**Documentation: Stock Breakout Analysis Tool**

**Overview**

The **Stock Breakout Analysis Tool** is a Streamlit-based web application that allows users to analyze stock price and volume data to identify breakout days. Breakout days are characterized by a significant increase in trading volume and price movement compared to historical averages. This tool also calculates the potential returns for a defined holding period following a breakout.

The application is built using the following technologies:

* **Python**: Core programming language.
* **Streamlit**: Framework for building interactive web applications.
* **yFinance**: Library for fetching historical stock market data.
* **Pandas**: Library for data manipulation and analysis.

**How the Code Works**

**1. Fetching Stock Data**

The function get\_stock\_data uses the yfinance library to fetch historical stock market data for a given ticker symbol within a specified date range. It calculates the 20-day rolling average volume to determine historical trading activity levels.

* **Input Parameters**:
  + ticker: Stock ticker symbol (e.g., "AAPL" for Apple).
  + start\_date: Start date for fetching data.
  + end\_date: End date for fetching data.
* **Output**:
  + A Pandas DataFrame containing historical stock data with an additional column for the 20-day average volume.

**2. Analyzing Breakouts**

The analyze\_breakouts function identifies days where trading volume and price movements meet predefined thresholds. It calculates the returns for a specified holding period after each breakout.

* **Input Parameters**:
  + data: The stock data (Pandas DataFrame) with volume and price information.
  + vol\_threshold: Percentage increase in volume required for a breakout.
  + price\_threshold: Percentage increase in price required for a breakout.
  + holding\_period: Number of days to hold the stock after a breakout.
* **Output**:
  + A DataFrame containing breakout days and calculated returns.

**3. Interactive Web Application**

The application uses Streamlit to create an interactive user interface.

* **User Inputs**:
  + Ticker: Stock ticker to analyze.
  + Start Date and End Date: Date range for analysis.
  + Volume Breakout Threshold (%): Volume increase threshold for identifying breakouts.
  + Price Change Threshold (%): Price increase threshold for identifying breakouts.
  + Holding Period (days): Number of days to hold the stock post-breakout.
* **Process**:
  + When the user clicks the "Generate Report" button, the app fetches stock data using get\_stock\_data.
  + It identifies breakout days and calculates returns using analyze\_breakouts.
  + Results are displayed in a data table, and a CSV download button is provided for exporting the report.

**How to Use**

1. **Launch the App**: Run the following command in the terminal:
   * streamlit run app.py
2. **Input Parameters**:
   * Enter the stock ticker symbol (e.g., "AAPL").
   * Select the start and end dates for analysis.
   * Adjust the volume breakout threshold, price change threshold, and holding period.
3. **Generate Report**:
   * Click "Generate Report" to view the analysis in a table.
   * Download the results as a CSV file.

**Idea and Benefits**

The tool provides actionable insights for traders and investors by:

* Identifying high-potential trading opportunities through breakout analysis.
* Automating the data processing and analysis workflow.
* Allowing customization of thresholds and holding periods to fit different trading strategies.

This approach can be extended to include additional features such as technical indicators, machine learning models for predictions, or integration with trading platforms.

References:

* ChatGPT: For Streamlit, yfinance usage and cleaning the code.
* Streamlit config: https://docs.streamlit.io/