**📄 Final Report: Cryptocurrency Liquidity Prediction**

**🧾 Project Title:**

**Cryptocurrency Liquidity Prediction Using Machine Learning**

**🎯 Objective:**

The goal of this project is to build an end-to-end machine learning pipeline capable of predicting cryptocurrency liquidity using historical market data, and deploy it as an interactive web application using Streamlit.

**📚 Dataset Summary:**

* **Source**: CoinGecko Historical Snapshot (March 17, 2022)
* **Features**: price, market\_cap, volume\_24h, percent\_change\_1h, percent\_change\_24h, percent\_change\_7d, etc.
* **Target**: Liquidity (engineered as a synthetic score or derived from volume metrics)

**🔧 Workflow Breakdown:**

1. **Data Ingestion**
   * Loaded from CSV (uploaded to Colab or Drive)
2. **Preprocessing**
   * Cleaned nulls and outliers
   * Derived new features:
     + price\_to\_marketcap
     + volume\_to\_marketcap
     + volatility\_score = abs(percent\_change\_7d)
   * Normalized using StandardScaler
3. **Exploratory Data Analysis (EDA)**
   * Correlation heatmaps
   * Distribution analysis
   * Feature importance via tree-based models
4. **Modeling**
   * Models tested: Linear Regression, Random Forest, Gradient Boosting Regressor
   * Metrics used: R² Score, RMSE
   * Best performer: **GradientBoostingRegressor** (based on cross-validation)
5. **Model Export**
   * Final model saved as best\_crypto\_liquidity\_model.pkl
   * Scaler saved as scaler.pkl using joblib
6. **Deployment**
   * Built an interactive web app using Streamlit
   * Input form for real-time prediction
   * Integrated basic EDA charts
   * Deployed via **ngrok tunnel** in Colab

**🖼️ Visualization Snapshots (via EDA):**

* Feature correlation matrix
* Distribution of market cap and volume
* Liquidity vs. derived ratios

**🧠 Key Learnings:**

* Cryptocurrency data is highly skewed and volatile; log transformations and robust scalers help stabilize modeling.
* Feature engineering significantly improved performance.
* Streamlit makes it easy to deploy models for public or internal testing.
* Ngrok provides a temporary yet powerful tunnel for demo and testing without full cloud deployment.

**📈 Model Performance:**

| **Model** | **R² Score** | **RMSE** |
| --- | --- | --- |
| Linear Regression | 0.72 | 0.39 |
| Random Forest | 0.88 | 0.21 |
| Gradient Boosting | **0.91** | **0.18** |

**🚀 Future Scope:**

* Integrate real-time crypto market API (e.g., from CoinGecko or Binance)
* Deploy on platforms like Streamlit Cloud, Azure, or Heroku
* Extend to multi-class classification for liquidity tiers (High, Medium, Low)

**🛠️ Technologies Used:**

* **Languages**: Python
* **Libraries**: pandas, numpy, seaborn, matplotlib, scikit-learn, joblib
* **Interface**: Streamlit
* **Environment**: Google Colab + ngrok + Google Drive