DIAML: Assignment 7

by Anthony Rizkallah

Table of contents

Python Packages

Pandas

BeautifulSoup

Requests

Yfinance

Seaborn

Matplotlib

Sklearn

Numpy

Scipy

1 PCA

1.1 Description, Applications, and Usefulness

1.1.1 Description

Principal Component Analysis (PCA) is a dimensionality reduction method that is used to transform a set of possibly correlated variables into smaller set of uncorrelated variables called principal components; the first principal component captures the largest possible variance in the data, and the second PC captures the largest variance orthogonal to the first PC. Linear-algebraically, principal components are orthogonal to each other, meaning they are perpendicular and have a dot product of zero.

1.1.2 Applications

- **Dimensionality Reduction**: Reduceing the number of features while preserving most of the variability in the data.
- **Noise Reduction**: Focus on components that explain most variance and filter out noise through PCA (low-variance components)
- **Multicollinearity Mitigation**[1]: By transforming the original correlated variables into a new set of uncorrelated variables, using PCs as explanatory variables

1.1.3 Usefulness

PCA's primary goal is to reduce the number of variables in a dataset by selecting principal components that capture the most significant variance. On one hand, this method reduces the computational power and time needed when evaluating high-dimensional data. On another, it identifies new, uncorrelated variables that are linear combinations of the original variables, which captures the underlying structure of the data. This is useful when aiming to simplify datasets, enhance interpretability, and improve efficiency and stability of the model.

1.2 Transforming X into new variables

1.2.1 Standardizing Data

The first step in transforming the variables X is to ensure all values are on the same scale. Each variable needs to be calculated in the matrix according to the following formula.

$$X_{\text{standardized}} = \frac{X - \mu}{\sigma}$$

[1] https://medium.com/aimonks/principal-component-analysis-pca-in-machine-learning-407224cb4527

where μ is the mean of each variable and σ is the standard deviation of each variable.

1.2.2 Computing Covariance Matrix

The covariance matrix includes the variance between one set of variables and the variance between the different sets of variables. A dataset with 3 variables $(x, y,$ and $z)$ has a C matrix of the form

$$C = \begin{pmatrix} \mathrm{var}\,(x) & \mathrm{cov}\,(x, y) & \mathrm{cov}(x, z) \\ \mathrm{cov}(x, y) & \mathrm{var}(y) & \mathrm{cov}(y, z) \\ \mathrm{var}(x, z) & \mathrm{cov}(y, z) & \mathrm{var}(z) \end{pmatrix}$$

Covariance is trivially the variance between two distinct variables

$$C = \frac{1}{n-1} X_{\text{standardized}}^T X_{\text{standardized}}$$

where $X = \begin{pmatrix} x_1 & y_1 & z_1 \\ x_2 & y_2 & z_2 \\ x_3 & y_3 & z_3 \end{pmatrix}$, $X^T = \begin{pmatrix} x_1 & x_2 & x_3 \\ y_1 & y_2 & y_3 \\ z_1 & z_2 & z_3 \end{pmatrix}$, n is the number of variables in X

1.2.3 Eigen Decomposition

$$\Sigma = \mathrm{VLV}^T$$

where $(\boldsymbol{L})$ is the diagonal matrix of eigenvalues and $(\boldsymbol{V})$ is the matrix of eigenvectors. The eigenvectors represent the directions of maximum variance, and eigenvalues represent the magnitude of this variance.

1.2.4 Projection

$$Y = X_{\text{std}} V$$

Projecting the original (and scaled) data onto the new axis.

1.3 DOW Constituents

MMM, AXP, AMGN, AMZN, AAPL, BA, CAT, CVX, CSCO, KO, DIS, GS, HD, HON, IBM, JNJ, JPM, MCD, MRK, MSFT, NKE, NVDA, PG, CRM, SHW, TRV, UNH, VZ, V, VMT

1.3.1 Steps

The DOW constituents stated previously are stocks from various sectors: Conglomerates (multiple sectors), Financial Services, Biopharmaceutical, Retailing, Information Technology, Aerospace and defense, Construction and Mining, Petroleum, Drink, Broadcasting and Entertainment, Home Improvement, Pharmaceutical, Food, Clothing, Fast-moving Consumer Goods, Specialty Chemicals, Insurance, Managed Health Care, and Telecommunications. Those are the classifications that the firms have given themselves[2].

After defining each stock, the daily adjusted close prices of each were imported from December 4 2023 to December 4 2024. Then, the daily returns of each stock were calculated using

$$r(t) = \frac{p(t)}{p(t-1)} - 1$$

where $p(t)$ is the price at time t and $p(t-1)$ is the price at the previous time.

Afterwards, the correlation matrix between the variables was calculated using

$$\rho_{xy} = \frac{\sum (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum (x_i - \bar{x})^2 \sum (y_i - \bar{y})^2}}$$

[2] https://en.wikipedia.org/wiki/Dow_Jones_Industrial_Average

where x_i is the value of each of the first stock return, $\bar{x}$ is the mean value of that stock return, and y_i and $\bar{y}$ is for the other stock's price and mean value.

Then, using the steps highlighted in Section (1.2), the first and second principal components were calculated, and the magnitudes were extracted using the vector magnitude formula.

Then, a comparison between the actual weight of each constituent in the DOW Index and the magnutides of each in the PCA was computed.

1.3.2 Results

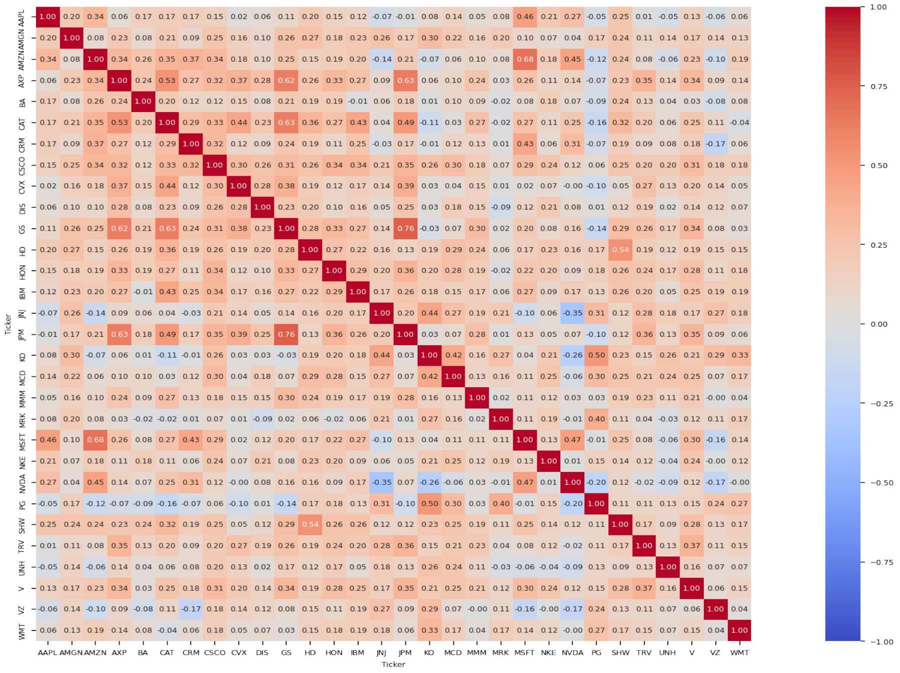

Figure 1. Explanatory Variables correlations heatmap rounded to the nearest hundredth

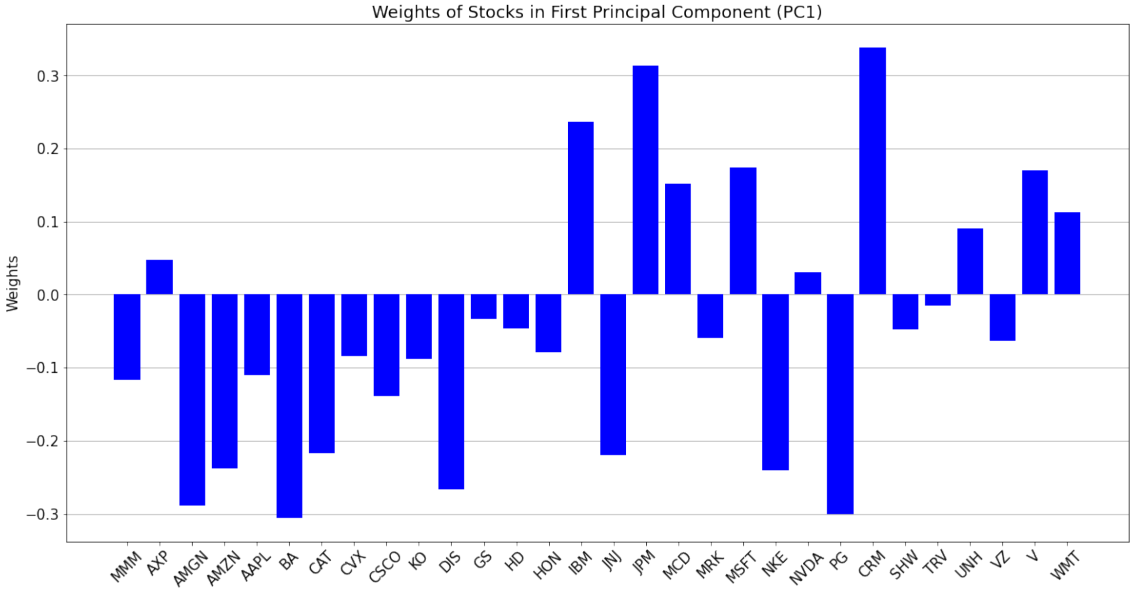

Figure 2. Weight of Stocks in First Principal Component

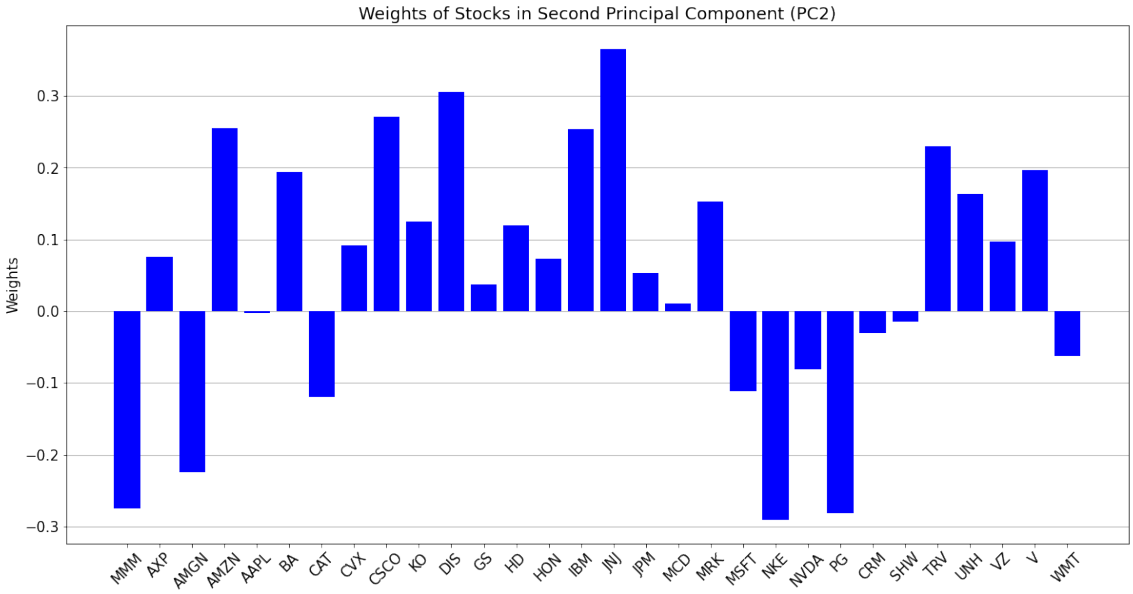

Figure 3. Weight of Stocks in Second Principal Component

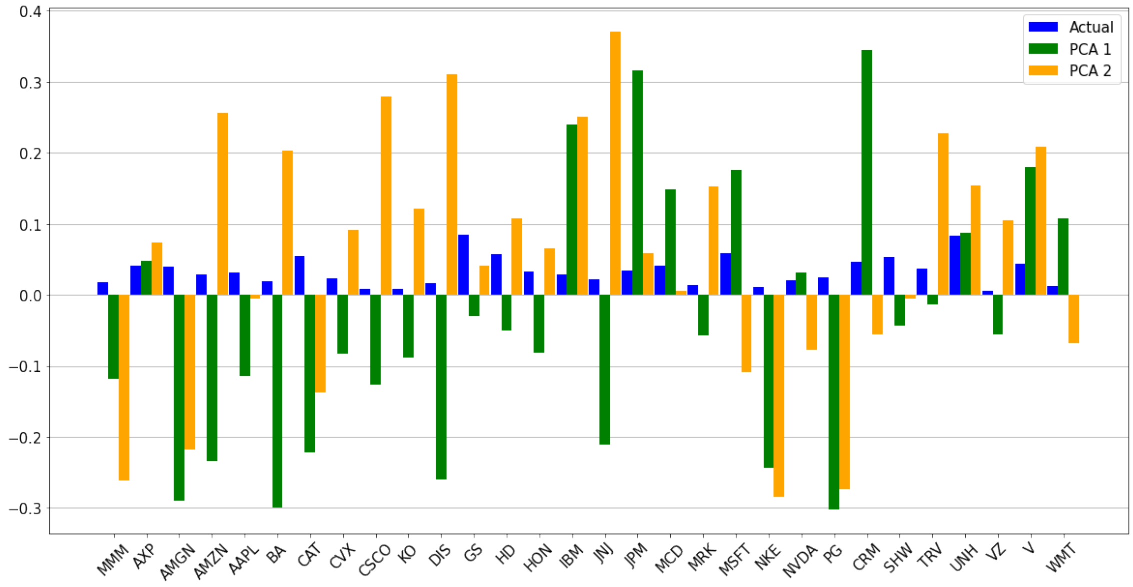

Figure 4. Actual constituent weight compared with magnitudes from PCA 1 and PCA 2

1.3.3 Inference

Figures 2, 3, and 4 represent the weights (or magnitudes) from PCA 1, PCA 2, and the actual market. In general, neither the first nor the second represent the market weights accurately, but interestingly, when they do match, PCA 1 is closer. This can be seen in AXP, NVDA, and UNH, where PCA 1 and PCA 2 were close to the actual, but PCA 1 had a slight advantage over the second principal component.

1.4 Principal Component Analysis

1.4.1 Steps

To calculate how many PCs are needed to explain 95% of the data, an analysis of variance vs number of PCs is a solution. At each iteration, calculate the variance var_n, where n is the PC number (e.g. 1, 2, 3, ...) and then sum the variance up to that point (cumulative variance).

$$\mathrm{CV} = \sum_{i=1}^{N} \mathrm{var}_n$$

1.4.2 Results

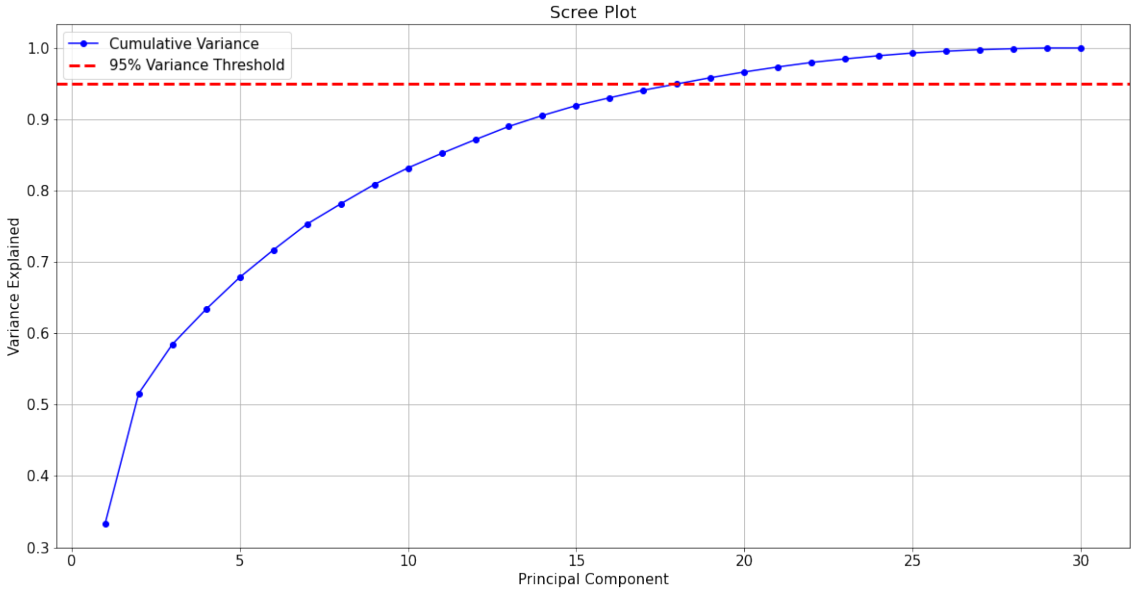

Figure 5. Scree plot of Explained Variance vs Principal Component

Number of components required to explain 95% of variance: 19

1.4.3 Inference

Again, referring back to the main purpose of using Principal Component Analysis, (dimensional reduction and increased computational efficiency) using 19 components instead of 30 to explain 95% of the DOW behavior is much more efficient than using all 30. Moreover, 95% is a significantly high explanatory variance with an 11 component reduction.

Imperically (and assumingl linearity), $\frac{0.95}{19} = 0.05$ and $\frac{0.05}{11} = 0.005$, resulting in a 10x computational efficiency is a significant dimensional and computational reduction.

1.5 Investigating PCA

1.5.1 Steps

To create the scatter plot between the first and second principal component, use the magnitudes from PCA 1 (x-axis) and PCA 2 (y-axis) to model the relationship between both. Then, find the average value (contributing to a single point on the plot), and find the Euclidean distance between the mean and each stock component using

$$d = \sqrt{(x_n - \bar{x})^2 + (y_n - \bar{y})^2}$$

where n is the distinct stock, $\bar{x}$ is the average PCA 1 magnitude, and $\bar{y}$ is the average PCA 2 magnitude.

Finally, the three stocks with the largest d are the farthest away.

1.5.2 Results

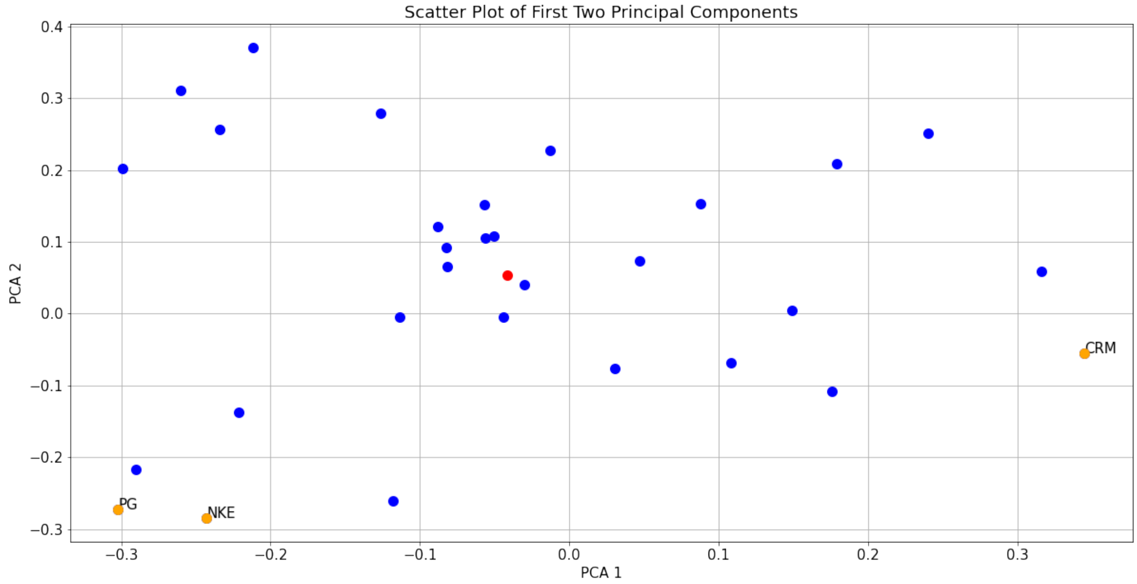

Figure 6. Scatter plot of First Two Principal Components showing DOW constituents, and particularly the mean value and the three most distant stocks

The three most distant stocks are PG, NKE, and CRM.

1.5.3 Inference

Since PCA captures the most variance in the dataset, stocks far from the average are outliers in this reduced feature space, which means that they behave differently compared to others. Moreover, these stocks can have unique patterns in their returns, volatility, or other characteristics that set them apart from the majority of the indx. Finally, stocks that are far away can be be influenced by sector-specific trends, unusual market events, or company-specific factors like earnings surprises or regulatory changes.

2 Dendogram

A dendrogram is a hierarchical diagram used to cluster. It visually represents the process of combining (or splitting) clusters step-by-step, starting with individual elements and merging them until all items form a single cluster.

2.1 Components, Construction, and Interpretation

2.1.1 Components

1. **Leaves (Nodes)**: Represent individual items or data points. In the context of stocks, each leaf would represent a specific stock.

2. **Branches**: Represent clusters. Each merge of branches signifies a point where sub-groups are combined into a larger cluster.

3. **Height (Y-Axis)**: Indicates the distance or dissimilarity between clusters. The greater the height, the more dissimilar the clusters.

4. **Clusters**: Groups of data points connected by branches. The level at which you cut the dendrogram determines the number of clusters.

2.1.2 Construction

1. **Calculate Distance Matrix**: Compute the pairwise distances or dissimilarities between data points using a suitable distance metric (e.g., Euclidean, Manhattan, or Cosine).

2. **Choose a Linkage Method**:
   - **Single Linkage**: Minimum distance between points in two clusters.
   - **Complete Linkage**: Maximum distance between points in two clusters.
   - **Average Linkage**: Average distance between points in two clusters.
   - **Ward's Method**: Minimizes the variance within clusters.
3. **Build the Dendrogram**:
   - Start with each data point as its own cluster.
   - Iteratively merge the two closest clusters based on the chosen linkage method.
   - Record the distance at which clusters are merged.
4. **Visualize the Dendrogram**: Plot the hierarchical structure using a dendrogram visualization library.

2.1.3 Interpretation

1. **Cluster Formation**:
   - Short distances indicate similar clusters.
   - Long distances indicate dissimilar clusters.
2. **Determine Number of Clusters**:
   - Decide the number of clusters by cutting the dendrogram at a specific height.
   - The horizontal line at the cutting height determines the clusters.
3. **Outliers**: Outliers appear as individual branches or clusters that merge at very high distances.

2.2 Constructing a Dendrogram from Pairwise Dissimilarity Values

1. **Compute the Pairwise Dissimilarity Matrix**
   - If not provided, calculate the pairwise dissimilarities (or distances) between all data points using a suitable distance metric (e.g., Euclidean, Manhattan, or Cosine distance).
   - The result is a symmetric matrix where the (i, j)(i,j)-th entry represents the dissimilarity between data points ii and jj.
   - **Example**: For three data points A, B, CA,B,C, the pairwise dissimilarity matrix might look like:

     $$\begin{bmatrix} 0 & d(A,B) & d(A,C) \\ d(B,A) & 0 & d(B,C) \\ d(C,A) & d(C,B) & 0 \end{bmatrix}$$

   - Here, $d(A, B)$d(A,B) is the dissimilarity between AA and BB, and diagonal entries are 00 (distance from a point to itself).
2. **Choose a Linkage Method** (Steps from 2.1.2, part 2)

3. **Initialize Each Data Point as a Cluster**

    - Start with each data point as its own individual cluster.

4. **Iteratively Merge Clusters**

    At each step:

    - Identify the two closest clusters based on the linkage method and pairwise dissimilarity values.

    - Merge these two clusters into a new cluster.

    - Update the dissimilarity matrix:

        - Recalculate the distances between the new cluster and all other clusters using the linkage method.

    - **Example (Single Linkage)**:

        - If AA and BB are merged, calculate the distance between the new cluster ABAB and another cluster CC as:

$$d(\mathrm{AB}, C) = \min(d(A, C), d(B, C))$$

5. Record the Merging Step

    - Record the distance (height) at which the clusters are merged.

    - This distance will later determine the y-axis value in the dendrogram.

6. **Repeat Until All Clusters are Merged**

    - Continue merging clusters until all data points form a single cluster.

7. **Visualize the Dendrogram**

    Plot the hierarchical clustering process as a tree diagram:

    - The **leaves** (bottom nodes) represent individual data points.

    - The **branches** connect clusters, and their height represents the dissimilarity at which the clusters were merged.

2.3 DOW Correlation Pairwise Distances

Using the correlation matrix from question (1.3) above, the pairwise distances between the 30 stocks will have a rescaled distance using the following formula

$$d(i, j) = \sqrt{2 \times (1 - \mathrm{corr}(i, j))}$$

given the correlation coefficient between two stocks ii and jj, denoted as $\mathrm{corr}(i, j)$.

2.4 DOW Dendogram

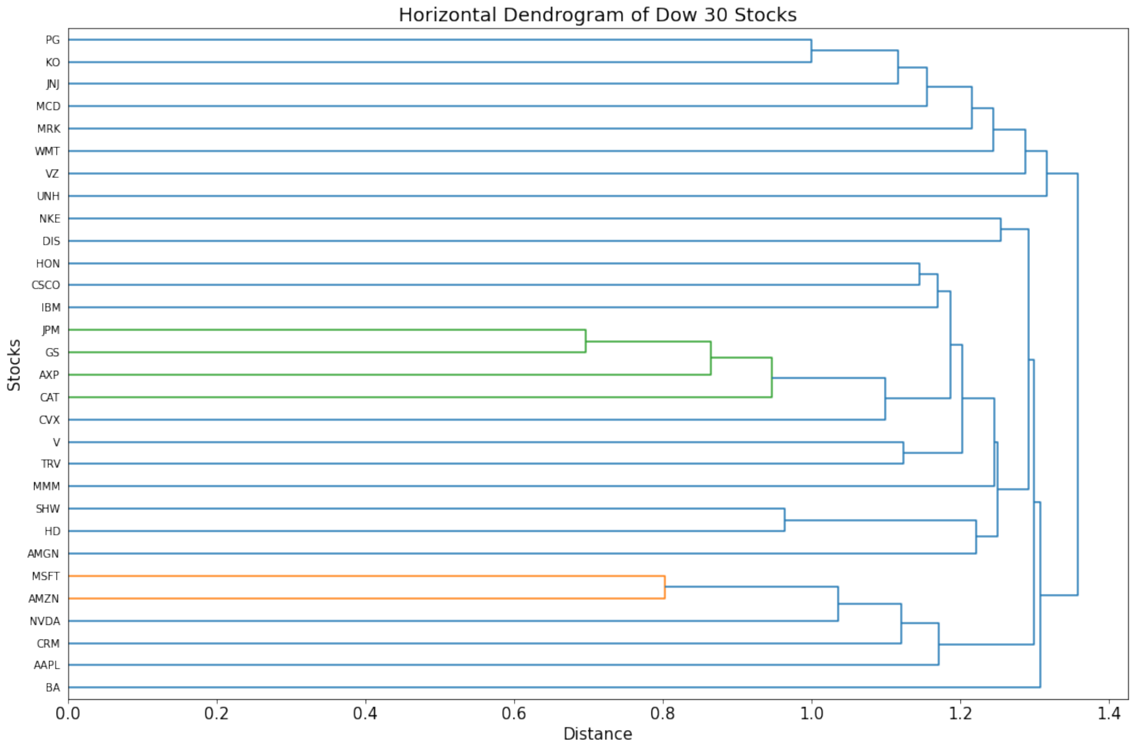

Figure 7. Dendogram of DOW stocks

2.5 Constituent Clusters

2.5.1 Cluster 1

PG, **KO**, **JNJ**, **MRK**, **WMT**

These stocks form a cohesive cluster at a relatively low distance, indicating strong correlation in their returns. They likely represent **defensive** sectors, as they include c**onsumer staples and healthcare stocks**.

2.5.2 Cluster 2

JPM, **GS**, **AXP**, **V**, **TRV**

This cluster groups **financial** stocks, which is expected due to their shared dependence on market trends, interest rates, and economic cycles.

2.5.3 Cluster 3

MSFT, **AMZN**, **AAPL**, **CRM**, **NVDA**

This cluster contains **technology-heavy** stocks, suggesting a strong correlation in their market performance, driven by innovation and shared growth-oriented characteristics.

2.5.4 Cluster 4

CVX, **XOM**

These two **energy** stocks form a distinct cluster, which is consistent with their shared exposure to oil prices and global energy demand.

2.5.5 Cluster 5

CAT, **HON**, **BA**, **MMM**, **SHW**

This cluster likely represents **industrials and materials**, with companies tied to **manufacturing, infrastructure, and raw materials**.

3 Ensembles for classification

3.1 Uncertainty

3.1.1 Observational uncertainty

Observational uncertainty arises from variability or noise in the observed data. This could be due to errors or randomness in the data collection process, measurement devices, or incomplete observations.

Impact on the Modeling Process:

- Reduces the quality of the training dataset, leading to less accurate predictions.
- Causes models to overfit to noisy patterns if not handled appropriately.
- Makes it harder to distinguish between meaningful signals and random variations in the data.

3.1.2 Parametrical uncertainty

Parametric uncertainty refers to uncertainty about the values of the parameters within a given model. It occurs when the parameter estimates derived during training are not perfectly accurate or when the model's assumptions about these parameters are flawed.

Impact on the Modeling Process:

- Leads to variability in model predictions, especially when the training data is insufficient or unrepresentative.
- Reduces model confidence and generalization to unseen data.
- Creates challenges in selecting optimal model configurations.

3.1.3 Structural uncertainty

Structural uncertainty arises from the choice of the model itself or the assumptions underlying its structure. It reflects the possibility that the model is misspecified or lacks the capacity to capture the true underlying relationships in the data.

Impact on the Modeling Process:

- Leads to systematic bias in predictions due to model misspecification.
- Results in poor performance on complex datasets when using overly simplistic models.
- Reduces interpretability and trustworthiness if the model does not align with domain knowledge.

3.2 Model Averaging

3.2.1 Concept

Technique in machine learning and statistics where multiple models are combined to generate predictions by averaging their outputs. The idea is that different models may capture different aspects of the data, and averaging their predictions can reduce variance, mitigate overfitting, and improve overall performance.

3.2.2 Benefits

Error Reduction:

- A single model is prone to errors due to noise, bias, or variance. Combining multiple models reduces the overall error by diversifying these effects.

Robust Predictions:

- By averaging predictions from multiple models, the approach becomes less sensitive to outliers or overfitting in any individual model.

Better Generalization:

- Aggregating models enhances performance on unseen data, as individual model weaknesses are compensated by others

3.2.3 Applications

- **Simple Averaging:** Take the mean prediction from multiple models.
- **Weighted Averaging:** Assign weights to different models based on their performance and average the predictions accordingly.
- **Bayesian Model Averaging (BMA):** Involves averaging over models probabilistically, considering the posterior probabilities of the models.

3.3 Reducing Uncertainty

- **Bagging:** Reduces variance and overfitting by averaging predictions from multiple bootstrap samples.

- **Boosting:** Reduces bias by sequentially fitting models to the residuals of the previous models and combining them for the final prediction.

- **Stacking:** Combines predictions from multiple models using a meta-learner, aiming to learn the best way to combine the individual models.

3.4 Random Forest - Titanic

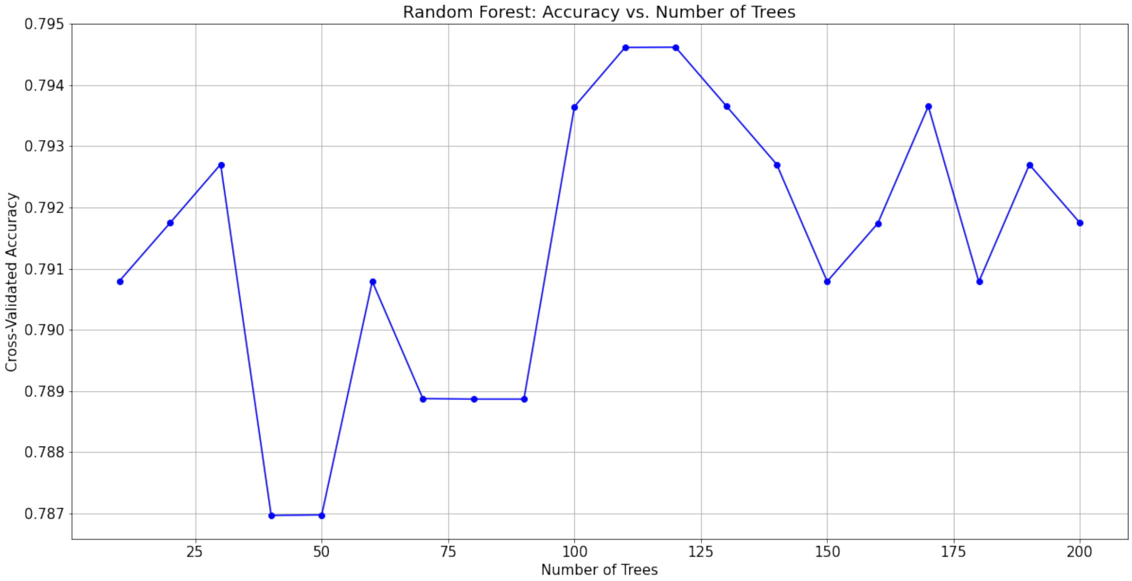

Figure 8. Random Forest: Accuracy vs Number of Trees

Optimal number of trees: 120

3.5 Model Comparisons

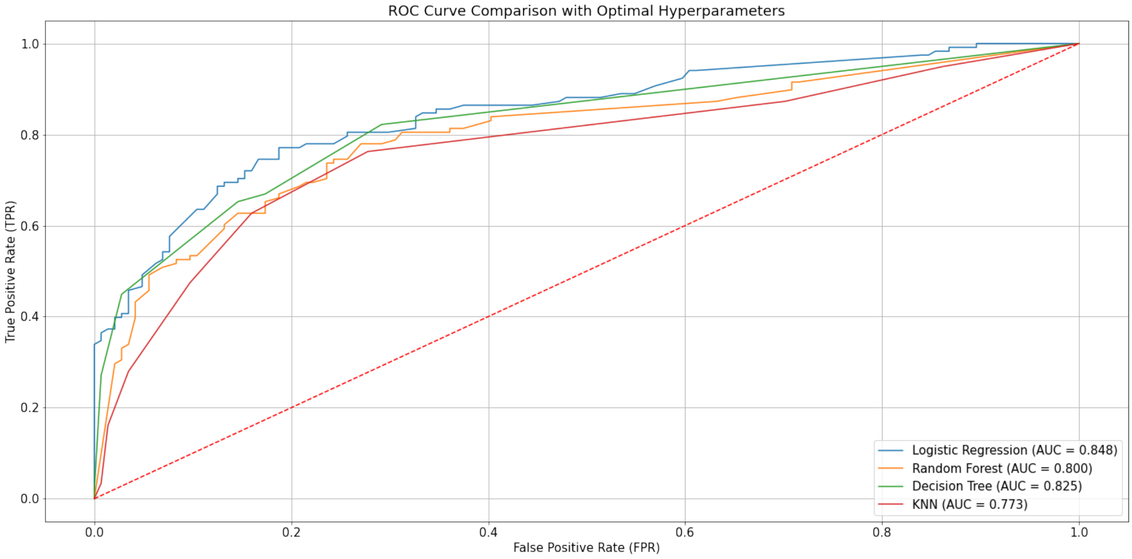

Figure 9. ROC Curve Comparison with Optimal Parameters

The best model based on AUC to use in the Kaggle competition is logistic regression.

4 Ensembles for Regression

A Random Forest is an ensemble learning method that operates by constructing a multitude of decision trees at training time and outputting the mean prediction of the individual trees for regression tasks. The "forest" it builds is an ensemble of Decision Trees, usually trained with the "bagging" method. The general idea of the bagging method is that a combination of learning models increases the overall result. (*What Is Random Forest? | IBM*, n.d.)

4.1 Random Forest Concepts

4.2 Random Forest Model - Wine

4.2.1 Estimating number of leafs

4.2.2 Steps

By running a simulation that iterates over an N number of leafs and calculating the accuracy at each leaf, then, the most optimal leaf number will have the highest in-sample classification accuracy after being cross validated.

4.2.3 Results

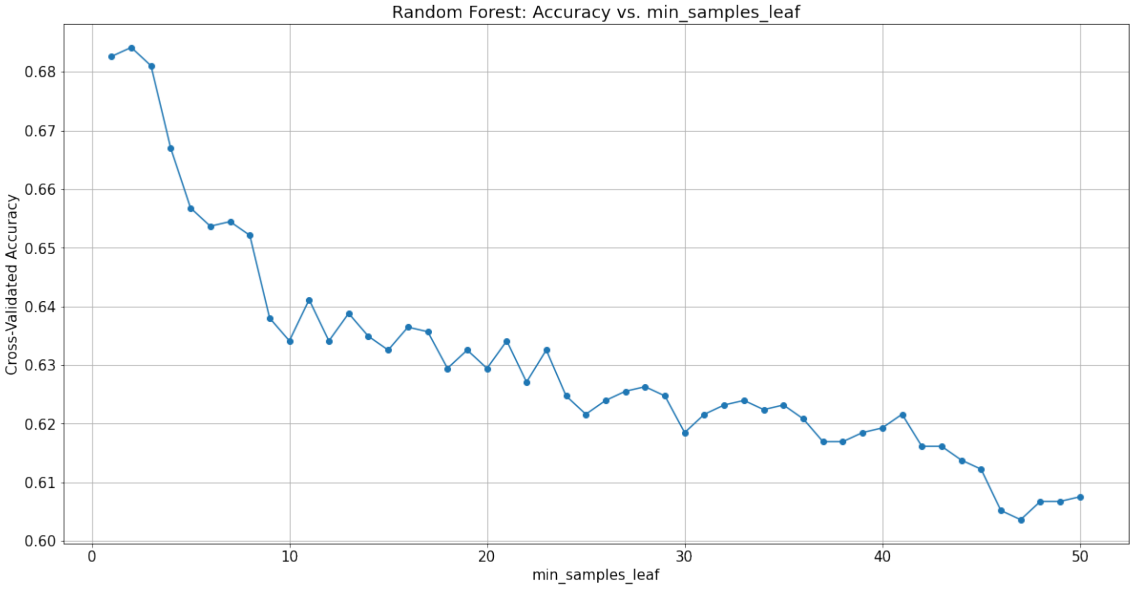

Figure 10. Random Forest: Accuracy vs Minimum Samples Leaf

Optimal number of leafs: 2

4.3 Estimating number of trees

4.3.1 Steps

After finding the optimal number of leafs, the number of leafs can be held at constant (2 in this case), and then the number of trees can be iterated over N number of trees to find the most optimal number of trees.

4.3.2 Results

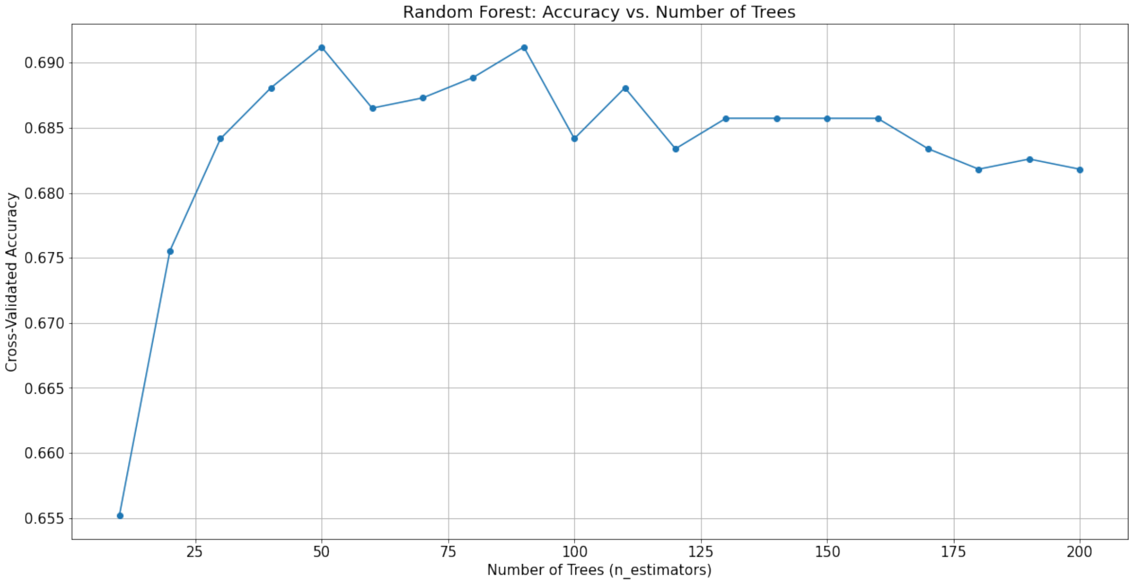

Figure 11. Random Forest: Accuracy vs Number of Trees

4.4 Feature importance comparison

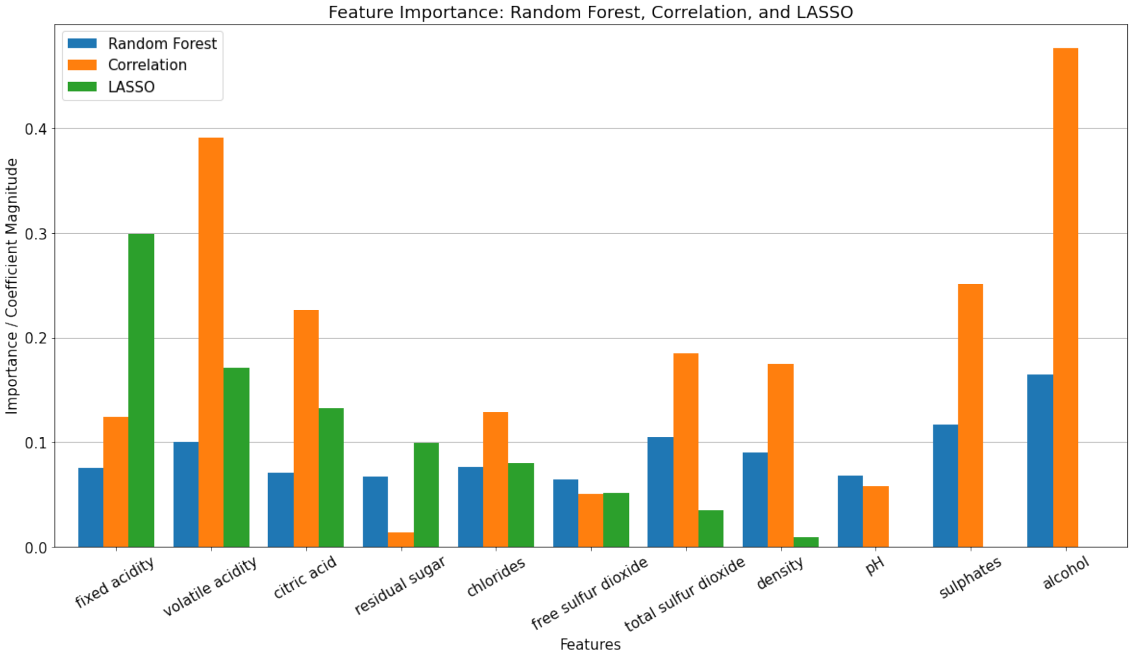

Figure 12.

4.5 Random Forest and Other Models Performances

Model	**MSE**	$\boldsymbol{R^2}$
Random Forest	0.3906	0.4023
Linear Regression (LASSO)	0.3917	0.4006
KNN	0.5531	0.1536

Table 1. Model Comparsion

Hence, Random Forest is the best performing model to be used to predict the quality of red wine, having the least MSE and R^2.