Crypt(Ech)o: Tracing How Headlines Ripple Through Markets

Overview

Finance in politics. Politics and finance. The two concepts, though distinct, are inseparable. Fluctuations in one will almost always entail change in the other. But by how much? And for how long? To what extent?

This project involves building an end-to-end data pipeline which integrates cryptocurrency market data with financial news and macroeconomic indicators to provide holistic market insights. In the following sections, we will discuss the necessary requirements and steps for you to build this pipeline.

Data Acquisition

There are a couple of free real-time data sources that you can use to build your pipeline. Our recommended sources for fetching the news, and crypto market data are News API and CoinCap, respectively. These, however, are not necessarily the sources you must use: feel free to explore around the internet for other free sources as well. Some other sources that might be useful to you are listed below:

- New York Times Newswire API
- Blockchain transactions
- Binance WebSocket API

These free resources usually have some sort of usage limit in place, so efficiency in development and testing and is a must.

The data we expect you to acquire from the news is ones that might influence the financial market; such as politics, or finance news itself.

> ⓘ Your data acquisition solution should preferably implement some sort of incremental data fetching strategy, such that it wouldn’t retrieve redundant or duplicated data.

Data Processing

The processing step is where the magic happens. This phase includes a number of tasks that need attention.

- We expect you to build a modular ETL framework with clear separation of extraction, transformation, and loading stages.
- Since you’ll be acquiring data from at least two sources, integrating them properly is one of the key project requirements. How, when, and where this integration occurs, is up to you.
- Extra points for implementing some measure of data validation and quality check.
- The intermediate file type storage to use is up to you.
- Think in scale. Although the current development version of your pipeline won’t have much data to deal with, it will eventually. How will your pipeline handle terabyte-scale data? What about frequently updated ones?
- How much tolerance should your pipeline have against errors? Where should it fail, warn the user, or completely ignore an error?
- Store the processed data on a database of your choice—our recommendation would be PostgreSQL, but you can also choose another database should you feel the need to do so.

Data Storage

Design a multi-layer storage solution:

- Raw data lake
- Processed data warehouse
- Real-time serving layer (if you see the need for one)

How will your storage solutions handle the evolution of your data schema over time? What would happen if we decide to add extra data sources, fields, etc?

Deliverables

- The codebase which includes your solution to the presented problem
- Proper documentation on various aspects of the codebase and their usage
- Prepared sample demo for presenting during the code review session
- Bonus points for containerizing your solution
- Writing unit tests for your code is another plus

Additional Notes

- There is a fair amount of open-endedness in the problem description. This is on purpose, so that we may be able to measure your creativity in handling the challenge: bonus points for any innovative solutions that you come up along the way.
- Automate as much as possible. Manual intervention should be kept to a minimum.
- Using other tools and services is completely acceptable as long as you understand how they work. We do not prohibit the use of AI tools such as ChatGPT, Gemini, etc. but you must demonstrate that you have sufficient understanding of the solution you're providing, and have not just copy pasted code from other sources.
- We don't expect a fully functional solution from you that covers every aspect of the project description. Instead, we want to witness how you handle the challenges you'll face along the way, how quickly you adapt to unfamiliar technologies, and measure your architectural thinking capabilities in solving the various problems presented in this document. The result isn't the only thing that matters.
- Keep in touch! Feel free to ask for clarifications from the team whenever necessary.