Visualizing Relationships Between Economic Markets

Gabriel Keith
Computer Science
University Colorado Boulder
Boulder CO United States
gake9229@colorado.edu
Student ID: 109577570
Course: CSCI 4502-001

Robert Hellums
Computer Science
University Colorado Boulder
Boulder CO United States
rohe5584@colorado.edu
Student ID: 109478647
Course: CSCI 4502-001

Sarthak Shukla
Computer Science
University Colorado Boulder
Boulder CO United States
sash9861@colorado.edu
Student ID: 109612817
Course: CSCI 4502-001

Lucas Derr
Computer Science
University Colorado Boulder
Boulder CO United States
lude5778@colorado.edu
Student ID: 109609052
Course: CSCI 5502-001

ABSTRACT

This project will explore the effect of different economic markets on each other using a Correlation Graph format and a Clustering Analysis format. The goal is to mine time-series data from several economic markets, starting from the United States' major stock markets, bond markets, commodities markets, and an economic performance index. This data will be used to build a correlation matrix representing correlations between each market, as well as optimally tuned clustering models to organize the data into clusters. A correlation graph will be built using this matrix. These data sets will be considered nodes of the graph while the correlation score between each node will be the edges of the graph. Using a graph-based model will allow visualization of a network of datasets. Visualizing these data sets will give a specific insight not found in other models. Correlation graphs can easily represent the strength of connections between nodes and using this technique on several economic markets will allow quick analysis of these relationships. Alongside the correlation graph, clusters of data sets will be generated using cluster analysis techniques. This serves to affirm the conclusions and information discerned from the correlation graph, and possibly generate new insights. This clustering analysis was accomplished using optimally tuned k-means and agglomerative clustering algorithms across several relevant python libraries. As a result of both the network and clustering analysis, the following insights were generated: an economic performance index metric, various network visualizations, an overlay of clusters in the dataset, and a dendrogram displaying the hierarchical relationships of the dataset.

INTRODUCTION

For individual investors it is important to understand when one's investment is at risk, quickly and easily without data analysis skills. Changes in certain markets can affect markets where one's investment may lie. This project's goal is to build an easily understandable visualization tool that presents the correlations between markets to everyday investors.

For this project to succeed several challenges must be overcome. The initial gathering of data sets to build the graph requires access to information and data from several stock, bond, and commodities markets. Additionally to acquire a general economic indicator, United States unemployment, GDP, inflation, and budget deficit data must be gathered. Following data acquisition, this project must successfully integrate each data set into a standard format that can be used to generate correlation vectors. Successfully finding a clustering method for all the data sets will need to be accomplished. Finally and most importantly, conclusions and insights must be garnered from the visualization and clustering of data.

The project relies heavily upon the key concept that information can be gained from the correlation of economic markets. We also assume that economic markets have clusters of highly correlated markets. Developing this correlation graph gives a general guide to market relationships.

RELATED WORK

1 Relation between Financial Market Structure and Real Economy: Comparison between Clustering Methods

This paper, written by Nicoló Musmeci, Tomaso Aste, and T. Di Matteo in 2015, compares the results of different clustering methods with the Directed Bubble Hierarchical Tree (DBHT) on N=342 stocks from the New York Stock Exchange. Each dataset includes daily closing prices from 1997-2012. The goal of using clustering methods on the stock market data is to create a tree structure that arranges each stock into a hierarchical structure. Different methods, such as Minimum Spanning Tree and Planar Maximally Filtered Graph can be applied to the tree structure to give greater insight into correlations between different stocks and potentially improve portfolio performance.

The proposed clustering methods that are compared to the DBHT are Single Linkage, Average Linkage, Complete Linkage, and K-medoids. There are three factors that are used in evaluating the clusters: measuring heterogeneity using disparity, measuring clustering similarity using the Adjusted Rand Index, and measuring cluster-industry overexpression using a hypergeometric test.

From the results, the Average Linkage, Complete Linkage, K-medoids, and DBHT all had relative homogeneity in regard to cluster size. The Simple Linkage was the only algorithm that had a strong heterogeneity in cluster size, with a single giant cluster containing 318 stocks (90% of all stocks). In addition, the three clusters with the highest adjusted Rand Index were DBHT (0.419), k-medoids (0.387), and Complete Linkage (0.387). The authors conclude that these three clustering methods outperform the other two when varying the number of clusters. When comparing the results of the clusters compared to their respective Industrial Classification Benchmark subsectors, the results are largely the same: DBHT, K-medioids, and Complete Linkage outperform the other two methods.

Our clustering analysis differs from those shown in this article due to a vastly different dataset and in the evaluation of the success of the clusters. While our dataset contains 29 indices from different economic markets, their dataset contains 342 stocks from the New York Stock Exchange. In addition, while our dataset runs from 2012-2022, theirs is from 1997-2012. This means the results of the analysis will be completely different due to datasets which have nothing in common. In addition, the study used heterogeneity, similarity, and overexpression while we elected to use inertia and silhouette score to measure the 'goodness' of each model.

2 The Economic Performance Index (EPI): an Intuitive Indicator for Assessing a Country's Economic Performance Dynamics in an Historical Perspective

In this paper, written by Vadim Khramov and John Ridings Lee in 2013, the authors formulate and evaluate for accuracy an intuitive indicator for assessing a country's economic performance. The aforementioned indicator is named the Economic Performance Index (EPI). This indicator looks to reflect the economy's performance by evaluating the economy's three main sectors: households, firms and government. This score is calculated by plugging inflation rate, unemployment rate, budget deficit as a percentage of GDP, and change in real GDP into a 'raw' and 'weighted' variation of a formula designed to generate an EPI score. In this paper the EPI score was generated and compared to historical economic events to verify the validity of the calculated score. In our own project we will use this methodology to calculate an Economic Performance Index for our own uses over a customized time series.[1]

METHODOLOGY

1 Cleaning and Piping Economic Health Data

The process of piping the economic health data was handled using the pandas python library. Initially all economic data was stored locally as csv files. While some of the data was available to download as a csv file, others were only available as a xls file and had to be converted to a csv after downloading. Sources for this data are as follows:

1. Inflation Rate (Monthly): https://beta.bls.gov/dataViewer/view/timeseries/CUUR0000SA0
2. Unemployment Rate (Monthly): https://data.bls.gov/cgi-bin/surveymost?bls (From link you must select: Unemployment Rate (Seasonally Adjusted) - LNS14000000)
3. GDP (Quarterly): https://apps.bea.gov/iTable/?reqid=19&step=2&isuri=1&categories=survey#eyJhcHBpZCI6MTksInN0ZXBzIjpbMSwyLDMsM10sImRhdGEiOltbImNhdGVnb3JpZXMiLCJTdXJ2ZXkiXSxbIk5JUEFfVGFibGVfTGlzdCIsIjUiXSxbIkZpcnN0X1llYXIiLCIxOTI5Il0sWyJMYXN0X1llYXIiLCIyMDIyIl0sWyJTY2FsZSIsIi05Il0sWyJTZXJpZXMiLCJRIl1dfQ==
4. Budget Deficit (Monthy 2015-2022): https://fiscaldata.treasury.gov/datasets/monthly-treasury-statement/summary-of-receipts-outlays-and-the-deficit-surplus-of-the-u-s-government

5. Budget Deficit (Monthly 1998-2022): https://www.fiscal.treasury.gov/reports-statements/mts/previous.html
6. GDP % Change (Monthly): https://data.bls.gov/timeseries/CUUR0000SA0L1E?output_view=pct_12mths

Using pandas, the stored csv files were converted into pandas data frames. Excess data and headers were discarded and each data frame was standardized to only hold the needed values and indexed by date. For this data we desired a data granularity of daily; however, as listed above, all available data for these economic indicators was either released monthly or quarterly. To resolve this issue, we created and inserted into the data frames a row associated with every date in between when data was released. From here we used a b-spline function to interpolate and fill all the remaining dates. Spline functions are piecewise polynomial functions commonly used in interpolation problems. The b-spline function used to interpolate in this case was a cubic spline function. Once all values were filled in, the data frames were cropped to match the greatest time series for which all data frames had values. Once all data frames were aligned they were joined together. Once this combined data frame was created we were prepared to calculate an EPI score.

2 Use Khramov and Lee's Methodology to Calculate EPI for Relevant Economies

Included in our network is a node that represents the economic performance of the United States over the same time series as our economic market data. This node reflects economic performance for the United States by using The Economic Performance Index (EPI). The Economic Performance Index was calculated by us for our desired time series using Vadim Kharamov and John Riding's methodology as outlined in their paper *The Economic Performance Index (EPI): an Intuitive for Assessing a Country's Economic Performance Dynamics in a Historical Perspective* [1]. This paper describes how to calculate the EPI using the following equation:

$$EPI = 100 - |Inflation\ Rate| - Unemployment\ Rate - \frac{Budget\ Deficit}{GDP} + \%Change\ in\ Real\ GDP$$

Using the data frame containing cleaned and piped economic health data described in **Cleaning and Piping Economic Health Data** we plugged the values contained in each individual column into the equation described above to calculate the daily EPI score values. These values then populated a final column representing the EPI score for each day. Finally all columns except the EPI score column were dropped leaving a data table containing the EPI scores indexed by date.

3 Cleaning and Piping Economic Market Data

The process of piping the economic market data was handled using the pandas python library. Initially all economic data was stored locally as csv files. Sources for these csv files are as follows:

1. NASDAQ Composite Index: https://www.wsj.com/market-data/quotes/index/COMP/historical-prices
2. Dow Jones Industrial Average: https://www.wsj.com/market-data/quotes/index/DJIA/historical-prices
3. Russell 2000 Index: https://www.wsj.com/market-data/quotes/index/RUT/historical-prices
4. S&P 500 Index: https://www.wsj.com/market-data/quotes/index/SPX/historical-prices
5. S&P U.S. Treasury Bond Index: https://www.spglobal.com/spdji/en/indices/fixed-income/sp-us-treasury-bond-index/#overview
6. S&P U.S. Treasury Bill Index: https://www.spglobal.com/spdji/en/indices/fixed-income/sp-us-treasury-bill-index/#overview
7. S&P Municipal Bond Index: https://www.spglobal.com/spdji/en/indices/fixed-income/sp-municipal-bond-index/#overview
8. S&P 500 Bond Index: https://www.spglobal.com/spdji/en/indices/fixed-income/sp-500-bond-index/#overview
9. Dow Jones Aluminum Commodity Index: https://www.spglobal.com/spdji/en/indices/commodities/dow-jones-commodity-index-aluminum/#overview
10. Dow Jones Gold Commodity Index: https://www.spglobal.com/spdji/en/indices/commodities/dow-jones-commodity-index-gold/#overview
11. Dow Jones Lead Commodity Index: https://www.spglobal.com/spdji/en/indices/commodities/dow-jones-commodity-index-lead/#overview
12. Dow Jones Nickel Commodity Index: https://www.spglobal.com/spdji/en/indices/commodities/dow-jones-commodity-index-nickel/#overview
13. Dow Jones Silver Commodity Index: https://www.spglobal.com/spdji/en/indices/commodities/dow-jones-commodity-index-silver/#overview
14. Dow Jones Zinc Commodity Index: https://www.spglobal.com/spdji/en/indices/commodities/dow-jones-commodity-index-zinc/#overview

15. S&P GSCI Copper Commodity Index: https://www.spglobal.com/spdji/en/indices/commodities/sp-gsci-copper/#overview
16. Dow Jones Brent Crude Oil Index: https://www.spglobal.com/spdji/en/indices/commodities/dow-jones-commodity-index-brent-crude/#overview
17. Dow Jones Natural Gas Index: https://www.spglobal.com/spdji/en/indices/commodities/dow-jones-commodity-index-natural-gas/#overview
18. Dow Jones Petroleum Index: https://www.spglobal.com/spdji/en/indices/commodities/dow-jones-commodity-index-petroleum/#overview
19. S&P GSCI Biofuel Index: https://www.spglobal.com/spdji/en/indices/commodities/sp-gsci-biofuel/#overview
20. Dow Jones Commodity Cattle Index: https://www.spglobal.com/spdji/en/indices/commodities/dow-jones-commodity-index-all-cattle/#overview
21. Dow Jones Commodity Cocoa Index: https://www.spglobal.com/spdji/en/indices/commodities/dow-jones-commodity-index-cocoa/#overview
22. Dow Jones Commodity Coffee Index: https://www.spglobal.com/spdji/en/indices/commodities/dow-jones-commodity-index-coffee/#overview
23. Dow Jones Commodity Corn Index: https://www.spglobal.com/spdji/en/indices/commodities/dow-jones-commodity-index-corn/#overview
24. Dow Jones Commodity Cotton Index: https://www.spglobal.com/spdji/en/indices/commodities/dow-jones-commodity-index-cotton/#overview
25. Dow Jones Commodity Lean Hogs Index: https://www.spglobal.com/spdji/en/indices/commodities/dow-jones-commodity-index-lean-hogs/#overview
26. Dow Jones Commodity SoyBean Index: https://www.spglobal.com/spdji/en/indices/commodities/dow-jones-commodity-index-soybeans/#overview
27. Dow Jones Commodity Sugar Index: https://www.spglobal.com/spdji/en/indices/commodities/dow-jones-commodity-index-sugar/#overview
28. Dow Jones Commodity Wheat Index: https://www.spglobal.com/spdji/en/indices/commodities/dow-jones-commodity-index-all-wheat/#overview
29. EPI: Calculated in section: **Use Kramov and Lee's Methodology to Calculate EPI for Relevant Economies**

A custom data set class converted these files, and stored them locally in a pandas dataframe. Excess data was discarded and each data frame was standardized to only hold the needed values and indexed by date. For all markets that had multiple values we chose to use closing value or end of day value. The data sets were then re-indexed to fill in missing days such as weekends and holidays. The values were then padded with the last data value. This gave the effect of weekends holding the same value as the Friday closing value. The data sets were then cropped to a specific time frame and normalized using standard deviation and min-max methods. A custom graph class is given a list of economic markets to pipe and completes the previous process on every single dataset. The class then pipes the normalized data into a single dataframe. Using this data frame a correlation matrix is developed and passed into the correlation graph process.

4 Generation of Correlation Graph

Once the datasets have been piped, a correlation value between each will be generated and stored locally as a correlation vector in the form of a pandas data frame. The rows and columns in the correlation vector correspond to each market or economic index, while the cells contain the correlation value between the corresponding row and column.

To create the correlation graph itself we used the NetworkX python library. To define the network we needed a list of nodes, a list of edges, and the correlation vector itself. We converted the correlation vector pandas dataframe into a NumPy matrix and used the NetworkX "from_numpy_matrix" function to generate a network directly from the correlation vector. We then relabelled the nodes in the network as all the markets/economic indices in our correlation vector and weighted the edges between the nodes according to the correlation value.

Furthermore, the generation of this network results in a visualization that would be difficult to interpret as the number of nodes and edges create a very dense appearing network. To solve this issue we generated two more networks from the original correlation network: one to visualize positive correlations and one to visualize negative correlations. The method for creating a positive correlation network is very similar to creating a negative correlation network. For both positive and negative correlation networks, we used the original correlation network and removed edges that had a negative correlation to generate a positive correlation and vice versa to generate a negative correlation network. We also added the option of adding a minimum correlation threshold for positive and negative correlation networks. For the positive correlation network we chose to have a threshold of 0.7 because we noticed that correlation vector had significantly more positive correlations (that were on average above 0.7) than negative correlations. In terms of the negative correlation network we chose a minimum correlation threshold of -0.1, which allowed us to see the more significant negative correlations.

5 Visualization of Correlation Network

Once we have generated networks for the entire correlation network, positive correlation network, and negative correlation network, we can begin the visualization process. The visualizations were done with the NetworkX drawing python library. For the visualizations of all three networks we positioned the nodes in our visualization using the NetworkX spring layout. The spring layout allows us to position nodes using the Fruchterman-Reingold force-directed algorithm. The NetworkX documentation describes this layout as: “The algorithm simulates a force-directed representation of the network treating edges as springs holding nodes close, while treating nodes as repelling objects, sometimes called an anti-gravity force” [4]. We set the minimal distances between nodes in the network as 1.0, and mapped each node to its corresponding position. Moreover, in the visualization, each node’s circle size is variable depending on the degree of the node. The size of the node in a network is the degree of the node cubed, which allows us to see the node with the most edges to other nodes in each network much more easily. Lastly, since all the correlations should be between -1.0 and 1.0, the edges would all look the same making it difficult to interpret between stronger and weaker correlations. Therefore to solve this problem for the visualizations we scaled the correlation values for each network by adding 1.0 to each correlation value and then squaring that value. This allowed the length of the edges to be dependent on scaled correlation value, as well as making the edges colorings more intense based on how strong the correlation value is.

For the positive correlation network we chose to represent the edges with a matplotlib colormap that returns a gradient from green to blue. For the negative correlation network we chose to represent the edges with a matplotlib colormap that returns a gradient from purple to red. The whole correlation network uses a matplotlib colormap with gradients from purple to blue to green.

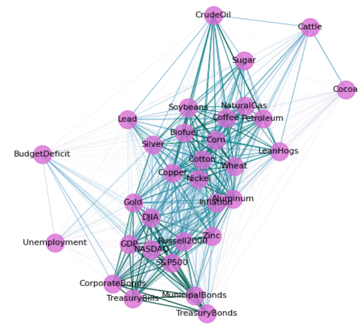

Figure 1: Visualization of Entire Correlation Network (number of nodes=32, number of edges=496)

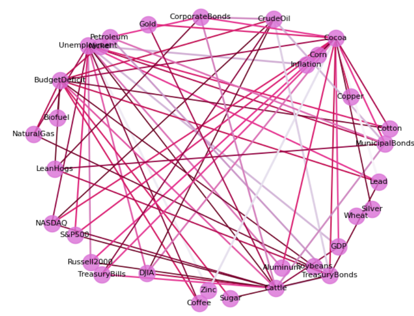

Figure 2: Visualization of Negative Correlation Network (min correlation=-0.1, number of nodes=32, number of edges=68)

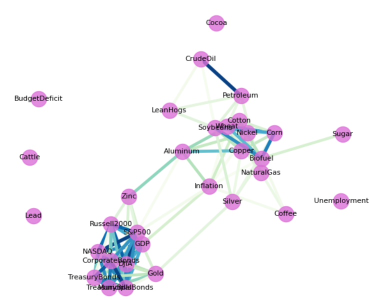

Figure 3: Visualization of Positive Correlation Network (min correlation=0.7, number of nodes=0.7, number of edges=101)

6 Clustering Analysis

In addition to generating the correlation network, our group opted to perform clustering analysis as a means of either reaffirming the insights generated from the network or generating new insights about the dataset.

The first and most important step of the analysis comes with deciding the appropriate machine learning algorithm and library to apply to the dataset. We elected to begin with tslearn's *TimeSeriesKMeans* class. The tslearn library is a machine learning library designed and optimized for time series data, so it works naturally with our dataset. We chose to start with K-Means because it is the most rudimentary clustering algorithm and could provide a baseline analysis to build on.

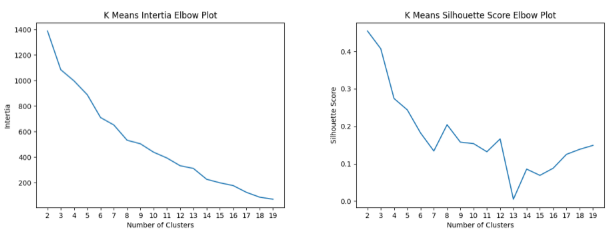

Figure 4: Result of K-Means Clustering (Euclidean Distance)

The initial result of K-Means is shown above. When running the K-Means algorithm, the two most critical hyperparameters are the number of clusters and the distance metric which is used to divide clusters. This run uses the default *euclidean distance* and measures both the inertia and the silhouette score for clusters of size 2 to 19. The way to find the optimal number of clusters is to locate the 'elbow' of the inertia plot- the intuition being finding the number of clusters where adding additional clusters is no longer worth the additional cost. The 'elbow' of this curve appears to be 8, with a resulting inertia of ~575 and a silhouette score of ~0.2. Further analysis is necessary to find the most optimal distance metric.

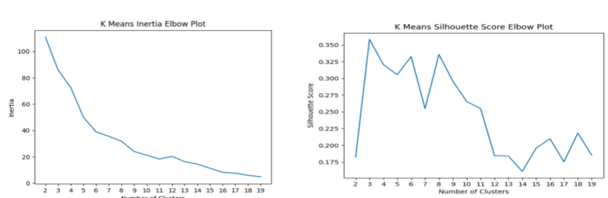

Figure 5: Result of K-Means Clustering (Dynamic Time Warping)

After analysis of all possible distance metrics, dynamic time warping yielded the best performance. Dynamic time warping is an algorithm that measures the similarity between temporal datasets by accounting for their varying speeds. Figure 5 details the performance of K-Means with dynamic time warping. There are two potential elbow points in the inertia plot: 6 clusters and 8 clusters. The silhouette plot reaffirms these two cluster sizes with peaks at each of these values. Overall, for a cluster size of 8, dynamic time warping K-Means strongly outperforms euclidean K-Means with an inertia of ~25 (compared to ~575) and a silhouette score of 0.325 (compared to ~0.2).

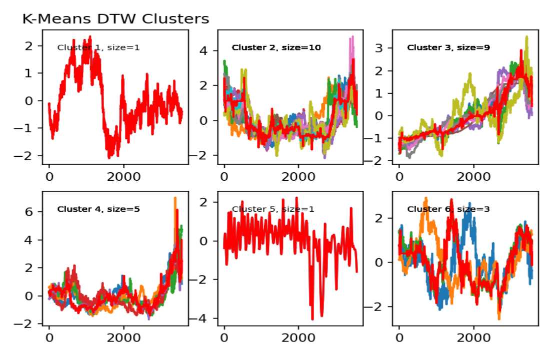

Figure 6: Overlay of Clusters Resulting from K-Means Dynamic Time Warping Model (6 clusters)

The plot above shows each of the clusters overlaid on a separate graph, with the centroid shown in red. Each cluster seems to account for a differently shaped graph, but cluster 2 and cluster 3 have sizes of 10 and 9 respectively. This accounts for 19 of the 29 indices shared between just two clusters, which seems overgeneralized. The 8 cluster model will show better results.

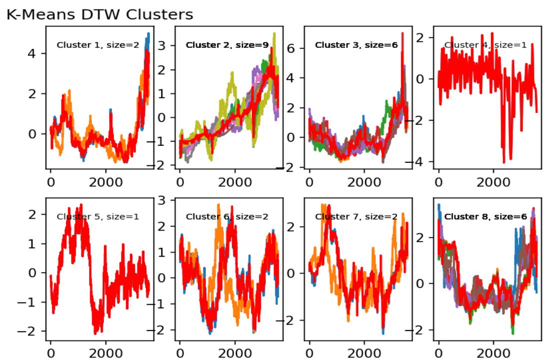

Figure 7: Overlay of Clusters Resulting from K-Means Dynamic Time Warping Model (8 clusters)

This shows the same plot as before, but with 8 clusters. While there is still a large cluster of size 9, the rest of the clusters are much more evenly distributed than in the 6 cluster model. Satisfied with this result, we elected to continue by performing hierarchical clustering using both scikit-learn's *AgglomerativeClustering* class and scipy's scipy.cluster.hierarchy library. This analysis would emulate that seen from Nicoló Musmeci, Tomaso Aste, and T. Di Matteo in their 2015 paper *Relation between Financial Market and Real Economy: Comparison between Clustering Methods* (see Related Work section). The goal of this endeavor was to both generate an optimal clustering model and also a dendrogram: a tree-based diagram which illustrates the hierarchical relationships between all entries in the data.

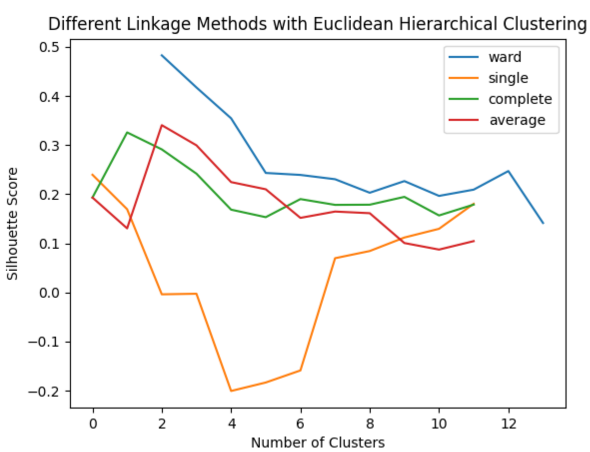

Figure 8: Comparison of Euclidean Hierarchical Clustering with Different Linkage Methods

The figure above shows the results of hyperparameter tuning on a hierarchical clustering model with a euclidean distance metric. The parameters tested are the linkage method and the number of clusters. The linkage methods differ in defining proximity between any two clusters at each time step. As is seen from the plot above, ward linkage outperforms all other linkage methods for all cluster sizes. The best cluster size appears to be 12, where a silhouette score of ~0.25 is shown. Although this result is satisfying, further analysis must be conducted on the distance parameter to ensure finding the best possible model.

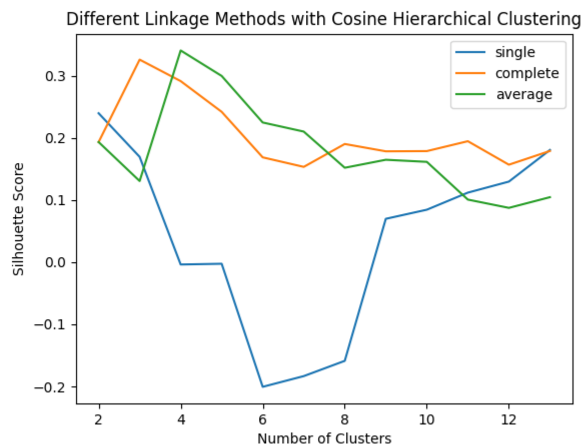

Figure 9: Comparison of Cosine Hierarchical Clustering with Different Linkage Methods

After testing the pearson, L2, L1, dynamic time warping, and cosine distance metrics on single complete, and average linkage (ward linkage only works for euclidean distance) for cluster sizes of 2-13, none of the models were able to outperform the ward linkage model using the silhouette score metric for any number of clusters. As a result, we elected to deploy the ward linkage model to generate our final dendrogram.

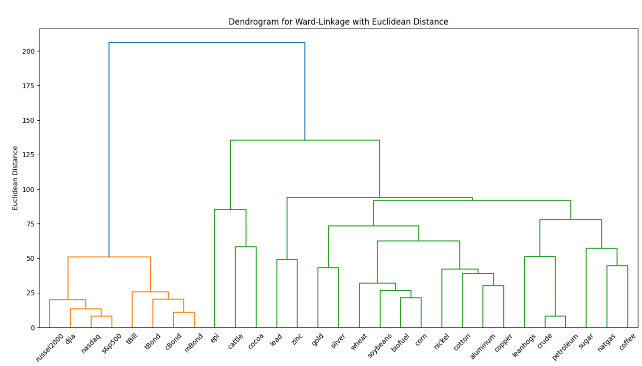

Figure 10: Final Dendrogram for Ward Linkage Hierarchical Clustering

A dendrogram depicting the clustering hierarchy for the entire dataset is shown above. The dendrogram depicts the clustering process by showing the order in which different clusters are formed based on the euclidean distance metric (y-axis). Each connection on the graph represents a cluster. To count the number of clusters for a specific distance, draw a line across the y-axis for that distance and count the number of lines it intersects. Each of those lines is a cluster. Some of the interesting insights which can be acquired from this dendrogram are: all stock indices are highly correlated with one another but not with the rest of the dataset; the epi metric does not appear to have a high correlation with any other index; the metal indices (nickel, aluminum, lead, zinc, copper, gold, and silver) are correlated with the agriculture indices (wheat, cotton, soybeans, corn, biofuel). After generating the dendrogram, we also wanted to confirm that the clusters made sense when graphed on top of one another. We applied the same function as was used in *Figure 10* and created the following plot.

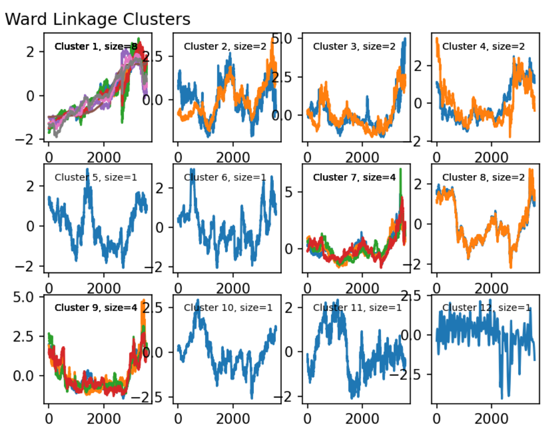

Figure 11: Overlay of Clusters Resulting from Euclidean Hierarchical Clustering with Ward Linkage

Upon initial inspection, each of the plots appears to have similarly shaped graphs. Out of the 12 clusters, five have a size of one and four have a size of two. The cluster of 9 depicts all the stock indices and the clusters of 4 reflect the agriculture and metal clusters depicted in *Figure 10*.

EVALUATION

In order to evaluate our correlation network and clustering methods for insights, we compared the relationships generated by our various visualizations with the real-world relationships that exist between relevant financial markets/economic indicators as revealed through individual research. Once meaningful relationships were identified for each visualization, we compared across each model to see if any relationship insights held. This should reveal which relationships are the most meaningful.

1 Correlation Network

To assist in identifying interesting relationships from our correlation network we examined a network containing positive correlations and negative correlations separately to make relationships easier to identify. When evaluating our correlation network a particular focus was placed on examining the nodes with the highest degree (the nodes possessing the most relationships) and nodes connected to edges representing the strongest correlations. Once unique groupings or extremely strong correlations were identified, individual research on relevant financial markets/economic indicators was performed to see if the relationships logically translated to real-world trends. While a number of groupings and relationships were revealed to be erroneous or coincidental, a similar number of groupings and relationships were revealed through research as valid, interesting, and relevant. These findings can be categorized at a high level, as extremely evident and less evident. Extremely evident findings include:

- All stock markets are highly positively correlated to each other. This is extremely evident as an understanding of how general investor confidence affects the performance of stock markets is all that is required to understand this relationship. [7]
- Similarly, bond markets are highly positively correlated to each other. This is easily understood through the relationship between the cost of borrowing money and the performance of bond markets. [8]
- Petroleum and crude oil are highly positively correlated to each other. This is extremely evident as the knowledge that petroleum is derived from crude oil is all that is required to understand this relationship.
- GDP is highly positively correlated with common investment vehicles (bonds, stocks, gold). This is easily understood through the long term relationship between GDP and gross investment. [9]
- Unemployment is strongly negatively correlated with all stock markets. As discussed above, stock market performance is intimately related to investor confidence. As unemployment has been

shown to damage investor confidence, this negative relationship is quite evident. [10]

Less evident findings include:

- Biofuel is highly positively correlated to corn, wheat, and soybeans. This is explained by research that reveals in the United States, corn and soybean oils are two of the most common inputs in biofuel. Furthermore, corn, soybeans, and wheat are considered substitute products in the field of economics and as such consumers will often choose the cheapest of the three in their purchasing decisions. This ties their prices together in the long term. [11]
- Aluminum, copper, gold, nickel, and silver are all highly positively correlated. Research reveals that these are all common inputs in electronics manufacturing, it is possible that their relationship stems from this commonality. [12][13]
- Bonds are strongly negatively correlated with many seemingly unrelated commodities. Given that inflation drives up the prices of commodities while damaging bond yields, this relationship can be explained by this. Research supports this idea. [14]
- EPI is strongly negatively correlated to bond markets and gold. This can possibly be explained by investors being less keen on low-risk, low-yield assets when they feel the economy is performing well. Research supports this. [15]

2 Clustering

The two best models generated from the clustering analyses were the K-Means using 8 clusters as well as dynamic time warping and the hierarchical clustering using ward linkage along with euclidean distance on 12 clusters. The clusters generated by each model are depicted below.

K-Means Clusters			
Cluster 1 Natural Gas Coffee	**Cluster 2** DJIA NASDAQ Russell 2000 S&P500 tBond tBill cBond mBond Zinc	**Cluster 3** Aluminum Copper Gold Nickel Wheat Cotton	**Cluster 4** Economic Performance Index (EPI)
Cluster 5 Cocoa	**Cluster 6** Sugar Lead	**Cluster 7** Cattle Leanhogs	**Cluster 8** Petroleum, Crude, Biofuel, Soybeans, Corn, Silver

Hierarchical Clustering Clusters			
Cluster 1 DJIA NASDAQ Russell 2000 S&P500 tBond tBill cBond mBond	**Cluster 2** Lead Zinc	**Cluster 3** Coffee Natural Gas	**Cluster 4** Gold Silver
Cluster 5 Sugar	**Cluster 6** Leanhogs	**Cluster 7** Aluminum Copper Nickel Cotton	**Cluster 8** Crude Petroleum
Cluster 9 Biofuel Soybeans Wheat Corn	**Cluster 10** Cattle	**Cluster 11** Cocoa	**Cluster 12** Economic Performance Index (EPI)

Figure 12: Cluster Labels for K-Means and Hierarchical Clustering

Many insights can be gathered simply by looking at the way each model clustered the data. The following sets of indices are clustered together in both algorithms: DJIA, NASDAQ, Russell 2000 S&P500, tBond, tBill, cBond, mBond; Aluminum, Copper, Nickel, Cotton; Crude and Petroleum; Biofuel, Soybeans, and Sorn. This implies that these sets of indices are highly correlated. In addition, both models had a cluster of size 1 for Cocoa and EPI, meaning those indices are not correlated to the other ones.

Some of the disparities between the models' clusters are also worth noting. An example is that K-Means puts cattle and Leanhogs into the same cluster while Hierarchical does not put them in the same cluster until there are only two clusters. These drastic differences in output of the models are due to both the different algorithms used by the models as well as their metric for determining whether each dataset is correlated. In K-Means, dynamic time warping matches points in each dataset with points in other datasets to account for the difference in speed between the two datasets. On the other hand, ward linkage in hierarchical clustering attempts to minimize the difference of the sum of squares as the result of each addition to a cluster. This means that ward linkage might be better at measuring the similarity between two datasets on a day-by-day basis, and dynamic time warping might be better at measuring long-term differences in the change of each dataset.

DISCUSSION

1 EPI Calculation

An issue with the calculation of the EPI values is that we had to interpolate several values. Due to the missing data values that we had to account for, we interpolated the data to find these values. However, when we are interpolating several values, the strength of data begins to diminish. In the scale (time series) of our project these interpolated values would not cause massive inaccuracies, since the data is from the last 10 years.

For future work where we incorporate expansion to other economic markets (specifically emerging markets) we would want to interpolate values such that the volatility of the EPI calculations are minimized. To accomplish this, we would want to use the weighted EPI formula instead of the raw EPI formula that we currently have used. Our intention with using the raw EPI formula was that for US markets, this calculation presents a low volatility risk which is sufficient to provide accurate results for our data from the last 10 years. However, with the expansion to other economic markets as well as if we were to use much older data, we would instead work with the weighted EPI calculation.

2 Correlation Network

The current correlation network does provide great insight into how the markets and stock indices are related, however we could improve on the methods in the correlation network. Firstly, instead of having a separate clustering model built from the correlation vector, we could have also run community detection algorithms on our network to identify clusters inside the networks and then repeat this for the positive correlation network and the negative correlation network.

We would propose using two clustering/community detection algorithms to apply on the network: Louvain community detection and Infomap community detection. Louvain community detection can be executed through the NetworkX python library, while Infomap community detection would be done with the Infomap python library. These two community detection algorithms differ in the way they calculate the best partition of a network to generate a community. The Louvain Community detection algorithm focuses on optimizing the modularity score of the partition of a network [5], while the Infomap community detection uses the Map Equation “which exploits the information-theoretic duality between finding community structure in networks and minimizing the description length of a random walker's movements on a network” [6].

Once we receive partitions from both community detection algorithms we can assess the modularity of the partitions and decide where we are getting the best clustering of our markets/stock indices. Furthermore, we can compare the results of the network community detection algorithms with the clustering models we have already conducted in the study.

3 Clustering

The problem with theoretical metrics such as silhouette score and inertia to evaluate clusters is that they do not demonstrate the degree to which the models are useful in a real-world application. The only way to truly test the success of each model would be to use the results of the models to make real-world predictions. There are two stock trading measurements that can be applied from the output of these models. The first is known as Momentum-based trading. In Momentum-based trading, stocks are bought based on their recent performance, with the intuition that their ‘momentum’ will cause their price to keep increasing. Performance can be measured compared to either the moving average or in the case of our model, the centroid generated by K-Means. Stocks are bought depending on their category, and since the clustering algorithms have generated ‘categories’ for us, we can use the output of the models for momentum-based trading. Another trading method is called Betting Against Beta. In this method, stocks are bought and sold depending on their volatility (usually standard deviation), and the method of buying is applied to each industry separately. Instead of different industries, the output of the clustering model can be used. Overall, the clustering models have applications beyond being an interesting project and can be applied to make smart investment decisions, but further testing is required to determine whether they are useful in those cases [3].

It is worth considering the limitations of the input data when evaluating the clustering models. The dataset used to feed the clustering model contained 29 time series sets that go back 10 years in time (a little over 3000 entries each). While this is an appropriate dataset for the scope and goals of our project, it may not be enough to generate results worthy of real-world application. In the United States, there are over 5,000 stock indices, which means that using only 29 may not capture all factors at play when conducting correlation and clustering analysis. In addition, we chose to use data dating back 10 years. While this may be useful for gaining a generalized understanding of the long-term relationships between indices, the clustering may not capture understanding of short-term trends between indices. To conduct further clustering analysis, we would first expand the number of indices in the dataset to further capture the overall essence of the market. Then, we would test the dataset on different lengths of time (going back 5 years, 1 year, 1 month) and determine whether the results are significantly different.

CONCLUSION

This project was developed as an exploratory study into the visualization of economic markets. The goal was to deliver a visualization tool that could be used to easily find the relationships between economic markets. The correlation graph was successful in demonstrating known relationships between markets.

The overall project can be broken down into key steps. Initial cleaning and piping of data, construction of the EPI, generating correlation vectors, visualizing correlation graph, cluster analysis of data sets, and discovering insights from the delivered graphs.

Insights gained from the completion of the project include but not limited to, the high correlation of stock markets, high correlation of bond markets, high correlation of petroleum and crude oil, correlation of agriculture and biofuel, and the negative correlation of stocks, bonds, and gold to unemployment.

REFERENCES

[1] Khramov, Vadim, and John Ridings Lee. "The Economic Performance Index (EPI): An Intuitive Indicator for Assessing a Country's Economic Performance Dynamics in an Historical Perspective." *SSRN Electronic Journal*, 2013, https://doi.org/10.2139/ssrn.2358485

[2] Musmeci N, Aste T, Di Matteo T (2015) Relation between Financial Market Structure and the Real Economy: Comparison between Clustering Methods. PLOS ONE 10(3): e011601. https://doi.org/10.1371/journal.pone.0116201

[3] Lu, Yilang. *Application of Clustering Methods to Trading Strategies in the US Equity Market*. Imperial College London, 2018, https://www.imperial.ac.uk/media/imperial-college/faculty-of-natural-sciences/department-of-mathematics/math-finance/Lu_Yilang_01407813.pdf

[4] spring_layout—NetworkX2.8.8documentation. https://networkx.org/documentation/stable/index.html. Retrieved December 11, 2022 from https://networkx.org/documentation/stable/reference/generated/networkx.drawing.layout.spring_layout.html

[5] louvain_communities — NetworkX 2.8.8 documentation. https://networkx.org/documentation/stable/index.html . Retrieved December 11, 2022 from https://networkx.org/documentation/stable/reference/algorithms/generated/networkx.algorithms.community.louvain.louvain_communities.html

[6] Infomap - Network community detection using the Map Equation. https://www.mapequation.org/ . Retrieved December 11, 2022 from https://www.mapequation.org/infomap/

[7] Lee A. Smales. 2016. Investor Sentiment and Stock Market Returns. SSRN Electronic Journal. DOI:https://doi.org/10.2139/ssrn.2749518

[8] How do rates affect bond performance? | PIMCO. PIMCO. Retrieved December 11, 2022 from https://global.pimco.com/en-gbl/marketintelligence/navigating-interest-rates/how-do-rates-affect-bond-performance

[9] 2021. Introduction to U.S. Economy: Business Investment. Congressional Research Service. Retrieved December 11, 2022 from https://sgp.fas.org/crs/misc/IF11020.pdf

[10] Selim Koray Demirel and Seyfettin Artan. 2017. The Causality Relationships between Economic Confidence and Fundamental Macroeconomic Indicators: Empirical Evidence from Selected European Union Countries. International Journal of Economics and Financial Issues. Retrieved December 11, 2022 from https://www.econjournals.com/index.php/ijefi/article/view/5356

[11] Biofuels Factsheet | Center for Sustainable Systems. University of Michigan Center for Sustainable Fuels. Retrieved December 11, 2022 from https://css.umich.edu/publications/factsheets/energy/biofuels-factsheet

[12] 2022. What Commodities Are the Main Inputs for the Electronics Sector? Investopedia. Retrieved December 11, 2022 from https://www.investopedia.com/ask/answers/042015/what-commodities-are-main-inputs-electronics-sector.asp

[13] 2021. Four Factors Behind the Metals Price Rally. IMF Blog. Retrieved December 11, 2022 from https://www.imf.org/en/Blogs/Articles/2021/06/08/four-factors-behind-the-metals-price-rally

[14] 2021. Intermarket Relationships: Following the Cycle. Investopedia. Retrieved December 11, 2022 from https://www.investopedia.com/articles/fundamental-analysis/09/intermarket-relations.asp

[15] 2022. How Changes in Economic Growth Affects Bonds. the balance. Retrieved December 11, 2022 from https://www.thebalancemoney.com/bonds-and-the-economy-417070

APPENDIX

1 Honor Code

On our honor, as University of Colorado Boulder students we have neither given nor received unauthorized assistance.

-Robert Hellums, Gabriel Keith, Sarthak Shukla, Lucas Derr

2 Individual Contributions

Robert Hellums: Identified key economic markets and collected all relevant data. Identified economic performance calculation methodology and collected all relevant data. Cleaned and piped data relevant to economic performance score. Calculated economic performance score to be passed into the main data pipeline. Researched economic markets in service of analyzing models.

Gabriel Keith: Developed data pipeline that cleaned and formatted data from csv files. Pipeline ended with the calculation of the correlation vector.

Sarthak Shukla: From the correlation vector created by Gabriel, Sarthak created the 3 correlation networks in our model as well as all the visualizations for the correlation networks.

Lucas Derr: Performed all clustering analyses. Also did initial data processing and analysis before the expansion of the dataset by Gabe and Rob.