**Final Report Document**

**Project Title: Cryptocurrency Liquidity Ratio Prediction**

**1. Objective**

**To develop a RESTful API using Fast API that predicts the liquidity ratio of a cryptocurrency based on input features like 24h change, 7d change, 24h volume, and market cap.**

**2. Pipeline Architecture**

**Embed the architecture diagram here and explain:**

* **API flow**
* **Input schema**
* **Model loading and prediction**
* **Error handling**

**3. Technology Stack**

| **Component** | **Tech** |
| --- | --- |
| **Language** | **Python 3.10** |
| **Framework** | **FastAPI** |
| **ML Model** | **scikit-learn / Pickle** |
| **Data Handling** | **Pandas, NumPy** |
| **Deployment** | **Uvicorn** |
| **Environment** | **Virtual env (flask)** |
| **4. Directory Structure Overview**  **FLASKPROJECT3/**  **├── artifacts/**  **│ ├── trained\_model.pkl**  **│ ├── processed\_full.csv**  **│ └── ...**  **├── flask/**  **├── log/**  **├── notebook/**  **├── src/**  **├── app.py**  **├── criptos.py**  **├── requirements.txt**  **├── template.py** |  |
| **5. Input/Output Example**  **{**  **"24h": 1.2,**  **"7d": 1,**  **"24h\_volume": 2.34,**  **"mkt\_cap": 1**  **}** |  |
| **Output:**  **{**  **"predicted\_liquidity\_ratio": 0.0010132737664297676**  **}** |  |
|  |  |
| **6. Testing and Validation**   * **Unit tested using sample inputs** * **Model performance evaluated on test set**   **7. Error Handling**  **Handled using:**   * **HTTP Exception** * **Custom exception and logging modules** |  |
| **8. Deployment**  **Run using:**  **uvicorn app:app --reload** |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |