CryptoSpark: Scalable Big Data Analytics Pipeline for Cryptocurrency Trading

---

Team Members: Akshit Tyagi, Chao Zheng, Dongmei Han, Niranjan Rao Saraf Srinivas Rao, Victor Dumaslan

Instructor: Sangjin Lee

Course: DATA 228 – Big Data Technologies And Applications

University: San Jose State University

Date: 5/10/2025

Abstract

In this project, we designed and implemented a scalable big data analytics pipeline focused on cryptocurrency market behavior using the Binance Full History dataset. Our workflow begins with uploading raw Parquet files—containing minute-level candlestick data—to Amazon S3 for distributed and cost-efficient storage. We then perform exploratory data analysis (EDA) on this raw dataset using PySpark, enabling us to uncover trading trends, price movements, and volume patterns at scale. Following EDA, we apply structured preprocessing and feature engineering techniques to enrich the dataset with computed fields such as daily returns, moving averages, volatility indicators, and buy pressure ratios. These enhanced features serve as input to a set of machine learning models, including linear regression, logistic regression, and XGBoost, to model price behavior and detect significant market patterns. This end-to-end pipeline—leveraging cloud storage, distributed computation, and visual analytics—offers a robust framework for deriving insights and predictive intelligence from large-scale cryptocurrency data.

1. Introduction

1.1. The Problem

Cryptocurrency markets are inherently fast-moving, highly volatile, and continuously generating vast volumes of trading data. While this data holds valuable insights, analyzing it at scale poses significant challenges due to its velocity, volume, and variety. Traditional tools often fall short when it comes to managing such complexity efficiently.

1.2 Opportunity

Historical trading data offers a powerful window into market dynamics and investor behavior. With the right computational tools, it is possible to identify trends, detect anomalies, and even build predictive models. However, the scale and structure of this data demand robust, distributed processing systems to transform raw information into actionable insights.

1.3 Our Goal

This project aims to build a scalable data analytics pipeline capable of handling high-frequency cryptocurrency trading data. Using open-source technologies like Apache Spark and cloud platforms like AWS, we explore methods for preprocessing, analyzing, and modeling market data to better understand the behavior of digital assets.

1.4 What We Do

Our end-to-end pipeline begins with uploading Binance's minute-level historical trading data into Amazon S3 for cloud-based storage. We then perform exploratory data analysis (EDA) using PySpark in VSCode, followed by preprocessing and feature engineering. Finally, we apply machine learning models—including linear regression, logistic regression, and XGBoost—to uncover predictive patterns and support informed decision-making in crypto trading environments.

2. Dataset Overview

We utilized the Binance Full History dataset, a large-scale collection of historical trading data from Binance.com. It contains over 35 GB of minute-level candlestick records for the top 1,000 cryptocurrency trading pairs, stored in Parquet format and indexed by time.

Each record includes the following attributes:

Features	**Description**
open_time	Starting time of the 1-minute interval
open	Opening price
high	Highest price during the interval
low	Lowest price during the interval
close	Closing price
volume	Total traded base asset volume
quote_asset_volume	Traded quote asset volume
number_of_trades	Count of executed trades
taker_buy_base_asset_volume	Volume bought by takers (base asset)
taker_buy_quote_asset_volume	Volume bought by takers (quote asset)

The dataset covers multiple years of trading data.

3. Methodology

Our methodology follows a structured pipeline designed to process and analyze large-scale cryptocurrency trading data efficiently. The core components include data ingestion, exploratory analysis, preprocessing, feature engineering, and machine learning.

3.1 Data Ingestion and Storage

We began by uploading the Binance Full History Parquet files to an Amazon S3 bucket using the AWS CLI. This allowed us to store and access the dataset in a scalable, distributed manner. S3's integration with Apache Spark enabled seamless data loading for analysis.

3.2 Exploratory Data Analysis (EDA)

The purpose of this EDA is to reduce the complexity of the dataset, identify a representative trading pair, and explore its price behavior, trading activity, and market sentiment.

3.2.1 Dataset Overview and Preparation

The dataset consists of 1-minute OHLCV trading data for over 1000 cryptocurrency pairs, covering the period from 2017 to 2022. The raw data contains more than 1 billion records, making direct exploration computationally expensive.

To enable scalable analysis, we applied early-stage data reduction by aggregating the raw minute-level data to daily-level summaries using PySpark. For each (symbol, date) combination, we calculated:

- avg(open), avg(close)
- max(high), min(low)
- sum(volume), sum(number_of_trades)
- sum(taker_buy_base_asset_volume)

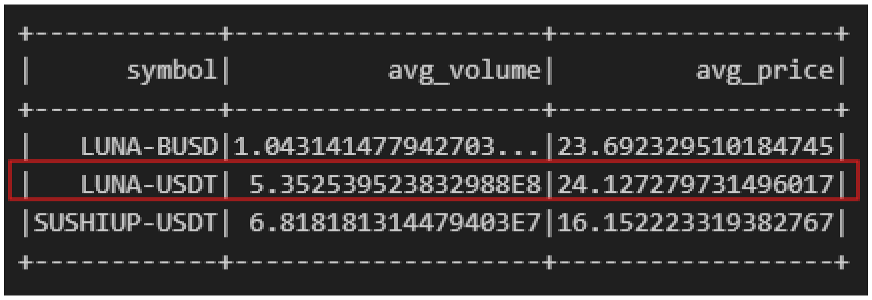
This reduced the dataset to a manageable size while preserving key trading dynamics.

3.2.2 Symbol Selection

We calculated the average daily volume and average closing price for each trading pair and applied the following filters:

- Average volume > 1,000,000
- Average close price > $10

This ensured we focused on active and liquid markets, eliminating low-activity pairs that may not reflect meaningful patterns. Based on these criteria, the top three candidates were shown in Figure. 1:



```
+-----------+-------------------+------------------+
|     symbol|         avg_volume|         avg_price|
+-----------+-------------------+------------------+
|  LUNA-BUSD|1.043141477942703...|23.692329510184745|
|  LUNA-USDT| 5.352539523832988E8|24.127279731496017|
|SUSHIUP-USDT| 6.818181314479403E7|16.152223319382767|
+-----------+-------------------+------------------+
```

Figure 1: Top 3 symbols by average daily volume and average closing price

LUNA-USDT was chosen due to its exceptionally high average daily volume (~535 million), making it a suitable candidate for exploring price trends, volatility, and market signals.

3.2.3 Price Trend and Volatility

We analyzed daily price movement for LUNA-USDT using 7-day and 30-day moving averages. As shown in Figure 2, these revealed strong upward trends through 2021, followed by a sharp crash in mid-2022, after which prices remained near zero.

Volatility was calculated as (max_high - min_low) / avg_open. Volatility remained low for most of the period but spiked sharply during the crash, indicating high uncertainty and market panic. Post-crash, both price and volatility stayed flat.

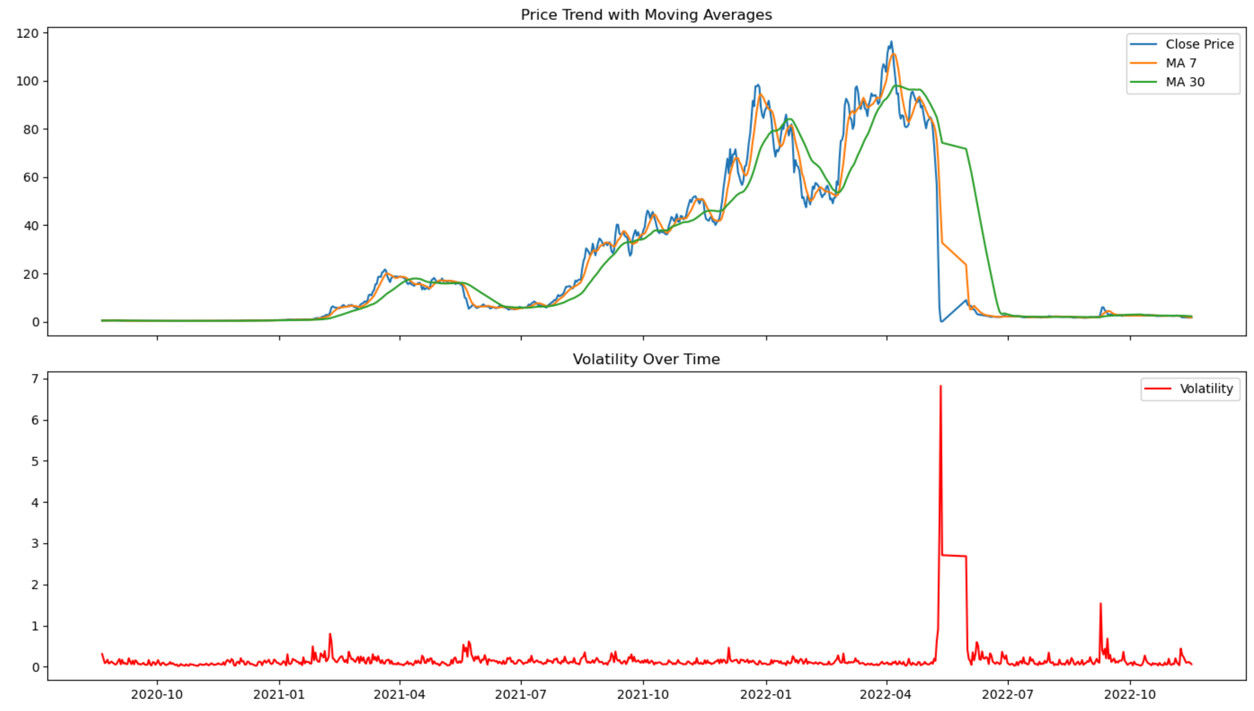


Figure 2: Price Trend and Volatility over time

3.2.4 Volume and Price Correlation

We examined the relationship between total daily trading volume and daily price return. The computed correlation coefficient was -0.566, indicating a moderate negative correlation. This suggests that higher trading volume often coincided with price declines, likely driven by sell pressure.

As we can see from Figure 3, volume spikes typically occurred during or before price drops, with few instances of high volume driving price gains.

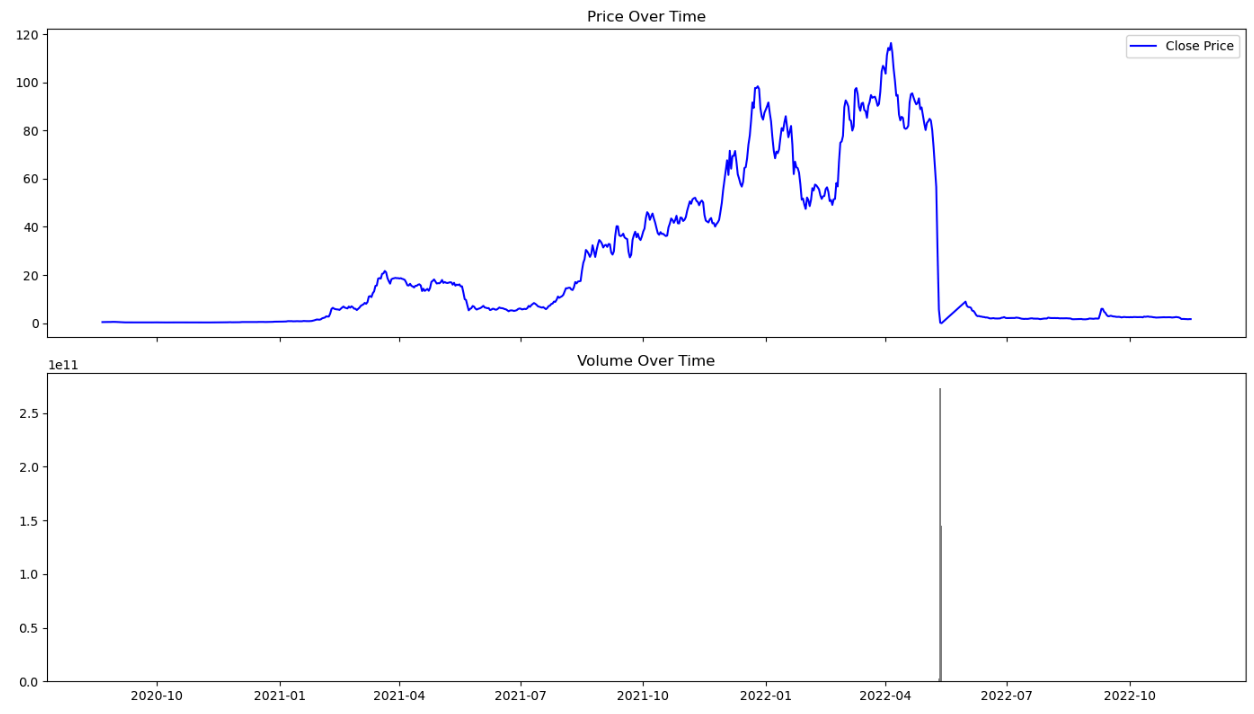


Figure 3: Volume Patterns & Price Behavior of LUNA-USDT

3.2.5 Buyer Behavior and Bullish Signal Detection

We computed the taker buy ratio (buy_volume / total_volume) to estimate buyer dominance. A 7-day moving average was applied to smooth the ratio and reduce noise. We defined a bullish signal as any day where:

- Smoothed taker buy ratio > 0.6
- Daily price return > 0

However, as shown in Figure 4, no days satisfied both conditions, indicating that even when buyers dominated the volume, it did not result in positive price movement. This suggests weak market confidence or lack of follow-through, highlighting the limits of using buyer ratio alone as a bullish indicator.

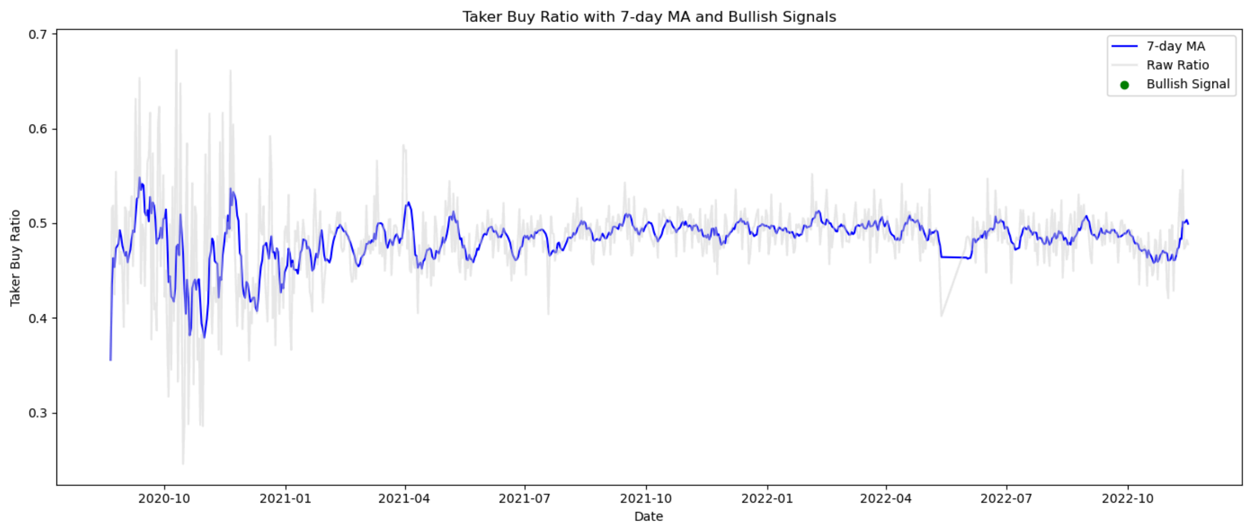


Figure 4:Taker Buy Ratio & Market Sentiment of LUNA-USDT

3.3 Data Cleaning

We dropped unnecessary columns (quote_asset_volume, number_of_trades) and removed rows with missing or invalid values in critical fields: open, high, low, close, and volume.

3.4 Feature Engineering

We generated several derived features to support machine learning:

Daily Return:

- Calculated as (Close - Previous Close) / Previous Close.
- Measures the day-to-day percentage price change and is a core signal for trends.

Volatility:

- Typically the rolling standard deviation of returns over a window (e.g., 7 or 30 days).
- Captures market uncertainty or instability.

ma_7 (7-day Moving Average):

- Average closing price over the past 7 days.
- Smooths short-term fluctuations and highlights recent trends.

ma_30 (30-day Moving Average):

- Average closing price over the past 30 days.
- Highlights longer-term trends and is less sensitive to short-term noise.

3.5 Machine Learning Model:

We applied machine learning models using Spark MLlib:

- **Linear Regression**:

    The engineered dataset containing features such as daily return, volatility, and moving averages was used to train a Linear Regression model using PySpark MLlib. The dataset was split into training and testing sets using a 80/20 ratio. Standard PySpark VectorAssembler and StringIndexer were used to prepare input features for modeling. The Linear Regression model was trained with default parameters, aiming to predict the continuous target variable: daily return. The model was later evaluated to assess its effectiveness in capturing price movement trends.

- **Logistic Regression**:

    We implemented the Logistic Regression model using PySpark MLlib to classify the direction of cryptocurrency returns (positive vs. negative). The target variable was derived by labeling returns as either 1 (gain) or 0 (loss). Prior to training, the dataset was transformed using Vector Assembler to combine relevant numerical features, and StringIndexer was used to convert labels into indexed format.

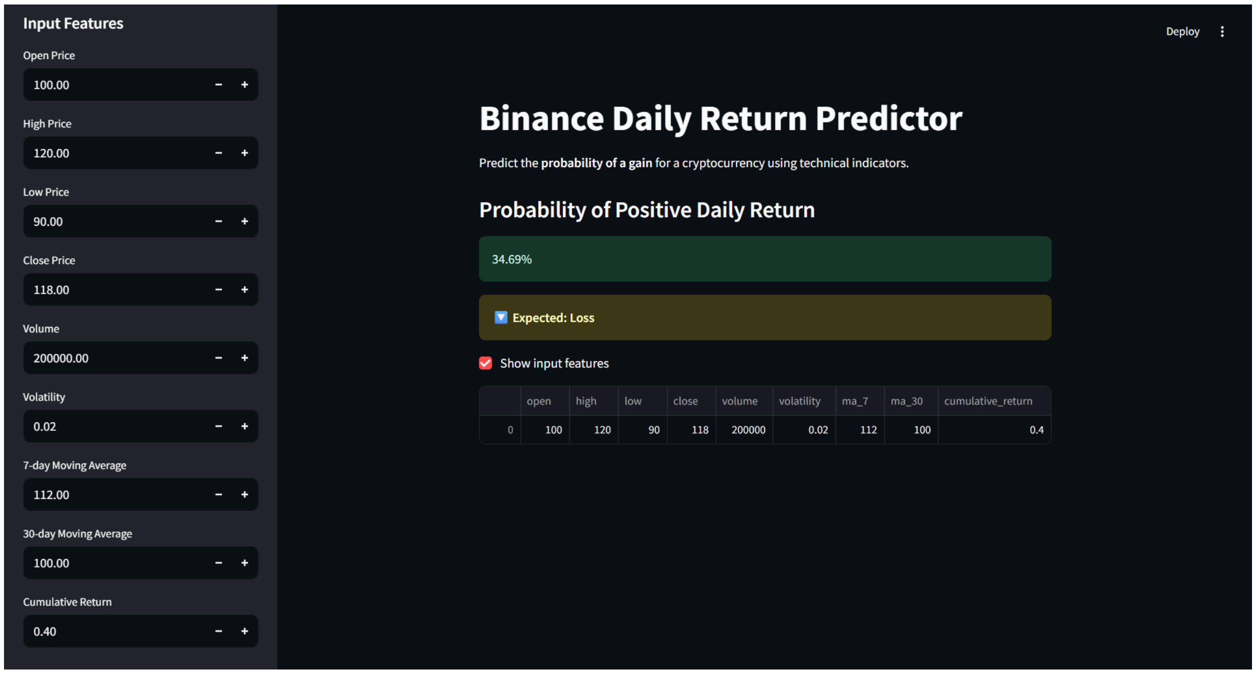
The data was split into 70% training and 30% testing, and the model was trained using default hyperparameters. Logistic Regression served as a simple baseline classifier to assess the ability of engineered features to separate upward and downward market movements.

- **XGBoost:**

  In this project, we implemented the XGBoost Regression model using PySpark to predict short-term price movements of various cryptocurrencies. XGBoost is a gradient boosting algorithm well-known for its high performance on structured/tabular data. It works by building an ensemble of weak prediction models (typically decision trees) in a sequential manner to minimize error.

3.6 Streamlit-Based UI:

With these models being generated, we also developed an **interactive user interface (UI) using Streamlit,** which allowed users to input their own information. Some of them include Open, High, Low, Close, Volume, Moving Averages, and more. Once the user has inputted their information, the UI returns the **predicted daily return** in percentage format. This percentage is an indicator of how well a certain cryptocurrency did. For example, if the percentage returned is below 50%, then that means there was a negative return, or a loss. Ultimately, this UI was designed for ease-of-use, making the model accessible for both technical and non-technical users.



Binance Daily Return Predictor

Predict the **probability of a gain** for a cryptocurrency using technical indicators.

Probability of Positive Daily Return

34.69%

🔽 **Expected: Loss**

✓ Show input features

	open	high	low	close	volume	volatility	ma_7	ma_30	cumulative_return
0	100	120	90	118	200000	0.02	112	100	0.4

Figure 5: Streamlit UI showing input sliders and predicted return display

4. Results and Evaluation

4.1 Linear Regression:

- The model was evaluated using R^2 score and Root Mean Squared Error (RMSE)
- R^2 Score: 0.0922 → very low, indicating weak ability to explain return variability
- RMSE was relatively high → showing large deviation between predicted and actual returns
- Prediction vs. Actual Scatter Plot:
  - Most predictions clustered near zero
  - Indicates model underfit the data and failed to capture price movement trends

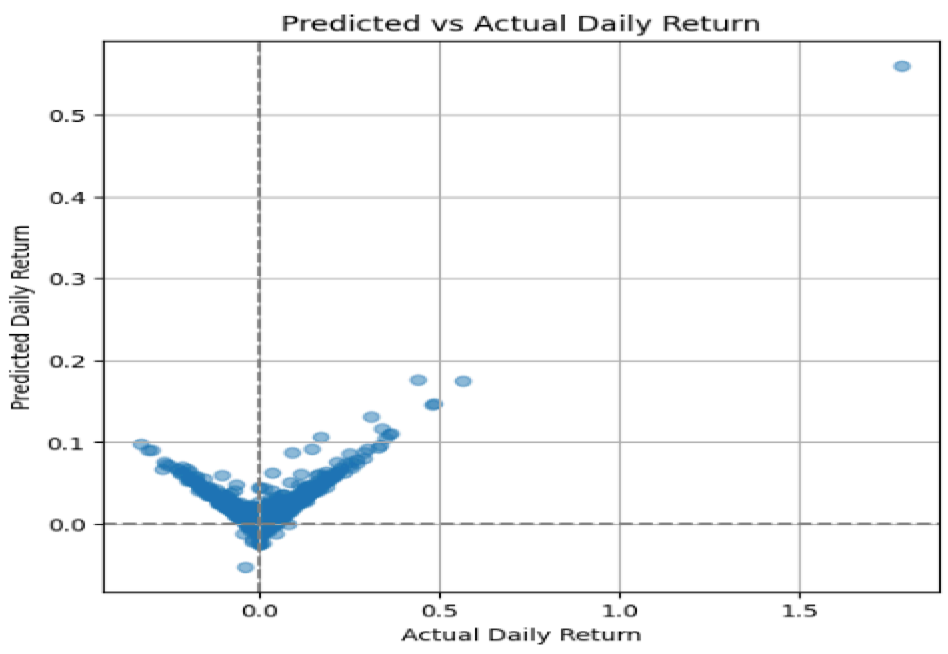


Figure 6: Scatter plot showing Predicted vs. Actual Daily Returns

- Histogram:
  - Actual returns have a wider distribution than the predicted returns
  - Indicates that the predicted returns are more biased towards zero

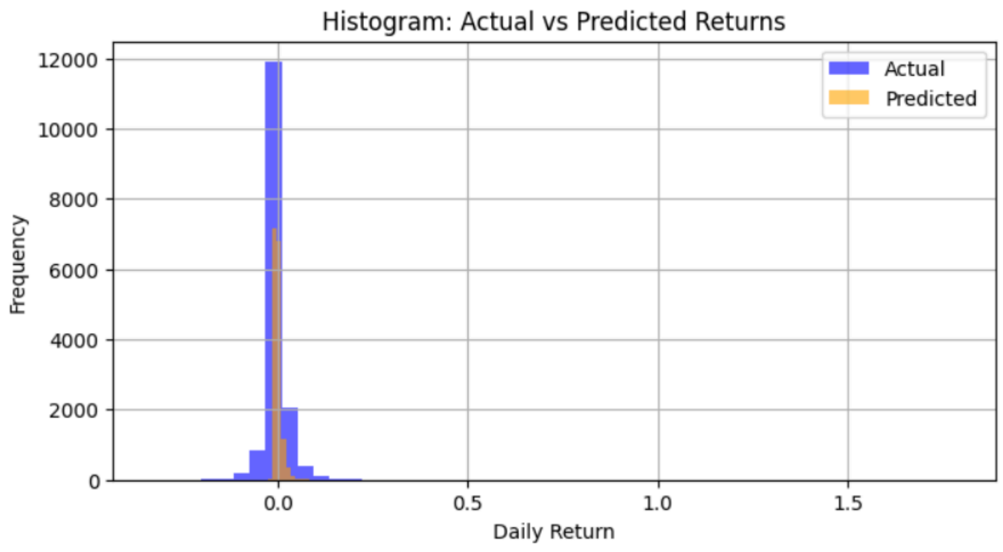


Figure 7: Histogram for visualizing actual and predicted returns

Linear Regression served as a baseline, but is not suitable for modeling volatile and complex crypto return behavior

4.2 Logistic Regression:

- **Performance Metrics:**
  - Accuracy: ~66.7% — indicates the model correctly classified two-thirds of test samples.
  - Precision: ~63.5% — reflects the proportion of predicted positives that were actually positive.
  - Recall: ~66.7% — shows the model's ability to identify positive returns.
  - F1 Score: ~58.0% — the harmonic mean of precision and recall; moderate balance.
- **ROC Curve Analysis:**
  - The ROC-AUC score is 0.6984, suggesting moderate classification performance.

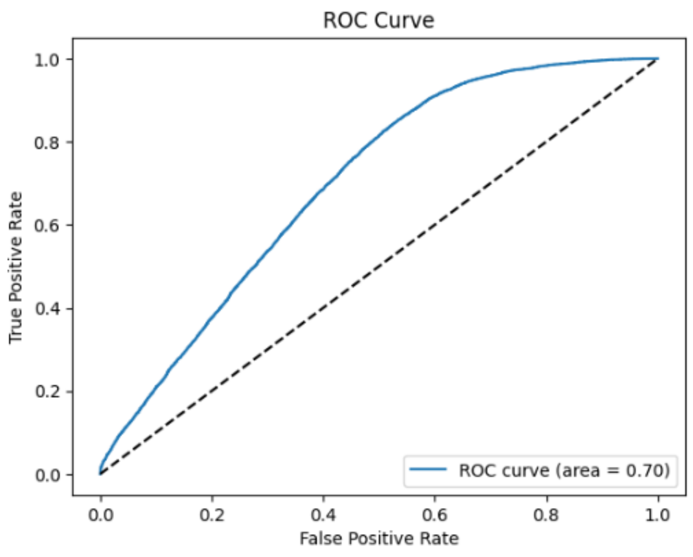


Figure 8: ROC Curve for Logistic Regression model

- **Confusion Matrix Analysis:**
    - Majority of negative-return samples were correctly classified.
    - Positive-return samples were often misclassified, leading to a performance imbalance.

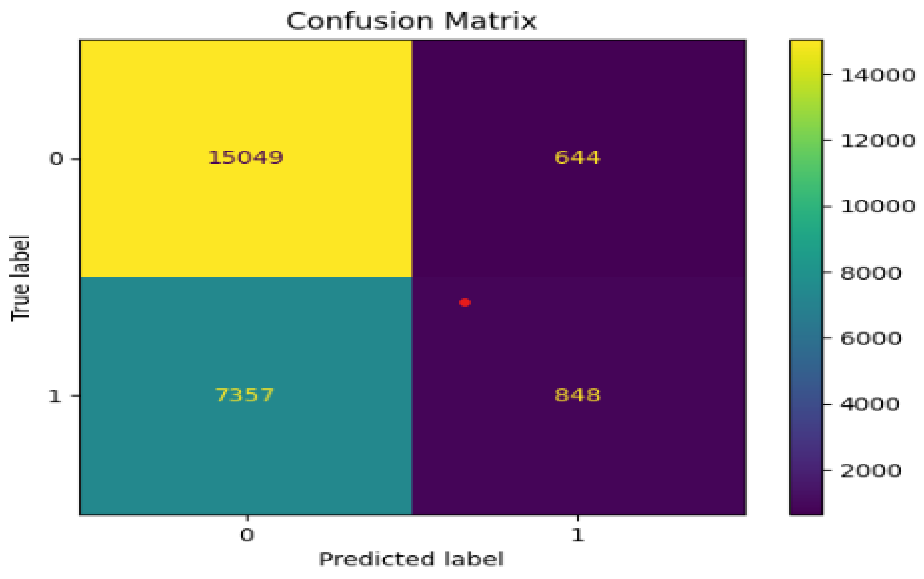


Figure 9: Confusion Matrix showing prediction distribution

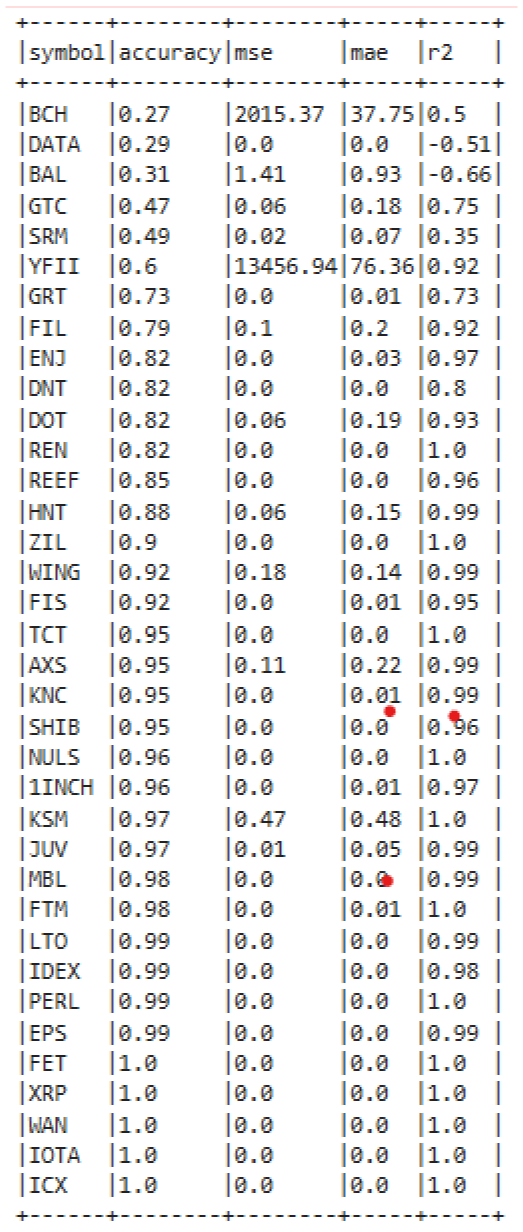
Logistic Regression performs reasonably as a baseline classifier. It struggles with non-linear patterns, indicating the need for more complex models like tree-based methods.

4.3 XGBoost:

The XGBoost model was evaluated per cryptocurrency using the following metrics:

- **R² Score (Coefficient of Determination)**: Measures how well the model explains variance in the target.

- **Mean Squared Error (MSE)**: Penalizes large errors heavily, reflects prediction accuracy.
- **Mean Absolute Error (MAE)**: Measures average absolute prediction error.
- **Accuracy** (custom threshold-based): Indicates directional correctness (up/down movement).



```
+------+--------+--------+-----+-----+
|symbol|accuracy|mse     |mae  |r2   |
+------+--------+--------+-----+-----+
|BCH   |0.27    |2015.37 |37.75|0.5  |
|DATA  |0.29    |0.0     |0.0  |-0.51|
|BAL   |0.31    |1.41    |0.93 |-0.66|
|GTC   |0.47    |0.06    |0.18 |0.75 |
|SRM   |0.49    |0.02    |0.07 |0.35 |
|YFII  |0.6     |13456.94|76.36|0.92 |
|GRT   |0.73    |0.0     |0.01 |0.73 |
|FIL   |0.79    |0.1     |0.2  |0.92 |
|ENJ   |0.82    |0.0     |0.03 |0.97 |
|DNT   |0.82    |0.0     |0.0  |0.8  |
|DOT   |0.82    |0.06    |0.19 |0.93 |
|REN   |0.82    |0.0     |0.0  |1.0  |
|REEF  |0.85    |0.0     |0.0  |0.96 |
|HNT   |0.88    |0.06    |0.15 |0.99 |
|ZIL   |0.9     |0.0     |0.0  |1.0  |
|WING  |0.92    |0.18    |0.14 |0.99 |
|FIS   |0.92    |0.0     |0.01 |0.95 |
|TCT   |0.95    |0.0     |0.0  |1.0  |
|AXS   |0.95    |0.11    |0.22 |0.99 |
|KNC   |0.95    |0.0     |0.01 |0.99 |
|SHIB  |0.95    |0.0     |0.0  |0.96 |
|NULS  |0.96    |0.0     |0.0  |1.0  |
|1INCH |0.96    |0.0     |0.01 |0.97 |
|KSM   |0.97    |0.47    |0.48 |1.0  |
|JUV   |0.97    |0.01    |0.05 |0.99 |
|MBL   |0.98    |0.0     |0.0  |0.99 |
|FTM   |0.98    |0.0     |0.01 |1.0  |
|LTO   |0.99    |0.0     |0.0  |0.99 |
|IDEX  |0.99    |0.0     |0.0  |0.98 |
|PERL  |0.99    |0.0     |0.0  |1.0  |
|EPS   |0.99    |0.0     |0.0  |0.99 |
|FET   |1.0     |0.0     |0.0  |1.0  |
|XRP   |1.0     |0.0     |0.0  |1.0  |
|WAN   |1.0     |0.0     |0.0  |1.0  |
|IOTA  |1.0     |0.0     |0.0  |1.0  |
|ICX   |1.0     |0.0     |0.0  |1.0  |
+------+--------+--------+-----+-----+
```

Figure 10: Per-symbol performance metrics of the XGBoost model across various cryptocurrencies. High-performing tokens show strong accuracy and low error values.

4.4 Summary:

Model	R^2/AUC	Accuracy	RMSE	Comments
Linear Regression	0.0922	-	High	Underfitted, poor predictions
Logistic Regression	0.6984	66.7%	-	Decent baseline, limited power
XGBoost	Varies	-	Low	Best performance, non-linear

5. Future Works

Based on the work that the group has done, there are many different things that we can do to improve the performance of the overall project. For example, addressing the model overfitting issues, and misleading predictions. In future iterations, we could have aimed to enhance model generalization by experimenting with different methods, as well as using regularization and using advanced assembly tuning, which minimizes overfitting and improves reliability. Furthermore, by integrating SHAP or LIME to make the model predictions more transparent and explainable, we would be able to improve interoperability as well.

In this project, the process was through a batch pipeline due to our dataset being from 2017 to 2022. However, by extending it to allow streaming, for example using Spark's Structured Streaming, we would be able to make predictions on live crypto feeds, and deploy a real-time analytics pipeline. Lastly, to make the project train faster and speed up performance, launching a multi-node Spark cluster would be beneficial, especially for EDA, feature engineering, retraining, etc.

6. Conclusion

By leveraging Amazon S3 as storage and Spark as our computation, we were able to develop an end-to-end big data analytics pipeline. This pipeline was used to process and model 35 GB of data, which was in parquet format. This ensured that our storage was fast and scalable, and with the help of S3, allowed distributed data access. With the help of PySpark, exploratory data analysis (EDA) was able to perform very well, which helped us uncover trends in different areas including volatility and daily return. Other than those features, moving averages (for 7 days and 30 days) were also created to give more meaning to our dataset. These new features, including the ones given, helped develop multiple different models. Some of these models include linear regression, logistic regression, and XGBoost, with XGBoost having the best performance. Overall, this project was able to demonstrate the potential of scalable analytics for generating actionable insights in volatile cryptocurrency markets.

7. References

[1] Jorijn Smit, "Binance Full History Dataset," *Kaggle*, [Online]. Available: https://www.kaggle.com/datasets/jorijnsmit/binance-full-history.