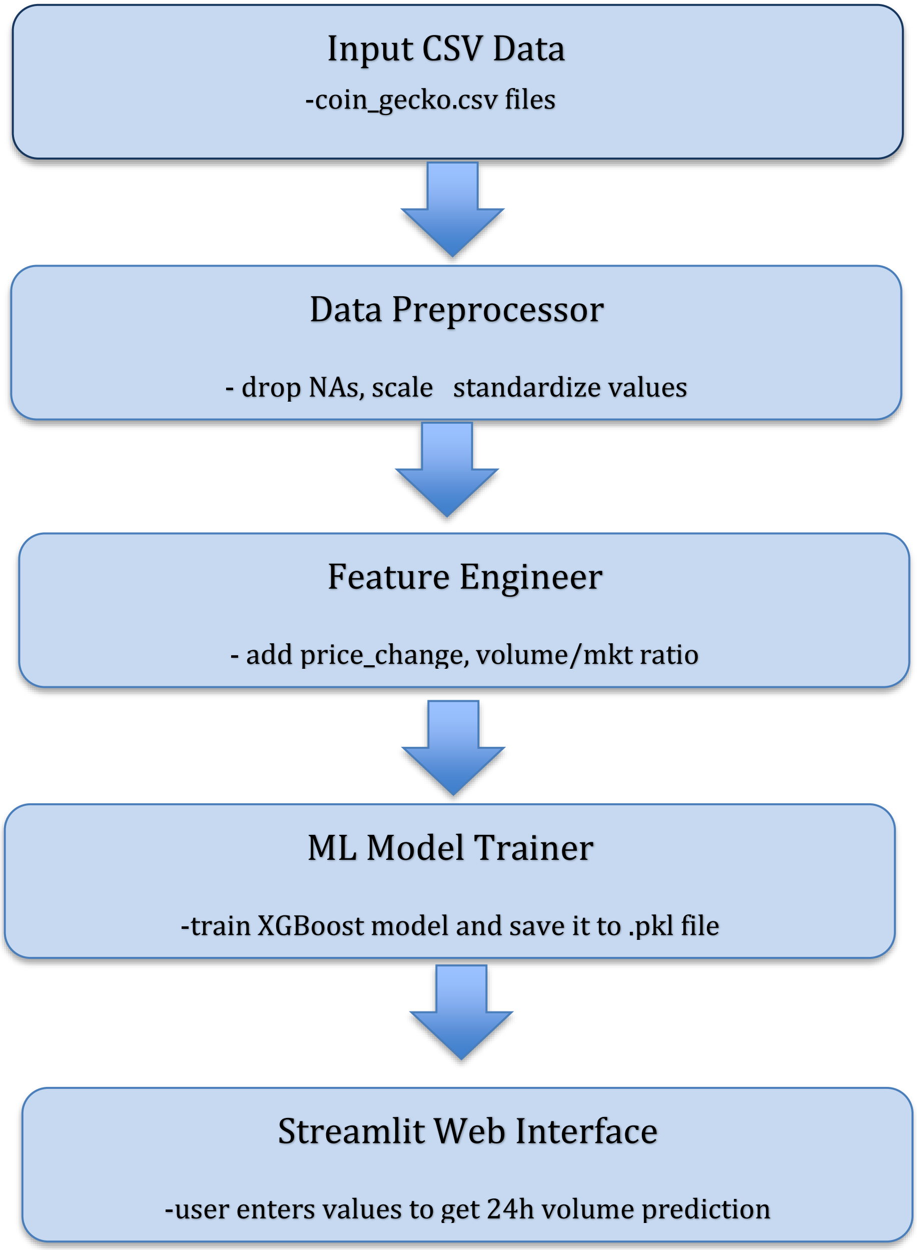
- The goal of this system is to predict the 24-hour trading volume (liquidity) of a cryptocurrency based on market features like price, market cap, and short-term price changes.
- This helps traders, analysts, and platforms identify liquidity risks in advance.

2. System Overview

This is a machine learning-based regression system. It:

- Reads real cryptocurrency market data from CSV files
- Cleans and preprocesses the data
- Extracts meaningful features (like price change score, volume ratios)
- Trains an ML model (XGBoostRegressor)
- Predicts liquidity (24h volume)
- Provides results via a Streamlit web app

3. System Architecture Diagram


Input CSV Data
-coin_gecko.csv files
Data Preprocessor
- drop NAs, scale standardize values
Feature Engineer
- add price_change, volume/mkt ratio
ML Model Trainer
-train XGBoost model and save it to .pkl file
Streamlit Web Interface
-user enters values to get 24h volume prediction

4. Main Components

Component	Description
coin_gecko_2022-03-16.csv, coin_gecko_2022-03-17.csv	Raw market data files (e.g., price, volume, change %)
crypto_liquidity_model1.py	The main training and analysis script: EDA, feature engineering, model training
xgb_model.pkl	Saved trained model (XGBoostRegressor)
volume_scaler.pkl	Scaler to reverse-transform predicted volume (if needed)
app.py	Streamlit frontend to take inputs and show predictions
EDA Report Crypto Liquidity	EDA report with charts like correlation, distributions

5. Data Flow

1. Load historical market data from CSV files
2. Clean & normalize numeric features
3. Engineer new features (price_change_score, volume_to_marketcap)
4. Split data and train model (XGBoost)
5. Save model and scaler
6. Streamlit interface allows real-time user inputs to predict 24h volume

6. Technology Stack

Layer	Tool Used
Programming	Python 3.x
ML Model	XGBoost (via xgboost)
Data Handling	Pandas, NumPy
Visualization	Matplotlib, Seaborn
UI	Streamlit
Deployment	Localhost / .py files

7. Key Features

- Clean preprocessing pipeline
- Lightweight and interpretable XGBoost model
- Visual EDA built-in
- Easy deployment using Streamlit
- Accurate predictions with RMSE ~0.35 and R^2 ~0.92