## **Project Report: Full-Scale Stock Analytics with PySpark and Pandas**

## Objective

This project demonstrates a comprehensive stock analytics workflow using PySpark for scalable data processing and Pandas/Matplotlib for visualization. It is designed for teaching students how to extract actionable insights from time-series financial data.

### Dataset

- Source: a.us.txt historical stock data for a U.S. company
- Columns: Date, Open, High, Low, Close, Volume
- Format: CSV with headers, daily frequency
- Technologies Used

# Analytics Performed

#### 1. Daily Return & Volatility

- Computed daily percentage change in closing price
- Calculated 21-day rolling standard deviation
- Insight: Measures short-term movement and risk

#### 2. Moving Averages

- 20-day and 50-day moving averages
- Insight: Highlights trend direction and momentum

#### 3. Cumulative Return

- Log returns aggregated over time
- Insight: Shows total growth of investment

#### 4. Volume Analysis

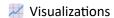
- Identified top 10 volume spikes
- Insight: Detects unusual trading activity

#### 5. Price Extremes

- Extracted highest and lowest closing prices
- Insight: Useful for benchmarking and range analysis

#### 6. Monthly Trends

- Grouped data by month to compute average close
- Insight: Reveals seasonal or cyclical patterns



All plots are generated using Pandas and Matplotlib for clarity and teaching impact.

# **Project Structure**

- ✓ How to Run
  - 1. Clone the repository:

git clone https://github.com/your-username/stock-analytics.git cd stock-analytics

2. Install dependencies:

pip install -r requirements.txt

3. Launch Jupyter Notebook:

jupyter notebook

4. Run stock\_analysis.ipynb step-by-step

### **Solution** Educational Value

This project is ideal for:

- Teaching time-series analysis
- Demonstrating PySpark window functions
- Visualizing financial metrics
- Introducing reproducible data science workflows

Let me know if you'd like help writing the README.md, adding a license, or preparing a GitHub Pages dashboard. I can also help you extend this to multiple stocks or integrate Streamlit for interactivity.