FULL LEGAL NAME	LOCATION (COUNTRY)	EMAIL ADDRESS	MARK X FOR ANY NON-CONTRIBUTING MEMBER
Alejandro Piedra Alvarado	Costa Rica	1alepiedra7@gmail.com	
Peter Esekhaigbe	Nigeria	petersonese304@gmail.com	
Jude Chukwuebuka Ugwuoke	United States	jcu0005@auburn.edu	

Statement of integrity: By typing the names of all group members in the text boxes below, you confirm that the assignment submitted is original work produced by the group (excluding any non-contributing members identified with an "X" above).	
Team member 1	**Alejandro Piedra Alvarado**
Team member 2	**Peter Eronmosele Esekhaigbe**
Team member 3	**Jude Chukwuebuka Ugwuoke**

Introduction

Financial markets are becoming increasingly sophisticated, a direct result of the need for complex analytical techniques. Machine learning (ML) has been a game-changer in quantitative finance by powering data-driven decision-making, thus allowing investment managers to dig deeper into huge datasets and extract useful insights. This report examines the three main ML methods for trading strategy development: LASSO regression, hierarchical clustering, and Principal Component Analysis (PCA). Each of the methods performs a specific function: LASSO regression increases the predictive accuracy through the selection of the most significant financial variables, Hierarchical Clustering solves the problem of finding the best mix of assets by grouping them, and PCA helps make sense of complex datasets by making them less complex through the reduction of dimensionality. Utilization of these models by professionals in the finance sector can lead to the advancement of portfolio construction, risk assessment, and the forecasting of markets which thus reaffirms the role of ML in investment strategies.

Step 1

- **Category 1: Topic: LASSO regression.**
    - **Basics:**
        - Definition: LASSO (Least Absolute Shrinkage and Selection Operator) regression is a linear regression model with L1 regularization. Unlike OLS, which minimizes only the sum of squared errors, LASSO adds a penalty term proportional to the sum of absolute values

of coefficients. This leads to feature selection as some coefficients shrink to zero, removing irrelevant predictors.

- Classification:
  - Type: Supervised Learning
  - Subcategory: Regression (Regularized Linear Regression)
  - Key Property: Performs both regression and feature selection
  - Common Use Cases in Finance: Asset pricing, factor selection, risk modeling, and portfolio optimization

- **Keywords:** "tags" that identify this model.

#LASSORegression
#MachineLearningFinance
#LinearRegression
#FeatureSelection
#SparseModels
#Regularization
#AlphaTuning
#QuantFinance
#RiskModeling
#PredictiveAnalytics

- **Category 2. Topic: Hierarchical clustering.**

  - **Basics:**
    - Definition: Hierarchical clustering is a technique in unsupervised learning that gradually pools items in the same group (clusters) by looking at their pair similarity or differences. The final outcome is a hierarchy of organizations that can be depicted as a tree structure (commonly called a dendrogram) (Hastie et al.).

      By examining this dendrogram, analysts can decide on the ideal number of clusters based on their objectives. This method does not require prior knowledge of the number of clusters, which makes hierarchical clustering particularly suited for exploratory data analysis. It becomes paramount in situations where an analysis targets understanding how the items may combine into general groups or when the analyst suspects that the data may be indicating different organizational levels (the existence of smaller subgroups within larger groups).

    - Classification:
      - **Agglomerative (Bottom-Up) Approach**
        - **Process**: Each data point begins as its own cluster. The algorithm calculates the distance (or similarity) between the clusters, and the two closest groups are combined in each iteration. This is illustrated in the figure below:
        - **Hierarchy Formation**: Repetitively merging all the different clusters into one will result in one unique cluster, and the nested hierarchy will be built.
        - **Dendrogram**: The graphic representation of a sorted group of clusters, demonstrating how long a pair of groups unites into one larger group.

- **Divisive (Top-Down) Approach**
  - **Process:** All data points are grouped into a single cluster. The method divides the cluster into smaller licensed groups according to the chosen dissimilarity measure.
  - **Hierarchy Formation**: The splitting continues level by level, producing a series of narrower partitions, until each point is unattached (or until an alternate stopping criterion is reached).
  - **Dendrogram:** Even though less commonly done through the agglomerative approach, the top-down methodology can still show splits using the dendrogram, showing the points where the significant splits happen.

- **Linkage Methods.** When deciding how to merge or split clusters, the algorithm uses a linkage criterion that quantifies the distance between clusters:
  - **Single Linkage**: Using the cluster-to-cluster distance will mean choosing the shortest possible distance between any pair of points in different clusters.
  - **Complete Linkage**: In the other way, when two clusters are the farthest apart, the maximum distances are used as distances between the clusters.
  - **Centroid Linkage**: The method focuses on the distance between the center of gravity (centroid) of two clusters.
  - **Average Linkage**: It takes an average of the distances between each point in one cluster and every point in the other.
  - **Ward's Method**: It brings the minimum distance between the nearest cluster and the maximum distance between the two clusters together. The merger is done with the smallest value, so the total sum of squares remains as small as possible.

- **Distance Metrics.** Through hierarchical clustering, varying distance or similarity measures may be utilized to analyze the degree of correlation of data points:
  - **Euclidean Distance**: This approach determines the straight-line distance between two points mathematically expressed on a 2D Cartesian plane.
  - **Manhattan Distance:** It is the total of the absolute differences of these coordinates, hence, it is called "city block distance".
  - **Cosine Similarity:** The measure is often used in high-dimensional contexts (e.g., textual data) and is given by the cosine of the angle between two vectors.
  - **Canberra Distance:** This metric takes into account relative differences and is the most commonly used for ordered categorical data.

- **Hamming Distance:** The quantification of the number of differing places between two strings is a perfect way to represent categorical or binary variables.
- **Maximum Distance (Chebyshev):** It gives rise to the biggest absolute difference that exists in the dimensions of the point across all dimensions.

- **Dendrogram Interpretation.** The result of the dendrogram can be interpreted as a powerful technique:
  - **Vertical Lines & Cluster Merges**: Each vertical line (or node) represents a point of density between two clusters, and the height indicates how different the two clusters were.
  - **Cutting the Dendrogram:** To obtain the clusters, just draw a horizontal line across the chosen height on the dendrogram. The number of clusters corresponds to how many vertical segments the horizontal line intersects.
  - **Granularity Control:** Lower cut lines yield more, smaller clusters, while higher cuts produce fewer, larger groups.

- **Keywords:** "tags" that identify this model.
  - **Hierarchical Clustering**: A clustering technique that organizes data points into a nested sequence of groups, forming a tree-like structure called a dendrogram.
  - **Clusters:** Groups of data points that are similar or near one another, generally the clusters are identified by the help of distance or dissimilarity
  - **Dendrogram:** A graphical representation of a tree that shows how individual points or small clusters come together step by step. This is the process that discloses the hierarchy in the data.
  - **Linkage Criteria** specify the criteria or methods for the calculation of distances between clusters and are used in the merger process, such as single, complete, average, and Ward's.
  - **Distance Metric:** A systematic approach (like Euclidean or Manhattan) that is used to say how different data points are from each other, which is very important in precisely defining cluster boundaries.
  - **Ward's Method:** A linkage criterion that is specific to this, and that makes the clusters to be in by making the least increase in within-cluster variance, may result in a more balanced grouping.
  - **Agglomerative/Divisive:** These are two primary dynamics of hierarchical clustering. The agglomerative one marches forward in its build of clusters from the ones of the smallest points, making backward. The divisor will usually split one large cluster into separate small clusters.
  - **Nested Clusters:** Subgrouping structures are created from existing clusters such as sub-clusters in larger clusters; thus, they are more informative than clusters at the higher level in the dataset.

- **Outlier Resilience:** The hierarchical clustering capability of potentially acting separately or focusing on outliers, limiting their negative impact on the overall cluster structure.
- **Cut Level:** It is the level of significance in dendrograms at which nodes are recursively cut, with the last cut being that of the fusion of the final merged nodes, where the clustering assignment of the nodes is established, and it is definitive.
- **Data Mining:** The term that refers to a larger set of processes that involve the extraction of patterns and insights from large or complex datasets; these processes include exploratory data analysis (EDA), which also is a common application of data mining.
- **Visualization:** The act of converting data in different forms into a graph or a chart, for example dendrograms and/or scatterplots, to help readers grasp the message conveyed by cluster analysis results more fully and correctly.

- **Category 3. Topic: Principal Component Analysis (PCA)**
  - **Basics:**
    - Definition: Principal Component Analysis is a feature extraction technique that reduces the dimension of statistical data while also extracting and representing relevant information about the data as a set of orthogonal principal components. This methodology handles high-dimensional datasets by transforming features in datasets into representable principal components. These principal components effectively explain the original data points in the dataset. In simple terms, principal component analysis does two important things: dimensionality reduction and feature extraction.

      Mathematically, PCA relies on the concept of eigen spaces: the eigen-decomposition of positive semi-definite matrices and the singular value decomposition (SVD) of rectangular matrices. Eigenvalues and eigenvectors are essential for the transformation of rectangular matrices of original data.

      Principal Component Analysis is applicable for data reduction and image compression. and patternrecognition. The PCA methodology is applied in the different fields,such as finance,healthcare, and forensics

      In finance, PCA ismainly used for interest-rate modelling across different maturities,helping investors determine which bonds or stocks to invest. It compares thequality of bonds based on the information garnered from their yield curves. In most cases, only three principal components (PC1, PC2, and PC3). PC1 is measures the *parallel shift* in the yield curve (which is the most dominant economic factor to consider). PC2 measures the *slope changes—an indication* of major changes in the specific sectors. Lastly, PC3 explores minor variations in the sector-specific patterns.

    - Classification:
      Principal Component Analysis is classified based on the data type and structures

- **Linear PCA**: This approach applies linear transformation to data features that have linear relationships. Generally, the data features can be represented in linear equations with different weights and independent (highly correlated) variables.
- **Non-linear PCA:** Unlike the linear PCA, this approach is useful for non-linear datasets with non-linear data structures.
- **Advanced PCA:** This approach deploys advanced techniques in handling complex data forms, such as datasets with a large number of outliers and feature selection issues, datasets with missing data, and/or large numbers of variables.

■ **Keywords**

Principal Component Analysis, PCA, eigenvalues, eigenvectors, loadings, weights, dimensionality, multicollinearity, correlation, covariance matrix, explained variance, identity matrix, scree plot, biplot

Step 2

★ **Category 1. Topic LASSO Regression.**

a. **Advantages:**

- **Feature Selection**: Automatically eliminates irrelevant variables, reducing overfitting.
- **Handles High-Dimensional Data**: Effective performance in scenarios where the number of predictors exceeds the number of observations, such as in factor models
- **Prevents Multicollinearity**: Unlike OLS, LASSO mitigates the impact of highly correlated variables, preventing model distortion.
- **Enhances Interpretability**: Models are more interpretable due to sparsity (zero coefficients).
- **Computationally Efficient**: LASSO offers a faster alternative to stepwise regression for feature selection tasks.

b. **Computation:** Refer to “MLiF_GWP1_g8507.ipynb” for the computational implementation details.

c. **Disadvantages:**

- **Over-Shrinking**: LASSO tends to underperform when all variables are relevant but exhibit small coefficients, as it excessively reduces their magnitudes.
- **Computational Cost:** Tuning hyperparameters via cross-validation can render LASSO slower than OLS regression.
- **Collinearity Sensitivity:** If two highly correlated variables exist, LASSO arbitrarily selects one and discards the other, lacking a systematic selection mechanism.

- **Bias in Coefficients**: Unlike Ridge regression, LASSO does not always provide unbiased coefficient estimates.

d. **Equations:**

- LASSO minimizes the following cost function:

$$\min_{\beta} \sum_{i=1}^{n} (y_i - X_i\beta)^2 + \lambda \sum_{j=1}^{p} |\beta_j|$$

The L1 regularization term promotes sparsity by driving some coefficients precisely to zero. In contrast, the L2 regularization (used in Ridge regression) penalizes large coefficients without eliminating them entirely. With L1 regularization, the penalty applies uniformly across all coefficients, irrespective of their size. As λ increases, less influential predictors are excluded, retaining only the most significant features. This property positions LASSO as an effective method for feature selection.

L1 regularization term: $\lambda \sum |\beta_j|$

L2 regularization (used in Ridge regression): $\lambda \sum |\beta_j^2|$

Where:

- **λ** is the regularization parameter (determines sparsity).
- **β** are the regression coefficients.
- The second term enforces L1 regularization, shrinking some coefficients to zero.

e. **Features:**

- Performs automatic variable selection.
- Works well when p>n (more features than observations).
- Reduces overfitting in high-dimensional settings.
- Handles outliers better than Ridge regression (L1 penalty).

f. **Guide:**

- Inputs

  - Data from Yahoo Finance for the following tickers, utilizing only “Close” prices:
    - SPY: S&P 500 ETF
    - EURUSD=X: Euro / US Dollar Exchange Rate
    - GLD: Gold ETF
    - MINT: Short-Term Bond ETF
    - TIP: Treasury Inflation Protected Securities ETF
    - USO: Oil ETF
    - VNQ: Real Estate ETF
    - XLF: Financials ETF
    - XLY: Consumer Discretionary ETF
    - XME: Metals & Mining ETF
    - ^IXIC: Nasdaq Composite Index

- ^VIX: The CBOE Volatility Index.
- ^TNX: 10-Year Treasury Yield

- Feature Matrix (X):
  - Comprises macroeconomic indicators and asset returns, including returns of the listed assets (excluding SPY).
  - Returns are calculated as the percentage change in price from the previous day, shifted one period to predict future returns.
- Target Variable (Y):
  - Future returns of the SPY ETF serve as the target variable.

- Outputs
  - **Optimal Alpha (α):** The best regularization parameter for the LASSO model selected through cross-validation. Can also be referred to as lambda (λ).
  - **Sparse Coefficients:** Estimated coefficients for each feature after LASSO regularization, with sparsity, highlighting the most predictive macroeconomic variables.
  - **Predicted SPY Returns**: Forecasted SPY returns derived from the fitted LASSO model.
  - **Model Evaluation Metrics:**
    - **R^2 Score:** Measures how well the model explains the variance of the target.
    - **RMSE (Root Mean Squared Error):** Quantifies the difference between the predicted and actual SPY returns.

g. **Hyperparameters:**

- **Alpha (α):** regularization parameter that controls the strength of LASSO regularization.
  - The larger the value of alpha, the stronger the regularization, leading to more coefficients being shrunk to zero.
  - Tuning is performed using LassoCV, which applies cross-validation across a range of α values.
  - The function np.logspace(-6, 6, 13) generates 13 values of α / λ logarithmically spaced between 10^{-6} to 10^{6}. ensuring a comprehensive search range. Cross-validation identifies the value minimizing the chosen error metric.
- **Cross-Validation Folds (cv)**: controls how many folds of cross-validation are used to evaluate the model during the alpha tuning process.
  - The model uses 5-fold cross-validation, a number that balances bias and variance by providing a reliable trade-off between stability and efficiency..
- **Standardization (StandardScaler)**:

- While not directly a hyperparameter in the LASSO model itself, it is crucial to ensure the features are standardized (zero mean, unit variance) before applying LASSO.

- **Train-Test Split**:
  - Data is divided into training and testing sets (80-20 split), influencing model evaluation though not a traditional hyperparameter.

h. **Illustration:**

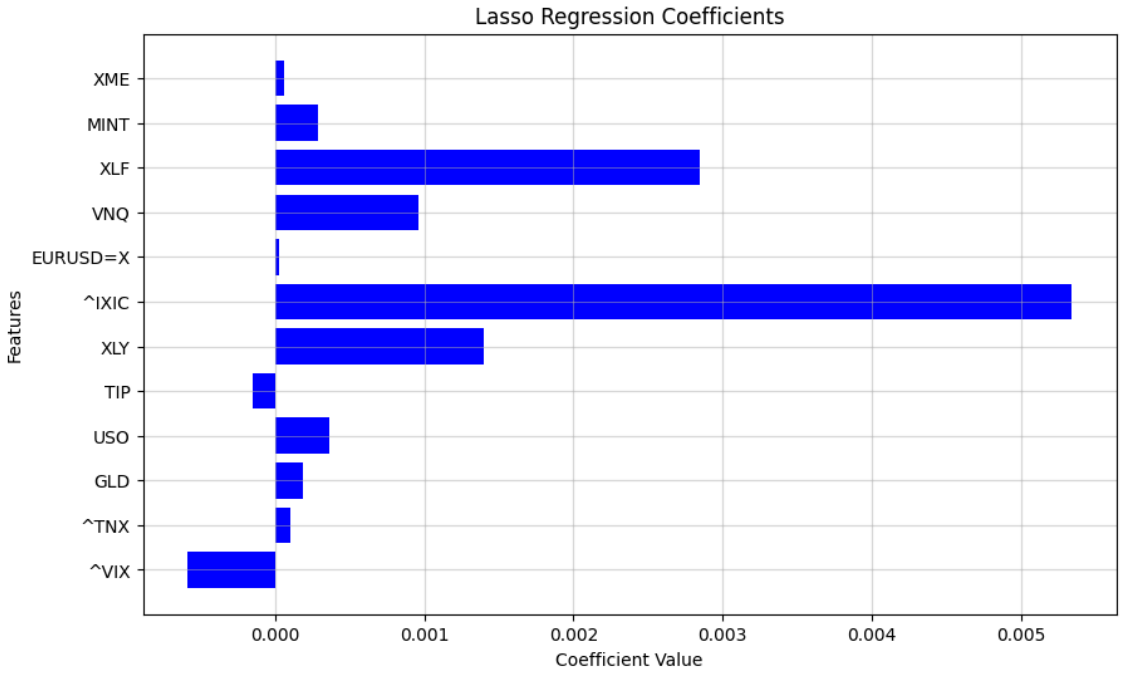

Fig 1. Lasso Regression Coefficients

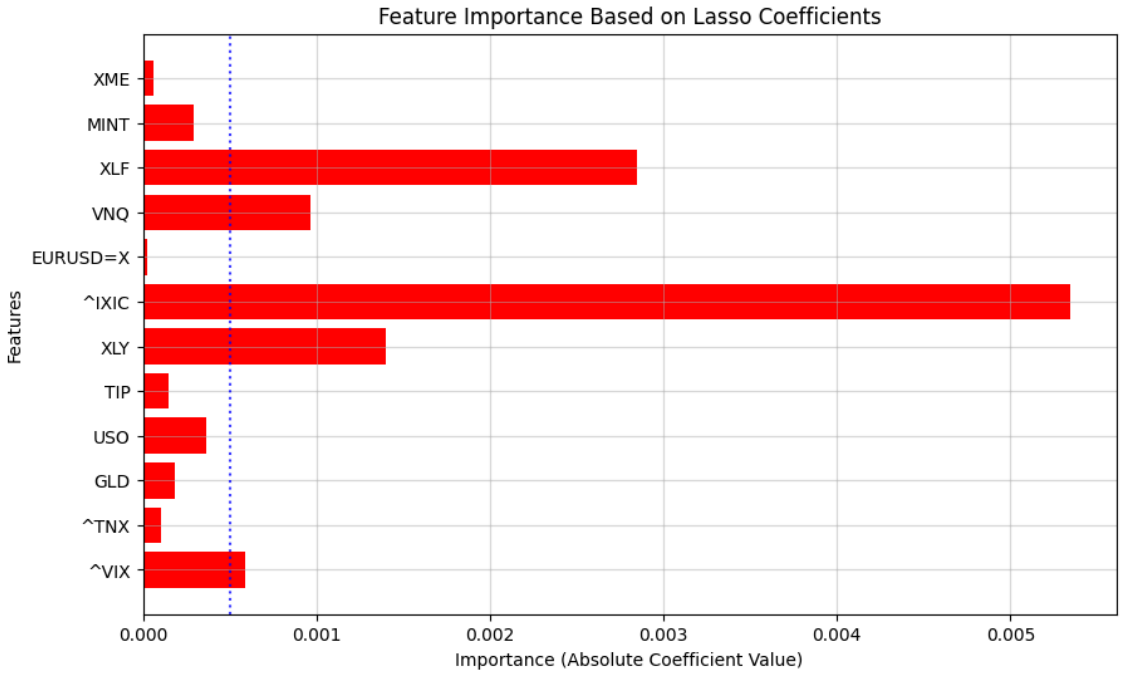

Fig. 2. Feature Importance Based on Lasso Coefficients

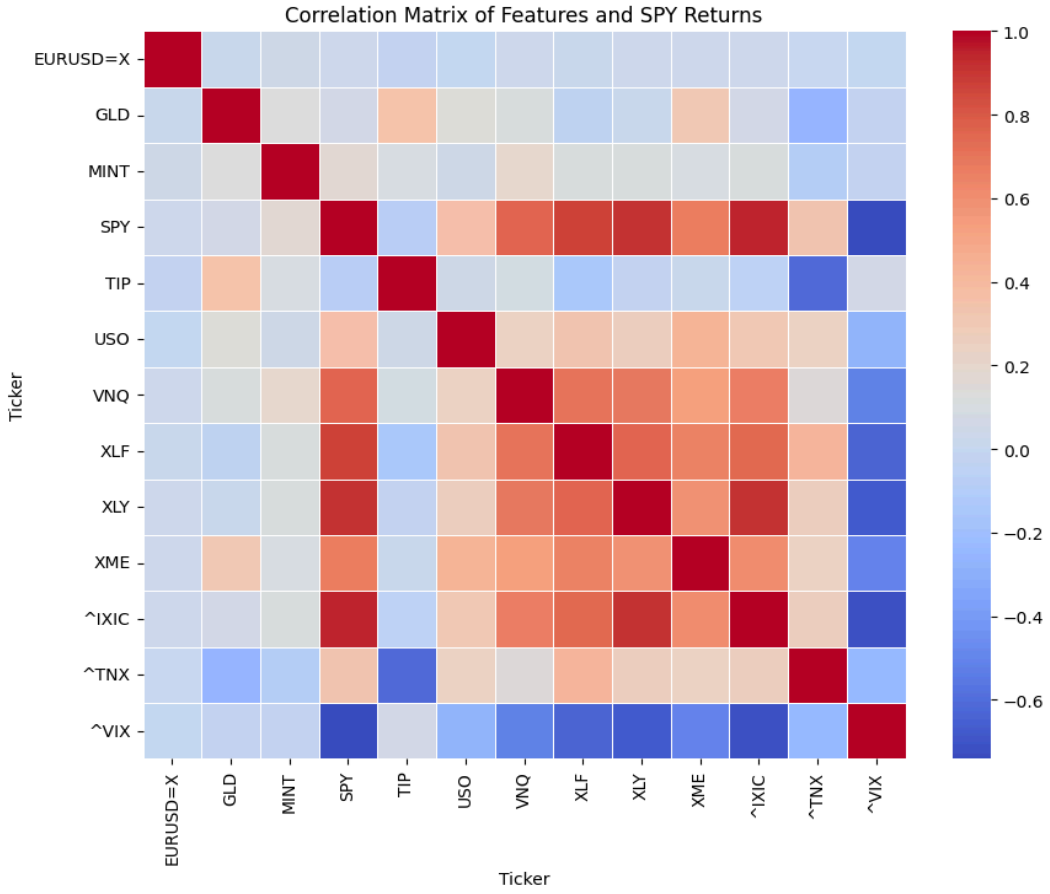

Fig. 3. SPY Returns Correlation Matrix

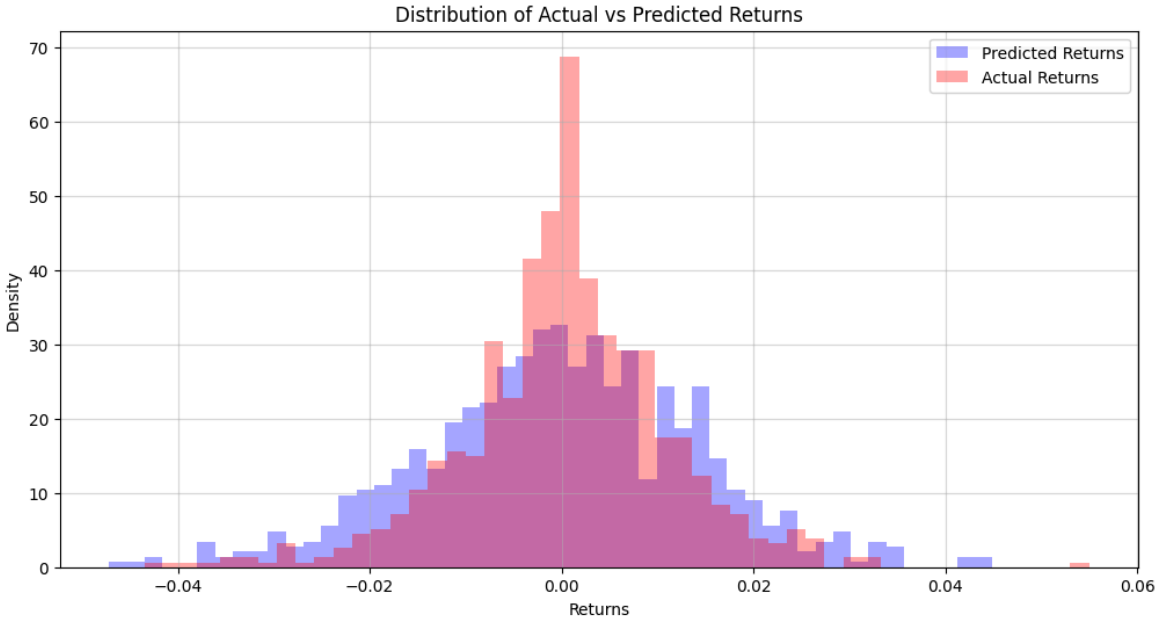

Fig. 4. Distribution of Actual vs Predicted Returns

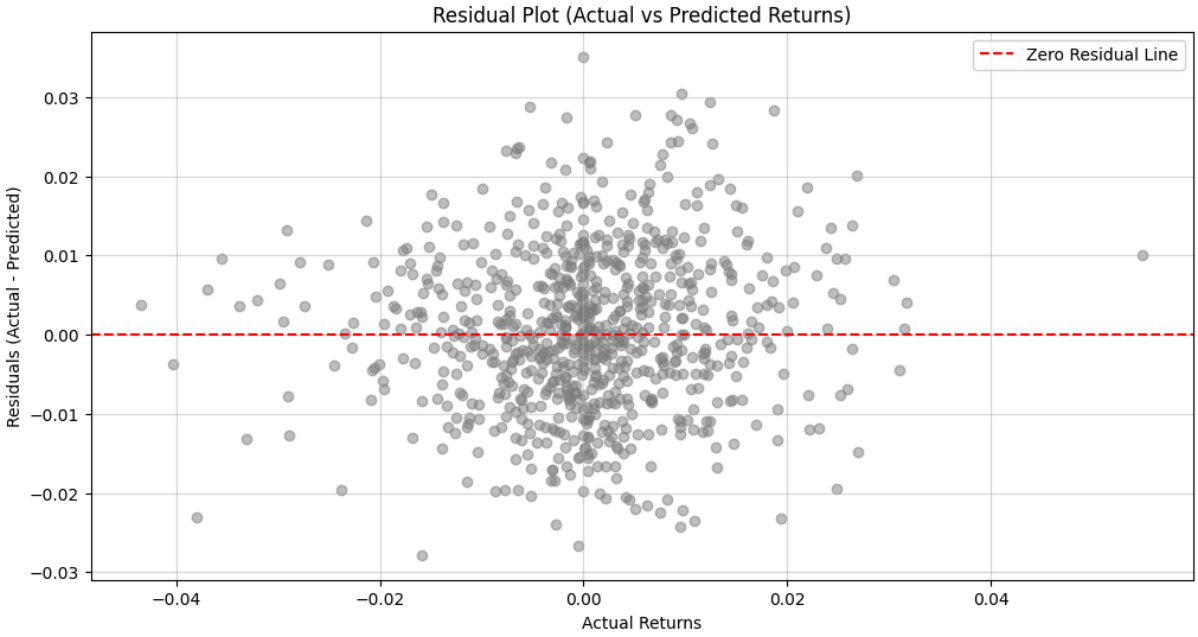

Fig. 5. SPY Returns Correlation Matrix

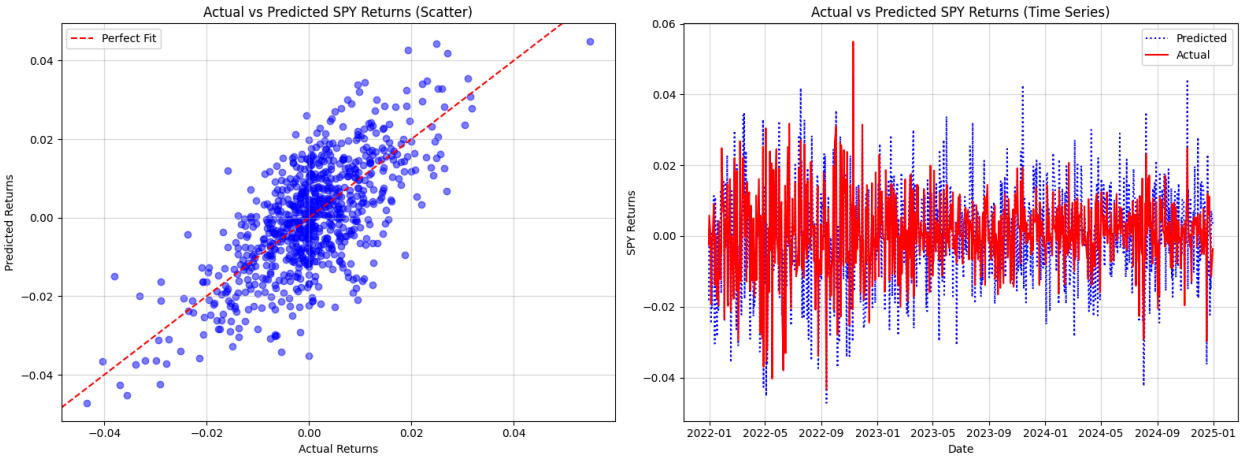

Fig. 6a. Actual Returns vs Predicted SPY Returns (Scatter Plot);

Fig. 6b. Actual Returns vs Predicted SPY Returns (Time Series).

i. **Journal:**

- Chan-Lau, Jorge A. “Lasso Regressions and Forecasting Models in Applied Stress Testing.” *IMF*, 5 May 2017, www.imf.org/en/Publications/WP/Issues/2017/05/05/Lasso-Regressions-and-Forecasting-Models-in-Applied-Stress-Testing-44887.

Jorge A. Chan-Lau’s paper, *"Lasso Regressions and Forecasting Models in Applied Stress Testing"*, provides a strong empirical and theoretical foundation for the use of Lasso regression in financial modeling. The study highlights Lasso’s effectiveness in high-dimensional settings, particularly when the number of explanatory variables exceeds the number of observations—a common challenge in financial econometrics. The paper focuses on model selection and forecasting in stress testing, demonstrating how Lasso helps address multicollinearity, overfitting, and the curse of dimensionality in predictive models.

This project similarly employs Lasso regression to enhance model interpretability and predictive accuracy in financial applications. Like Chan-Lau’s work, it emphasizes automated feature selection, robust cross-validation techniques, and the ability to capture key economic drivers without overfitting. By applying Lasso in the context of asset pricing, risk modeling, and factor selection, this study builds upon the principles outlined in the IMF research, reinforcing the methodology’s relevance to modern financial forecasting and portfolio optimization.

Some relevant conclusions, derived directly from Chan-Lau´s journal (p29):

"Lasso regressions are likely to outperform traditional statistical models such as ordinary least squares in forecasting the performance indicators required in applied stress testing. The advantage of Lasso-type estimators lies in the fact that they can handle the complications arising from the high dimensional nature of stress tests.

Typically, the stress test specifies a large set of primary explanatory variables for which there are only few observations. This leads to a situation where the number of variables is of the same order of magnitude as the observations. Without regularizing (or shrinking) the coefficients, statistical models would be very unstable. In contrast, regularization methods, like Lasso, are designed specifically to handle high dimensional problems, many of which present in different areas of finance and economics, including financial networks."

★ **Category 2. Hierarchical Clustering.**

Advantages (Hierarchical Clustering):

1. The generation of a dendrogram that depicts the merging and splitting of clusters is one of the most significant advantages. This visual illustration assists the analyst in denoting the formations of the natural data as well as the multi-layer relationships of clusters.
2. Hierarchical clustering does not require the imposition of geometric assumptions about the structure of a cluster (e.g., spherical) which leads to a precise estimation of the potential structures. It can detect elongated, irregular, or otherwise non-standard structures effectively.
3. Because it creates a complete hierarchy of cluster merges, users do not have to pre-determine the exact number of clusters. Examining the different cut levels along the dendrogram allows for the selection of the most suitable count based on the context or purpose of the analysis.
4. As clusters have a nested structure, it is possible to notice some deep patterns, or sub-clusters. For example, if a big cluster contains a group of very similar items, it can be separated as a sub-cluster in hierarchical clustering in the process of ascending the merging tree.
5. Hierarchical clustering is easy to understand conceptually; it either starts with every data point alone (agglomerative) or starts with all data points together (divisive). This distinctness enables beginners to quickly learn the technique without being daunted by overly complex equations.
6. Several different methods of linkage, such as single, complete, average, and Ward's can be checked to find which one matches the user's data the best. This is a flexible method such that the researchers can change the linkage they choose based on the data characteristics and thus improve the cluster quality.
7. In agglomerative schemes, local outliers might only affect the clustering in a localized manner. It is also possible that they affect the convergence process, but still, the overall hierarchy is generally stable unless outliers are very far apart.
8. Due to its design, hierarchical clustering generates groups in a sequential merging process. This feature becomes incredibly important when the data analysis requires information at both

coarse and fine levels. For example, the general groups are identified first, and then individual categories are investigated.

9. Regarding smaller data collections, the memory and computation need are normally controlled and the dendrogram is an idea that can be revealed without any tangle. The researchers that explore the middle data level are quite interested in it for their first study.
10. This is an automatic cut at any point in the existence of the dendrogram by the users. If the first cut gives out too many clusters, the users can modify the threshold and bring them all into one; on the other hand, if it shows something that might be too simple, then most of the time switching the inception point makes them a bit separate the outliers. This active interaction is encouraging from the point of view of customization without having to redo the full operation from zero.
11. Hierarchical clustering results in well-defined clusters at every level, meaning it can be compared with the outcomes of partition-based clustering techniques (for example, k-means) and can be done together in the researchers' procedure. Overlaying cluster labels and visibly noticing congruence or incongruence between the two approaches can help one see how well they align.
12. The dendrogram method which is not so complicated to explain is relatively straightforward to use so that decision-makers without statistical or computer science knowledge can understand it. The idea that far-from-each-other “branches” merge is logical, thus making it easy to explain why clustering occurs.
13. The distance metric can be any distance measurement; thus, the algorithm is not bound to just one. The nature of the data can determine whether the algorithm uses Euclidean, Manhattan, cosine, or even correlation distances. This is particularly useful in fields such as text analysis or genomics.

Computation

We made a cluster analysis of real historical stock market data from SPY (S&P 500 ETF) and QQQ (Nasdaq-100 ETF). For the time frame of 2018-2023, we acquired adjusted closing prices from Yahoo Finance. The dataset was subjected to preprocessing using StandardScaler from scipy-learn to guarantee the standard normal distribution with an average of 0 and a standard deviation of 1, which facilitates the meaningful distance calculations.

For the clustering process, we adopted the Ward’s linkage method along with Euclidean distance, which is the one that ensures the minimum variance within the clusters. This procedure, on the other hand, guarantees that all the attributes are thus equal during the distance calculation. The hierarchical clustering model was generated via the SciPy linkage function, which resulted in a dendrogram that gives a visual representation of how data points are merged into clusters. Based on the dendrogram structure resulting from the above process, we set a cut-off threshold of 10 and used the fcluster function in order to determine the conclusive cluster assignments. The cluster labels that resulted from this process were then added to the original dataset.

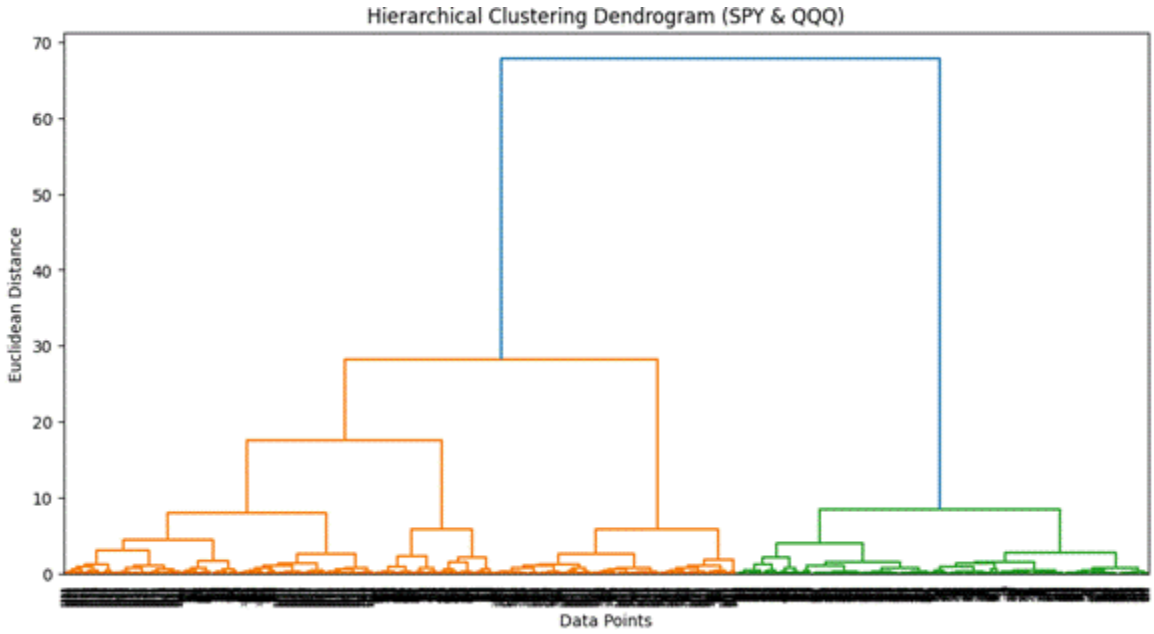

Fig. 7. Dendogram

The dendrogram above represents how the QQQ and SPY data points are linked together historically. The horizontal axis indicates every employee's odd observation in the data file, while the vertical axis shows the distance of cluster merging. Merging clusters lower on the Y-axis suggest high similarity scores. For instance, SPY and QQQ prices are clustered together at the very beginning of the analysis where a short distance between them is observed. The graver merger of two considerably larger groups at a higher distance indicates a trend across wider market conditions.

One of the main benefits of hierarchical clustering is the ability to set the cluster threshold. Making a cutoff at a lower level creates many more detailed groups, while with the higher one, the data is binned into broader categories. This property makes the hierarchical approach to clustering very effective for exploratory studies where the best number of clusters is unknown.

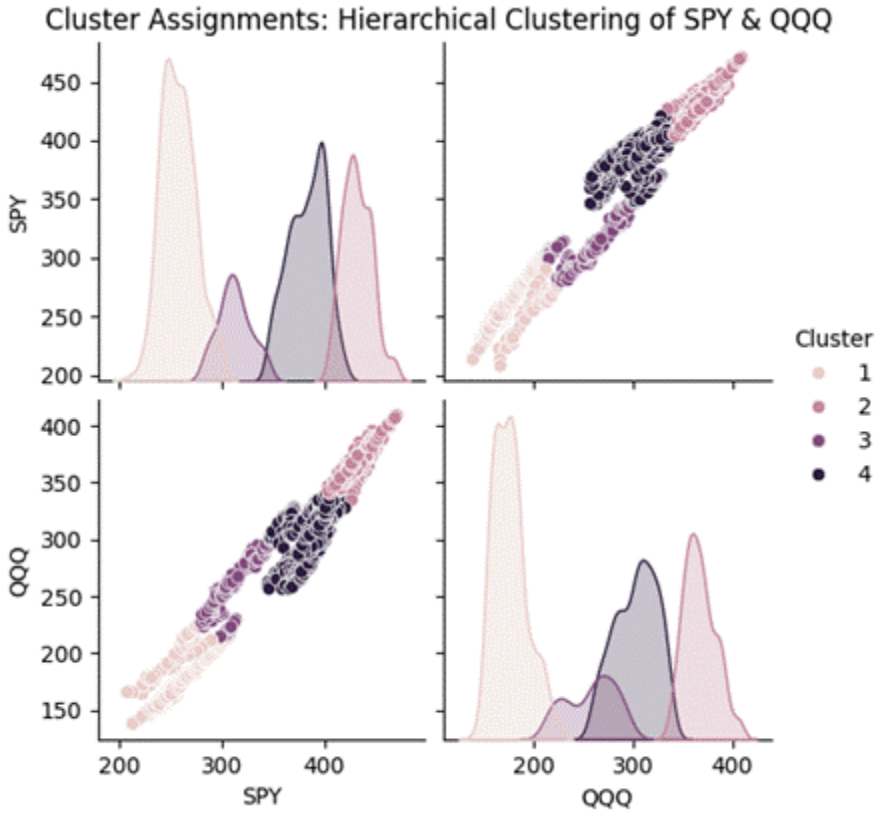

Fig. 8. Cluster Assignments

The pairplot above marks each distinct data point, representing a day's price observation and assigning different colors for different clusters. In addition, the diagonal subplots elucidate the kernel density assessment for each group, demonstrating how the prices are plenteously disseminated within each set.

The off-diagonal scatter plots illustrate the separation of clusters along two different parameters. Data points that are located in the same cluster mostly have similar prices on both the given indices. For instance, a certain cluster could indicate events where both SPY and QQQ went through a bull market. In contrast, another cluster could represent the periods of market corrections or the low-volatility phases of the market. This footage confirms the clustering structure by showing how hierarchical clustering can accurately segregate market situations based on historical price data.

Disadvantages (Hierarchical Clustering):

1. For substantial datasets, computing and updating the necessary distance matrix can be time-consuming and memory-intensive, often making hierarchical clustering infeasible without specialized optimizations (Jain et al.).

2. Traditional implementations maintain a proximity matrix of size n × n where n is the number of observations. This matrix must be updated after each merge or split step, which can quickly exceed available resources for very large data.
3. Although minor anomalies may have limited impact, strongly deviant outliers can cause incorrect merges or merges that deviate from expected groupings, especially in certain linkage methods like single linkage (which can produce “chaining” effects).
4. Users must visually inspect or apply heuristics to decide the “cut level” in the dendrogram. There is often no definitive statistical test to dictate the best cut, so domain knowledge and experience frequently guide the final choice.
5. Hierarchical clustering is not well-suited for constantly evolving datasets because adding a new data point effectively means re-running or re-calculating the merges. This is time-consuming in high-velocity environments.
6. The entire merging process is greedy. If a suboptimal merge occurs at an earlier stage, it can cascade, producing questionable groupings in subsequent levels—there is no guarantee of a global optimum.
7. In many real-world applications, variables might be numerous (thousands or more). Distances in high-dimensional spaces can become less meaningful, leading to degraded performance or meaningless clusters.
8. While it can use different distance metrics, mixing categorical, ordinal, and continuous data still poses challenges. It can be tricky to choose or design a consistent distance measure that works for all variable types.
9. When performing top-down clustering, the first split can significantly drive subsequent splits. A poor initial partition might lead to suboptimal final groupings.
10. True hierarchical approaches that meticulously recalculate cluster distances after each merge can have a time complexity of O (n^2 logn) or worse, lacking the scalability needed for tens or hundreds of thousands of observations.
11. Some linkage methods implicitly assume that the distances or the similarities could be a monotonic function. Hence, if a dataset contains any non-monotonic relationships, the merges that such methods do might not really reflect real-world groupings.
12. Automated methods (like the elbow method, the silhouette coefficients, or the gap statistics) are typical of partition-based clustering. In the context of dendrogram cutting, they can be more complicated to apply and less intuitive to use.
13. Single linkage may yield a silkworm-like (chaining in the jargon) group of data, while complete linkage is very prone to outliers. Certain well-known biases that each of the linkages has might also cause the clusters to be disproportionate, depending on the shape of the dataset.
14. Random noise in data can disrupt computed distances and hamper the formation of meaningful clusters. When features are extremely noisy, it can blur distinctions between groups.
15. Despite its flexibility, hierarchical clustering does not solve every clustering challenge. For extremely large datasets or real-time analytics, alternative algorithms (like k-means variants or density-based methods) might offer more practical performance or additional features.

Equations

Hierarchical clustering is based on defining the method of measuring the distance between groups of data points and on how these groups are merged through successive iterations. Below are some of the common linkage criteria (that is, how distances are computed between clusters) and distance metrics (that is, how distances between data points are measured) applied in hierarchical clustering.

Name	Equation	Explanation
Single Linkage	$d(C_u, C_v) = \min_{\{x_i \in C_u, x_j \in C_v\}} dist(x_i, x_j)$	Let C_u and C_v represent two clusters, each containing data points $\{x_i\}$ and $\{x_j\}$ respectively. We need a rule for determining $d(C_u, C_v)$, the distance between these two clusters. This distance is the **smallest pairwise** distance among any points in C_u and C_v. This often produces a "chaining" effect, where clusters can elongate.
Complete Linkage	$d(C_u, C_v) = \max_{\{x_i \in C_u, x_j \in C_v\}} dist(x_i, x_j)$	The distance is defined as the **largest pairwise** distance between any points in the two clusters. This can lead to more compact, spherical clusters but may make them more sensitive to outliers.
Average Linkage	$d(C_u C_v) = \frac{1}{(C_u) \times (C_v)} \sum_{x_i} \sum_{x_j} dist(x$	The distance is the **average** of all pairwise distances between points in C_u and C_v. This often provides a middle ground between single and complete linkage.
Centroid Linkage	$d(u, v) = \|\|cu - cv\|\| 2$	The distance between clusters is then the **Euclidean distance** between their centroids.
Ward's Method		Compared to regular distance measures, Ward's method merges clusters by calculating the amount of increment in total within-cluster variance. This frequently results in more balanced clusters in the real world as it minimizes the squared distances of each participant from their like group centroids.

Distance Metrics		
Euclidean Distance	Squareroot $((x_2 - x_1)^2 + (y_2 - y_1)^2)$	This refers to the shortest distance between two points in Cartesian space.
Manhattan Distance	Dist (x,y) = $\sum_i \lvert x_i - y_i \rvert$	
Maximum Distance	dist(x,y) = $\max_i \lvert x_i - y_i \rvert$	
Cosine Similarity	sim(a,b) = $(a \cdot b)$ /($\lvert\lvert a \rvert\rvert \cdot \lvert\lvert b \rvert\rvert$)	
Canberra Distance	$\sum_i \lvert a_i - b_i / (\lvert a_i - b_i \rvert)$	
Hamming distance	dist(x,y)=i$\sum_I$(xi ≠yi)	Counts the number of coordinates where two vectors differ. For instance, for binary or categorical data

Features of the model (Hierarchical Clustering)

Handling Missing Values

1. The algorithm can recognize patterns in incomplete data and estimate data features without relying on the complete set of records.
2. It even permits the computation of pairwise distance on incomplete data, allowing for clustering despite some missing entries.
3. Some techniques make it possible to use the data as they are without the need for inventing values, therefore, the original data structure can be preserved.
4. Pairwise direct comparisons allow for a resilient clustering mechanism that can be operational even when specific data points are unavailable.
5. A combination of these two makes it possible to use a method that is more tolerant of variable data, and to compose the dataset as a whole by not getting rid of any way to capture information.
6. Hierarchical clustering can include missing data through enhanced distance measures that disregard or weigh less the dimensions without input. It is compatible with the methods that automatically compensate for the missing data and thus minimize the bias risk that could originate from informal techniques.

Dendrogram and Other Features

1. The method produces a dendrogram that is used to chart a visual tree showing how the separate data points merge into clusters which is intuitive and provide an overview of the clustering process.
2. Dendrograms allow users to follow the tree of clusters from the narrowest levels to the broader groupings.
3. They allow easy determination of the best number of optimal clusters by selecting a cut-off point for the vertical axis.
4. The nature of the visualization explicitly illustrates the amount of dissimilarity of clusters through the height of the merge lines, thus supporting its interpretation.
5. It can allow a wide variety of linkage methods (single, complete, average, Ward's) for hierarchical clustering, and the connections among the clusters can be defined flexibly.
6. The dendrogram's structure makes it easier to detect outliers. If points merge at such distances, this usually indicates anomalies.
7. Its nested arrangement allows for a deep understanding of the data's organization, demonstrating micro and macro patterns within the dataset.

Guide: List of Inputs and Outputs

Inputs:

Data Matrix or Distance Matrix:
A fundamental data representation is a prerequisite for hierarchical clustering. It can be the original data matrix displaying instances (rows) and features (columns) or a distance matrix showing the pairwise dissimilarities between data points. For a data matrix, the algorithm calculates distances based on a selected metric, while for precomputed distance matrices, it is possible to use them directly, thus avoiding the need for redundant computations and saving time.

Linkage Criteria Specification:
Users need to specify how to estimate the distance between clusters and the linkage criterion. The available choices are single linkage (which takes the least distance between two clusters), complete linkage (which takes the distance of the objects which are farthest apart), average linkage (which is the average of distances of all points within clusters), centroid linkage (which takes the centroid distance of clusters), and Ward's method (which the increase of total variance is minimized). This selection is decisive in the structure and the interpretability of the dendrogram generated.

Outputs:

Dendrogram:
The major output is a dendrogram, which is a branching diagram. It shows how individuals or smaller clusters become larger over time. The vertical part of the dendrogram demonstrates the level of

dissimilarity at which the combinations are made, allowing the analysts to see clearly the natural hierarchy in the data and to find an appropriate cutting point that best categorizes disparate groups from the whole dataset. Thus, the method is valuable for analyzing data.

Cluster Assignments:
When a decision is made about the point to "cut" the dendrogram, each observation is given a cluster label under this threshold. The ultimate cluster assignment is done, and the data set is partitioned into groups of similar observations, which can be analyzed later on for patterns or insights pertinent to the research.

Hyperparameters that need tuning.

Number of Clusters (Dendrogram Cut Level)
This is frequently viewed as the primary "hyperparameter" by experts. Analysts do not state a cluster count at the start but instead analyze it post hoc using conditions such as the dendrogram or rules such as the silhouette coefficients, gap statistics, elbow method, etc.

Linkage Method
Using single, complete, average, or Ward's linkage can result in dramatically distinct outcomes. A common practice is to assess several linkage algorithms to find the one that best reflects the data's background.

Illustration:

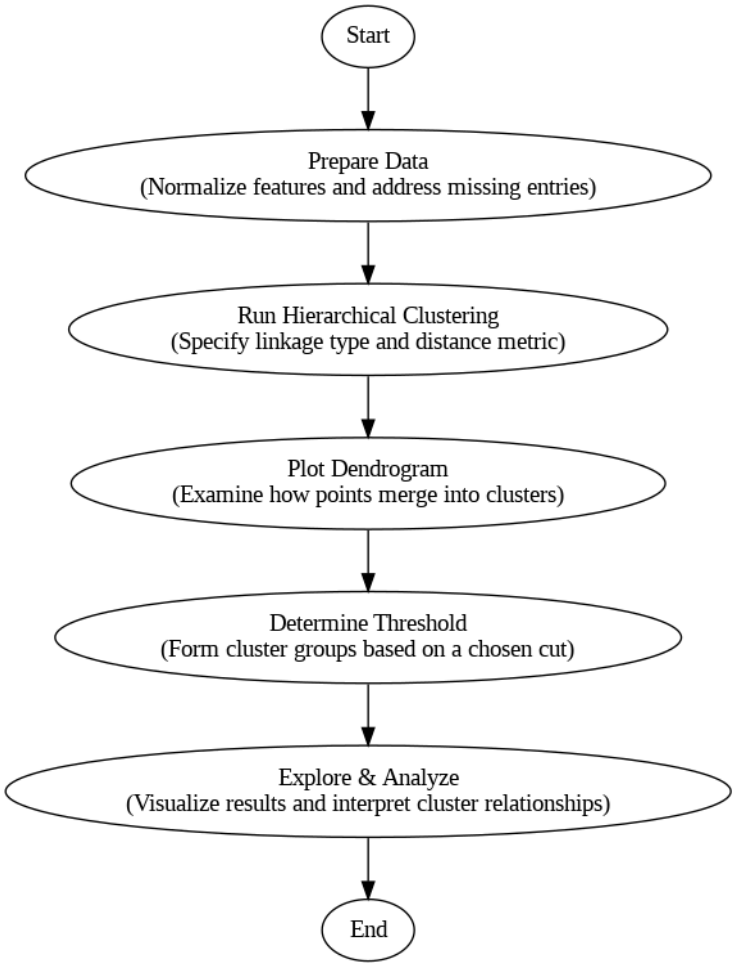

Fig. 9. Hierarchical Clustering Process Flow Diagram

Start: This step begins the clustering process, determining how the data will be prepared and how the analysis will be conducted.

Prepare Data: All data features undergo standardization or normalization to be brought to the same scale. Additionally, any missing values are taken care of because these values may disturb the outcome of the clustering process.

Run Hierarchical Clustering: In this step, users can choose the linkage criterion (e.g., Ward, single, and complete) and the distance metric (e.g., Euclidean, or Manhattan) according to their wishes. These elements essentially and gradually define how the clusters are recognized and fused.

Plot Dendrogram: A dendrogram is generated that visually represents how individual observations gradually fuse into larger clusters as the distance threshold rises. This visual representation helps identify the natural grouping within the data.

Determine Threshold: The dataset is divided into clusters through a dendrogram cut (i.e., a pre-determined distance level is elected). Observations below a certain distance are grouped into the same group.

Explore & Analyze: Visualizations such as pair plots or scatter plots can explain newly formed groups. In this phase, the focus is on the strong effectiveness within and between the groups, while anomalies and patterns are also particularly emphasized.

End: At this point in the workflow, the clusters are precisely defined, and the knowledge gained here sets the stage for the next steps, which may include reporting the results, performing further analyses, or merging the analysis results with the overall company data strategy.

Journal (Guam and Jiang)

Abstract/Introduction

In this research paper, the authors are handling the clustering of financial time series, namely, stocks to design a wide-ranging portfolio under Markowitz's risk-return framework. They propose a new similarity measure that is specially designed to help identify the dissimilarities among individual stocks, thus allowing for the selection of a diversified set of securities that minimizes the overall portfolio risk for a given return. At the same time, applying a group-ward hierarchical clustering technique, the paper indicates how adopting a portfolio method, total grouping of stocks would, through numerical examples, avoid being a costlier alternative to brute-force aids. The savings brought about by the net of the outputs from the model are always a plus point in the way of portfolios being competent users of data science with different oncogenic sensitivity to change.

Methodology

The fundamental innovation is a personalized distance choice that aims to find the main differences between a pair of stock return sequences. To do this, the authors use the hierarchical group-ward classification, which classifies stocks into groups with remarkably similar patterns. A proper application of this concept is to take one object (one stock) from each subgroup and make investors create a much bigger range for themselves than a random pick of shares or the shares of one group. The technique is tested through experiments with real-world stocks. The researchers affirm the methodology's success by applying the calculations of the mean return rates and the standard deviations (using the Markowitz methodology) as evidence that the clustering method of choosing stocks is the way to go. It is an excellent means of putting together an optimum portfolio.

Findings and Results

Tests conducted empirically disclosed that selection based on clustering reduces the level of risk for a particular return while also decreasing the amount of the required calculations in terms of the portfolio. The group-ward method differentiates stocks to a great extent. It thus is also the most time-efficient way of achieving the optimal solution on the efficient frontier compared to other techniques like the random selection of stocks or choosing from a single cluster. Based on applying the multiple clusters method of stock selection, the findings demonstrate a lower variance than just a single cluster, thus confirming the applicability of the new similarity metric. As a result, it is a more practical and efficient method, which will substantially improve the portfolio in real-world investment situations.

★ **Category 3. Topic: Principal Components.**

a. **Advantages:** The benefits of using this methodology are as follows:
   - **Dimensionality Reduction**: The main aim of the PCA is first to reduce the dimensionality of a multivariate dataset while retaining the essential variation present in the dataset. It essentially reduces the noise and complexity in the dataset and improves the model's performance.
   - **Simplify Data Visualization:** By simply reducing data dimension, it enhances visualization of high-dimensional data structures.
   - **Removes Multicollinearity:** Multicollinearity is the result of a high correlation between independent variables in the dataset, which describes the same underlying phenomena. PCA helps to remove multicollinearity from the dataset by transforming correlated data into uncorrelated principal components.
   - **Improves Model Computational Efficiency:** PCA helps reduce the number of model parameters, simplifies model training time, and consequently improves model performance.

b. **Computation:**

   To illustrate the Principal Component Methodology, we will consider the high-quality market (HQM) corporate bond spot rate data sourced from Fred Economics at different maturity dates (from 1 year to 50 years)

   According to Fred Economic Data, *Spot rate* for any maturity is defined as the yield maturity of a corporate bond that gives a single payment at maturity. This implies that we'll be dealing with a zero-coupon bond for the model computation. The HQM (High-Quality Market) methodology computes the spot rates of corporate bonds so as to ensure that interest rates remain consistent with other high-quality bonds. HQM helps identify strong bonds in the market by using different metrics such as bond per yield and spot rates. In this computation scenario, rates will be used.

We will consider spot rates with maturities of 1 year, 2 years, 3 years, 5 years, 7 years, 10 years, 15 years, 20 years, 25 years, 30 years, 40 years, and lastly, 50 years. For this scenario, we'll consider:

- 1–5 years as *Short-term* spot rates
- 7- 15 years as *Medium-term* spot rates &
- 20 - 50 years as *Long-term* spot rates

As discussed previously, the Principal Component Analysis Methodology is a dimensionality reduction technique that employs the use of eigen-decomposition of the covariance matrix of the dataset to generate principal components. Interestingly, two approaches were used in the computation of the PCA algorithm. Steps followed to implement dimensionality reduction:

- **First approach:**
  - **Perform general exploratory data analysis:** on the original dataset, detrend the dataset, and implement data visualization afterward. The dataset is detrended by using mean normalization.
  - **Derive covariance matrix of the dataset**
  - **Implement eigendecomposition** of covariance matrix, determines the eigenvalues, explained variance, and eigenvectors (factor loadings). Identify the top factor loadings and visualize them (Factor loadings for PC1, PC2, PC3). Factor loadings show the movement of the yield curve. The first factor loading (PC1) represents a parallel level shift in the yield curve In our computation, the interest rate encountered a decline from year 1 to year 7 by 0.031 points but positive upward movement thereafter. The second-factor loading (PC2) reflects the slope change; however, the slope continued to steep downwards from 1st year to the 50th year. Lastly, the third-factor loading (PC3) depicts curvature shifts across maturities. In the computation, there was a shift in the curve in the 10th year.
  - **Visualize scree plot:** a plot of declining explained variance against individual principal components. From the plot (as shown below), PC1 and PC2 represent 99% of the cumulative variance and are therefore significant, whereas PC3 up to PC12 is insignificant for modeling.
  - **Compute the principal components** of the top factor loadings (PC1, PC2, & PC3). Visualize the variance across different maturities (as shown below). PC1 has the highest volatility (from the model computation)
  - Additionally, a PCA algorithm was run on reconstructed data (both from the original dataset and the factor loading), and similar results were obtained for all of these cases.
- **Second approach:**
  - On a second approach, we ran the above-mentioned steps on the rescaled dataset (i.e. dataset standardized using StandardScaler). *StandardScaler* is a library in the *skearn.preprocessing* module used for feature scaling. The PCA model is then fitted (*using PCA().fit*) on the rescaled data to generate principal components.

- Derive the correlation matrix for the loadings showing the uncorrelated relationship amongst the principal components.
- Finally, we ran the PCA algorithms using only the first 2 principal components since they represent the highest cumulative variance of the dataset. Upon graphing these PCs on a biplot, we observed that:
    I. PC1 and PC2 represent 99.31% of the variance; PC1 possesses the highest proportion—93.0% of the variance. This is pretty similar to previous approaches.
    II. From the biplot (as shown in the Fig,), we can see long-term interest rates are highly correlated, even more than short-term rates. This is because the angle between them is smaller. Comparing short-term and long-term interest rates, we can conclude that they are positively correlated as the angle between them is wider. The mid-term interest rates seem not to be correlated with both the short- and long-term interest rates.

c. **Disadvantages:**

Some of the challenges associated with using Pricncipal Component Analysis Methodology are as follows:

- **Sensitivity to Outliers:** Variation in the dataset affects the PCA algorithm. Hence, data requires standardisation or feature scaling before running a PCA. Applying simple standardisation techniques such mean-subtraction of feature scaled help mitigate high variance in the dataset
- **Curse of Dimensionality Reduction:** While the goal of PCA is to reduce the dimensions in the dataset into the most significant information, other relevant information may be lost in the process.
- **Limited to Linearity:** Principal Component Analysis implements linear transformation on original features generating principal components. Non-linear variable relationships and missing data values complicate the data structures and tend to require more complex forms of PCA algorithms, such as: Kernel PCAs, Incremental PCAs, Probabilistic PCAs etc. Non-linear PCA algorithm may not perform well on the dataset.
- **Complicates Interpretability of the Extracted Information:** The transformed principal components are linear combinations of original features, making it difficult to interpret the results. The reconstruction of the dataset from factor loadings at the tail end of algorithm implementation complicates the interpretability of the results.
- **Poor performance on Highly Correlated Datasets:** The essence of PCA is primarily to reduce dimensions, extract features, and lastly to remove correlations in the original dataset. Thus, PCA may perform poorly on these kinds of datasets.

d. **Equations:**

Mathemically, Principal Component Analysis depends upon the eigen-decomposition of positive semi-definite matrices and upon the singular value decomposition (SVD) of rectangular matrices. It is determined by eigenvectors and eigenvalues. *(Sidharth et. 2017,)* There are equations for each step of the computation process— from standardisation of the dataset to generation of principal components. To explain how the model works using equations, we'll document the equation in each procedural step:

- **Standardization of the Dataset:** Generally, the essence of standardisation is to remove sensitivity to outlers. Standardisation is performed simply by two steps:
  i. ***Mean (Z-score) normalization*** (i.e. the ratio of the daa point difference from the mean and standard deviation). Mathematically, it computed as:
  - $X_{scaled} = \frac{X - \mu}{\sigma}$

  $where$:
  $X_{scaled} = Standardised\ data\ matrix$
  $X = Orignal\ data\ matrix$
  $\mu = Mean\ of\ the\ dataset$
  $\sigma = Standard\ deviation$

  Ii. **StandardScaler fit** from sklearn.preprocessing can be applied to also standardise the dataset.

- **Compute Covariance Matrix:** The goal of using a covariance matrix is to indicate the directions along which variance is maximized. Mathematically, Covariance is computed as:

$$X_{cov} = \sum_{i}^{n} \frac{X_{scaled} X^{T}}{m}$$

$where$:
$X_{cov} = Covariance\ of\ data\ matrix$
$X_{scaled} = Standardised\ data\ matrix$
$X^{b} = Transpose\ of\ the\ standard\ matrix$
$m = Numnber\ of\ samples.$

- **Eigen-decomposition:** In this step, we derive eigenvalues and eigenvectors to analyze the $n\,x\,n$ square matrices. Eigenvalues and eigenvectors are numbers and vectors that responsible for square matrix analysis.
  Let X be a $n\,x\,n$ square (co variance) matrix The eigenvalue of X, λ is determined by a characteristic equation:

$$determinant(X - \lambda I) = 0$$

$where$:

$I = Identity\ matrix$ (a diagonal matrix of 1s and off-diagonals of 0s.)

The eigenvector $\bar{x}$ is computed as:

$$X\bar{x} = \lambda\bar{x}$$

Normalize the eigenvectors and the dataset X into $X_{transformed}$ such that:

$$X = \bar{x}^{T} \lambda x$$

$where$:

$X = Original\ Data\ matrix$

$\bar{x}^{T} = transposed\ eigenvector$

The first set of columns of the eigenvectors form the *factor loadings* required for generating principal components (*PC1, PC2, & PC3*).

- **Derive Explained Variance:** Explained variance is simply the proportion of the total variance explained by each eigenvalue in the eigenvectors (factor loading). Mathematically, it is computed as

$$Explained\ Variance = \frac{\lambda_i}{\sum_{i=1}^{n} \lambda_i}$$

- **Derive Reduced Data:** Here, a a new dataset with reduced dimensions is generated from the combination of a transformation matrix (containing k number of principal components). Consider,, the transformation matrix $W = (pc1, pc2,..., pck)$

$$X_{reduced} = WX$$

$where$:

$X = Original\ Data\ matrix$

$X_{reduced} = Reduced\ Data\ Matrix$

Note: $X_{reduced}$ should be uncorrelated (orthogonal) matrix

e. **Features of Principal Component Analysis (PCA)**

The main features of the PCA model are:

- **Dimensionality Reduction**: The main aim of applying the PCA model to any dataset is to reduce the number of variables in the dataset while retaining essential information about the dataset.
- **Linear transformation:** High-dimensional dataset is linearly transformed into reduced dataset using eigenvalues and eigenvectors

- **Orthogonality of resulting principal components:** The linear transformation of the dataset generates uncorrelated (orthogonal) principal components that are representative of the original dataset.
- **Standardisation:** The standardisation of the dataset at the initial stage of model implementation, helps mitigate sensitivity to outliers and noise, and detrend original dataset.
- **Reduced Interpretability of the Data:** The resulting principal components from the PCA model are not easily interpretable.

f. **Guide:**

- **Input: Standardised Data Matrix**
  The fundamental requirement for Principal Component Analysis is the ***Standardized Data matrix*** with multiple variables. Eigenvalues and eigenvectors help transform the data into simple output.

- **Output: Principal Components**
  The output of the PCA model are principal components of a reduced data matrix that clearly retains the information in the initial (high-dimensional) dataset.

g. **Hyperparameters:** Some of the hyperparameters that may require tuning are as follows:

- **Scaling methodology:** The method of scaling or standardization of the original dataset could either be done by *Mean subtraction (Z-score normalization)* or *Feature scaling (using StandardScaler.fit).* Both methods were applied in the two approaches in the Jupyter notebook model computation
- **Eigen-decomposition Solving method:** Eigen-decomposition can be carried out using either the conventional numpy computation *np.lig.eig()* on the covariance matrix or by using the sciki learn *svd_solver* library,. In addition, the *svd_solver* library could be tuned in different ways depending on the dimension of the dataset. Solver options such as “auto” selectthe best solver, “full” is used for full eigendecomposition of small-scaledatasets “argpack” applies for sparse eigendecomposition of large datasets.
- **Number of Principal Components:** Deciding on the number of principal components to retain can be done using *the explained variance* ratio as a tuning strategy. If only the first 2 factors explain the variance in the dataset, the PCA can be tuned to model on 2 principal components using *pca(n_components=2).*
- **Transformation method:** While the linear PCA transformation method is the only PCA type described in this report, there are other PCA methods that apply to non-linear and robust data structures. For a non-linear (kernel) PCA model, we’d have to stipulate the *kernel* to use (e.g. "linear," “poly”, etc.).

h. **Illustration:**

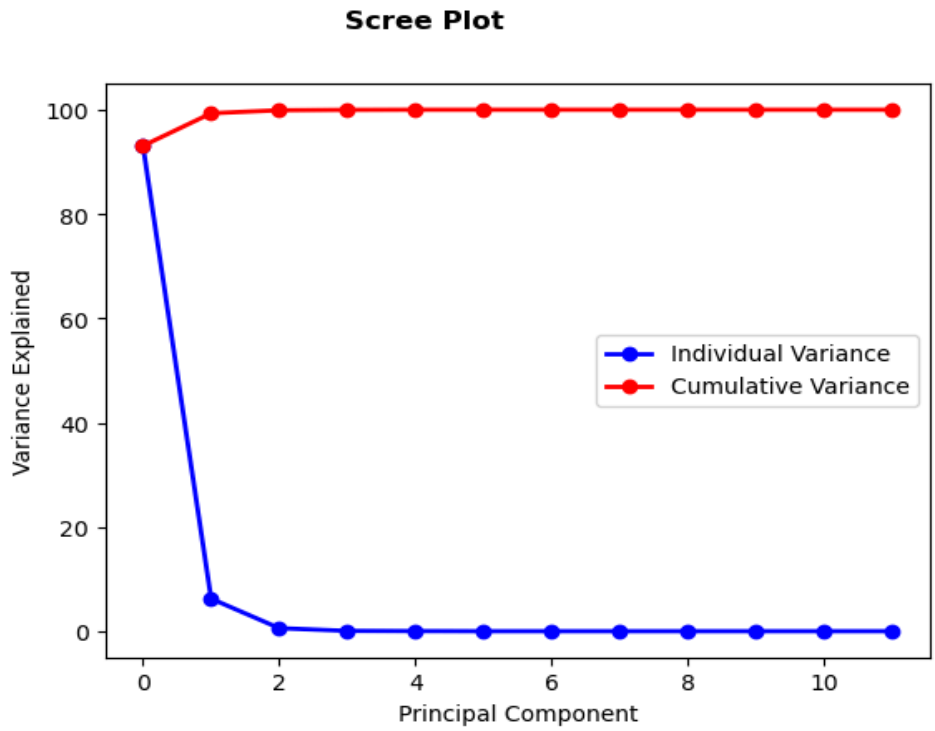

Fig. 10. Scree Plot

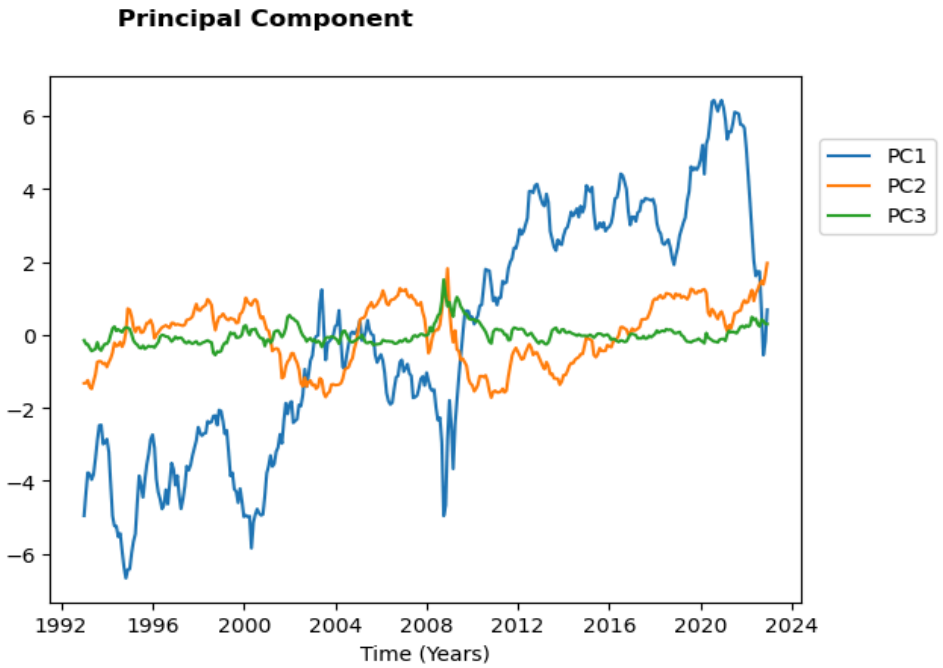

Fig. 11. Principal Component Plot

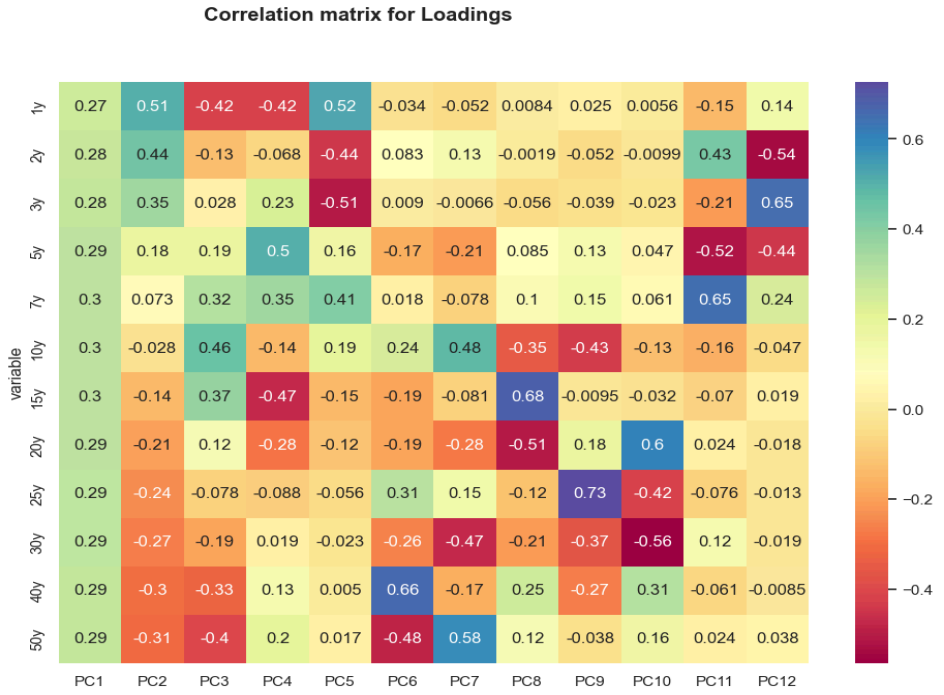

Fig. 12. Correlation of Loadings

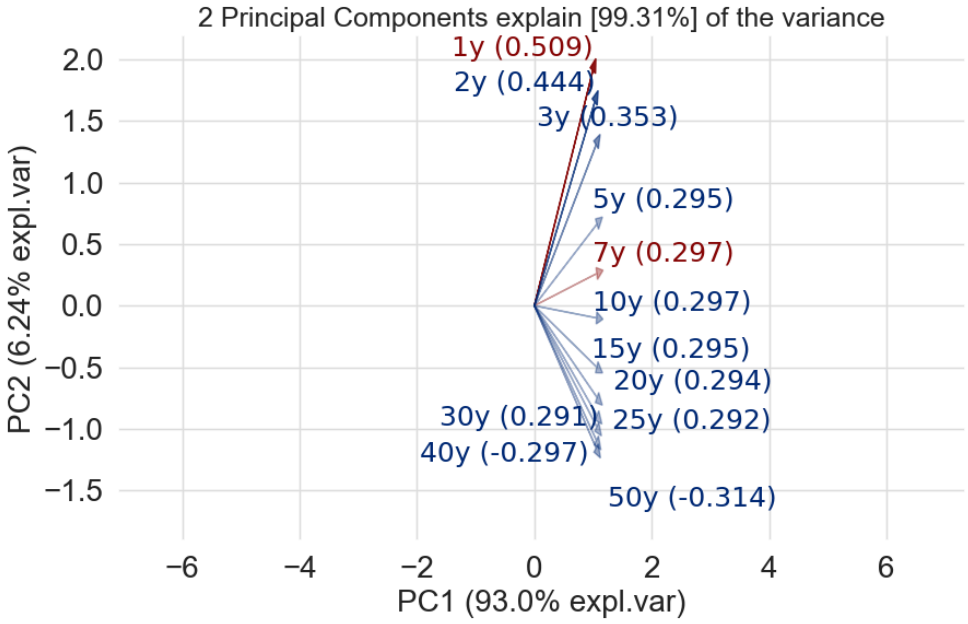

Fig. 13. Biplot

i. **Journal:**

- Mishra, S., Sarkar, U., Taraphder, S., Datta, S., Swain, D., & Saikhom, R. et al. (2017). Multivariate Statistical Data Analysis- Principal Component Analysis (PCA). International Journal of Livestock Research, 7(5), 60-78. http://dx.doi.org/10.5455/ijlr.20170415115235

This journal article describes the computational procedure for the PCA model. It elucidates and clearly defines all of the terms and parameters required to generate principal components for any dataset. It also includes the mathematical formulations for each term and parameter required in the model computation.

Generally, for the model computation, it applies the PCA algorithm for made-up data matrix without coding. The model computation follows similar steps as ones discussed overleaf, however, the model calculations were implemented mechanically using mathematical computations.

The article posits that the PCA is a non-parametric analysis - as there are no parameters to tweak or coefficients to tune to suit the user's experience. This implies that any form of PCA—linear or non-linear—is rigid, and the user must select all of the parameters and decide on the most appropriate assumption before running the algorithm. Secondly, the article also states that the first k-dimensions explain the dynamics of the dataset, which is obviously consistent with the result of our modl implementation in the Jupyter notebook.

Step 3: Technical Section

In this section, we will discuss the general process of hyperparameter tuning in machine learning models, with a focus on the 3 selected models: LASSO regression, Hierarchical Clustering, and Principal Components. Hyperparameter tuning is a critical step in optimizing model performance, as it involves selecting the best set of parameters that control the learning process. These parameters are not learned from the data but are set before training the model.

Hyperparameter Tuning in Machine Learning

Hyperparameter tuning is the process of finding the optimal values for the parameters that govern the behavior of a machine learning algorithm; unlike model parameters (e.g., coefficients in linear regression, hyperparameters are not learned during training but are set by the researcher. The goal of tuning is to find the hyperparameters that result in the best model performance, typically measured by metrics such as accuracy, mean squared error (MSE), R^2, etc.

A practical method for hyperparameter tuning optimally configures a model, thus ensuring possible performance and keeping overfitting and underfitting low and minimal. Methods in use for hyperparameter tuning include:

(i) The grid search is a computerized mechanism that depth checks every combination of a group of values formed by priorly defined hyperparameters. Although comprehensive, it is very cost-inefficient as it uses a lot of computing resources.

(ii) The approach of random search is a randomized selection of combinations of hyperparameters based on some specified ranges [the approach] is much more computationally efficient than a grid search it also helps identify strong combinations of hyperparameters in a shorter time than others.

(iii) Bayesian optimization takes a probabilistic approach to make use of the data gained thus far in the experiment to explore more the hyperparameter space pointing out the most fruitful parameter settings to test inserting previously collected performance data.

(iv) Automated Machine Learning (AutoML) is an automatic engineering process of hyperparameter selection and mining, building it entirely on a machine schedule that takes place without manual for a not necessarily optimized selection of tools. Thus, the problem of optimizing the design parameters is simplified.

Common techniques for hyperparameter tuning specific to our models include:

1. **Cross-Validation:** often used in conjunction with hyperparameter tuning to ensure that the model generalizes well to unseen data. In k-fold cross-validation, the data is split into k subsets, and the model is trained k times, each time using a different subset as the validation set and the remaining data as the training set. The average performance across the k folds is used to evaluate the hyperparameters.

2. **Dendrogram Inspection**: This method utilizes the dendrogram, an illustrative tree diagram produced through hierarchical clustering. Analysts determine the natural "cut-off points" by observing the significant changes in the vertical distances (heights) between the cluster merges. The height at which the analyst "cuts" the dendrogram determines the number of clusters. This relies on the judgment of domain experts and the context of the data. Therefore, it is flexible in cases where numeric methods (such as the elbow method or silhouette analysis) cannot convincingly suggest an optimal number of clusters.

3. **Defining number of Principal Components:** The number of principal components is the most essential parameter that may require tuning in a PCA model. Modelers use the explained variance ratio for each factor loading component to determine the number of principal components to be used in the models. Essentially, all the preceding computation steps, ranging from the dimensionality of the data matrix to standardisation and transformation methods, all of these factors contribute to final decision-making on the number of principal components to be used in the model.

LASSO Regression: Its most critical hyperparameter is the regularization parameter, often denoted as λ or alpha (α). This parameter controls the strength of the L1 penalty, which shrinks some coefficients to zero, effectively performing feature selection. The choice of λ is crucial because it balances the trade-off between model complexity (number of non-zero coefficients) and model fit (how well the model explains the data). For a more detailed discussion of the hyperparameters used in LASSO regression,

including the regularization parameter (α), cross-validation for lambda selection, and the importance of standardization, refer to Category 1. LASSO Regression. Section g. Hyperparameters. In our implementation, we used 5-fold cv to tune the α parameter. The optimal α was selected based on the cross-validation results, and the model's performance was evaluated using metrics such as R^2 and RMSE (Root Mean Squared Error). By carefully tuning the hyperparameters, we were able to build a LASSO regression model that balances model complexity and predictive accuracy, making it a powerful tool for financial applications such as asset pricing and risk modeling.

Hierarchical Clustering: Although hierarchical clustering does not involve conventional hyperparameters like predictive models, making several key decisions helps as "hyperparameters":

(i) Distance Metric: A function that measures the similarity or dissimilarity between data points. The most popular metrics, such as Euclidean distance, Manhattan distance, and Cosine distance, can heavily affect clustering results.

(ii) Linkage Criterion: The method for measuring and joining distances between clusters is defined as:

Single linkage: Minimum distance between clusters
Complete linkage: Maximum distance
Average linkage: Average of all pairwise distances
Ward's method: Minimizes intra-cluster variance

(iii) Number of Clusters: In contrast to those techniques that rely on predefined clusters, hierarchical clustering is, however, not dependent on a defined number of clusters beforehand, but it selects a cut-off point of the dendrogram. The decision methods include:

Elbow Method: Point where the reduction rate of variance significantly declines is determined.
Dendrogram Inspection: The natural divisions of the clusters are identified visually.
Gap Statistic/Silhouette Score: Methods of statistical metrics used to show how many clusters are the most appropriate.

Principal Component Analysis: Although PCA is often seen as *non-paametric analysis—one that has no parameters for tuning* (Mishra et. al, 2017). However, quantitative analysts may decide tweak a number of methods through which computational steps are executed. Some these include:

- **Number of Principal Components:** The tuning strategy applied here is determined factors like variance explained and scree plot. Variance explained is the percentage of total variance retained in each component. The Scree plot, on the other hand, helps visualise the elbow point—the point where the explained variance starts to flatten out. The number of components before the elbow represents the *k* number of principal components required for modelling.
- **Other tuning techniques:** The scaling technique can be performed with methods like mean subtraction, feature scaling, min-max normalization on the original dataset before generating principal components. Additionally, eigendecomposition could also be perfromed either by the numpy computational function or the scikit-learn solver method.

Step 4: Marketing Alpha

In this section, we will highlight the advantages and features of the selected machine learning techniques and demonstrate how these methods can be leveraged to create robust trading strategies for investment managers to showcase the potential of machine learning in finance and how it can be used to generate alpha, that is excess returns above a benchmark.

The Power of Machine Learning in Finance

ML has revolutionized the field of quantitative finance by enabling the development of sophisticated models that can uncover complex patterns in financial data. Traditional statistical methods often struggle with high-dimensional datasets, where the number of predictors (features) exceeds the number of observations. ML techniques, such as LASSO regression, Hierarchical Clustering, and Principal Components excel in these scenarios by automatically selecting the most relevant features and reducing overfitting.

LASSO Regression is a powerful machine-learning technique that combines the benefits of linear regression with feature selection. By adding an L1 regularization term to the regression objective function, LASSO shrinks the coefficients of less important features to zero, effectively removing them from the model. This makes LASSO particularly well-suited for financial applications, where the number of potential predictors (such as macroeconomic indicators, asset returns, and volatility measures) can be very large.

Key Advantages

1. LASSO selects relevant predictors, reducing overfitting and improving model interpretability, crucial in finance.
2. It excels in high-dimensional data, making it ideal for factor models and asset return prediction.
3. By addressing multicollinearity, LASSO ensures stable models by selecting one variable from correlated features.
4. Its computational efficiency and cross-validation for hyperparameter tuning make it practical for large financial datasets.
5. LASSO's sparse models enhance interpretability, helping investment managers identify key drivers of asset returns.

Real-World Applications

- Asset Pricing: LASSO can be used to identify the most relevant factors driving asset returns, helping investment managers build more accurate pricing models.
- Risk Modeling: By selecting the most important risk factors, LASSO can improve the accuracy of risk models, enabling better portfolio risk management.

- Portfolio Optimization: LASSO can be used to construct sparse portfolios by selecting a subset of assets that provide the best risk-adjusted returns, reducing transaction costs and improving portfolio performance.

Hierarchical Clustering: important machine learning method used in financial analytics, allowing investment managers to find natural clusters within the market data, improve portfolio diversification, and strengthen risk management. Hierarchical clustering is different from classical clustering procedures that define the number of clusters beforehand. Its main advantage is the visual representation of the data that it provides through a dendrogram. As the data does not have to be predetermined, the analyst can identify the inherent structure in the data. Thus, this method enables the following: exploratory data analysis, portfolio structuring, and asset classification (segregation of stocks and bonds into categories)

1. Market Regime Identification

- Advantage: Using hierarchical clustering, a unique market regime can be very well identified. The method groups such periods by applying the same strategy to time series data as by adding three values: volatility, returns, and macroeconomic conditions.
- Real-World Application: Investment managers utilize segmentation of financial databases to identify bull, bear, and sideways market conditions leading to effective trading mechanisms.

2. Portfolio Diversification and Asset Allocation

- Advantage: This method also provides insights into which class of the asset is related to the other, and hides multiple classes on one, thus diversifying and ensuring safety through group clustering based on similar returns.
- Real-World Application: Risk-based portfolio construction is the area specifically where hierarchical clustering is widely used to form both three traditional major categories (stocks, bonds, and cash) and two alternative ones (real estate and commodities such that they are positively or negatively correlated).

3. Algorithmic Trading Strategy Optimization

- Advantage: Traders can leverage statistical clustering techniques to autonomous trading strategies. They cluster stocks, demanding similar price moves.
- Real-World Application: Statistical arbitrage funds exploit the method of hierarchical clustering to gather assets with a similar risk-return profile and ultimately optimize the pairs trading and mean-reversion strategies.

4. Credit Risk and Customer Segmentation

- Advantage: Clustering techniques provide certainly many-sided indices to break ``the elegance of the analysis, that is, a scoring of quality, stability or risk the prediction that fits the most the global risk of the bank's products' risk and then taking the retail customer and consumer product productivity for the two firms, Chrysler and Ford(the bloated one).

- Real-World Application: Banks and fintech companies ignore professional customers and offer heavy duty commercial base aids to financial engineers who can quickly improve quality of life for them that will consequently generate better returns.

5. Mergers & Acquisitions (M&A) and Industry Classification

- Advantage: Studies the existence of bubbles in certain industries, performing the so-called Internet Search and extracting such data from the Philip Morris web pages, such as the news articles: This step identifies the group's selection, which is the Time Premium, the Lifetime of the Company, the price paid, the Investment Company Consortium, etc..
- Real-World Application: The hierarchical clustering method was applied to a large set of already collected data, both publicly available and in-house, such as corporate fundamentals, financial ratios, and revenue structures, to determine the optimal M&A targets.

6. Sentiment Analysis in Financial News & Social Media

- Advantage: Sort out all relevant articles promoting or decrying a Companies products on social media platforms, earning reports and news excerpts according to emotion context by means of the principal component analysis.
- Real-World Application: Hedge funds control so-called hedging funds or have made the others holding the same kind of stock by sending misinformation via email on social media pages with the objective of getting rid of a team's huge short position as soon as possible.

Principal Component Analysis**.** The Marketing Alpha for PCA is solely determined by its key advantages and added value it brings in real-world applications. Some of the areas where PCA is applied is listed below:

1. **Market Segmentation:**
   Key advantages: PCA is useful for filtering redundant information and retaining.
   Real-world application: PCA is used to improve the clustering of customer market segment simply by reducing the noise in the dataset.

2. **Portfolio Risk Management:**
   Key advantages: PCA adds value essentially by reducing the dimension of the multivariate dataset.
   Real-world application: PCA is a powerful statistical tool for helping investors limit their investment portfolio to only the most profitable financial assets in their portfolio. Consequently, it helps investors manage risks.

3. **Credit Risk Modelling:**
   Key Advantages: PCA essentially extracts only the most import data features in the dataset.

Real-world application: PCA is relevant for retain financial information about loan debtors (such as loan history, income debt levels, credit scores, etc. This effectively help credit institutions measure the credit worthiness of their customers and the inherent credit risk.

4. **Fraud and Regime Change Detection:**
   Key Advantages: One essential feature of PCA is its sensitivity to outliers. Outliers can be detected in a high-dimensional dataset (e.g. list of financial transactions) using this methodology. Real-world applications: Outliers such as abnormal spending behaviour and suspicious transaction comments of borrowers could be detected and explained in the form of an explained variance ratio by the PCA methodology. Additionallly, macro-economic and/or sector-specific changes could be detected by as outlier on the interest yield curve by PCA.

Step 5: Learn More

- Chan-Lau, Jorge A. "Lasso Regressions and Forecasting Models in Applied Stress Testing." *IMF*, 5 May 2017, www.imf.org/en/Publications/WP/Issues/2017/05/05/Lasso-Regressions-and-Forecasting-Models-in-Applied-Stress-Testing-44887.
  - **Description:** The study elaborate on the use of LASSO regression in stress testing and financial forecasting which adds value to the existing knowledge regarding its adeptness to tackle high-dimensional datasets with higher predictive power corrections. In this paper, LASSO's regularization techniques are highlighted as the means by which financial institutions can reduce risk by way of selecting the most appropriate economic indicators for modeling.

- Feng, Guanhao, et al. *Annual Meetings of the American Finance Association, the 2016 Financial Engineering and Risk Management Symposium in Guangzhou*. NBER, 2018, www.nber.org/system/files/working_papers/w25481/w25481.pdf.
  - **Description:** Analysis of the fundamental components and processes of risk management and asset valuation shows that present-day machine learning science and technology and its quantitative methods enhance classic financial models. The study presents concrete data on the advantages of implementing ML techniques in the financial industry to make more accurate predictions and proper judgment.

- Guam, He-Shan, and Qing-Shan Jiang. 'Cluster Financial Time Series for Portfolio'. *2007 International Conference on Wavelet Analysis and Pattern Recognition*, vol. 2, 2007, pp. 851–56. *IEEE Xplore*, https://doi.org/10.1109/ICWAPR.2007.4420788.
  - **Description:** This paper introduces hierarchical clustering as a method for grouping financial assets for maximum portfolio diversification. It features a new method of computing similarity for the clustering of stock return sequences, which aids investors in selecting portfolios with the least possible risk and maximum diversification.

- Hastie, Trevor, et al. 'The Elements of Statistical Learning: Data Mining, Inference, and Prediction'. *Math. Intell.*, vol. 27, Nov. 2004, pp. 83–85. *ResearchGate*, https://doi.org/10.1007/BF02985802.
    - **Description:** This book is based on key machine-learning techniques such as regression, classification, and clustering. Its great theoretical structure and practical applications make it indispensable for financial analysts and data scientists.

- Jain, A. K., et al. 'Data Clustering: A Review'. *ACM Computing Surveys*, vol. 31, no. 3, Sept. 1999, pp. 264–323. *Semantic Scholar*, https://doi.org/10.1145/331499.331504.
    - **Description:** This paper applies a comprehensive assessment of the clustering method, including the hierarchical clustering method, k-means, and density-based approaches. This evaluation highlights the advantages and disadvantages of several clustering algorithms and offers a detailed explanation of their applicability in various fields, such as finance and economics.

- Mishra, S., Sarkar, U., Taraphder, S., Datta, S., Swain, D., & Saikhom, R. et al. (2017). Multivariate Statistical Data Analysis- Principal Component Analysis (PCA). International Journal of Livestock Research, 7(5), 60-78. http://dx.doi.org/10.5455/ijlr.20170415115235

    Description: This article gives a comprehensive report on principal component analysis and its relevance as a statistical technique for multivariate data analysis. It highlights the definition and mathematical formulas for various terms in the model computation. It also implement the model on made-up dataset to further explain the concept of PCA.

- Nuwan Weeraratne, Lyn Hunt, Jason Kurz et al. *"Challenges of Principal Component Analysis in High-Dimensional Settings when n<p"*, 11 March 2024, PREPRINT (Version 1) available at Research Square https://doi.org/10.21203/rs.3.rs-4033858/v1

    Description: This report highlights some of the challenges of using principal component analysis on finite data samples (where the number variable *n* is is lower than the number of *p* principal components). For such cases, the article proposes number of advanced methods to improve the accuracy of the PCA algorithm by using sparse and well-conditioned covariance estimation.

Conclusion

By applying LASSO Regression, Hierarchical Clustering, and PCA to financial modeling, a strong model for analysis and decision-making is achieved. The factor selection is accurate using LASSO, and the chance of overfitting diminishes as the model becomes easier to interpret. With the help of Hierarchical Clustering, the hidden structures characteristic of financial assets are discovered, leading to the possibility of risk diversification and, hence, the optimization of the portfolio. PCA is about manipulating the data to represent the dataset and show the relation between data points and their usability. These techniques provide firms with comprehensive market intelligence.