FROM STOCK TO SCHEMAS

DIVERSIFYING PORTFOLIOS WITH GRAPH DATABASES

04.14.2025

RYAN FARHAT-SABET, MAIA KENNEDY, HANNAH MACDONALD, KRISHNA TUMMALAPALLI
UNIVERSITY OF CALIFORNIA, BERKELEY

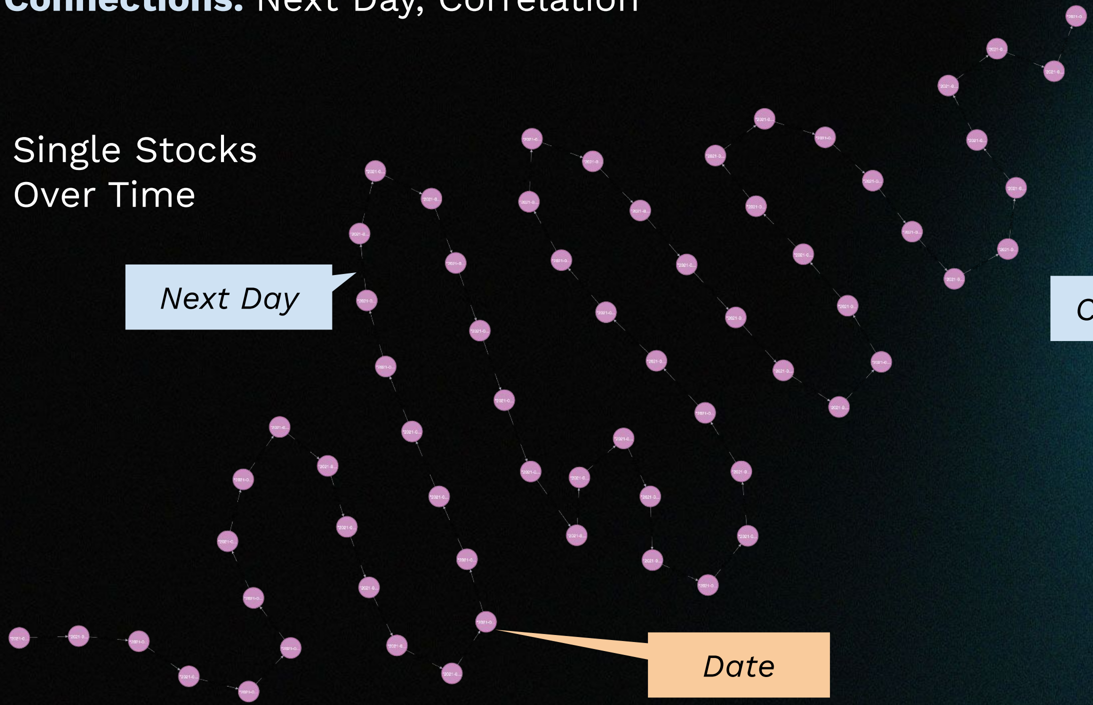
HOW DO I CREATE A DIVERSIFIED STOCK PORTFOLIO TO MINIMIZE RISK EXPOSURE?

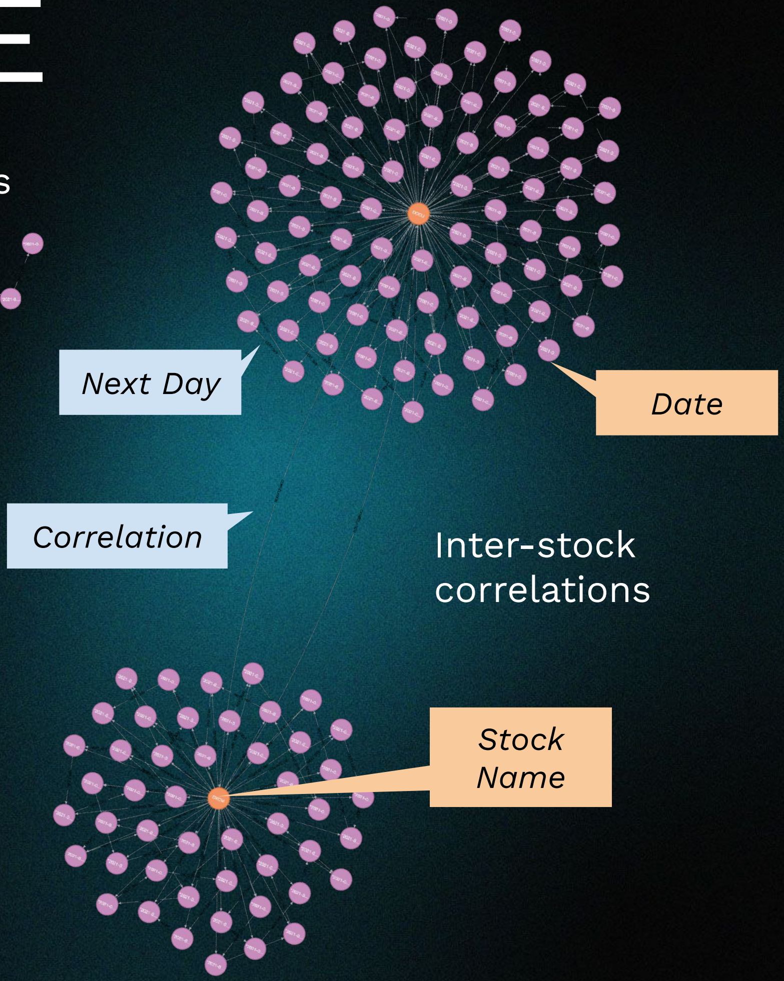
1. Translate tabular data into nodes & relationships
2. Visualize relationships in graph form
3. Leverage native Neo4j algorithms to uncover correlations, clusters, and centrality of stock influence
4. Facilitate real-time updates (Redis) and dynamic schema changes with time (MongoDB)

ABOUT OUR BASE

Nodes: Stock Name, Dates, Close Price, Volume of Trades
Connections: Next Day, Correlation

Single Stocks
Over Time





Inter-stock
correlations

OUR ALGORITHMS

PEARSON CORRELATION

Which stocks are correlated with one another?

JACCARD SIMILARITY

Which stocks are dissimilar or helpful for a portfolio spread?

LOUVAIN MODULARITY

Can we group stocks into meaningful clusters?

BETWEENNESS

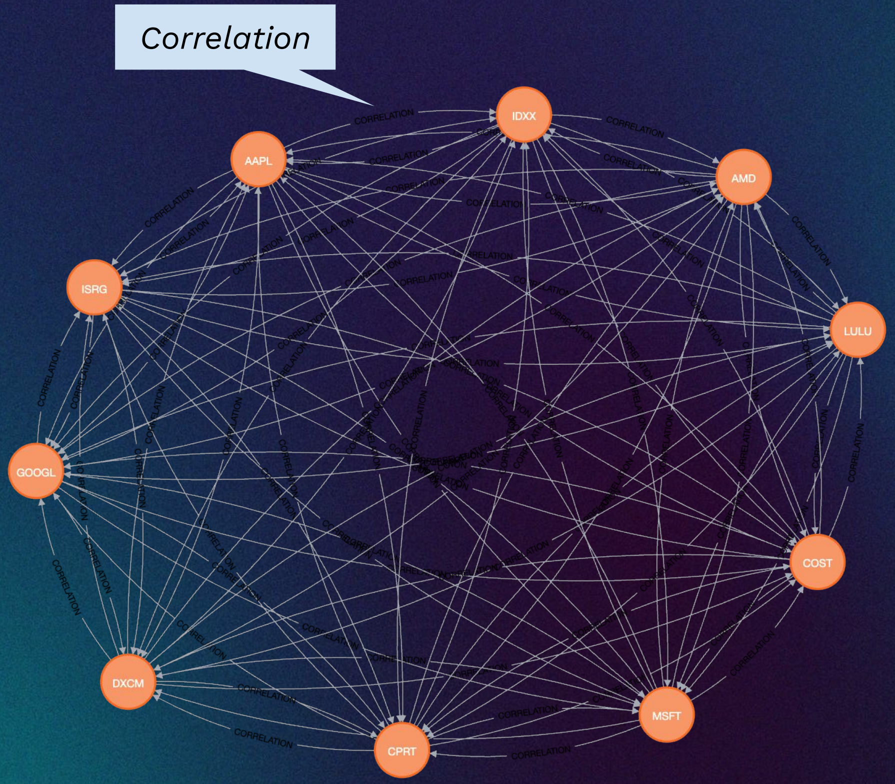
Which stocks are most connected to other stocks?

PAGE RANK

Which stocks have the most influence over other stocks?

Why not relational database?

- Require deep ...



PEARSON CORRELATION

- This algorithm looks at the linear relationship between two continuous variables
- Helps identify stocks that move similarly in price over time
- The Pearson Correlation was calculated based on stock closing prices
- Graph displays the stocks as nodes and their connections if the correlation was greater than 0.8

PEARSON CORRELATION CONT.

Stock1	Stock2	Correlation
AAPL	ADBE	0.9590710112246545
AAPL	ADP	0.8308901906072841

Stock1	Stock2	Correlation
AAPL	CERN	-0.04381570064937407
AAPL	EA	0.0800269926865635

- Findings:
- Strong Positive Correlations (> 0.8) show Similar Stocks
- Low Correlations (< -0.2 and 0.2) show Unrelated Stocks
- Strong Negative Correlations (> - 0.8) show Inverse Moving Stocks

LOUVAIN MODULARITY

	ticker	community	intermediate_community
0	BIIB	14	[14, 14]
1	BKNG	15	[15, 15]
2	CERN	18	[18, 18]
3	CSX	26	[26, 26]
4	CTSH	28	[28, 28]
5	ADBE	29	[29, 29]
6	AVGO	29	[29, 29]
7	CRWD	29	[29, 29]
8	CTAS	29	[29, 29]
9	DLTR	29	[29, 29]
10	DOCU	29	[29, 29]

Step 1: Organized data by stock & date

Step 2: Calculated correlation based on stock closing prices

Step 3: Created connections between stocks only if they had a strong correlation (> 0.8)

Step 4: Detect clusters of stocks or communities using the Louvain algorithm

Step 5: Noted intermediate communities to show how stocks moved through sub-groups during clustering

JACCARD SIMILARITY

- Identify similar clusters of stocks based on volume of trading
- Nodes
  - Stock
  - StockTradingDay
  - volumeCategory
- Relationships
  - IN_VOLUME_CATEGORY
  - HAS_VOLUME_CATEGORY

Step 1: Create volumeCategory nodes based on trading volume.

- HighVolume: > 10M per day
- MediumVolume: 1M to 10M per day
- LowVolume: < 1M per day

Step 2: Link each StockTradingDay node to volumeCategory node using IN_VOLUME_CATEGORY relationship

Step 3: Link each Stock node to volumeCategory node using HAS_VOLUME_CATEGORY relationship

Step 4: Create pairs of stocks with jaccard similarity

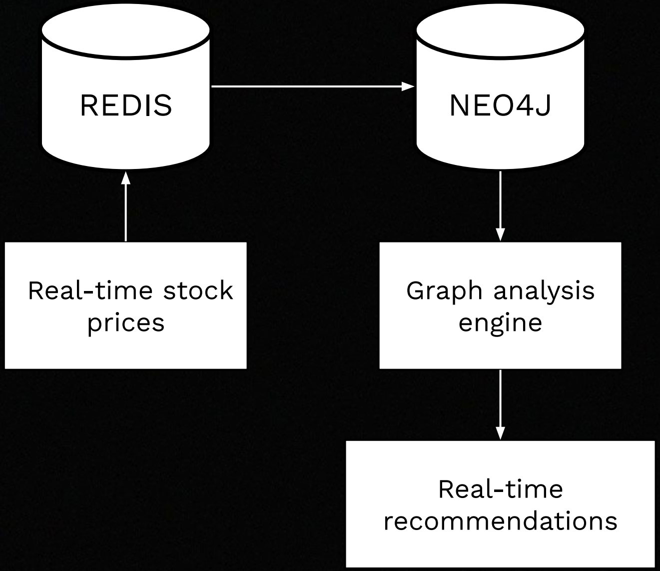
- 1 - always in the same bucket
- 0 - never in the same bucket

REDIS

<u>Use Case</u>: real-time stock prices and risk-based recommendations

- Neo4j would connect to Redis for real-time stock prices to provide recommendations
  - Calculate different graph algorithms in real time
  - Real-time portfolio suggestions
  - Faster than querying a database or data warehouse

- **Why not relational database?**
  - Redis is faster than traditional relational databases



MONGODB

Why a document store?

- Business Use Case: Dynamically update document structure (Keys & Values)
- Schema less - each ticker can have different structure based on similarities
- Helps with real-time analytics by facilitating quick iteration for risk recommendations

Why not relational database?

- Forces us to have same structure for every ticker.
- Need for complex queries to retrieve data (multi table joins)

{'_id': ObjectId('67fd9819b07b767971278d25'),
'ticker': 'AAPL',
'jaccard_similar': ['MSFT', 'INTC', 'NVDA', 'AMD'],
'jaccard_dissimilar': ['CSCO', 'CSX', 'ATVI', 'CTSH', 'BIDU', 'CRWD'],
'betweenness_score': 66.0,
'pagerank_score': 1.4397808463255954,
'louvain_community': 77,
'pearson_similar': ['ADBE', 'ADP', 'ALGN', 'AMD', 'ANSS', 'ASML', 'AVGO', 'CDNS', 'CDW', 'CHTR', 'CMCSA', 'COST', 'CPRT', 'CRWD', 'CTAS', 'DOCU', 'DXCM', 'EBAY', 'FB', 'GOOG', 'GOOGL', 'IDXX', 'INTU', 'ISRG', 'LULU', 'MELI', 'MRNA', 'MRVL', 'MSFT', 'NVDA', 'ORLY', 'PAYX', 'PEP', 'QCOM', 'REGN', 'SNPS', 'SPLK', 'TEAM', 'TSLA', 'VRSK', 'XLNX'],
'pearson_dissimilar': ['ATVI', 'FOX', 'KHC', 'PCAR', 'PDD', 'TCOM', 'WBA']}

THANK YOU

APPENDIX

Raw Tabular Data

	Date	Open	High	Low	Close	Adj Close	Volume	Name
0	2021-05-03	132.039993	134.070007	131.830002	132.539993	132.117294	75135100	AAPL
1	2021-05-04	131.190002	131.490005	126.699997	127.849998	127.442261	137564700	AAPL
2	2021-05-05	129.199997	130.449997	127.970001	128.100006	127.691475	84000900	AAPL
3	2021-05-06	127.889999	129.750000	127.129997	129.740005	129.326233	78128300	AAPL
4	2021-05-07	130.850006	131.259995	129.479996	130.210007	130.015213	78973300	AAPL
5	2021-05-10	129.410004	129.539993	126.809998	126.849998	126.660225	88071200	AAPL
6	2021-05-11	123.500000	126.269997	122.769997	125.910004	125.721642	126142800	AAPL
7	2021-05-12	123.400002	124.639999	122.250000	122.769997	122.586334	112172300	AAPL
8	2021-05-13	124.580002	126.150002	124.260002	124.970001	124.783043	105861300	AAPL
9	2021-05-14	126.250000	127.889999	125.849998	127.449997	127.259331	81918000	AAPL