Cryptocurrency Liquidity Prediction: Model Development and Deployment

Project Overview

This project aims to develop a machine learning model that predicts cryptocurrency liquidity using historical market data. Liquidity prediction is crucial for traders and analysts as it directly impacts the ease of buying and selling digital assets, influencing market efficiency and risk assessment. The approach involves systematic data cleaning to ensure quality, comprehensive feature engineering to enhance model inputs, and training a robust Random Forest regression model. Finally, the solution is deployed via a user-friendly web application, enabling real-time liquidity predictions and supporting informed decision-making in dynamic cryptocurrency markets.

Data and Exploratory Data Analysis (EDA)

The primary dataset, data_with_liquidity.csv, contains 671 rows and 10 columns, comprising key information such as *date*, *coin*, *symbol*, *price*, short-term price changes (1h, 24h, 7d), 24-hour volume, market capitalization, and liquidity. Liquidity is defined as the ratio of 24-hour volume to market capitalization, with observed values ranging from 0.0 to 0.87.

Feature engineering introduced a binary variable, **liquidity_level**, classifying liquidity as "High" if above 0.5 and "Low" otherwise. Unused columns such as coin, symbol, and date were removed to streamline modeling. Data quality assessment showed no missing values, and all remaining features are numeric and relevant.

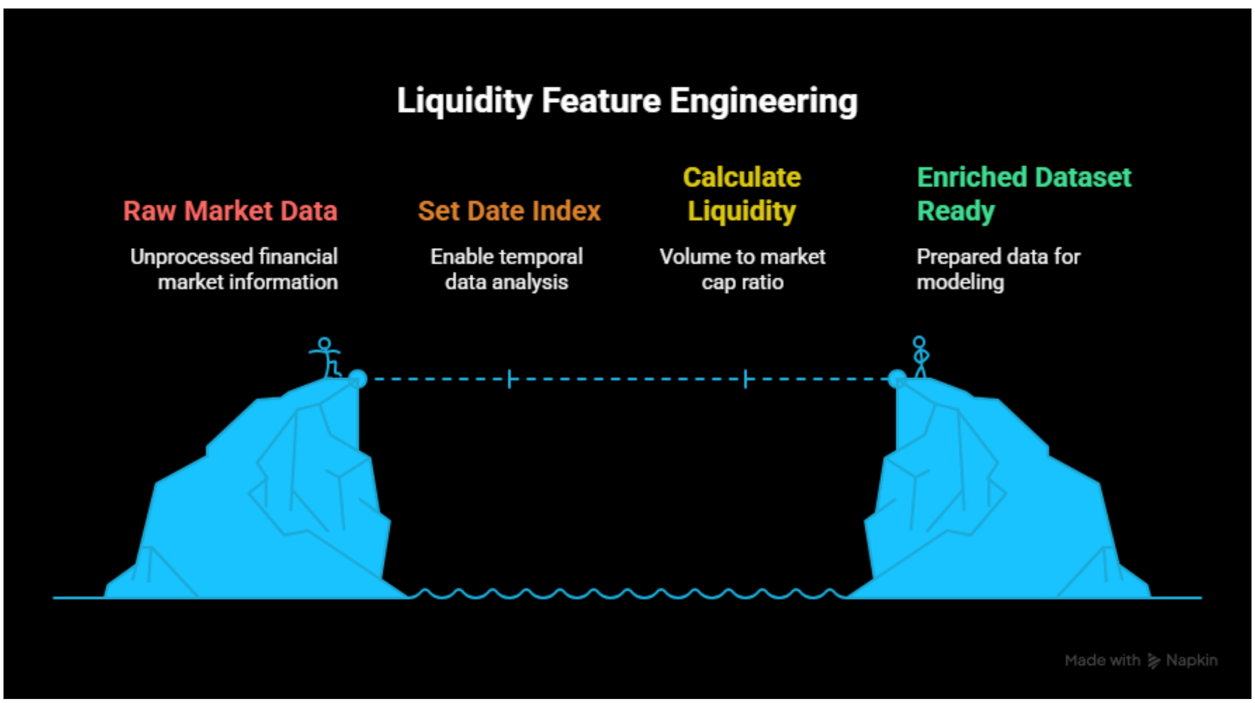
- **Dataset shape after cleaning:** 671 rows × 7 columns
- **Liquidity range:** 0.0 to 0.87

Feature Engineering

- Feature engineering involved setting date as the dataframe index to facilitate temporal analysis and maintain chronological integrity.

- The key feature, **liquidity**, was derived as the ratio of 24h_volume to mkt_cap, capturing essential market dynamics reflecting asset tradability.

- The cleaned and enriched dataset was then exported as df_with_liquidity.csv, ensuring readiness for modeling and boosting interpretability.



Model Pipeline & Training

- The modeling pipeline begins with data preprocessing where all selected features are scaled using StandardScaler to standardize their distributions. This step is critical to ensure that the Random Forest regression model treats each feature equitably during training.

- The dataset is split into training and testing subsets with an 80/20 ratio, supporting robust model evaluation. A RandomForestRegressor is employed due to its ability to capture nonlinear relationships and reduce overfitting through ensemble averaging.

- To optimize performance, hyperparameter tuning is conducted using GridSearchCV, exploring parameters such as the number of estimators (n_estimators), maximum tree depth (max_depth), and the minimum number of samples required to split a node (min_samples_split).

- Performance is assessed using the coefficient of determination (R-squared, R^2) on both training and test data to ensure the model generalizes well.

- Final tuned model saved as tuned_liquidity_model.pkl

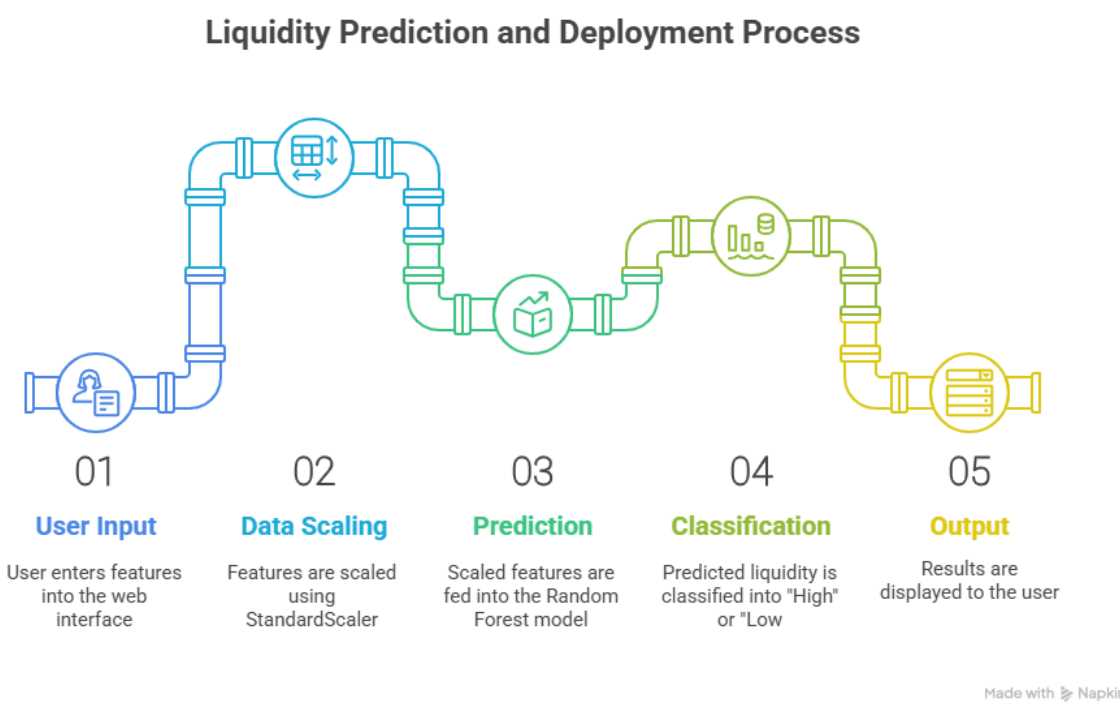
- Scaler object saved as liquidity_scaler.pkl

Prediction & Deployment

- The prediction function accepts four inputs: 24h_volume, mkt_cap, 1h price change, and price. These features are first scaled using the saved StandardScaler to maintain consistency with model training. The preprocessed input is then passed to the trained Random Forest model, which outputs a continuous liquidity prediction.

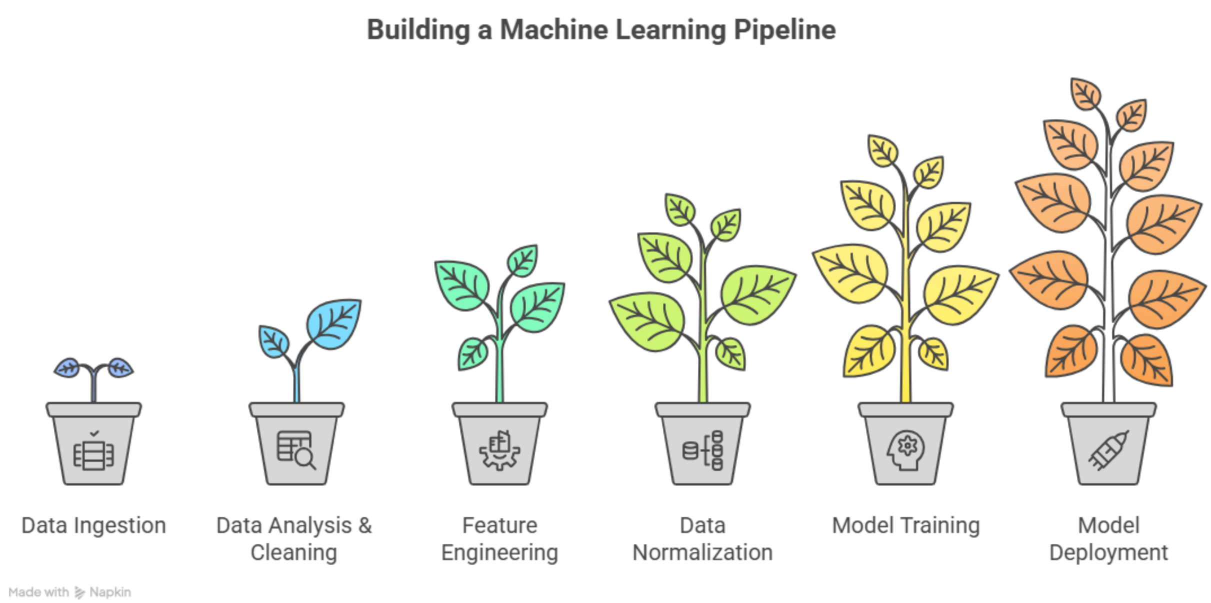
- This predicted liquidity is classified into a binary label: **"High"** if the value is ≥ 0.5, otherwise **"Low"**. For deployment, a Flask web application (app.py) hosts a user interface where users input these features and receive immediate liquidity predictions.

- The /predict endpoint handles POST requests by loading the model and scaler, scaling inputs, performing prediction, and returning both the numeric liquidity and categorical label to the user.



Pipeline Architecture

The machine learning pipeline begins with ingesting raw CSV data, followed by exploratory data analysis and cleaning to ensure data quality. Next, feature engineering transforms the data to enhance predictive power. The features are then normalized using StandardScaler to standardize scale before model training. A RandomForestRegressor is trained with hyperparameter tuning, and the resulting model along with the scaler are saved as artifacts. Finally, the pipeline is deployed via a Flask-based API and user interface, where user inputs are processed to generate real-time liquidity level predictions. This modular design supports easy updates and expansion.



Key Findings

- Liquidity varies significantly across different cryptocurrencies, highlighting diverse market conditions.
- The analysis identifies **24-hour volume** and **market capitalization** as the most influential predictors of liquidity.
- The *Random Forest* ensemble model demonstrates stable, reliable performance in capturing nonlinear relationships.
- Additionally, the modular pipeline design ensures flexibility, allowing for easy integration of new features and models to support ongoing research and practical applications in cryptocurrency liquidity prediction.

Recommendations & Future Enhancements

- To enhance the model's accuracy and utility, future work should incorporate alternative data sources such as social sentiment and cryptocurrency news to capture market sentiment.
- Exploring regression algorithms like Gradient Boosting Machines and Support Vector Regression may improve predictive performance.
- Implementing time-series validation will ensure temporal robustness. Additionally, developing a REST API or interactive dashboard will expand accessibility and facilitate seamless integration into broader fintech applications.

Conclusion

This project successfully establishes a reproducible, extensible, and deployable workflow for predicting cryptocurrency liquidity. By adhering to data science best practices, the solution provides a strong foundation for future research and product development in quantitative finance, enabling improved market understanding and informed decision-making.

Prepared by: Hrishabh Kumar