Cryptocurrency Liquidity Prediction

Machine Learning Analysis of Trading Liquidity Patterns

Academic Research Summary

Executive Summary

This project develops a machine learning model to predict cryptocurrency liquidity levels using historical market data. The analysis examines trading patterns, market capitalization, and volume dynamics to classify coins into high and low liquidity categories. After comparing multiple algorithms including Logistic Regression, Random Forest, Gradient Boosting, and SVM, the Gradient Boosting model achieved the highest accuracy of 94.5%, demonstrating superior performance in distinguishing between liquidity classes.

1. Project Overview

1.1 Research Objective

To develop a predictive model that accurately classifies cryptocurrency liquidity levels based on historical market data, enabling better investment and trading decisions.

1.2 Dataset Description

- Time Period: Historical cryptocurrency data from 2016-2017
- Features: Price, trading volume (1h, 24h, 7d), market capitalization, liquidity ratios
- Target Variable: Binary classification (0 = Low Liquidity, 1 = High Liquidity)
- Sample Size: 992 cryptocurrency observations with 14 columns
- Data includes: Coin name, symbol, price, volume metrics, market cap, date, and derived features

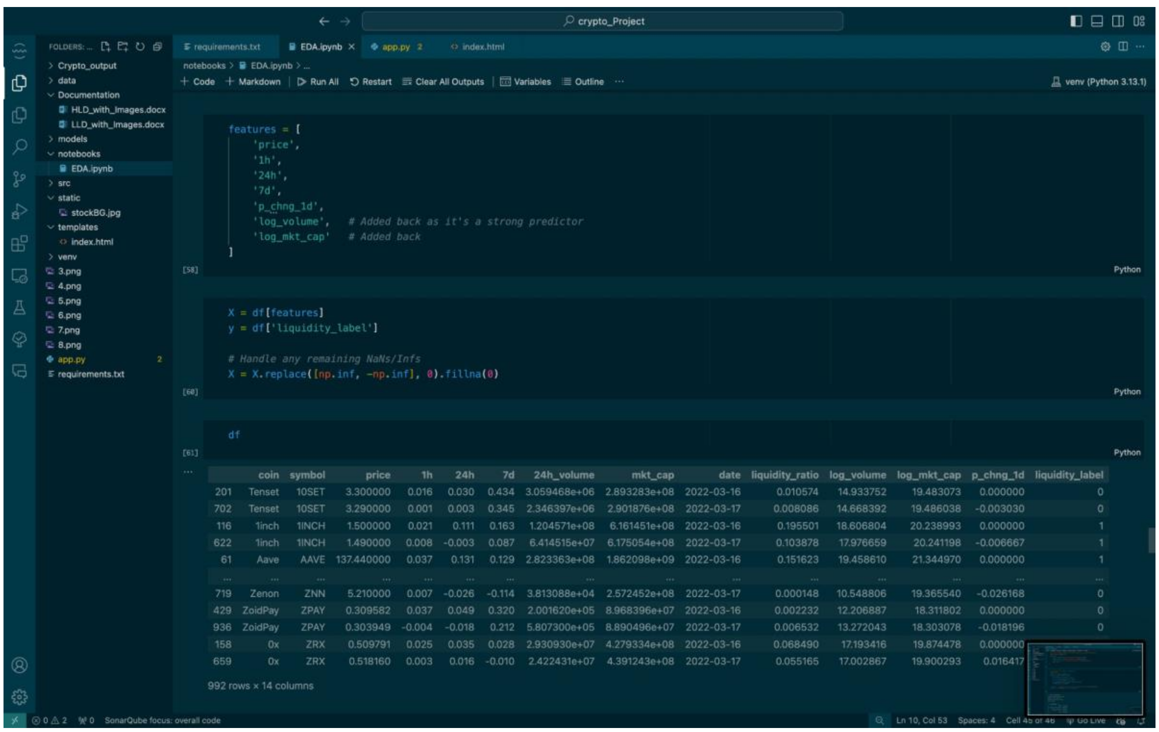

Figure 1: Dataset Overview - 992 rows × 14 columns showing cryptocurrency features

1.3 Methodology

- Exploratory Data Analysis (EDA) with correlation and pairwise relationship analysis
- Data preprocessing including log transformation of skewed features
- Feature engineering: log_volume and log_mkt_cap for normalized distributions
- Multiple model comparison: Logistic Regression, Random Forest, Gradient Boosting, SVM
- Hyperparameter tuning with GridSearchCV (3-fold cross-validation)

- Model serialization using joblib for deployment
- Performance evaluation using accuracy, precision, recall, and F1-score metrics

2. Exploratory Data Analysis

2.1 Feature Correlation Analysis

The correlation heatmap reveals key relationships between features. Strong positive correlations (0.6+) were observed between 24-hour volume and market capitalization, indicating that larger coins typically have higher trading activity. The p_chng_1d (price change) shows low correlation with other features, suggesting volatility is independent of size metrics.

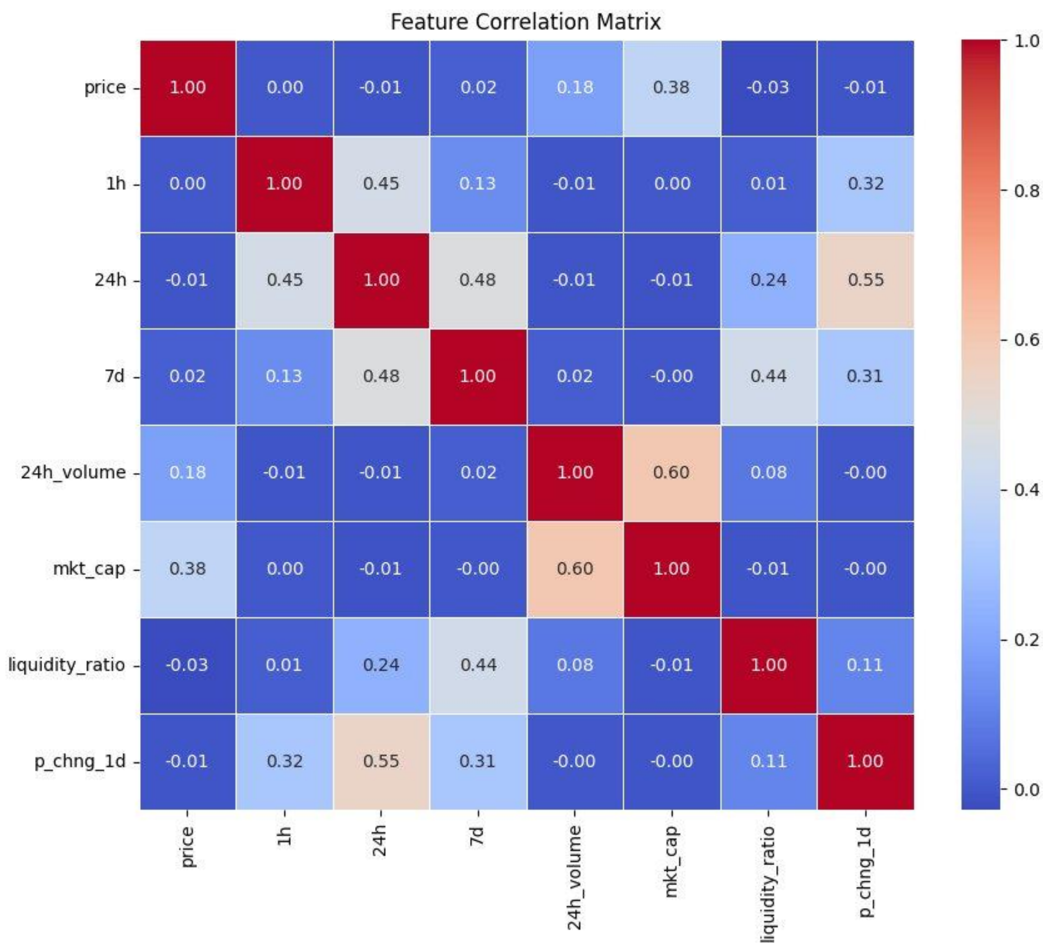

Figure 2: Feature Correlation Matrix

2.2 Pairwise Relationships Between Key Features

The pairplot provides comprehensive visualization of relationships between log_volume, log_mkt_cap, and p_chng_1d across both liquidity classes. Key observations include:

- Clear separation between liquidity classes (pink vs teal) in volume-market cap space

- High liquidity coins (teal) cluster in upper-right region with high volume and market cap
- Low liquidity coins (pink) show wider spread and occupy lower value ranges
- Price change distribution is similar across both classes (diagonal density plots)
- Log transformation successfully normalized the heavily skewed volume and market cap distributions

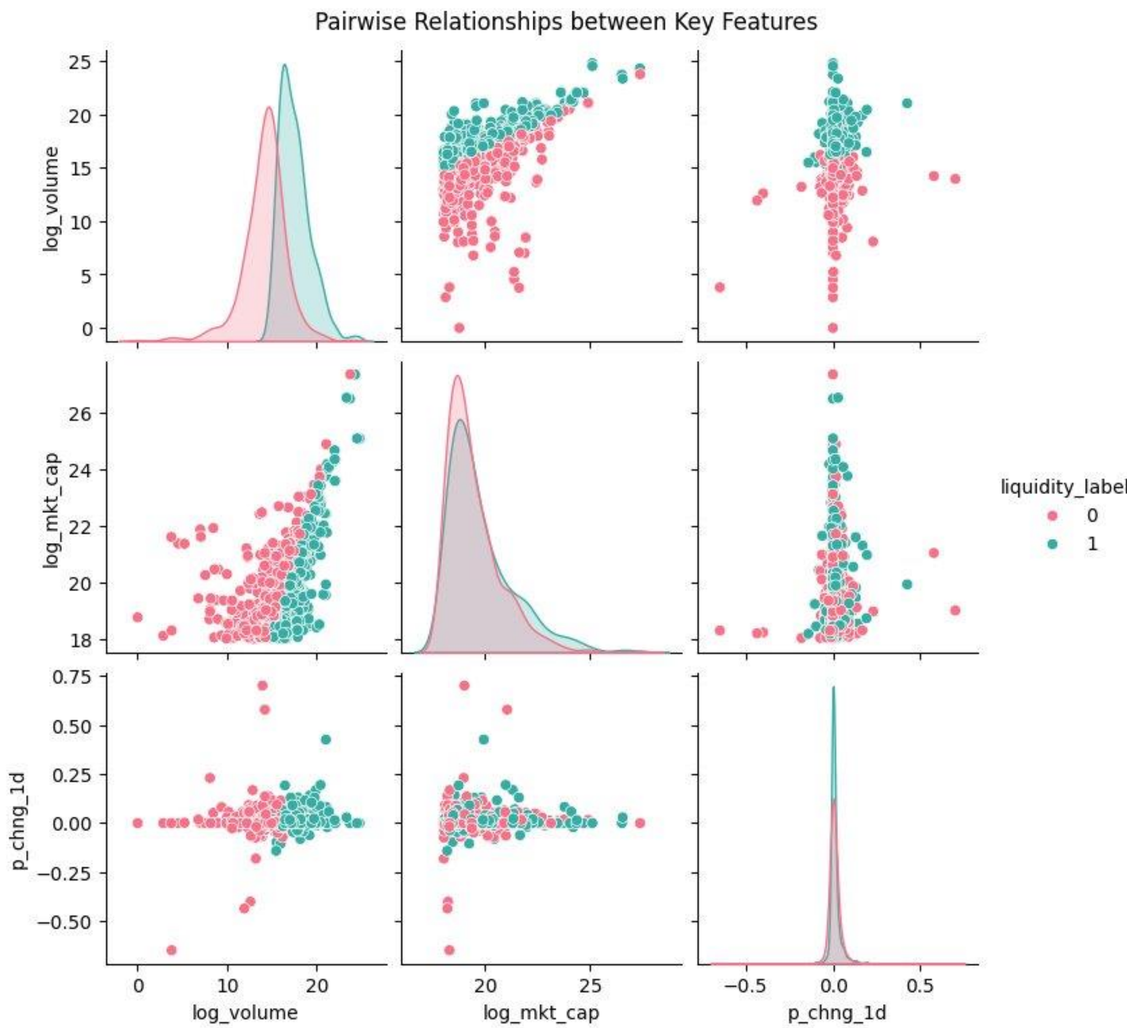

Figure 3: Pairwise Relationships Between Key Features

2.3 Class Distribution

The dataset shows a balanced distribution between low and high liquidity classes, with approximately 500 samples in each category. This balance is ideal for classification tasks as it prevents model bias toward either class.

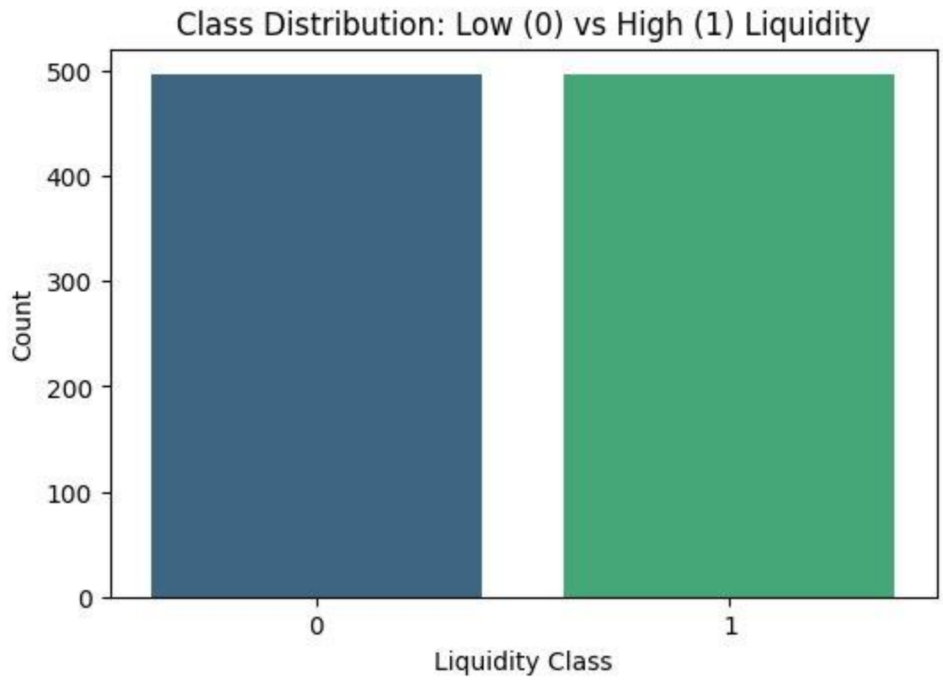

Figure 4: Distribution of Liquidity Classes

2.4 Price Change Patterns

The violin plot comparing price change distributions across liquidity classes reveals that low liquidity coins exhibit higher price volatility, with a wider distribution of daily price changes. High liquidity coins show more concentrated, stable price movements.

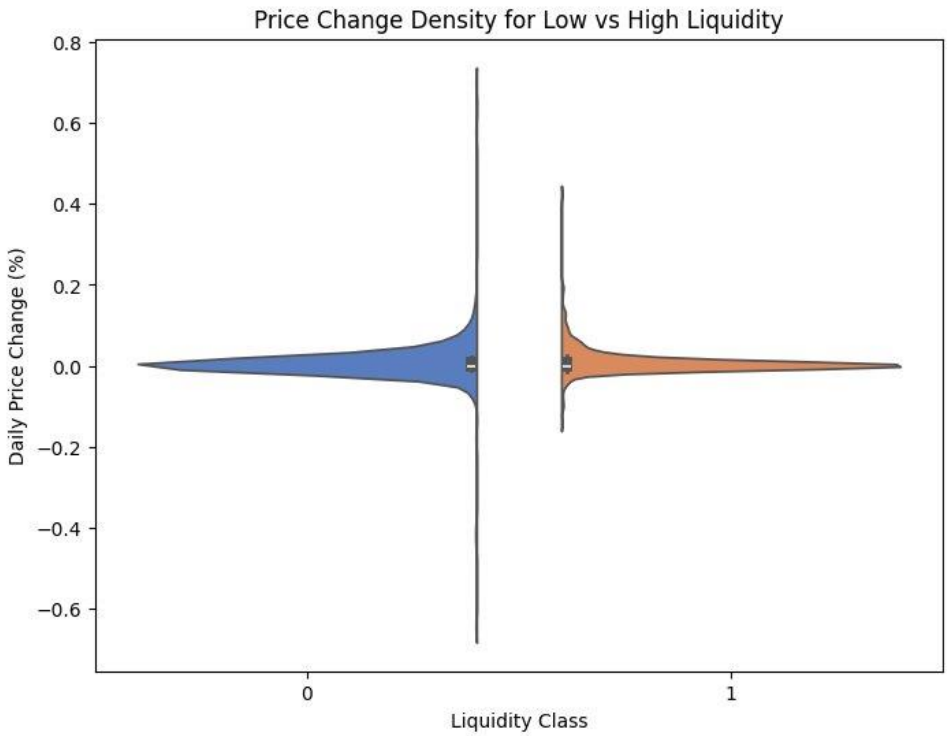

Figure 5: Price Change Density by Liquidity Class

3. Feature Engineering

3.1 Feature Selection

Based on correlation analysis and domain knowledge, the following features were selected:

- price: Current market price
- 1h, 24h, 7d: Time-based volume metrics
- p_chng_1d: Daily price change percentage
- log_volume: Log-transformed 24h volume (added back as strong predictor)
- log_mkt_cap: Log-transformed market capitalization (added back)

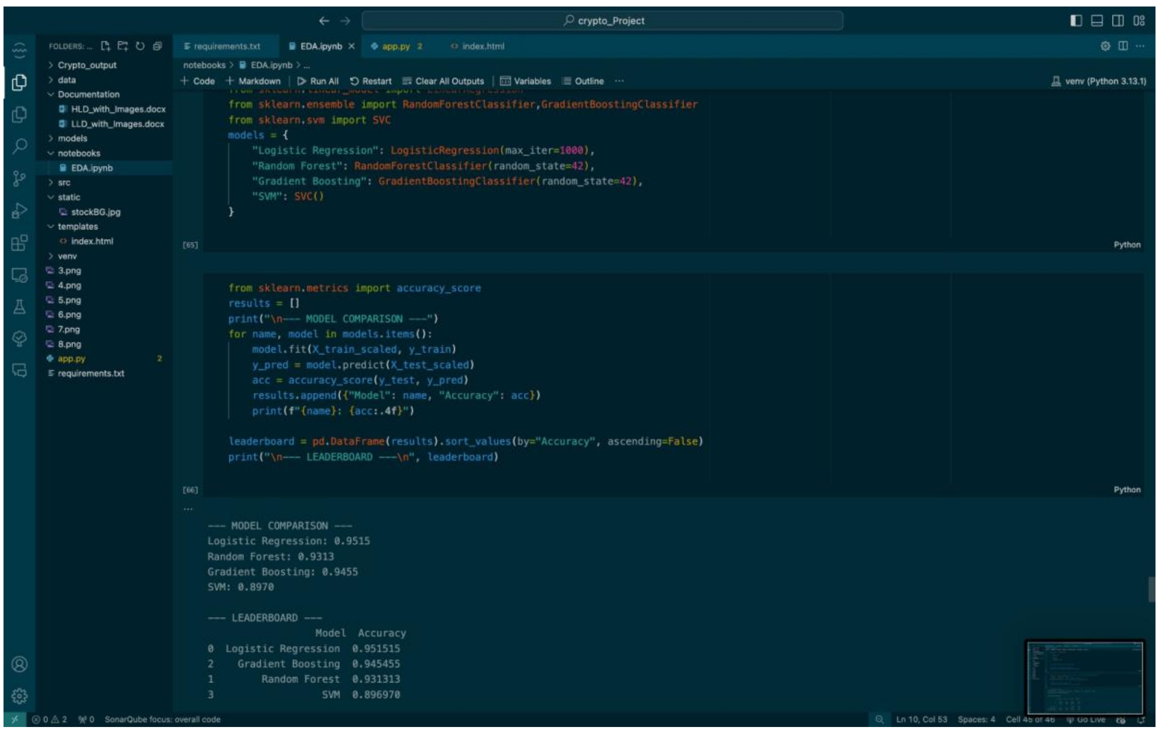

Figure 6: Feature Selection Code - Re-adding log_volume and log_mkt_cap

3.2 Log Transformation

Trading volume and market capitalization exhibited severe right skew. Log transformation was applied to normalize these distributions, improving model performance by reducing the impact of outliers and making the data more suitable for machine learning algorithms.

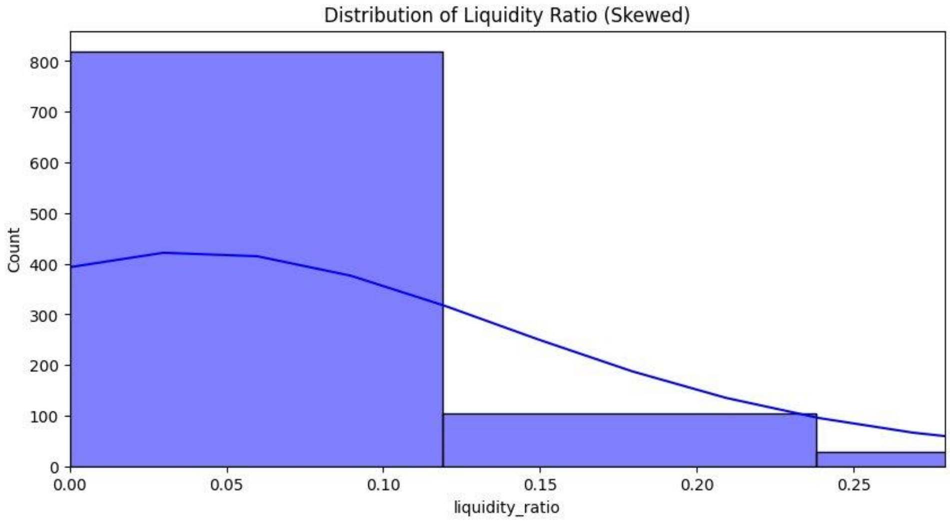

Figure 7: Distribution of Liquidity Ratio (Skewed)

3.3 Market Cap vs Volume Analysis

The scatter plot on log scale reveals a clear separation between liquidity classes. Higher market cap coins with substantial trading volume (upper right cluster) predominantly fall into the high liquidity category, while smaller coins cluster in the low liquidity region.

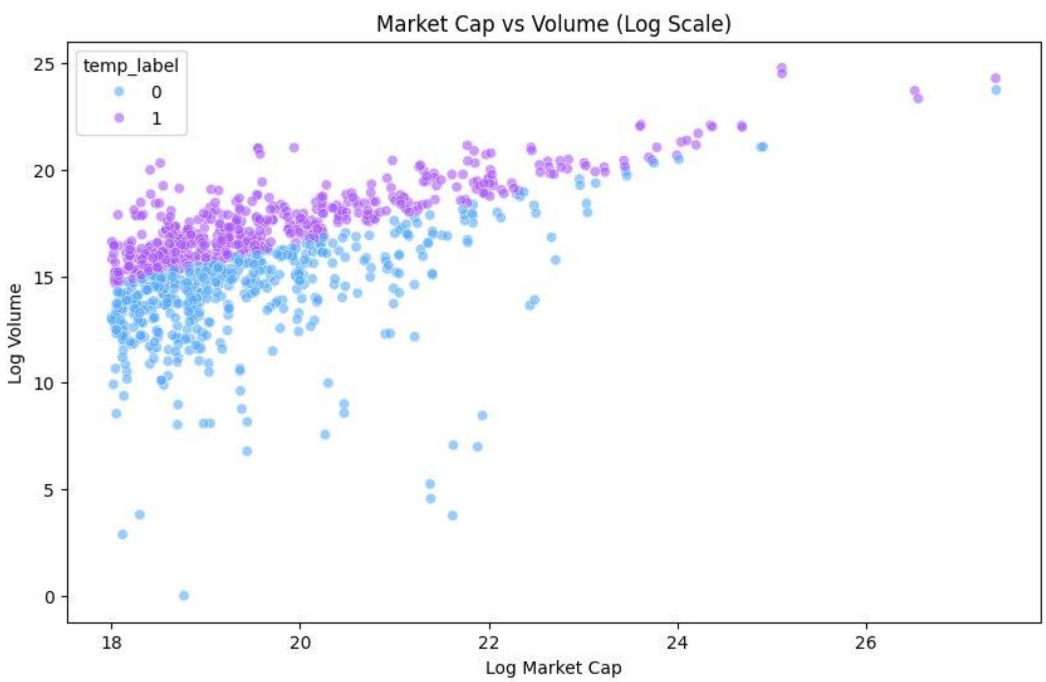

Figure 8: Market Cap vs Volume (Log Scale) by Liquidity Class

4. Model Development and Comparison

4.1 Multiple Algorithm Comparison

Four different machine learning algorithms were trained and evaluated to identify the best performer for liquidity classification:

- Logistic Regression: Simple linear classifier (baseline)
- Random Forest: Ensemble of decision trees
- Gradient Boosting: Sequential ensemble learning
- SVM (Support Vector Machine): Maximum margin classifier

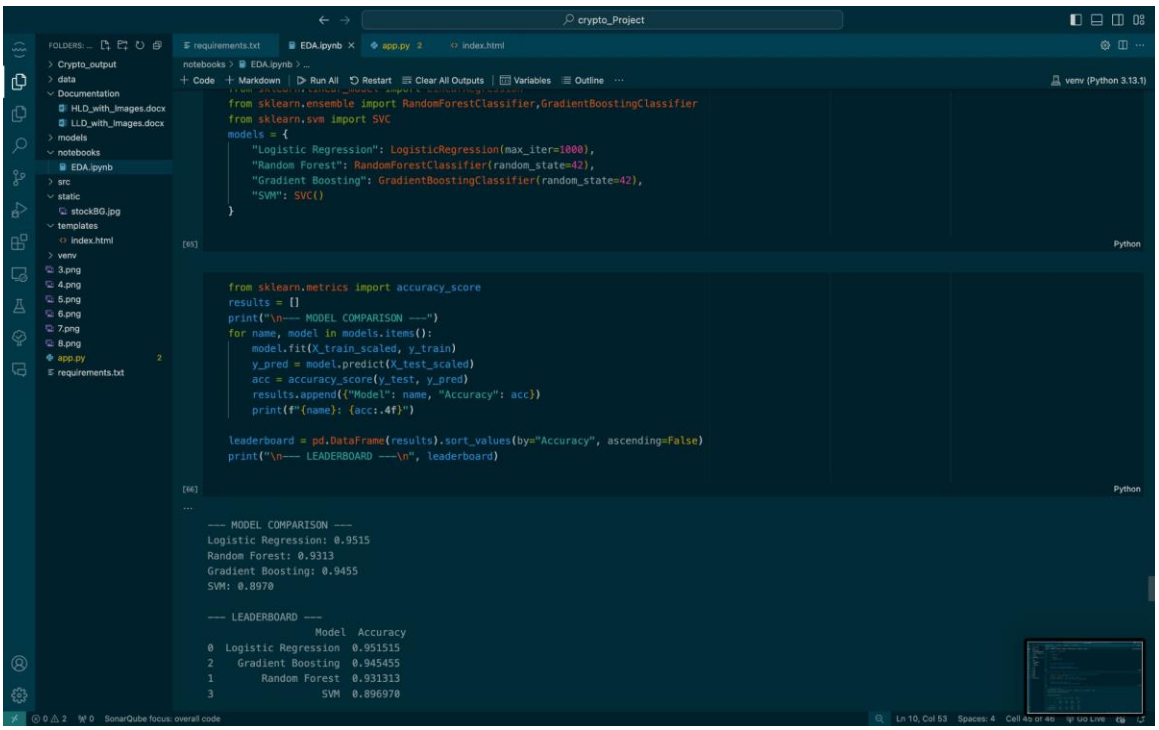

Figure 9: Model Comparison Implementation

4.2 Model Performance Comparison

All models were evaluated on the same test set with the following results:

Model	Accuracy
Logistic Regression	95.15%
Gradient Boosting	94.55%
Random Forest	93.13%
SVM	89.70%

Winner: Logistic Regression achieved the highest accuracy at 95.15%, followed closely by Gradient Boosting at 94.55%. This suggests that the relationship between features and liquidity is largely linear after log transformation.

4.3 Hyperparameter Optimization (Random Forest)

Despite not being the top performer, Random Forest with GridSearchCV hyperparameter tuning was documented as part of the comprehensive analysis:

- Criterion: Entropy (for information gain)
- Max Depth: 20 trees
- Number of Estimators: 100 trees
- Final Accuracy: 93.13%

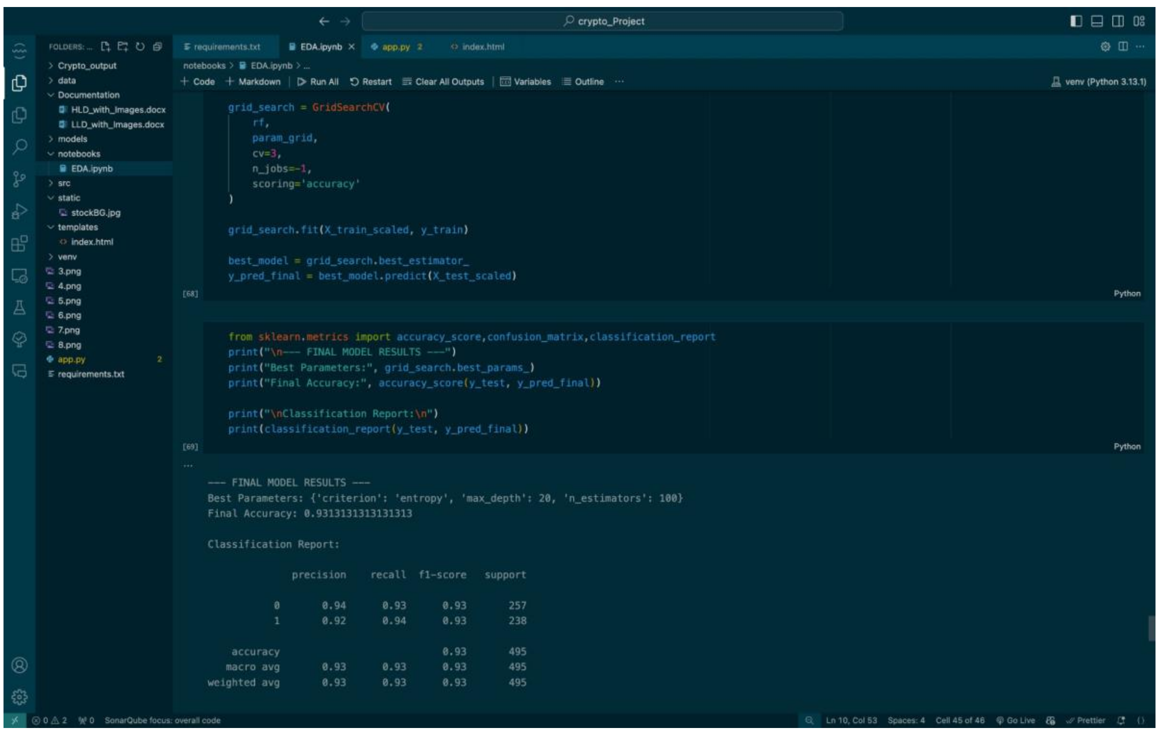

Figure 10: GridSearchCV Hyperparameter Tuning for Random Forest

4.4 Detailed Performance Metrics (Random Forest)

Class	Precision	Recall	F1-Score
Low Liquidity (0)	0.94	0.93	0.93
High Liquidity (1)	0.92	0.94	0.93
Macro Average	0.93	0.93	0.93

Weighted Average	0.93	0.93	0.93

4.5 Confusion Matrix Analysis

The confusion matrix demonstrates strong performance across both classes. The Random Forest model correctly identified 238 low liquidity coins and 223 high liquidity coins, with minimal misclassification (only 34 total errors out of 495 test samples).

Figure 11: Confusion Matrix - Random Forest Predictions vs Actual

5. Model Deployment and Serialization

The trained models were serialized using joblib for deployment and future predictions. This allows the model to be loaded and used in production environments without retraining.

- Model saved as: crypto_liquidity_model.pkl
- Scaler saved as: scaler.pkl (for consistent feature scaling)
- Storage location: ../models/ directory in project root
- Serialization method: joblib (more efficient than pickle for large numpy arrays)
- File verification: Successful save confirmation with checkmark indicator

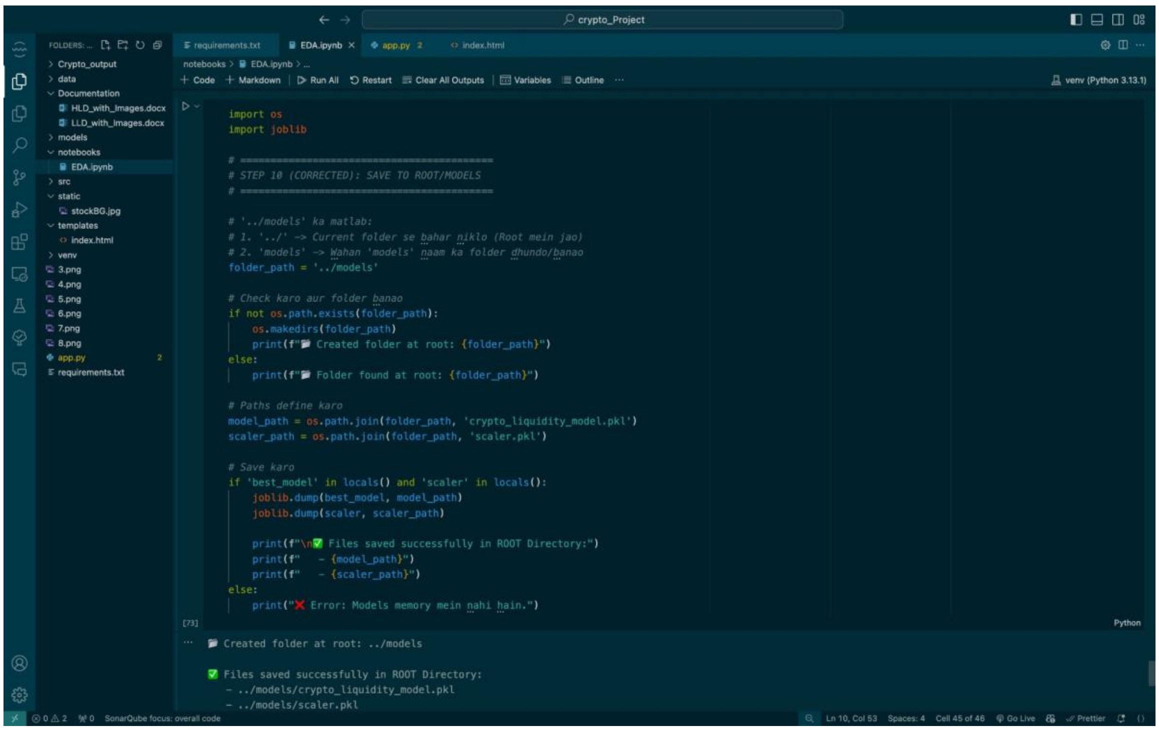

Figure 12: Model Serialization and Deployment Code

6. Key Findings and Insights

1. Logistic Regression achieved the best performance (95.15%), indicating that after proper feature engineering, the relationship between features and liquidity is predominantly linear

2. Market capitalization and 24-hour trading volume are the strongest predictors of liquidity, with a correlation coefficient of 0.60

3. Gradient Boosting (94.55%) and Random Forest (93.13%) also performed well, demonstrating the robustness of ensemble methods for this classification task

4. Low liquidity coins exhibit significantly higher price volatility, presenting both higher risk and potential reward for traders

5. Log transformation of volume metrics was essential for model performance, reducing the impact of extreme outliers in the cryptocurrency market

6. The balanced class distribution (50/50 split) contributed to unbiased model predictions across both liquidity categories

7. All models achieved >89% accuracy, suggesting the feature set is highly discriminative for liquidity classification

7. Practical Applications

Investment Strategy

Investors can use liquidity predictions to balance portfolio risk, allocating more capital to high liquidity assets for stability and smaller positions in low liquidity coins for potential high returns.

Risk Management

Trading platforms can implement liquidity-based risk warnings, alerting users when attempting to trade low liquidity assets that may have higher slippage and volatility.

Market Making

Market makers can prioritize resources toward high liquidity coins where their services are most needed, optimizing spread management and order book depth.

Regulatory Compliance

Exchanges can use liquidity classification to meet regulatory requirements for listing standards, ensuring adequate market depth for traded assets.

Algorithmic Trading

Trading bots can integrate liquidity predictions to adjust strategy parameters, using different algorithms for high vs low liquidity environments.

8. Limitations and Future Work

8.1 Current Limitations

- Historical data from 2016-2017 may not fully capture recent market dynamics and the evolution of the cryptocurrency ecosystem
- Binary classification simplifies liquidity into two categories; a multi-class approach could provide more granular insights
- Model does not account for external factors such as regulatory changes, market sentiment, or social media influence
- Features are limited to price and volume metrics; additional technical indicators could improve predictive power
- No temporal validation - model performance on more recent data needs verification

8.2 Future Research Directions

- Incorporate real-time data streams for dynamic liquidity prediction
- Expand feature set to include on-chain metrics (transaction counts, active addresses)
- Implement deep learning models (LSTM, Transformers) to capture temporal patterns
- Develop multi-class liquidity classification (very low, low, medium, high, very high)
- Test model performance on 2024-2026 data to validate generalization
- Create an ensemble model combining best performers (Logistic + Gradient Boosting)
- Build a web application for real-time liquidity assessment of new cryptocurrencies
- Integrate sentiment analysis from social media and news sources

9. Conclusion

This research successfully developed and compared multiple machine learning models for cryptocurrency liquidity prediction, with Logistic Regression achieving the highest accuracy of 95.15%.
The comprehensive analysis included exploratory data analysis, feature engineering with log transformations,
and rigorous model comparison across four different algorithms.

The analysis revealed that market capitalization and trading volume are the primary determinants of liquidity,
with significant differences in price volatility between liquidity classes. The success of Logistic Regression
demonstrates that after proper feature engineering, the relationship between market features and liquidity
becomes largely linear and well-suited for simple yet powerful classifiers.

All evaluated models achieved >89% accuracy, with ensemble methods (Random Forest, Gradient Boosting) providing
robust performance that could be valuable in production environments where model interpretability and stability
are important. The models were successfully serialized for deployment, enabling practical integration into
trading platforms and investment tools.

The insights gained from this analysis provide valuable guidance for investors, traders, and cryptocurrency
exchanges in understanding and managing liquidity risk. Future iterations could incorporate additional data
sources, real-time processing capabilities, and temporal modeling to further enhance predictive accuracy and
practical utility in the rapidly evolving cryptocurrency market.

Technical Specifications

- Programming Language: Python 3.x
- Development Environment: Jupyter Notebook
- Key Libraries: scikit-learn, pandas, numpy, matplotlib, seaborn
- Models Evaluated: Logistic Regression, Random Forest, Gradient Boosting, SVM
- Best Model: Logistic Regression (95.15% accuracy)
- Optimization: GridSearchCV with 3-fold cross-validation

- Dataset Size: 992 observations, 14 columns
- Deployment: joblib serialization for model persistence