

Pipeline Architecture for Crypto Liquidity Prediction

Generated by [Avinesh Masih](#) – View on GitHub: [AVINESH MASIH](#)

1. Data Ingestion

- **Source:** CSV ([Dataset](#))
 - **Tools:** pandas, requests, SQLAlchemy
 - **Output:** Raw data frame
-

2. Data Preprocessing

- **Handling missing values**
 - **Outlier detection and removal**
 - **Feature engineering:**
 - volatility = abs(returns)
 - **Scaling** (if required)
 - **Tools:** pandas, numpy, scikit-learn
-

3. Feature Selection

- Select key numerical features:
 - price, market_cap, volume_24h, returns, volatility
 - Optional: correlation analysis or feature importance
-

4. Model Training

- **Model used:** Random Forest Regressor (or XGBoost, Linear Regression)
 - **Training pipeline:**
 - train_test_split
 - model.fit(X_train, y_train)
 - **Tools:** scikit-learn, joblib (for model saving)
-

5. Model Evaluation

- **Metrics:** RMSE, MAE, R^2 score
- **Validation:** Cross-validation (e.g., KFold)

- Save best model using `joblib.dump()`

6. Model Deployment

- **Deployment Framework:** Streamlit
 - **Frontend UI:** User inputs price, volume, market cap, returns
 - **Backend:** Loads model, makes predictions
 - **Bonus:** Add author info, social links, hover effects, and branding
-

Technologies Used

Component	Tool
Data Processing	pandas, numpy
Modeling	scikit-learn, joblib
Deployment	Streamlit
Visualization	Matplotlib / Seaborn (for EDA)

Diagram Overview

