NeuroStockViz: Visualizing Stock Correlations as Neural Network Structures

By Team 2: Ishant Kundra, Rahaan Gandhi, Tanishq Chopra

Introduction

Financial market dynamics are complex and influenced by various factors affecting individual stocks and the broader market. Investors, analysts, and researchers seek effective ways to understand these interrelationships to predict market movements and improve decision-making. Traditional financial visualization methods, such as line charts and correlation heatmaps, often fail to capture the intricate connections among stocks. To address this, our project, NeuroStockViz, introduces a novel approach that utilizes intuitive, neural network-inspired visualizations to represent stock correlations.

NeuroStockViz responds to the demand for clearer insights in financial analysis, particularly in how stocks within major indices like the Dow Jones Industrial Average influence each other, even during volatile market conditions. This report examines how visualizing stocks as interconnected nodes can enhance understanding, enabling users to identify key influencers and changing correlations over time for better investment decisions.

What is NeuroStockViz?

NeuroStockViz is an interactive, web-based visualization tool designed to represent stock market correlations through a novel, neural-network-inspired graphical approach. Unlike conventional static correlation matrices or basic charts, NeuroStockViz dynamically structures stocks as interconnected nodes, with connections (edges) between these nodes reflecting the strength and direction of their correlations. Each edge's weight is derived from Pearson's correlation coefficient, quantitatively measuring how closely stock price movements align over specific timeframes.

The tool provides users with real-time interactions to select specific stocks, industry sectors, correlation thresholds, and time intervals, offering a tailored exploration of financial data. The primary visualization dynamically adapts, revealing influential stocks, correlation clusters, and evolving market dependencies. Additional visualization features, including mini time-series charts and pairwise correlation heatmaps, provide further analytical depth and context.

Through NeuroStockViz, we aim to bridge the gap between financial analysis and advanced data visualization techniques, empowering users to effortlessly identify hidden market patterns,

influential stocks, and correlation dynamics previously challenging to decipher using traditional methods.

Research Question

- How do stock price movements within the US stock market influence each other?
- Can visualizing stock correlations in a neural network-like structure provide better insights into market dependencies?
- Which stocks are key influencers in the network, and how strong are their correlations with others?
- How does the correlation structure change over time, especially during market volatility?

Related Work

- **Correlation Heatmaps - Asset Correlations -** Portfolio Visualizer: This tool allows users to view correlations for stocks, ETFs, and mutual funds over a specified period. https://www.portfoliovisualizer.com/asset-correlations?utm_source=chatgpt.com

- **Graph Theory in Finance - An Alternative Way to Visualize Stocks' Correlations - DileQuante:** This article discusses using graph theory to model pairwise relations between stocks, introducing methods like Minimum Spanning Tree for visualization. https://dilequante.com/an-alternative-way-to-visualize-stocks-correlations/?utm_source=chatgpt.com

- **Machine Learning Models - Stock Market Forecasting with Differential Graph Transformer**: This article proposes a novel graph transformer architecture that updates stock correlation graphs dynamically to track volatile market conditions. https://medium.com/stanford-cs224w/stock-market-forecasting-with-differential-graph-transformer-62d095ebc821

Comparisons

Compared to Correlation Heatmaps (Portfolio Visualizer):

- **Visualization Method:** Traditional correlation heatmaps, like Portfolio Visualizer's, present static pairwise correlations. NeuroStockViz innovates by visualizing correlations dynamically as interconnected nodes, mimicking neural network structures, and making complex interdependencies intuitive.

- **Interactivity:** While Portfolio Visualizer provides limited interactivity, NeuroStockViz offers highly interactive exploration with real-time filtering by stock, sector, time frame, and correlation threshold.
- **Analytical Depth:** NeuroStockViz provides richer contextual information via integrated mini time-series charts and dynamic correlation heatmaps, enabling deeper analytical insights.

Compared to Graph Theory Visualization (DileQuante):

- **Graph Structure:** DileQuante uses traditional graph theory methods like Minimum Spanning Trees to visualize stock correlations. NeuroStockViz employs neural network-inspired structures, emphasizing weighted edges directly derived from Pearson's correlation coefficients.
- **Dynamic Representation:** NeuroStockViz's visualization adapts dynamically over user-specified timeframes, reflecting real-time changes in correlation, unlike the static or periodically updated visualizations typically presented in graph-theoretical approaches.
- **Usability and Clarity:** NeuroStockViz specifically targets user-friendly and intuitive interfaces, enhancing clarity through direct interaction and real-time updates, making it accessible even to non-technical financial analysts and students.

Compared to Differential Graph Transformer (Machine Learning Models):

- **Focus on Visualization:** The Differential Graph Transformer primarily uses graph-based architectures for predictive forecasting, dynamically updating correlations internally. In contrast, NeuroStockViz prioritizes external, user-interactive visualization, allowing direct exploration and immediate visual comprehension of correlation structures.
- **Educational and Exploratory Purpose:** NeuroStockViz explicitly aims at educational value, providing intuitive visual exploration to help users understand complex stock market dependencies without requiring advanced knowledge in machine learning.
- **User Engagement:** Unlike predictive models focused solely on forecasts, NeuroStockViz engages users actively by visually demonstrating real-time market dynamics and correlations, empowering users to derive their insights and hypotheses.

Novel Contributions of NeuroStockViz:

- **Neural Network-Inspired Structure:** A unique visualization approach leveraging neural network principles to intuitively represent complex market correlations.
- **Real-Time Interactive Exploration:** Extensive interactivity enabling personalized and dynamic exploration of stock correlations.
- **Educational Impact:** Facilitates greater comprehension and learning among financial analysts, students, and investors, bridging the gap between advanced data visualization and practical financial analysis.

By integrating intuitive visualizations, interactive dynamics, and educational insights, NeuroStockViz sets itself apart as a comprehensive, innovative tool for understanding and exploring financial market correlations.

Contributions

Our work contributes to the existing landscape of financial data visualization and analysis in several innovative ways:

1. **Neural Network-Inspired Visualization**:
   We introduce a novel visualization paradigm by structuring stock correlations as neural network-like graphs. This intuitive representation clearly illustrates complex market relationships, providing deeper analytical insights beyond traditional correlation matrices or charts.

2. **Interactive Real-Time Analytical Tool**:
   NeuroStockViz is an interactive, web-based platform allowing dynamic exploration of stock market correlations. Users can effortlessly identify influential stocks and correlation clusters in real-time, significantly enhancing analytical depth and decision-making efficiency.

3. **Educational and Practical Utility**:
   Designed with financial education and professional use in mind, our tool offers an accessible way for finance students, analysts, and investors to understand intricate stock market dynamics, facilitating both theoretical learning and practical investment analysis.

Requirement Analysis

Functional Requirements:

These requirements define the essential functions and features the NeuroStockViz tool must deliver.

1. **Visualization of Neural Network Structure**:
   - Display stock correlations visually as nodes (representing individual stocks) and edges (representing correlation strengths).
   - Each node must be labeled clearly with stock ticker symbols for easy identification.
   - Edges visually indicate correlation strength (e.g., thickness or opacity corresponding to Pearson's coefficient values).
   - Interactive elements enable users to hover over or select nodes and edges to display detailed correlation information and historical stock data.

2. **Interactive Stock and Sector Selection**:
    - Allow users to select an individual stock to serve as the primary focal point of the analysis.
    - Provide a filter to select specific industry sectors, enabling users to explore intra-sector and inter-sector correlations.
    - Automatically update the network visualization based on user selections, clearly differentiating correlations within and between sectors.

3. **Dynamic Real-time Updates Based on User Interaction**:
    - Enable users to adjust the correlation threshold, updating displayed correlations dynamically in real time to reflect these changes.
    - Allow the selection of different historical timeframes (e.g., last month, last quarter, last year) to observe evolving correlation patterns.
    - Seamless transition and instantaneous visual updates upon modifying interactive elements, providing a fluid user experience.

4. **Complementary Analytical Views**:
    - Integrate additional views such as the mini time-series charts, providing historical price trends for selected stocks.
    - Provide pairwise correlation heatmaps as a secondary visualization to give users alternative perspectives and validate findings from the neural network structure.

Non-functional Requirements:

These requirements ensure that NeuroStockViz is user-friendly, efficient, reliable, and secure.

1. **Responsive and Intuitive User Interface**:
    - Ensure compatibility with various screen sizes and devices (e.g., desktops, laptops, tablets).
    - Design intuitive navigation, clearly separating control panels (selection of stocks, sectors, thresholds, etc.) from visualization views.
    - Prioritize user-friendliness by reducing visual clutter, employing tooltips, legends, and clear instructions to guide users smoothly through interactions.

2. **Smooth Integration Between Frontend and Backend**:
    - Establish efficient communication protocols (e.g., RESTful APIs) between the front-end visualization built with D3.js and the backend Flask application.
    - Ensure asynchronous data fetching and processing to avoid latency and maintain real-time responsiveness during user interactions.
    - Implement robust error-handling mechanisms to gracefully manage data fetch failures or unexpected inputs without disrupting user experience.

3. **Accurate and Timely Correlation Data Processing**:

    - Utilize robust methods (Pearson's correlation coefficient) for correlation calculations, ensuring analytical precision and reliability.
    - Set up automated data retrieval and processing pipelines using Yahoo Finance APIs with periodic updates to keep correlation data accurate and current.
    - Optimize backend computations (potentially leveraging caching or efficient data structures) to support smooth, responsive updates to visualization elements even under high interactivity loads.

4. **Performance and Scalability**:
    - Ensure rapid visualization rendering and interaction response times, ideally within milliseconds to seconds, even with a large number of stocks.
    - Ensure scalability to accommodate future extensions to include additional indices (e.g., NASDAQ-100 or S&P 500), more extended historical periods, or larger datasets without significant degradation in performance.

5. **Security and Data Privacy**:
    - Adhere to data privacy standards and ensure secure transmission of data between front-end and back-end components (using secure HTTPS protocols).
    - Protect backend endpoints to prevent unauthorized access or misuse, maintaining overall system security.

System Overview

The NeuroStockViz system comprises two core components:

- **Frontend**: Built using HTML, CSS, JavaScript (D3.js) for interactive visualization.

- **Backend**: Python Flask backend that fetches and processes financial data using Yahoo Finance (yfinance) API and calculates correlations using pandas.

Algorithm Design

The main algorithm involves:

- Fetch historical stock data from Yahoo Finance.

- Calculating Pearson's correlation coefficient for pairs of stocks.

- Creating a correlation matrix.

- Structuring the correlations into a graph data structure, with stocks as nodes and correlation coefficients as weighted edges.

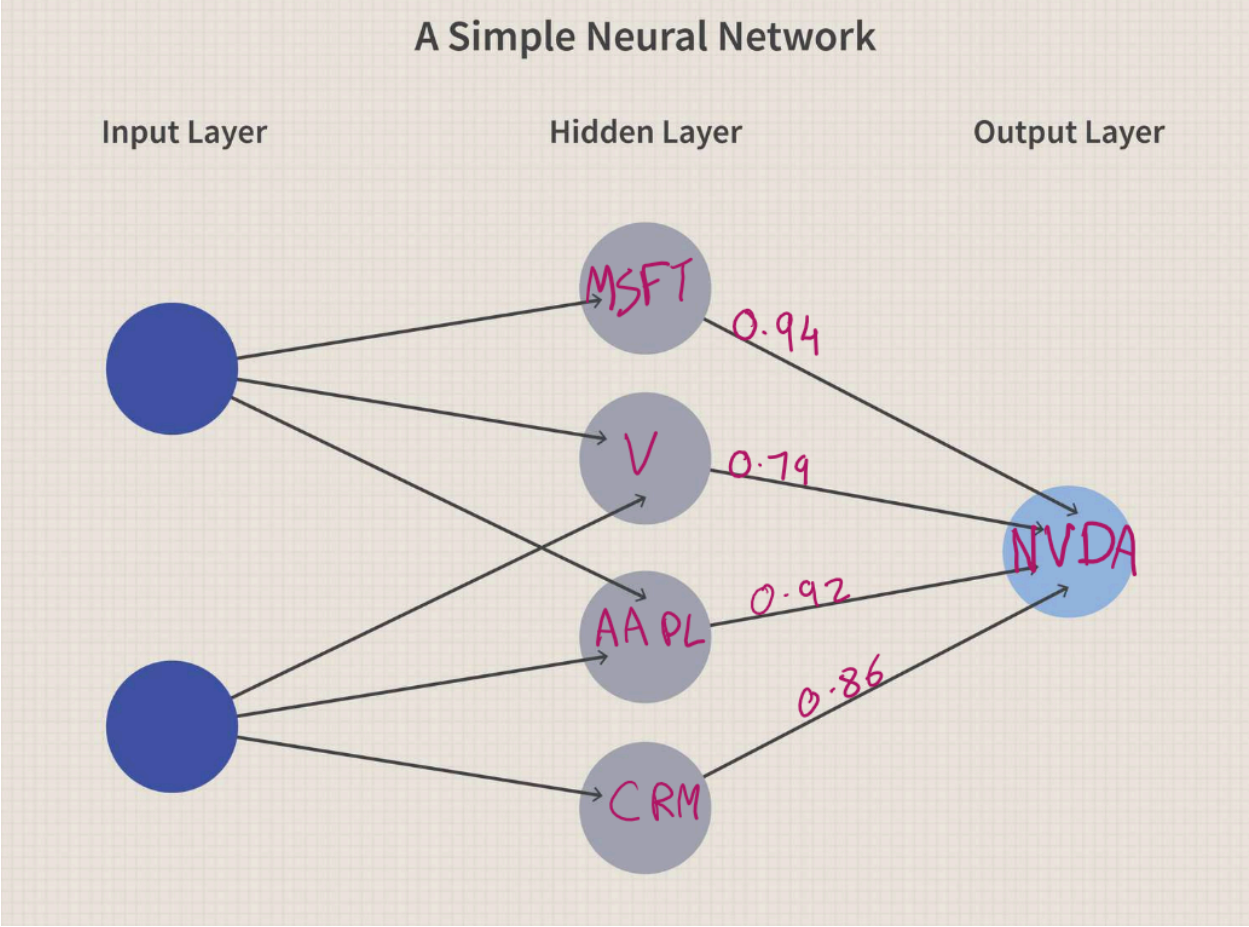
- Dynamically updating and rendering the graph based on frontend filters.

Interface/Interaction Design

Based on the low-fi prototype:

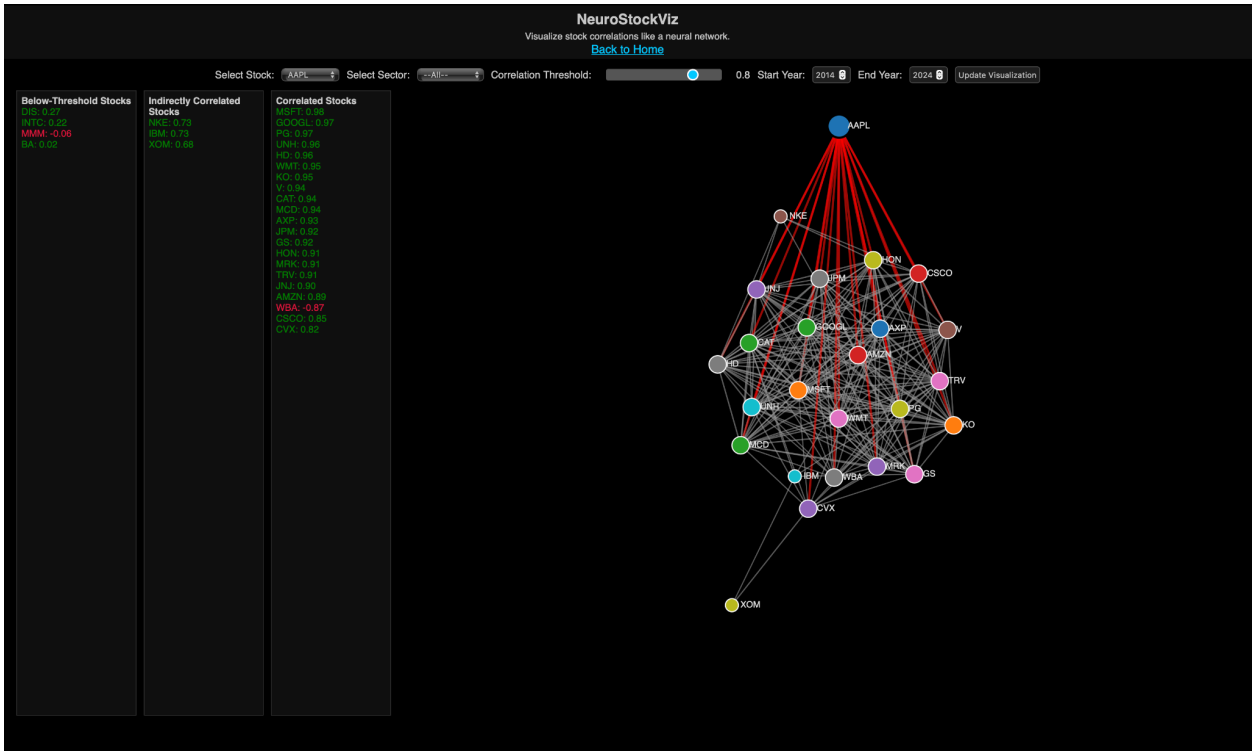
- **Input Selection Panel**: Users select the primary stock, sector, year range, and correlation threshold.
- **Visualization Panel**: Neural network-style interactive visualization; hovering reveals correlations and connected stocks; dynamically adapts based on user selection.
- **Time-Series Mini Chart**: Additional detailed view for selected stocks' historical performance.
- **Heatmap of Pairwise Correlations**: This offers another perspective on stock interactions.

Initial Low-Fi Prototype:



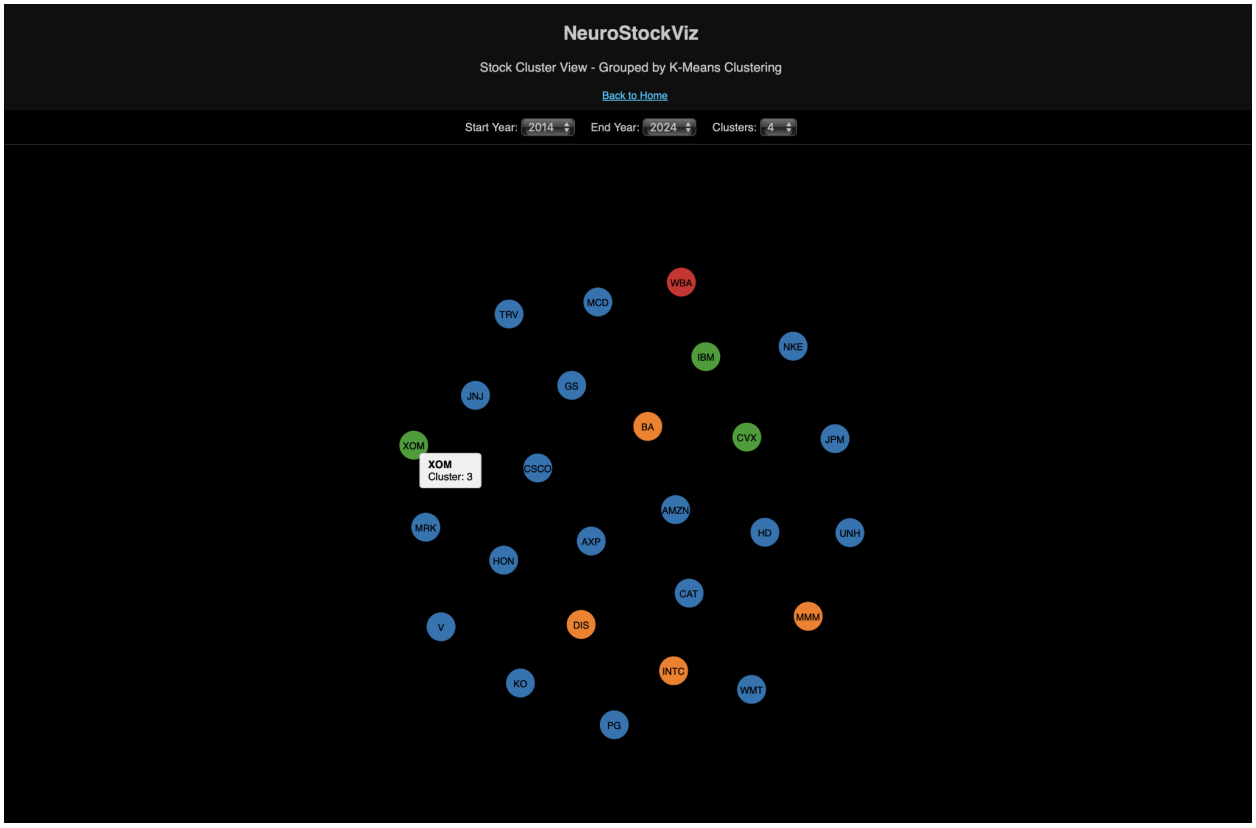
Actual Product:

Network visualization

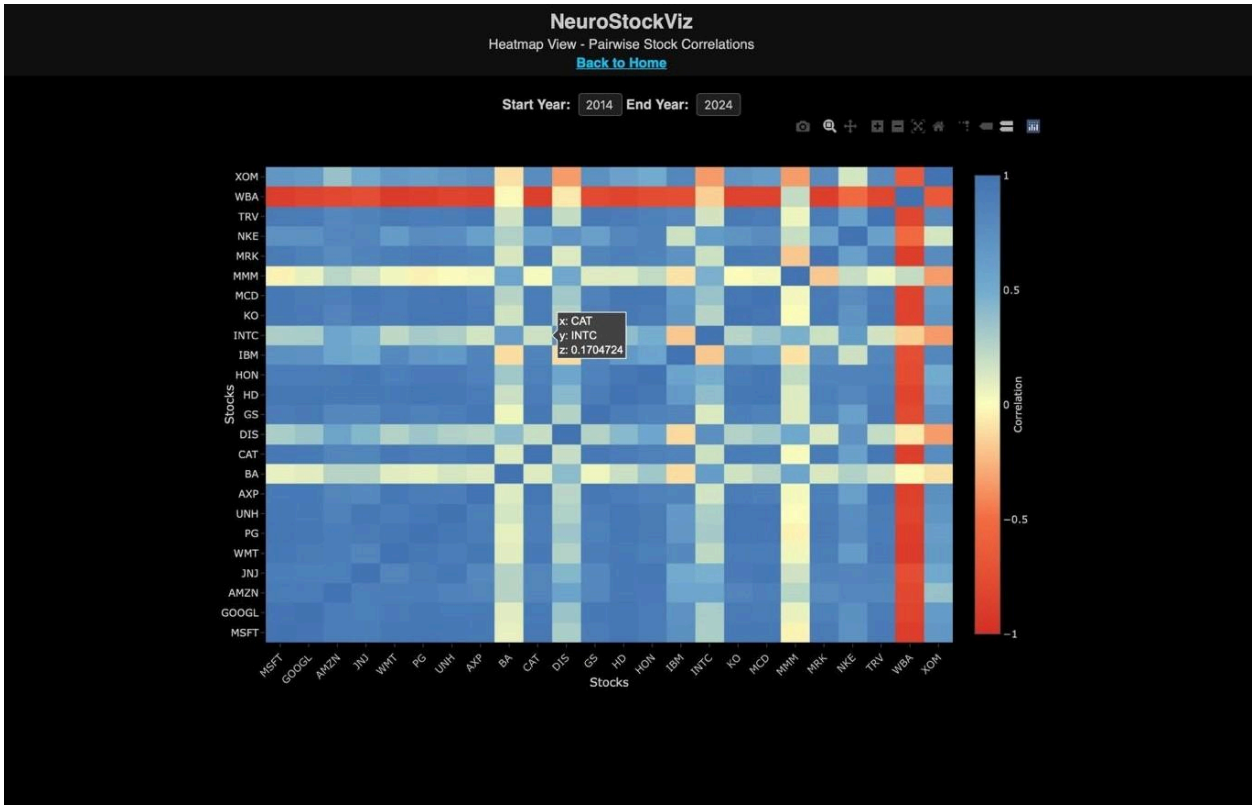


Candlestick Charts

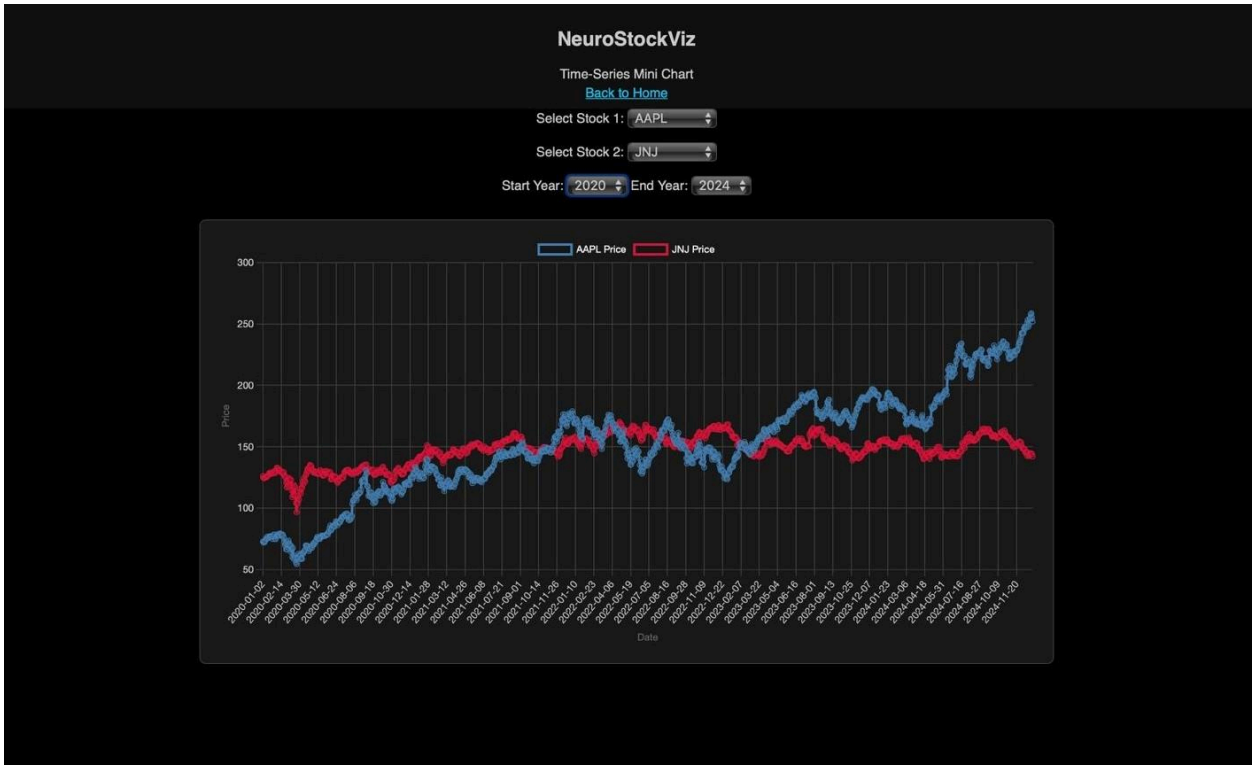
K-cluster



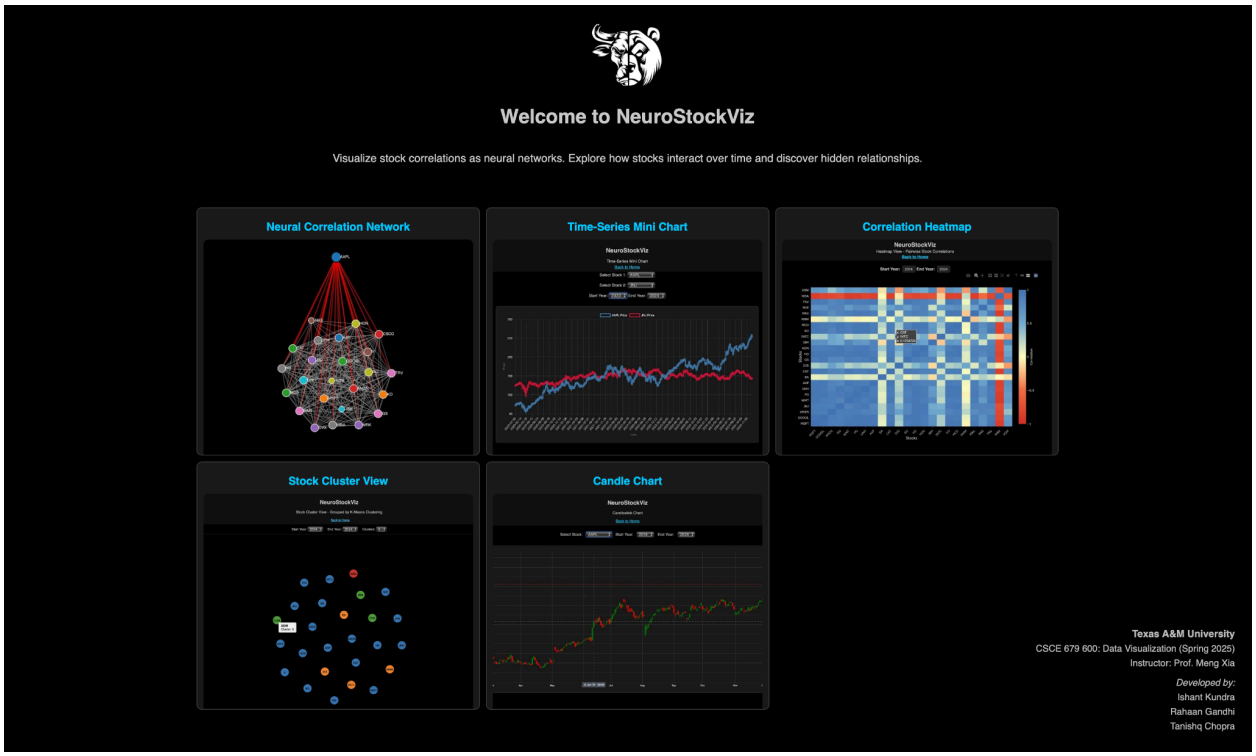
Heatmap



Multi-Line Chart



Home Page



Key Functions and Their Purposes:

Imports and Initial Setup

Purpose: Imports necessary libraries and initializes the Flask application, caching configuration, and stock data.

```python
from flask import Flask, render_template, request, jsonify, redirect
import yfinance as yf
import pandas as pd
from datetime import datetime
from sklearn.cluster import KMeans
from flask_caching import Cache

app = Flask(__name__)
cache = Cache(app, config={'CACHE_TYPE': 'SimpleCache'})
```

Stock and Sector Configuration

Purpose: Defines Dow Jones stock tickers and their corresponding sectors.

```python
# Dow Jones 30 components
stocks = [
    "AAPL", "MSFT", "GOOGL", "AMZN", "JNJ", "V", "WMT", "JPM", "PG", "UNH",
    "AXP", "BA", "CAT", "CSCO", "CVX", "DIS", "GS", "HD", "HON", "IBM",
    "INTC", "KO", "MCD", "MMM", "MRK", "NKE", "TRV", "WBA", "XOM", "DOW"
]

# Example mapping of stock tickers to sectors (you can load from CSV if needed)
stock_sectors = {
    "AAPL": "Technology",
    "MSFT": "Technology",
    "GOOGL": "Technology",
    "AMZN": "Consumer",
    "JNJ": "Healthcare",
    "V": "Financial",
    "WMT": "Consumer",
    "JPM": "Financial",
    "PG": "Consumer"
```

Functions:

`get_stock_data(ticker, start_date, end_date)`

Purpose: Fetches stock data from Yahoo Finance. Handles multi-index columns and ensures

correct column selection (Adj Close or Close).
Use: Crucial for accurate data fetching for correlation computation.

```python
def get_stock_data(ticker, start_date, end_date):
    data = yf.download(ticker, start=start_date, end=end_date, progress=False, auto_adjust=False, group_by="ticker")
    # Check if data has multi-level columns
    if isinstance(data.columns, pd.MultiIndex):
        if 'Adj Close' in data[ticker]:
            return data[ticker]['Adj Close']
        elif 'Close' in data[ticker]:
            return data[ticker]['Close']
    else:
        if 'Adj Close' in data.columns:
            return data['Adj Close']
        elif 'Close' in data.columns:
            return data['Close']
    raise KeyError(f"'Adj Close' and 'Close' columns are missing for {ticker}")
```

`compute_correlations(start_date, end_date, filtered_stocks)`

Purpose: Computes the Pearson correlation matrix among stocks within a selected period. Handles data preprocessing, filtering, and missing data.
Use: Core logic for generating correlation-based visualizations.

```python
def compute_correlations(start_date, end_date, filtered_stocks):
    df = pd.DataFrame()
    for ticker in filtered_stocks:
        try:
            df[ticker] = get_stock_data(ticker, start_date, end_date)
        except Exception as e:
            print(f"Error fetching data for {ticker}: {e}")      You, last week

    # Keep only stocks (columns) that have at least 80% non-null values
    df = df.dropna(axis=1, thresh=int(0.8 * len(df)))

    # Now drop any remaining rows with NaNs
    df.dropna(inplace=True)

    if df.shape[1] < 2:
        return pd.DataFrame()  # Not enough stocks left to compute correlations

    return df.corr(method='pearson')
```

Caching Decorator: `Compute_correlations_cached`

Purpose: Caches the correlation computation results for 5 minutes to improve performance and avoid redundant computations.

```python
@cache.cached(timeout=300, key_prefix='correlation_matrix')
def compute_correlations_cached(start_date, end_date, filtered_stocks):
    return compute_correlations(start_date, end_date, filtered_stocks)
```

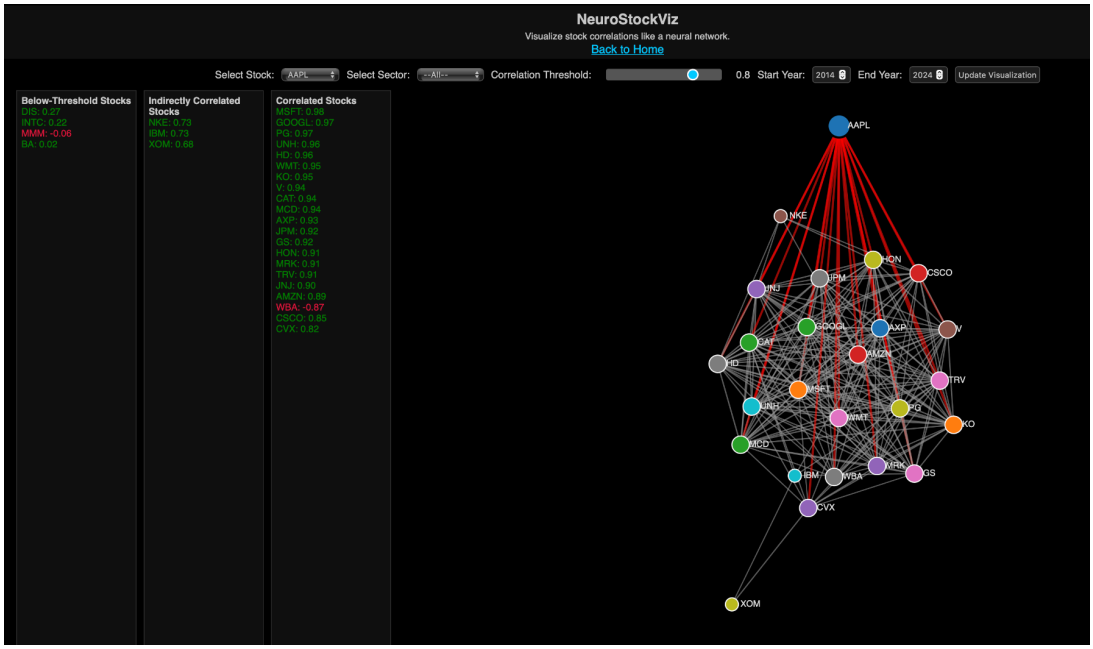
Route: Main and Navigation Routes
Purpose: Defines Flask routes to render different HTML templates/pages:

- `/home`: Home page
- `/visualization`: Correlation network visualization
- `/clusters`: Stock cluster visualization
- `/api/candlestick:` Provides data for rendering candlestick charts for a given stock within specified dates. Handles data downloading, preprocessing, and JSON formatting for front-end use.

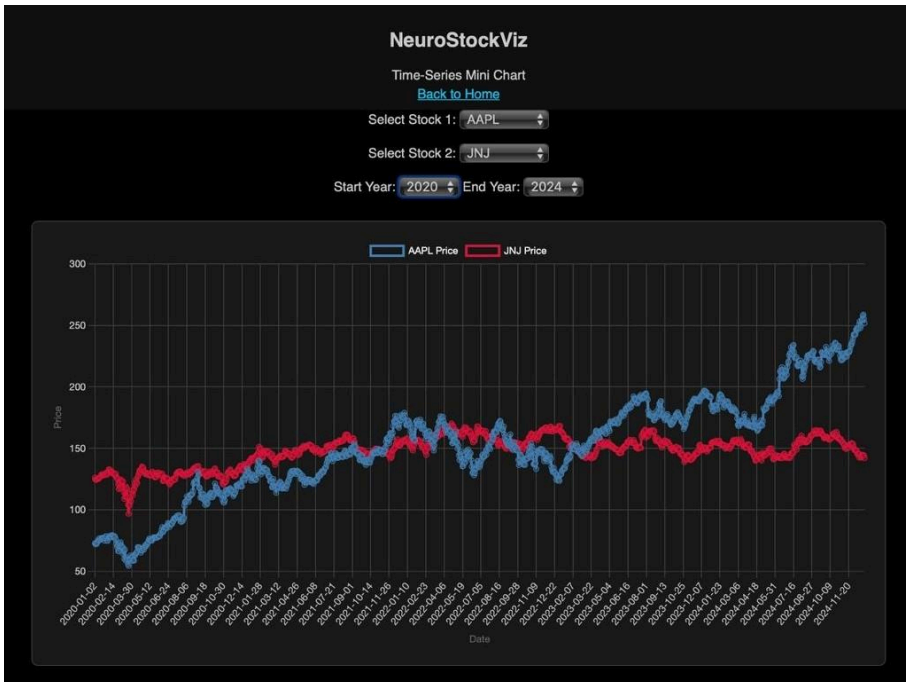
```python
@app.route('/')
def index():
    return redirect('/home')

@app.route('/home')
def home():
    return render_template('home.html')

@app.route('/visualization')
def visualization():
    return render_template('index.html')

@app.route('/clusters')
def cluster_view():
    return render_template('clusters.html')

@app.route('/api/candlestick')
def candlestick_data():
    ticker = request.args.get('ticker')
    start_year = request.args.get('start', '2014')
    end_year = request.args.get('end', '2024')

    start_date = f"{start_year}-01-01"
    end_date = f"{end_year}-12-31"
```

Screenshots of Actual Product:

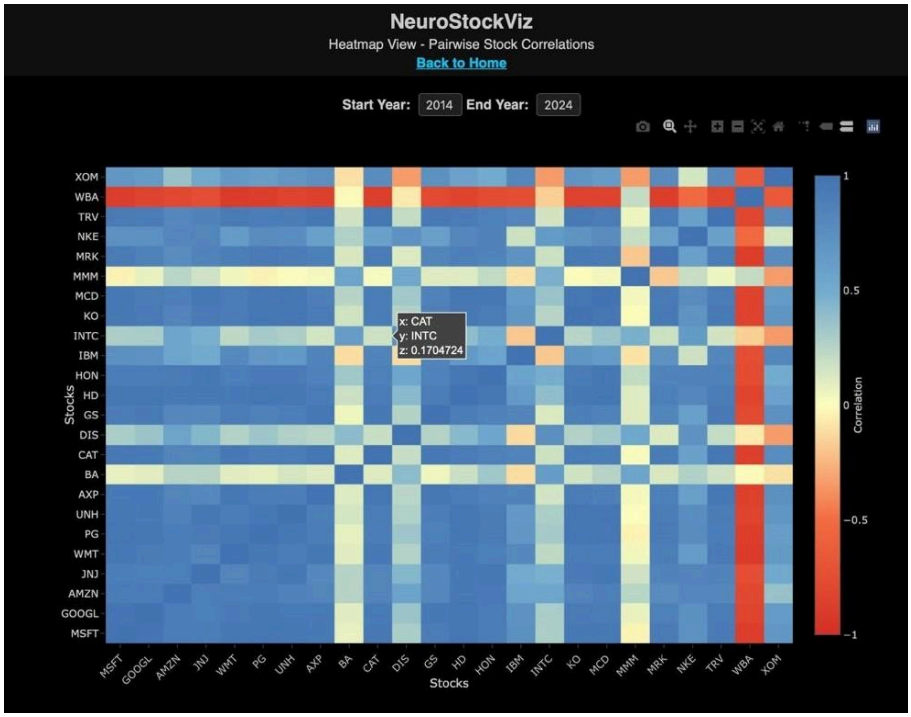
- The primary neural network visualization.



- Time-series mini-chart



- Correlation heatmap views.



Usage:

1. Main Dashboard (Homepage)

Use Case:

- **Overview and Navigation:** Investors, financial analysts, or students would use this main dashboard as a central hub. It visually summarizes available analytical tools and helps users quickly navigate to specific analysis types such as correlation networks, time-series charts, correlation heatmaps, clustering views, and candlestick charts. This ensures effortless exploration and intuitive access to detailed analytics.

2. Neural Correlation Network Visualization

Use Case:

- **Identifying Influential Stocks and Correlations:** Users such as portfolio managers and financial analysts can leverage this neural network visualization to explore complex relationships between stocks intuitively. By selecting a stock and adjusting the correlation threshold, users can visually identify the most influential stocks and their strongest correlations, thus uncovering hidden market patterns and dependencies to inform investment decisions or market predictions.

3. Time-Series Mini Chart

Use Case:

- **Historical Performance Comparison:** Individual investors or market researchers can utilize this feature to compare historical stock performance. Selecting two stocks and viewing their historical price trends over custom time intervals allows users to analyze comparative performance, identify divergence or convergence in price trends, and better understand temporal correlation patterns.

4. Correlation Heatmap

Use Case:

- **Broad Market Analysis:** This heatmap is particularly beneficial for quantitative analysts and researchers aiming to quickly identify and analyze pairwise stock correlations at scale. Users can rapidly detect clusters of highly correlated stocks or identify outliers that behave distinctly, which helps in portfolio diversification strategies, risk assessment, and strategic asset allocation.

5. Stock Cluster View (K-Means Clustering)

Use Case:

- **Sector and Behavioral Grouping:** Investors and academic researchers can utilize this clustering feature to group stocks based on historical price movement patterns. By employing K-Means clustering, users can uncover distinct groupings, analyze sector behaviors, and pinpoint groups of stocks that react similarly to market events, assisting in targeted investment strategies and market segmentation analyses.

6. Candlestick Chart

Use Case:

- **Detailed Price Movement Analysis:** Traders, analysts, and investors frequently utilize candlestick charts for precise and detailed insights into price movements, trends, support/resistance levels, and potential reversal signals. Selecting individual stocks and specific timeframes enables detailed, actionable insights that can directly inform short-term trading decisions and longer-term investment strategies.

Evaluation

Strategy:

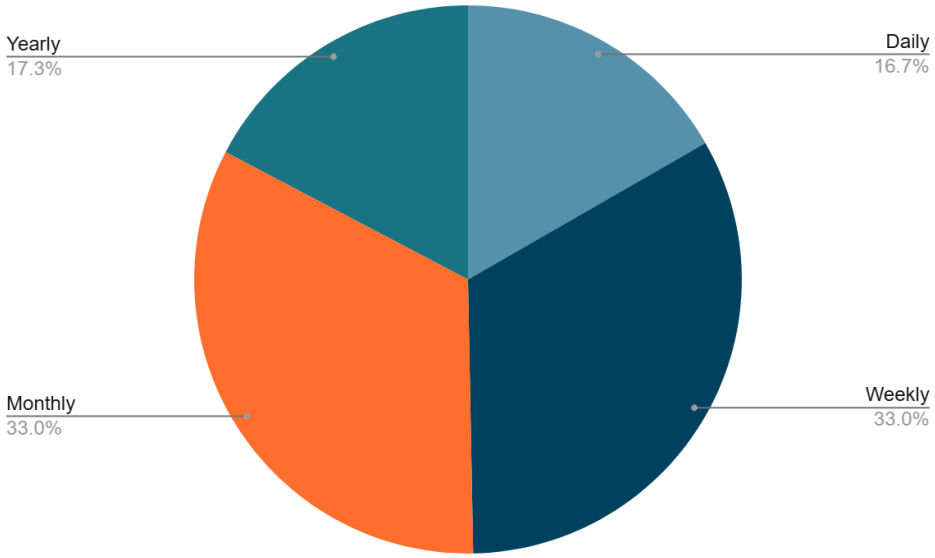
- Conduct usability testing with 10-15 users (financial analysts, students, investors).
- Feedback criteria: Intuitiveness, effectiveness, and clarity of visualization.
- Generate pie charts to visualize collected feedback data (Positive, Neutral, Negative responses).

Example evaluation aspects:

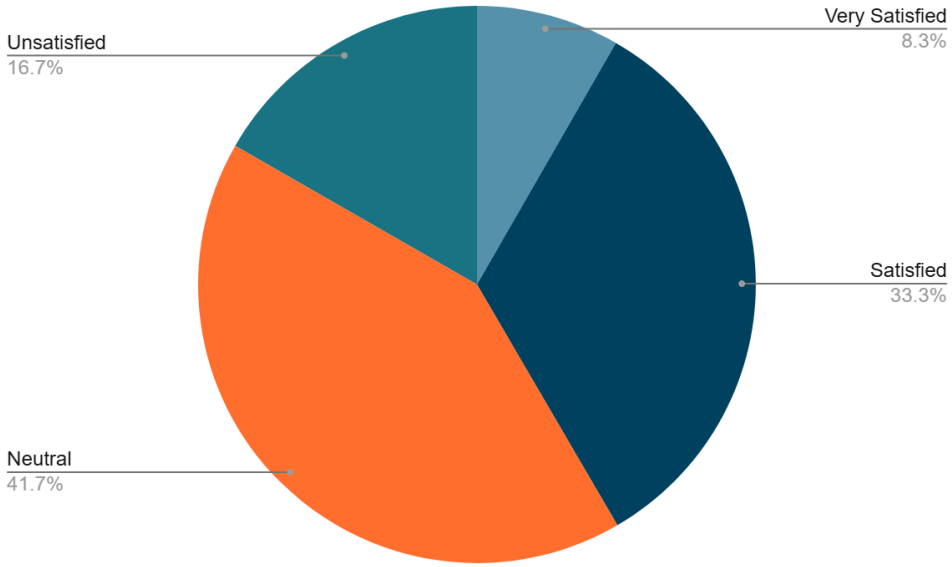
- Ease of Use
- Intuitiveness of Visualization
- Clarity of Information Displayed
- Relevance and Accuracy of Insights Provided

Survey Results:

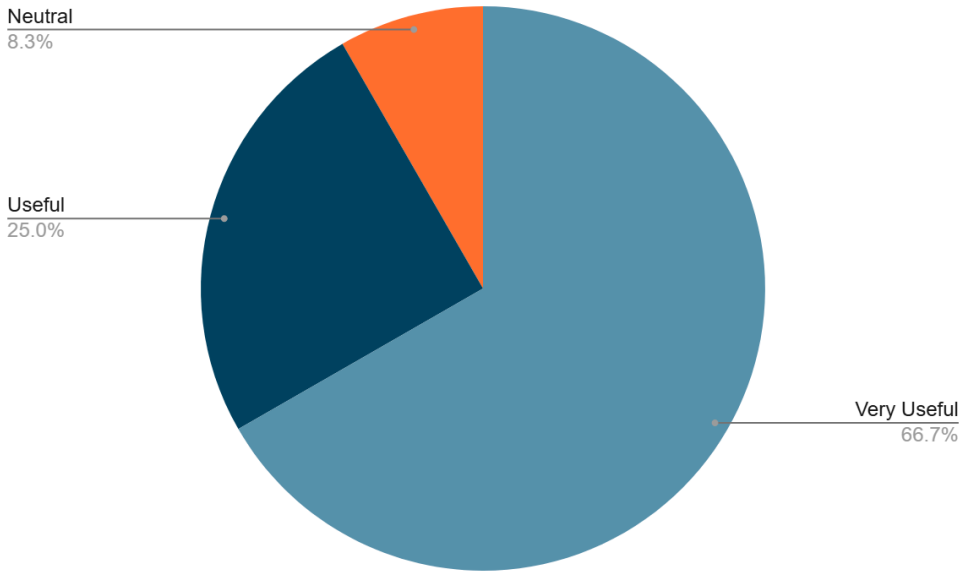
Frequency of Stock Correlation Analysis:



Satisfaction with Existing Tools:



Interest in Neural Network-Style Visualization:



Future Work

- Enable exporting visualizations as PNG/PDF.
- Implement historical comparison tables for performance benchmarking.
- Expand dataset coverage to include broader indexes such as S&P 500 or NASDAQ-100 stocks.

Group Members' Work Allocations

- **Ishant Kundra**:
    - Data Engineering, backend data pipeline.
    - Predictive insights, network modeling.

- **Rahaan Gandhi**:
    - Correlation calculation and exploratory analysis.
    - Interface and visualization interactivity development.

- **Tanishq Chopra**:
    - Descriptive analysis of stock correlations.
    - Frontend visualization implementation and UI enhancements.

Reflections

- **Technical Reflection**:
    - Challenges faced integrating frontend visualization with backend computation efficiently.
    - Improvements achieved in responsiveness through asynchronous data handling.

- **Team Dynamics Reflection**:
    - Effective collaboration strategies adopted.
    - Lessons learned from iterative design based on user feedback.

- **Project Management Reflection**:
    - Successes and shortcomings in adherence to the project timeline.
    - How task division impacted project progress positively or negatively.

References

1. Portfolio Visualizer. (n.d.). **Asset Correlations**. Retrieved April 9, 2025, from https://www.portfoliovisualizer.com/asset-correlations?utm_source=chatgpt.com

2. DileQuante. (n.d.). **Graph Theory in Finance: An Alternative Way to Visualize Stocks' Correlations**. Retrieved April 9, 2025, from https://dilequante.com/an-alternative-way-to-visualize-stocks-correlations/?utm_source=chatgpt.com

3. Stanford CS224W. (2022). **Stock Market Forecasting with Differential Graph Transformer**. Medium. Retrieved April 9, 2025, from https://medium.com/stanford-cs224w/stock-market-forecasting-with-differential-graph-transformer-62d095ebc821