Pipeline Architecture

Step-by-Step Breakdown

1. Data Collection

-Input data is sourced from two CSV files:

- coin_gecko_2022-03-16.csv
- coin_gecko_2022-03-17.csv

-These contain daily market statistics such as:

- Cryptocurrency name
- Price
- 1h, 24h, and 7d price changes
- Market capitalization
- 24h trading volume (target variable)

2. Data Preprocessing

- The data from both files is merged using pandas.
- Null values are removed to ensure data consistency (dropna()).
- Date columns are formatted correctly.
- All numerical columns (price, volume, mkt_cap, etc.) are normalized using StandardScaler to improve model performance and ensure uniform scale.

3. Feature Engineering

Two new derived features are created to enhance model learning:

- **price_change_score**: A weighted score calculated from short-term price changes:

price_change_score=0.2×1h+0.3×24h+0.5×7d\text{price_change_score} = 0.2 \times \text{1h} + 0.3 \times \text{24h} + 0.5 \times \text{7d}price_change_score=0.2×1h+0.3×24h+0.5×7d

- **volume_to_marketcap**: Measures liquidity by comparing 24h volume with market cap.

These features provide better insights into market movement and liquidity patterns.

4. Train/Test Split

- The preprocessed dataset is split using train_test_split() into:
  - 80% for training the model
  - 20% for testing and evaluating performance

5. Model Training

- The XGBoost regression model is selected (XGBRegressor) for its ability to handle complex, tabular data effectively.
- A GridSearchCV is used to tune model parameters (max_depth, n_estimators, learning_rate).
- Features used:
  - Scaled price
  - price_change_score
  - volume_to_marketcap
  - Scaled market cap
- Target: Scaled 24h_volume

6. Model Evaluation

After training, we check how well the model works:

- R^2 Score ≈ 0.92 → model explains 92% of liquidity variation
- RMSE and MAE are low → errors are small

This means the model is highly accurate.

7. Model Persistence

- The trained model is saved using pickle as xgb_model.pkl.
- The target scaler is also saved as volume_scaler.pkl to allow reverse transformation during prediction.

8. Deployment via Streamlit

- A simple web interface is created using Streamlit (app.py).
- Users provide four input values:
  - Price
  - Price Change Score
  - Volume-to-Market Cap Ratio
  - Market Cap
- The saved model (xgb_model.pkl) is loaded, and the model predicts the 24-hour trading volume.
- Results are shown immediately in the web interface.

Final Pipeline Flow