1

Using Graph Algorithms to Analyze S&P 500 Correlation Networks

Team Members: Indri Adisoemarta, Alec Heyde, Vaishnavi Vasudeva, Daniel Wang
Date: 4/17/25

Speaker: Alec

Hi there and welcome to our presentation about using graph algorithms to analyze S&P 500 correlation networks and our intended audience is investment professionals who focus on public market investing

Why Analyze S&P 500 Stock Correlations?

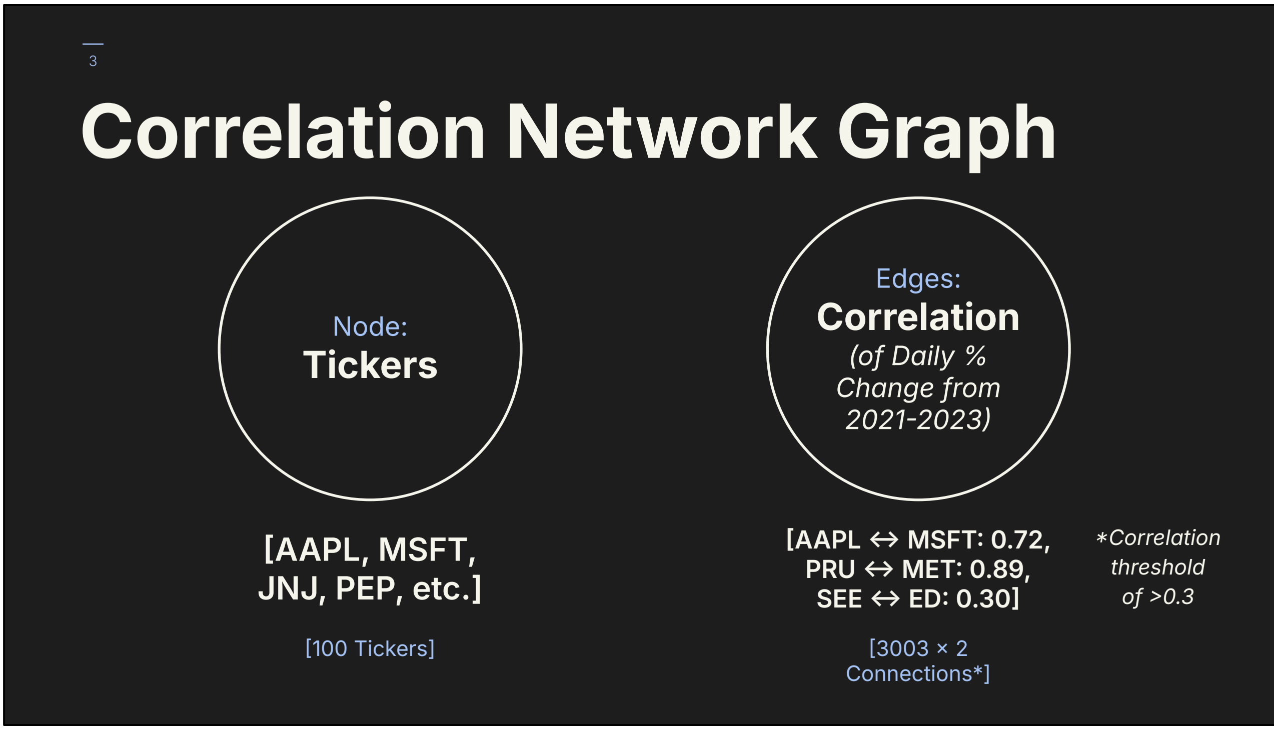
- Stock prices in the S&P often move together - sometimes even when companies are in different industries
- Graph analysis helps spot these patterns of movement more clearly than traditional tools
- These insights can support better portfolio diversification and highlight how momentum spreads

Speaker: Alec

One important dynamic in public markets is that stock prices often move together - not just within industries, but sometimes across them. For example, if there's major news about artificial intelligence, it might not just move NVIDIA's stock, but also software companies like Adobe or infrastructure players like Amazon. This kind of co-movement can be driven by sector sentiment, index trading, or narratives - and it is not always rooted in fundamental changes to a business

Traditional tools like correlations matrices are great, but they're flat – they don't show us the bigger structure of relationships. Graph analysis, on the other hand, lets us represent stocks as nodes and their relationships – like price correlations or news co-mentions as edges. This lets us zoom out and detect clusters or communities of stocks that move together. It's especially powerful when those clusters don't follow typical sector boundaries – that is something traditional classification might miss entirely

If you're building a stock portfolio, it's not enough to just pick stocks from different sectors. Graph analysis can show you whether those stocks are actually moving independently – or if they're secretly tied together through shared momentum. This can help you diversity more meaningfully and also understand how momentum can spread from a ‘narrative leader’ – like Tesla or NVIDIA – to related stocks, even in other sectors

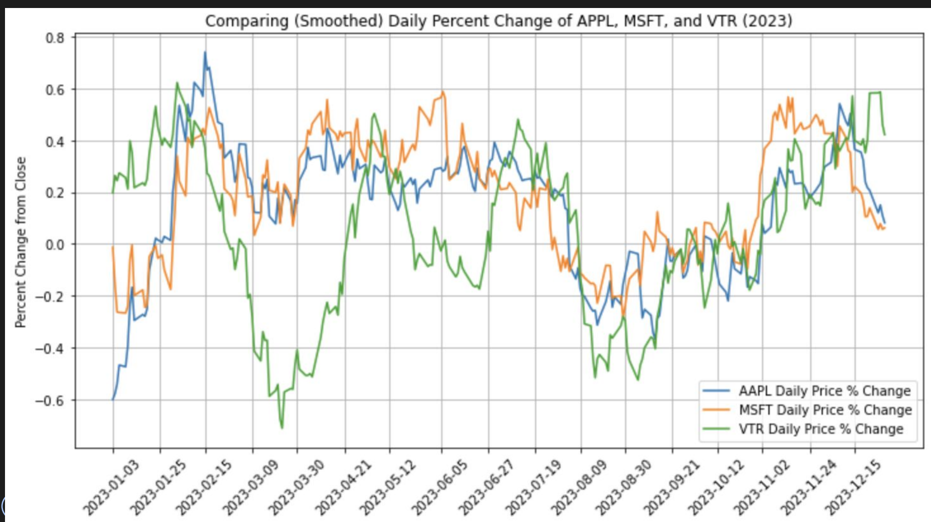


Rubric Notes: Slide(s) must have content explaining the design of at least 1 graph

Feel like this would be the explanation of how it was built / what are the main components

4

Correlation Network Graph



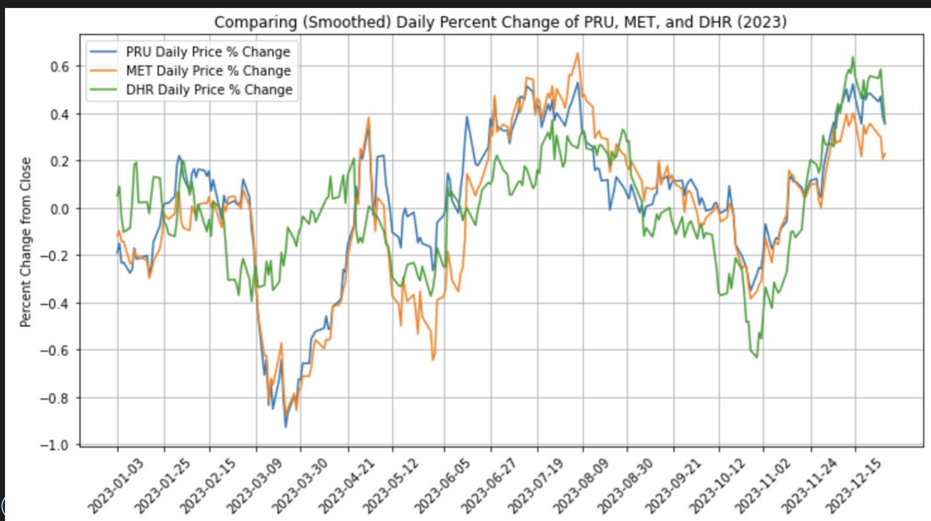
Correlation:

AAPL ↔ MSFT
0.72

AAPL ↔ VTR
0.30

5

Correlation Network Graph



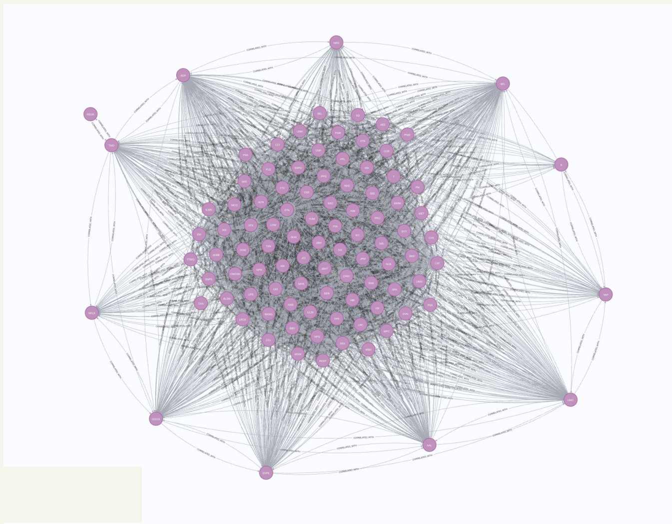
Correlation:

PRU ↔ MET
0.89

PRU ↔ DHR
0.30

6

Corr. Network Graph Result



Feel like this would be a screen shot of the graph so folks can visualize what we are talking about after understanding how it was created

Closeness Centrality Algorithm

Closeness Centrality (Harmonic Centrality): Measures the average of the shortest path distances between a node and all other nodes

- ➔ Nodes with high closeness have the shortest distances to other nodes
- ➔ High closeness indicates high correlation with many other S&P 500 companies' stock price movements

Name	Closeness
ITW	0.964286
REG	0.959184
BEN	0.948980
...	...
K	0.525510
DVA	0.476190
REGN	0.426871

Speaker: Indri
One of the main algorithms we plan to run on our graph is the Closeness Centrality Algorithm, specifically the Harmonic Centrality version. This algorithm measures the average of the shortest path distances between a node and all other nodes.
Nodes with high closeness have the shortest distances to other nodes. In our case, as we did 1 minus the correlation for our distance values, high closeness indicates high correlation with many other Fortune 500 companies' stock price movements. You can see the top 3 stock tickers ranked by closeness centrality value as well as the bottom 3 stock tickers in the table on the right.
Rubric Notes: Slide(s) must have content explaining at least 1 data science graph algorithm: 2 points

9

Business Significance

Closeness Centrality can act as a proxy for identifying influential companies in the S&P 500, and result in non-intuitive insights

Allows diversification of portfolio with stocks that have a variety of price movements

A relational database would not be able to run centrality graph algorithms as easily as we can using Neo4J

ITW: company that produces engineered fasteners and components, founded in 1912

REGN: American biotechnology company founded in 1988

Speaker: Indri
Now why does this matter for our business? Closeness Centrality can act as a proxy for identifying influential companies in the S&P 500, and result in non-intuitive insights. For example, the top closeness centrality ticker from our algorithm was ITW, the company described on the right. It is a company that produces engineered fasteners and components, founded in 1912. A lot of manufacturing companies probably use fasteners and engineered components, so it kind of makes sense that this ticker's movement is similar to a lot of other stock movements. REGN, the ticker with the lowest closeness centrality value, is a biotechnology company, whose fundamentals are probably very different from most other stocks in the S&P 500.

This information can help us diversify our portfolio with stocks that have a variety of price movements, allowing us, for example, to cancel high losses in some stocks with high gains in others.

A relational database would not be able to run centrality graph algorithms as easily as we can using Neo4J, and we'd miss out on these non-intuitive insights.

Rubric Notes: Slide(s) must have content explaining how it addresses at least 1 business example from the business case scenario: 2 points

- Slide(s) must have content explaining how a relational database is not a good fit for at least 1 business example from the business case scenario: 2 points

Thought that this slide could be used to explain whatever business example we choose to highlight for whatever algorithm we highlight and that would also serve as the proof point for why a relational database is not a good fit and therefore makes sense to have together, but separate of the algorithm explanation

MongoDB Business Cases

- Business Case 1: Storing News Articles and Sentiment for Stocks
  - MongoDB works well because:
    - It handles unstructured, text-heavy content
    - Documents can be nested
    - Fast to query

mongoDB

Speaker: Alec
For our MongoDB use case, we focus on storing **news articles and sentiment data** related to stocks — which is important in financial market analysis.

MongoDB, being a **document-based NoSQL database**, allows us to store large blocks of text — such as entire articles or comment threads — **natively** in JSON-like documents.

In MongoDB, documents can contain **subdocuments or arrays**, allowing for natural nesting of related information.For example, you could store: A stock ticker. A list of news articles. Each article containing metadata like the date, publisher, or raw text.

MongoDB is optimized for read-heavy workloads, especially when queries are shaped around document structures and indexes. This is important for real-time or near-real-time analytics pipelines — for example, you may want to: Pull the latest 100 articles for a stock and query only those with negative sentiment

10

Redis Business Cases

- Business Case 1: App shows real-time S&P 500 prices, correlations, and graph-based metrics
  - Redis works well
    - In-memory = ultra-fast read/write
    - Caches frequently accessed data
    - Ideal for dashboards, alerts
- Both MongoDB and Redis will not work well for analyzing graph relationships with stocks
  - Not suited for complex graph analysis
    - Redis and MongoDB lack graph traversal features

10

For this business case, imagine a financial app that needs to show real-time S&P 500 stock prices, correlation data between stocks, and the latest graph-based metrics. The key here is speed—users expect updates instantly, without any delays.

Redis is a great fit for this use case because it's an in-memory data store. That means data is stored in RAM, allowing for super fast reads and writes.

It's also commonly used for caching, so things like frequently accessed stock prices or calculations can be retrieved almost instantly. This makes Redis perfect for fast-loading dashboards and real-time alerts.

It's lightweight, simple, and fast—everything we'd want in a system that needs to push out real-time financial data to users.

However, when it comes to analyzing complex relationships between stocks using graph-based methods, Redis—and even MongoDB—aren't the best tools. They're not designed for deep graph traversal or complex network analysis.

For that kind of functionality, we'd need to look at something like a graph database, such as Neo4j."

A key insight from our work is that stock price movements often reflect more than just sector classifications. Using graph analysis, we uncovered relationships that wouldn't be obvious through traditional tools like correlation matrices or sector labels.

This approach can help investors build portfolios that are meaningfully diversified—not just on paper, but based on actual behavior in the market. For example, graph algorithms like closeness centrality helped us find companies that influence many others, regardless of their official industry.

Relational databases like SQL just aren't designed for this kind of network analysis. Without a graph database like Neo4j, these insights would be incredibly hard to access.

Looking ahead, our next step is to implement **label propagation**—a community detection algorithm. It will help us identify hidden groups of stocks that behave similarly over time, even if they come from very different sectors. That could open the door to entirely new ways of thinking about diversification and risk.