DS203 Final Project Report – Cryptocurrency Trend Analysis Using PySpark

Authors: Basit Shah and Bladys O. Perez

Platform: Google Colab (PySpark)

Dataset: 15-minute timeframe cryptocurrency data

Tools Used: PySpark, Google Colab

Problem Definition

The cryptocurrency market has experienced exponential growth over the past decade, leading to increased interest from both retail and institutional investors. However, the volatility and complexity of this market present significant challenges for traders and analysts. Understanding trading patterns, price movements, and market dynamics is crucial for making informed investment decisions.

This project aims to analyze a dataset containing 15-minute trading data for over 234 cryptocurrency pairs (e.g., BTCUSDT, ETHUSDT). The primary objectives are to:

1. Identify Trading Patterns: Analyze the trading volume and price movements to identify trends and anomalies.

2. Calculate Key Metrics: Derive important financial metrics such as *average prices*, *price ranges*, and *cumulative volumes*.

3. Provide Actionable Insights: Generate insights that can inform trading strategies and investment decisions.

By addressing these objectives, the project seeks to contribute to a deeper understanding of cryptocurrency trading dynamics, ultimately aiding traders in making more informed decisions.

Methodology

The analysis was conducted using PySpark, a powerful framework for big data processing. The following steps outline the methodology employed in this project:

1. Data Acquisition

The dataset was sourced from Kaggle, specifically the "Crypto Coins Prices OHLCV" dataset. It contains historical price data for various cryptocurrency pairs, including open, high, low, close prices, and trading volumes.

2. Data Preparation

Upon downloading the dataset, the following steps were taken to prepare the data for analysis:

- Schema Definition: A schema was defined to ensure that the data types were correctly interpreted. The schema included fields for the trading pair, datetime, open, high, low, close prices, and volume.
- Data Loading: The CSV files were read into a Spark DataFrame. This allowed for efficient processing of large datasets.
- Feature Engineering: New columns were created to extract relevant information from the dataset. For instance, the trading pair was derived from the filename, and additional columns for quote currency and coin were generated based on the trading pair.

3. Data Exploration

Exploratory data analysis was performed to understand the dataset better. This included:

- Descriptive Statistics: Basic statistics such as count, mean, standard deviation, and quantiles were calculated for the numerical columns to understand the distribution of prices and volumes.
- Filtering and Querying: Various queries were executed to filter the data based on specific conditions, such as volume thresholds and price ranges. This helped identify significant trading events and anomalies.

4. Data Transformation

Several transformations were applied to derive new metrics and insights:

- Price Range Calculation: A new column, `price_range`, was created to represent the difference between the high and low prices for each trading period. This metric is essential for understanding market volatility.
- Aggregate Functions: Aggregate functions such as average and maximum were used to calculate key metrics like average close price and maximum volume across the dataset.
- Window Functions: Window functions were employed to analyze trends over time. For example, moving averages and cumulative volumes were calculated to provide insights into trading activity.

Task Execution

Task A: Filtering (5 Queries)
- Filtered by volume > 1,000,000
- Filtered trades after "2022-01-01"
- Filtered with multiple AND conditions
- Used OR logic for extreme price points
- Combined complex AND + OR conditions

Task B: New Column
- Created a price_range column using: high-low to analyze volatility.

Task C: Aggregate Functions
- Calculated avg(close) for overall market movement.
- Found max(volume) to identify high-trade days.

Task D: Grouping
- Grouped by pair and computed avg(close) to assess average pair performance.

Task E: Sorting
- Sorted by volume to get the top 10 highest activity periods.

Task F: Joins
- Created a Left Join between result_df and average volume per pair.

Task G: Window Functions
- Used rank() to rank trades within each pair based on volume.
- Used lag() to compare the current close with the previous close.

Task H: Aggregate Window Functions
- Calculated 3-period moving average for close prices.
- Computed cumulative trading volume per pair.

Challenges Faced

- Dataset Size: Loading and merging multiple large files required careful memory management.
- Data Inconsistency: Some files had mismatched columns; resolved with schema inference and consistency filtering.
- Datetime Parsing: Standardized datetime fields using Spark SQL functions.

- Missing data: We extracted coin names from filenames in a PySpark dataset to enhance data analysis.

Results and Insights

- BTC and ETH pairs consistently showed high trading volumes.
- The 3-period moving average revealed upward or downward short-term momentum.
- Joining volume-based ranks helped visualize which pairs dominate user interest.
- The price_range metric exposed volatile vs. stable pairs.

Technologies Used

- PySpark
- Google Colab
- GitHub

Conclusion

This project demonstrated the power of PySpark in handling large-scale, real-world crypto trading data. From ingestion to insight, every task mapped a real business use case for trend identification, portfolio optimization, and behavioural clustering.

By completing each assigned task (A–H) with clarity and precision, the project showcases the technical ability to operate in a modern data pipeline using distributed computing frameworks like Spark.

Future Work

- Add more live crypto datasets with real-time ingestion using Kafka.
- Implement more ML models to forecast prices.
- Connect insights to dashboards using tools like Power BI or Streamlit.

Contributors

Basit Shah and **Bladys O. Perez**
- Data Science CO-OP Students