

HLD & LLD Documentation

High-Level Design (HLD)

Project Title:

Cryptocurrency Trend Analysis Using Machine Learning

Objective:

To analyze and predict trends in cryptocurrency prices using machine learning by combining and preprocessing data from different dates.

System Architecture:

1. Data Collection Module
2. Data Preprocessing Module
3. Exploratory Data Analysis (EDA)
4. Feature Engineering
5. Model Training & Evaluation
6. Prediction / Inference
7. GUI Deployment by **gradio** (locally)

Technology Stack:

- Language: Python
- Libraries:
 - pandas, numpy – Data handling
 - matplotlib, seaborn – Visualization
 - sklearn – Machine Learning
- Platform: Jupyter Notebook
- Data Source: CoinGecko CSVs (via GitHub)

Input/Output:

Input: CSV files from CoinGecko (March 16 & 17, 2022)

Output:

- Cleaned dataset
- Visualizations and trend insights
- Predictive model outputs (if implemented)

Assumptions:

- Datasets are similar in structure.
- Market trends are consistent over short periods (daily).

Risks/Constraints:

- Volatility of cryptocurrency data
 - Incomplete or inconsistent data
 - Limited data points (just 2 days)
-

Low-Level Design (LLD)

1. Data Collection:

```
df_16 = pd.read_csv("coin_gecko_2022-03-16.csv")
df_17 = pd.read_csv("coin_gecko_2022-03-17.csv")
df = pd.concat([df_16, df_17], ignore_index=True)
```

2. Data Preprocessing:

- Handling missing values (e.g., `df.dropna()` or `fillna()`)
- Removing duplicates
- Type conversion (e.g., converting price to float)
- Renaming columns for clarity

3. Exploratory Data Analysis (EDA):

- Distribution plots (e.g., `sns.histplot`)
- Correlation matrix (`df.corr()`)
- Top gainers/losers by % change

4. Feature Engineering:

- `price_change_percentage_24h`
- `market_cap_rank`
- `total_volume`

- Convert date/time fields if present

5. Model Training:

- Linear Regression, RandomForest, or DecisionTree
- Train/test split via train_test_split

6. Evaluation Metrics:

- Regression: MAE, MSE, RMSE, R^2
- Classification (if any): Accuracy, Precision, Recall

7. Hyperparameter Tuning:

- Process of selecting the best configuration settings for a machine learning model to improve its performance.
- In this project, after selecting a model adjusted hyperparameters

8. Model Testing & Validation:

- Validate its performance using **testing techniques** to ensure it generalizes well.
- Check how well the model performs on unseen data.

9. Model Testing & Validation:

- GUI Deployment by gradio framework (locally), by simple forms interface.