

REPORT OF ASSIGNMENT 7

Carnegie Mellon University Africa

Submitted By: Peace Ekundayo Bakare

Course: Data, Inference, and Applied Machine Learning

There are 4 Questions that are documented in total. Each Question is started on a new page. Please, scroll down even when you see a blank page.

Libraries Used:

1. pandas as peacepd
2. seaborn as peacesns
3. numpy as peacenp
4. matplotlib.pyplot as peaceplt
5. requests
6. yfinance as yfin
7. PCA from sklearn.decomposition
8. fcluster from scipy.cluster.hierarchy
9. scipy.cluster.hierarchy as peacesch
10. tabulate from tabulate
11. BeautifulSoup from bs4
12. pdist, squareform from scipy.spatial.distance
13. fcluster from scipy.cluster.hierarchy
14. ListedColormap from matplotlib.colors
15. train_test_split from sklearn.model_selection
16. RandomForestClassifier from sklearn.ensemble
17. roc_curve, RocCurveDisplay, roc_auc_score from sklearn.metrics
18. LogisticRegression, LinearRegression from sklearn.linear_model
19. DecisionTreeClassifier from sklearn.tree
20. KNeighborsClassifier, KNeighborsRegressor from sklearn.neighbors

There are 4 Questions that are documented in total. Each Question is started on a new page. Please, scroll down even when you see a blank page.

QUESTION 1 – PCA

1.1 Qualitative Description of Principal Component Analysis (PCA)

Principal Component Analysis (PCA) is an unsupervised machine learning technique particularly used for dimensionality reduction. It performs an **orthogonal transformation** to convert a set of possibly correlated variables into linearly uncorrelated variables called **principal components (PCs)**. The principal components are ordered by the amount of variance they explain, with the first component capturing the peak variance in the data.

PCA is particularly useful when dealing with high-dimensional data because it identifies patterns by focusing on the directions (or axes) of maximum variance.

In Mathematics, PCA is based on the eigen-decomposition of positive semi-definite matrices and the singular value decomposition (SVD) of rectangular matrices.[1]

The Goals of PCA

- (i) To extract the most relevant information from data tables.
- (ii) To reduce the size of data sets by keeping only this relevant information.
- (iii) To simplify the data set's description.
- (iv) To analyze the structure of the observations and the variables.

Applications in Machine Learning:

1. **Dimensionality Reduction:** PCA reduces the number of features in a dataset while retaining the most important information. This simplifies models, speeds up computations, and helps mitigate overfitting in machine learning.
2. **Noise Reduction:** PCA can filter out noise by discarding components that capture low variance (often considered noise). This is useful in signal processing, such as separating signals in applications like fetal ECG extraction.
3. **Provides an Eigenvalue Spectrum:** PCA produces an **eigenvalue spectrum**, where each eigenvalue represents the variance explained by the corresponding principal component. This spectrum can help identify the number of significant components to retain. It is particularly useful for determining the intrinsic dimensionality of the data or for visualizing the distribution of variance across components.

Why PCA Might Be Useful for Transforming Explanatory Variables:

There are 4 Questions that are documented in total. Each Question is started on a new page. Please, scroll down even when you see a blank page.

- **Dealing with Multicollinearity:** PCA transforms correlated variables into uncorrelated components, which is beneficial for machine learning models that assume independence between features (e.g., linear regression).
- **Improved Interpretability and Efficiency:** By reducing dimensionality, PCA allows easier visualization and processing of high-dimensional data while preserving the essential structure and variance of the data.
- **Captures Key Variance:** It ensures the primary patterns in the data are retained in fewer variables, which can improve model performance and reduce overfitting.

1.2 Mathematical Equations for PCA

Mathematical equations of PCA explaining how they transform the raw input data matrix X into a new set of variables.

1. To Construct the Data Matrix

X is the raw data matrix with N samples and M variables ($X \in \mathbb{R}^{N \times M}$).

2. To Standardize the Data

PCA standardizes data to prevent features that have larger scales from dominating.

$$Z = \frac{X - \mu}{\sigma}$$

where μ and σ are the means and standard deviations of the variables respectively

3. To Compute the Covariance Matrix

$$C = \frac{1}{N} X^T X$$

where C is the $M \times M$ covariance matrix capturing the relationships between variables

4. Eigenvalue Decomposition

The covariance matrix C is decomposed as:

$$C = V \Sigma^2 V^T$$

where V ($M \times M$) is the orthogonal matrix of eigenvectors ($V^T V = I$)

Σ^2 ($M \times M$) is the diagonal matrix of eigenvalues ($\sigma_1^2, \sigma_2^2, \dots, \sigma_M^2$), which represent the variance explained by each principal component.

5. Project Data into the Principal Component Space

$$Y = X V$$

There are 4 Questions that are documented in total. Each Question is started on a new page. Please, scroll down even when you see a blank page.

Where Y ($N \times M$) is a transformed data matrix with rows representing projections of X onto the major components. Each column in Y represents a major component, capturing the largest variance along its direction.

Interpretation of the Matrices

X : Original data matrix with raw input values.

C : The covariance matrix summarizes the pairwise connections (variance and covariance) between characteristics.

V : Eigenvector matrix with each column representing the direction of maximum variance (principal components).

The diagonal eigenvalue matrix (Σ^2) represents the variance explained by each primary component (also known as the eigenvalue spectrum).

Y : Transformed data represented as primary components, sorted by their contribution to the variance in X .


1.3 Correlation Matrix for PCA and Bar Graphs for First and Second Principal Components

I used some libraries and defined the Wikipedia URL for the webpage where the table is found. Part of the URL defined is the exact section of the page “Components”. The URL used is: https://en.wikipedia.org/wiki/Dow_Jones_Industrial_Average#Components

I have answered this question in steps.

Step 1: Web scraping and download of 1 year of daily adjusted close prices

In this step, I extracted ticker symbols for Dow Jones Industrial Average stocks obtained from the URL shared above using *BeautifulSoup*. I retrieved just 1 year (2019-01-01 to 2020-01-01) of daily adjusted close “Adj Close” prices for the tickers from Yahoo Finance using *yfinance*.

 Dow Jones Industrial Average Ticker Symbols:
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, WMT,

There are 4 Questions that are documented in total. Each Question is started on a new page. Please, scroll down even when you see a blank page.

Step 2: Calculation of Daily Returns

To calculate the daily percentage returns for each stock, I applied the `pct_change()` function.

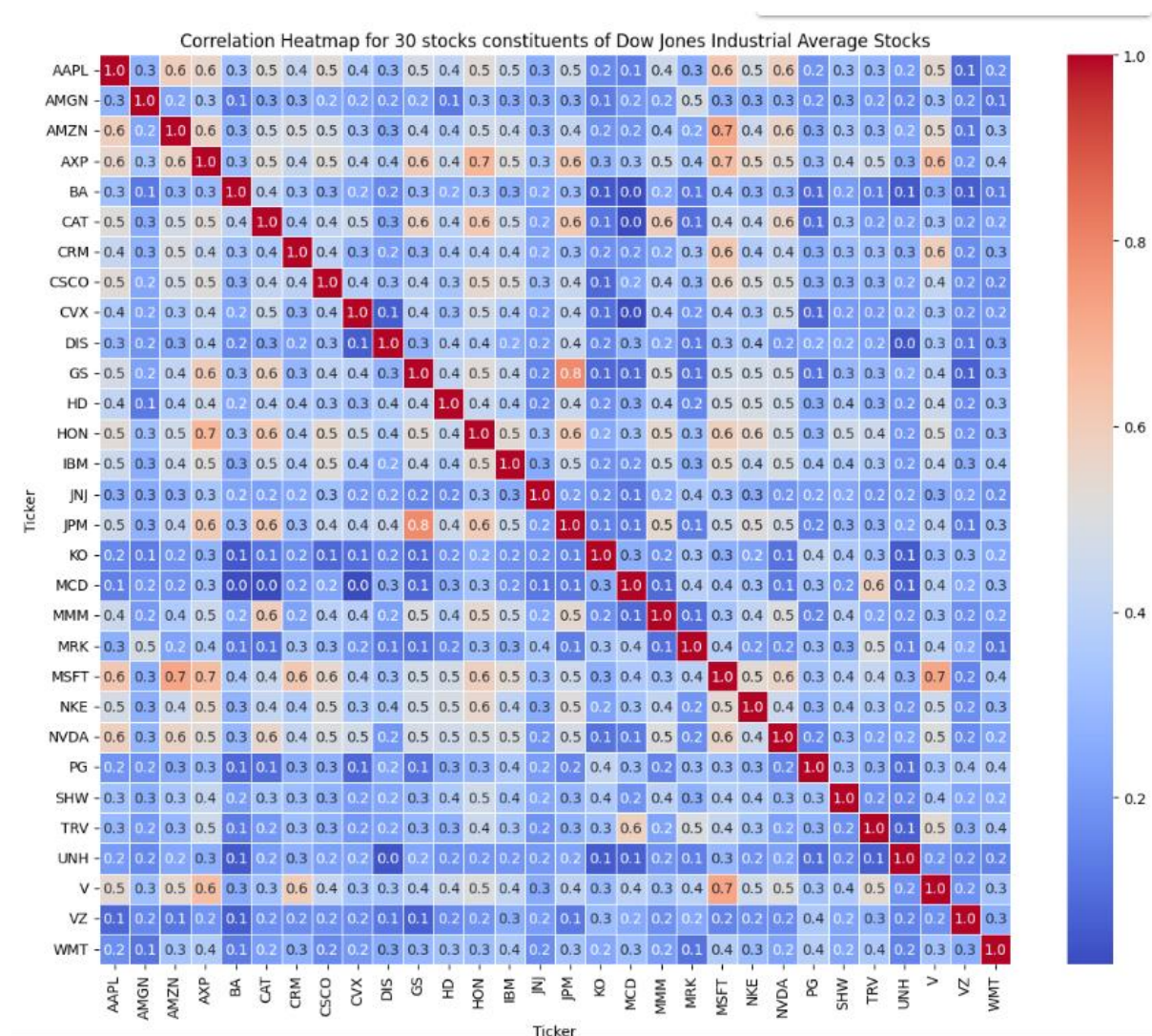
$$Return = \frac{P_t - P_{t-1}}{P_{t-1}}$$

This analyzes the variance and correlation of returns, not the prices. The return of normalized stock data makes them comparable across different stocks.

Step 3: Computation of the Correlation Matrix

I calculated the correlation matrix to assess the relationship between the daily returns of the stocks. Correlation matrix helps to understand how strong or weak the relationship between two stocks. Strong correlation (+1) suggests that the stocks moved in similar directions. This can be captured by PCA as principal components.

I have constructed an heatmap for the correlation matrix.



There are 4 Questions that are documented in total. Each Question is started on a new page. Please, scroll down even when you see a blank page.

Step 4: Performing Principal Component Analysis (PCA)

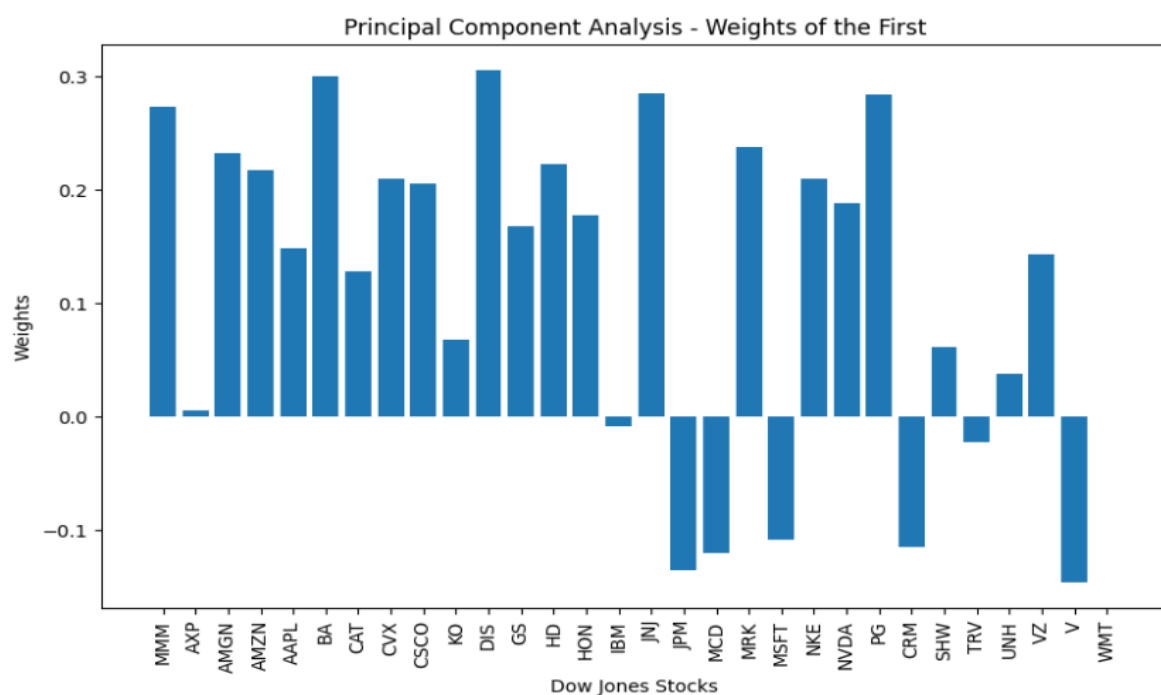
I applied PCA to the correlation matrix to extract the principal components. The *pca.fit()* method calculates the eigenvectors (components) and eigenvalues (the variance explained by each component). Calculation of the PCA helps to identify patterns in stock behaviors and this can be utilized for dimensionality reduction or clustering.

Step 5: First and Second Bar Graphs for Stock Weights

From the components array, I extracted the weights for the first and second principal components. The weights indicates the individual contributions of each principal component. The graphs have helped to see the stocks with the highest influence on each component.

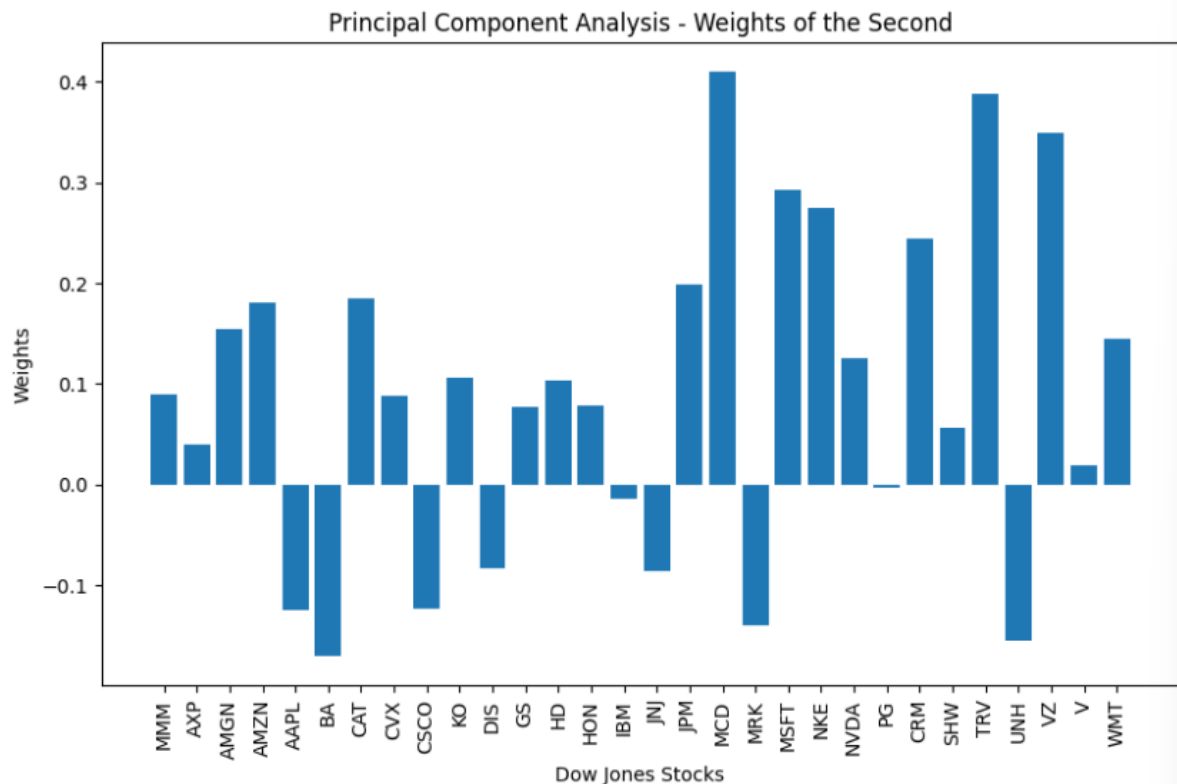
For each bar graph, I plotted Weights on the y-axis and “Dow Jones Stocks” on the x-axis.

Insights: The weights in this bar graph for First Component vary significantly. Stocks like the Amazon (AMZN), Nvidia (NVDA), and Microsoft (MSFT), exhibit dominant positive weights which suggest that they have strong influence on the component. On the flip side, stocks like Visa (V) and Walmart (WMT) have noticeable negative weights, suggesting an inverse relationship with the principal trend captured by the first component.



Principal Component Analysis of weights for the Second component

There are 4 Questions that are documented in total. Each Question is started on a new page. Please, scroll down even when you see a blank page.



Insights: The weights in this bar graph for Second Component shows a different distribution pattern. Stocks like Nvidia (NVDA) and Salesforce (CRM) display the highest positive weights, while stocks like Visa (V) and The Coca-Cola Company (KO) have noticeable weights. This implies that the second principal component captures a distinct variance dimension in the data, emphasizing the different stock's contributions.

Step 6: Explanation of the Market Similarity

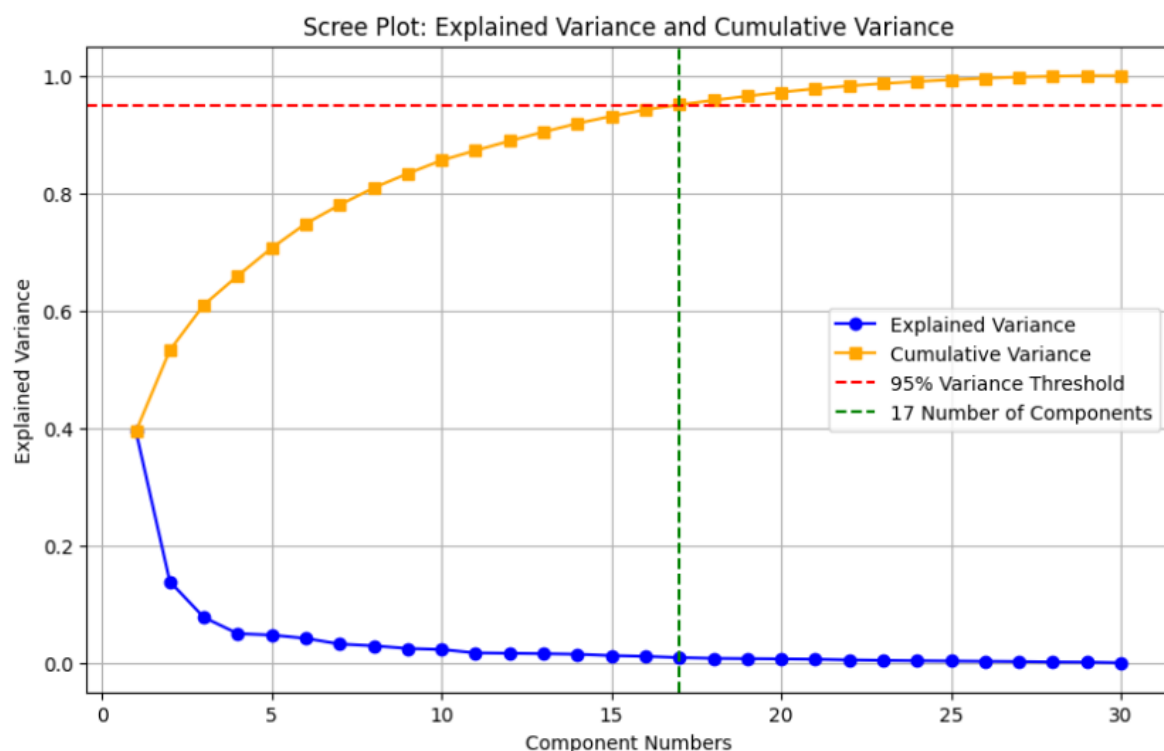
Comparing the first and second principal components to a market instance where all stocks have equal weights, the correlation coefficient between the components and the equal weights indicates how well each principal component reflects the broader market. The first principal component often captures the market behavior, comparing it to equal weights helps to determine if the component reflects the broad market movement or it is driven by specific sectors or stocks.

1.4 Explained Variance and Scree Plot

There are 4 Questions that are documented in total. Each Question is started on a new page. Please, scroll down even when you see a blank page.

I identified the minimum number of components needed to explain 95% of the total variance by calculating the cumulative variance which sums the explained variable of the principal components. The significance of obtaining 95% of the variance is to ensure that the most important patterns are captured without unnecessary complexity.

Also, I plotted the Scree Plot (line plot of eigenvalues). The y-axis shows the variance explained while the x-axis shows the principal component. The scree plot shows how much each principal component contributes to the total variance.

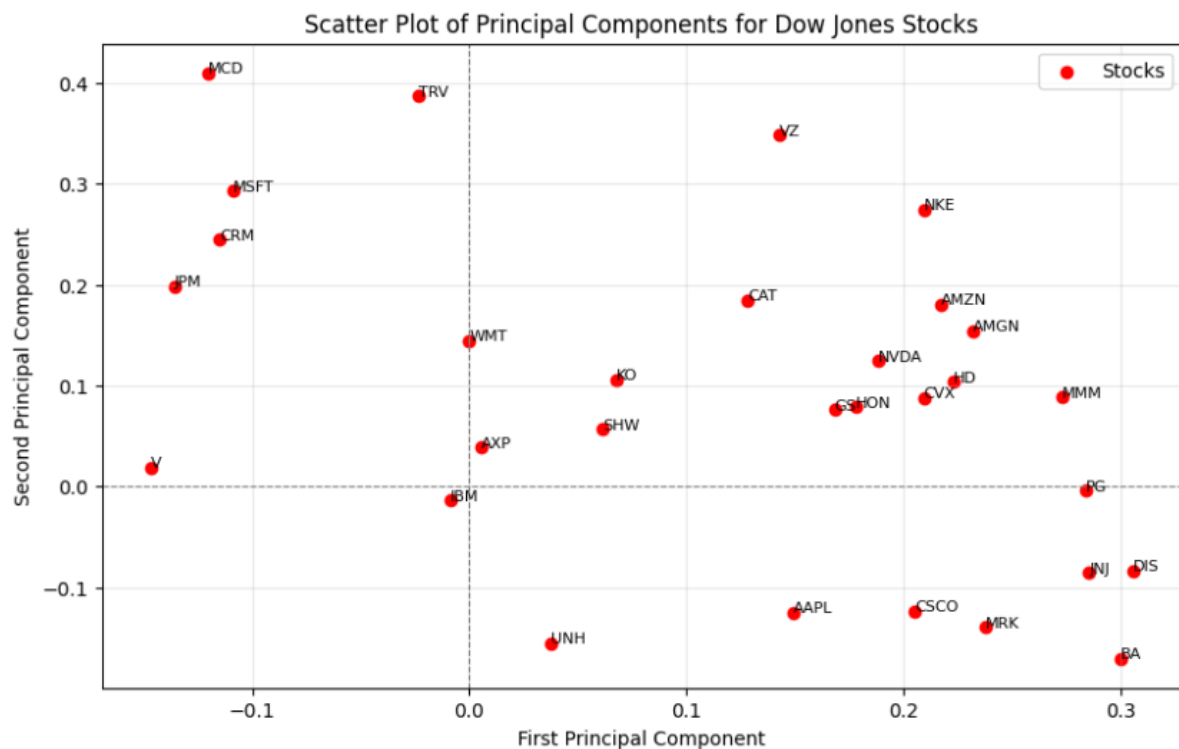


The Scree Plot suggests that just 17 components are needed to explain 95% of the variance in the data, although 30 components in total are in the dataset. This helps to eliminate redundancy in information and thereby simplifies the data while retaining most of the variability. It also aligns with the principle of using PCA to minimize complexity without significant loss in information.

1.5 Scatter Plot for the First Two Principal Components and Euclidean Distance

There are 4 Questions that are documented in total. Each Question is started on a new page. Please, scroll down even when you see a blank page.

I plotted the first two principal components against each other. The scatter plot shows how the stocks relate to each other based on the first two principal components.



I calculated the mean of all the 30 stocks for the first and second principal components and got the values:

First Principal Component (PCA1): **0.1149**

Second Principal Component (PCA2): **0.0901**

```
Mean of the First Principal Component, PCA1: 0.11492636454371195
```

```
Mean of the Second Principal Component, PCA2: 0.09006779410036962
```

I calculated the Euclidean distances from the mean and then I identified the three most distant stocks for each principal component.

For the First Principal Component (PCA1), the most distant stocks are:

- Visa (V),
- JP Morgan (JPM) and
- McDonalds Corporation (MCD)

For the Second Principal Component (PCA2), the most distant stocks are:

- McDonald's Corporation (MCD),

There are 4 Questions that are documented in total. Each Question is started on a new page. Please, scroll down even when you see a blank page.

- The Travelers Companies (TRV) and
- The Boeing Company (BA),

Most distant stocks for PCA1:

Stocks	Distance
V	0.261326
JPM	0.250783
MCD	0.235057

Most distant stocks for PCA2:

Stocks	Distance
MCD	0.319722
TRV	0.29727
BA	0.260516

QUESTION 2 – DENDOGRAM

There are 4 Questions that are documented in total. Each Question is started on a new page. Please, scroll down even when you see a blank page.

2.1 Description of the components of a Dendrogram, its construction and interpretation

A Dendrogram is a tree-like diagram that visualizes hierarchical clustering of data, showing the relationships between similar sets of data [2]

Dendrograms are tree-like diagrams that depict the layout of clusters generated by hierarchical clustering. They are made up of various fundamental components, such as branches, nodes, and leaves, which reflect data points' relationships and distances. A dendrogram is often constructed using agglomerative clustering methods, in which individual data points are gradually combined into bigger clusters based on similarity metrics. This method can be strengthened by applying optimal algorithms to improve interpretability, as shown in plant breeding data analysis.[3]

Components of a Dendrogram

- **Branches (Clades):** The branches represent the connections between clusters. This indicates the distance or dissimilarity.
- **Nodes:** These are points where branches split, showing the merging of clusters.
- **Leaves:** These are terminal points that represent individual data points or clusters.
- **Height:** This represents the dissimilarity or distance between clusters.

Construction Methods of Dendrograms

1. Compute pairwise distances (or dissimilarities) between all the observations
2. Use a hierarchical clustering algorithm e.g. single, complete, average linkage, to iteratively merge the closest clusters.
3. At each step, two clusters with the smallest distance are combined into a larger cluster until all the observations are grouped into one cluster.

How Dendrograms can be Interpreted

Clusters: Horizontal lines across the dendrogram identifies clusters while observations below a horizontal line form just one cluster.

There are 4 Questions that are documented in total. Each Question is started on a new page. Please, scroll down even when you see a blank page.

Similarity: Smaller heights indicate closer similarity between observations.

Hierarchy: The dendrogram shows nested relationships among clusters.

2.2 Steps for Constructing a Dendrogram

To construct a dendrogram from a collection of dissimilar pairwise values, a process involving hierarchical clustering method is used which allows for the visualization of the relationships among data points based on their dissimilarities.

1. Calculate Pairwise Dissimilarities – start by computing the dissimilarity matrix, which quantifies the pairwise distances between all the data points
2. Choose a Clustering Method – select an agglomerative or divisive clustering method. Agglomerative methods start with individual points and then merge them while divisive methods starts with the whole dataset and then splits them.[4]
3. Linkage Criteria – this defines the linkage criteria which could be single, complete, or average linkage, to determine the measure of the distance between the clusters [5]
4. Build the Dendrogram – this method iteratively merge clusters based on the chosen linkage criteria, updating the dendrogram at each step until all the points are merged into a single cluster [6]
5. Visualize the Dendrogram – at this stage, we represent the hierarchical structure of the dendrogram visually, where the height of the branches indicates the dissimilarity at which the clusters are merged.

2.3 Compute Pairwise Distances using Correlation Matrix

The pairwise distance matrix helps to quantify how similar or different each pair of stocks behaves. I determined the distance matrix using the formula

$$Distance = \sqrt{2 \times (1 - Correlation)}$$

Where correlation is the Pearson correlation coefficient between two stocks' returns. This was calculated in Question 1.

There are 4 Questions that are documented in total. Each Question is started on a new page. Please, scroll down even when you see a blank page.

After computing the distance matrix, I created a dataframe for it, making the distance matrix more interpretable.

Finally, I used the *tabulate()* function to print the dataframe to give a more visually appealing table. The matrix table generated is a huge one, but here is the screenshot of a portion of the table, showing a few stocks.

Pairwise Distance Matrix:

	MM	ASP	AMGN	AMZN	AMPL	BA	CAT	CVX	CSCO	KO	DIS	GS	HD	HON	IBM	JNJ	JPM	MCD	MRK	HSFT	NKE
MM	0	1.04716	1.25923	1.12743	1.04951	1.23178	0.90451	1.10247	1.09789	1.28502	1.24453	0.999331	1.10339	0.955465	1.03764	1.22946	0.954725	1.34785	1.34103	1.1522	1.08093
ASP	1.04716	0	1.14552	0.93368	0.945121	1.181	0.977417	1.07142	0.984575	1.19922	1.11997	0.897631	1.04938	0.810315	0.991428	1.14341	0.884579	1.15337	1.12825	0.826817	0.953888
AMGN	1.25923	1.14552	0	1.23253	1.20058	1.32132	1.2202	1.24357	1.23174	1.31698	1.27287	1.24266	1.32311	1.17044	1.21629	1.22428	1.2077	1.26622	1.02088	1.21112	1.21281
AMZN	1.12743	0.93368	1.23253	0	0.902343	1.19734	1.02625	1.1428	1.02592	1.27125	1.21964	1.07794	1.12251	1.0352	1.05311	1.20878	1.07447	1.27349	1.23765	0.744154	1.05133
AMPL	1.04951	0.945121	1.20058	0.902343	0	1.10405	0.983389	1.12322	0.970998	1.29387	1.10837	1.0192	1.07806	0.970782	1.04388	1.21494	1.04137	1.31076	1.21412	0.869592	1.01779
BA	1.23178	1.181	1.32132	1.19734	1.10405	0	1.09158	1.23601	1.20189	1.37679	1.2939	1.17919	1.22668	1.15085	1.21844	1.23996	1.17287	1.39438	1.32963	1.13426	1.20005
CAT	0.90451	0.977417	1.2202	1.02625	0.983389	1.09158	0	1.04829	1.07365	1.32819	1.22225	0.927088	1.09389	0.882294	1.04119	1.24775	0.886196	1.39786	1.33936	1.05578	1.06828
CVX	1.10247	1.07142	1.24357	1.1428	1.12322	1.23601	1.04829	0	1.1117	1.33002	1.37199	1.10035	1.1763	1.03621	1.12153	1.26598	1.08342	1.39979	1.27335	1.1024	1.14274
CSCO	1.09789	0.984575	1.23174	1.02592	0.970998	1.20189	1.07365	1.1117	0	1.35226	1.1715	1.0901	1.14576	0.954567	0.909239	1.16314	1.0557	1.23455	1.18183	0.94684	0.99778
KO	1.28502	1.19922	1.31698	1.27125	1.29387	1.37679	1.32819	1.33002	1.35226	0	1.20966	1.3524	1.28261	1.22538	1.27854	1.28058	1.30394	1.21404	1.14111	1.20925	1.24163
DIS	1.24453	1.11997	1.27287	1.21964	1.10837	1.2939	1.22225	1.37199	1.1715	1.20966	0	1.20860	1.12016	1.12378	1.22511	1.25990	1.05747	1.20095	1.32265	1.15904	1.13825
GS	0.999331	0.897631	1.24266	1.07794	1.0192	1.17919	0.927088	1.10035	1.0901	1.3524	1.20168	0	1.07934	0.971113	1.06304	1.3005	0.649339	1.3543	1.33555	1.03072	1.02877
HD	1.10339	1.04938	1.32311	1.12251	1.07806	1.22668	1.09389	1.1763	1.14576	1.28261	1.12016	1.07934	0	1.06224	1.11758	1.29201	1.04898	1.20964	1.29811	1.02618	1.01452
HON	0.955465	0.810315	1.17044	1.0352	0.970782	1.15085	0.882294	1.03621	0.954567	1.22538	1.12378	0.971113	1.06224	0	0.964136	1.15726	0.89488	1.15797	1.1677	0.913	0.947683
IBM	1.03764	0.991428	1.21629	1.05311	1.04388	1.21844	1.04119	1.12153	0.909239	1.27854	1.22511	1.06304	1.11758	0.964136	0	1.22474	1.03878	1.2788	1.21122	0.967359	1.08162
JNJ	1.22946	1.14341	1.22428	1.20878	1.21494	1.23996	1.24775	1.26598	1.16314	1.28058	1.25998	1.3005	1.29201	1.15726	1.22474	0	1.25578	1.32498	1.13772	1.17341	1.21789
JPM	0.954725	0.884579	1.2077	1.07447	1.04137	1.17287	0.886196	1.08342	1.0557	1.30394	1.05747	0.649339	1.04898	0.89488	1.03878	1.25578	0	1.32272	1.31474	1.04308	0.987763
MCD	1.34785	1.15337	1.26622	1.27349	1.31076	1.39438	1.39786	1.39979	1.23455	1.21404	1.20095	1.3543	1.20964	1.15797	1.2788	1.32498	1.32272	0	1.10769	1.13634	1.15304
MRK	1.34103	1.12825	1.02088	1.23765	1.21412	1.32963	1.33936	1.27335	1.18183	1.14111	1.32265	1.33555	1.29811	1.1677	1.21122	1.13772	1.31474	1.10769	0	1.10598	1.26913

Understanding what the distances mean

The distances between two stocks represent their level of dissimilarity in terms of their movements or direction. Each distance value is derived from their correlation.

- When **Distance = 0**, it means that the two stocks are **perfectly correlated** i.e. they move in the same direction with a correlation of 1.
- When **Distance > 0 but < 2**, it means that the two stocks have **some correlation** but are not perfectly correlated.
- When **Distance = sqrt (2)**, it means that the two stocks have **no correlation** i.e. their movements are independent of each other.

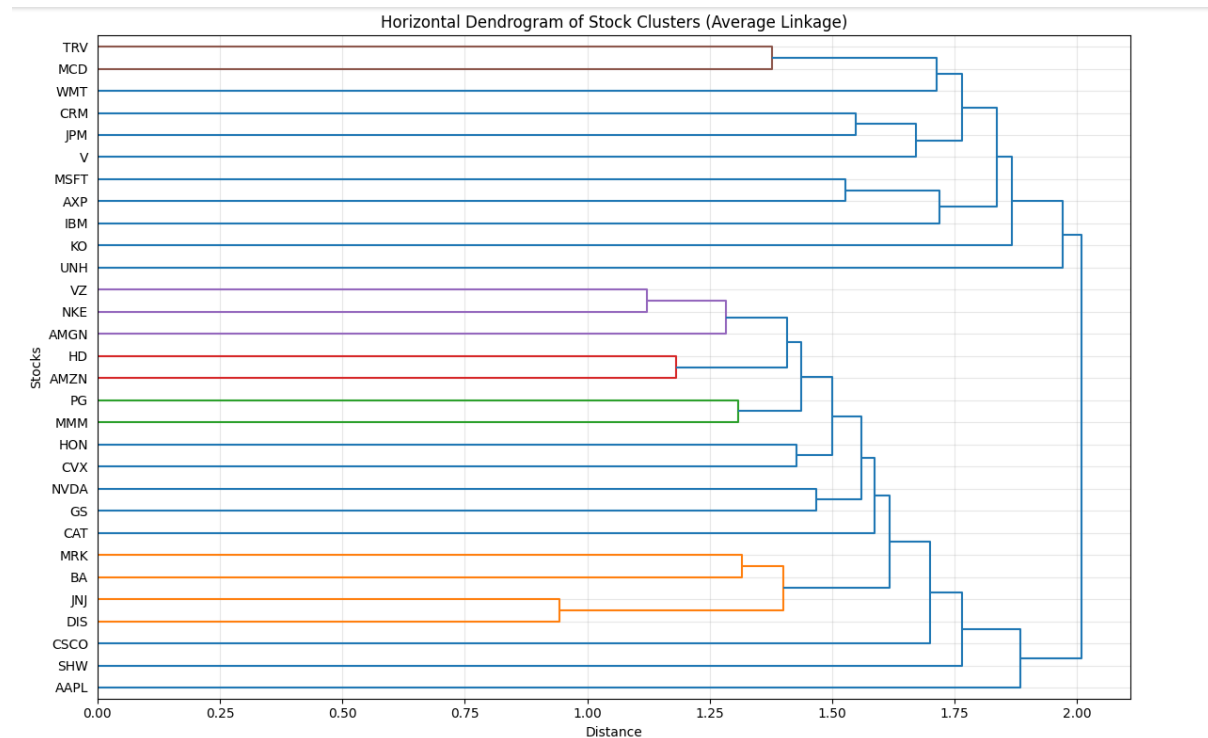
2.4 Create a Dendrogram using the Linkage Approach

I used the *linkage()* method from *scipy.cluster.hierarchy* to perform the hierarchical clustering. I passed the distance matrix from question 2.3 as input to calculate the

There are 4 Questions that are documented in total. Each Question is started on a new page. Please, scroll down even when you see a blank page.

linkage matrix. I used the **average** linkage method which calculates the distance between clusters by averaging the pairwise distances between the points in the clusters.

I used the *dendrogram()* function to plot a hierarchical tree, which shows the arrangement of the stocks in clusters.



Insight: The dendrogram helped to understand and identify the stocks that are most similar when their historical price movements are considered. It also visually demonstrate the relationship between different stock behaviors. The stocks with similar clusters tend to respond similarly to market movements while stocks in separate clusters behave distinctly. Stocks like TRV, MCD, WMT, etc. in the topmost cluster are grouped together with short branches suggesting high similarity in their behavior. Stocks like AAPL, CSCO, SHW, etc. in the bottom cluster are grouped together with longer branches, suggesting that there is a significant dissimilarity between the stocks when compared to the topmost cluster.

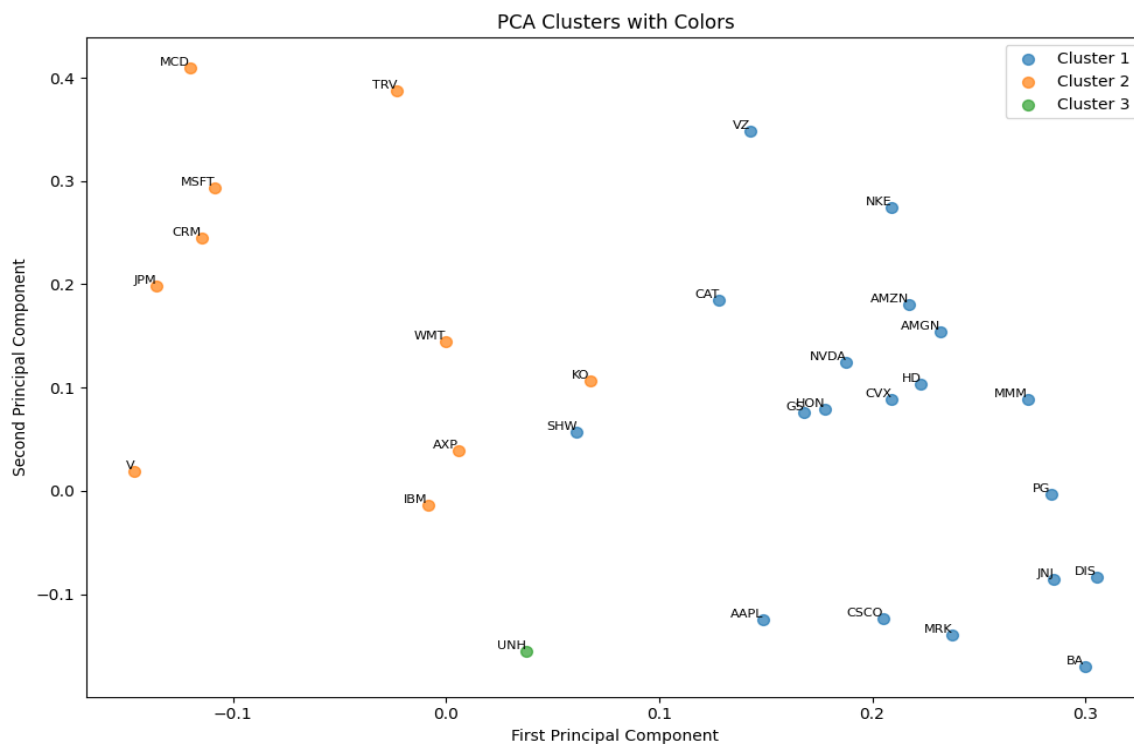
2.5 Creation of Clusters using inputs from Linkage Method in Dendrogram

There are 4 Questions that are documented in total. Each Question is started on a new page. Please, scroll down even when you see a blank page.

I used the *fcluster()* function to extract the three clusters from the hierarchical clustering linkage matrix. This ensures that the stocks are grouped into clusters based on their similarity.

Using PCA for Dimensionality Reduction, the first two principal components (PCA1 and PCA2) are used to project the stocks into a 2D scatter plot, representing each cluster with a different color.

Each cluster is plotted with color coding, with each point in the clusters annotated with its ticker symbol. This enables clarity, visibility and proper identification of each stock and cluster.



```
Cluster 1: ['MMM', 'AMGN', 'AMZN', 'AAPL', 'BA', 'CAT', 'CVX', 'CSCO', 'DIS', 'GS', 'HD', 'HON', 'JNJ', 'MRK', 'NKE', 'NVDA', 'PG', 'SHW', 'VZ']
Cluster 2: ['AXP', 'KO', 'IBM', 'JPM', 'MCD', 'MSFT', 'CRM', 'TRV', 'V', 'WMT']
Cluster 3: ['UNH']
```

Cluster 1: ['MMM', 'AMGN', 'AMZN', 'AAPL', 'BA', 'CAT', 'CVX', 'CSCO', 'DIS', 'GS', 'HD', 'HON', 'JNJ', 'MRK', 'NKE', 'NVDA', 'PG', 'SHW', 'VZ']

Cluster 2: ['AXP', 'KO', 'IBM', 'JPM', 'MCD', 'MSFT', 'CRM', 'TRV', 'V', 'WMT']

Cluster 3: ['UNH']

There are 4 Questions that are documented in total. Each Question is started on a new page. Please, scroll down even when you see a blank page.

Cluster 1 includes a varied range of stocks, including **technology** giants like Amazon (AMZN), and Apple (AAPL), as well as **industrial and consumer products** companies like 3M (MMM) and Caterpillar (CAT). This suggests that stocks in Cluster 1 have some comparable features which could reflect their performance in connection to economic cycles or their reaction to market occurrences. This cluster has diverse sector representation such as **Technology** (Amazon, Apple), **Industrial Manufacturing** (Caterpillar, 3M), **Consumer Goods** (Procter & Gamble), and **Energy** (Chevron) sectors.

Cluster 2 includes major financial institutions such as American Express (AXP) and JPMorgan (JPM), along with technology stocks such as Microsoft (MSFT) and Salesforce (CRM). The grouping of financials and tech in this cluster may suggest that these stocks exhibit similar behavior in certain market conditions, potentially related to macroeconomic factors like interest rates or inflation expectations. This cluster is dominated by Financial and Technology companies such as American Express, JPMorgan Chase, Travelers in the Financial sector, and Microsoft, Salesforce, IBM in the technology sector. Companies like Coca-Cola and McDonald's which represent **Consumer Foods** sector are also present.

Cluster 3 include only **UnitedHealth (UNH)**. This indicates that it behaves quite differently from the others, at least in terms of the features used for clustering (like stock performance, volatility, or returns). This may suggest that the healthcare sector has distinct characteristics in relation to the other stocks, perhaps driven by regulatory factors, consumer health trends, or healthcare-specific events.

QUESTION 3 – ENSEMBLES FOR CLASSIFICATION

3.1 Sources of Uncertainty and their influence on modelling process in machine learning

Uncertainty in machine learning models can come from several sources, significantly impacting the modeling process. To improve model reliability and performance, it is important to understand uncertainties. The three primary sources of uncertainties are input uncertainty, data uncertainty, and model uncertainty.

1. Input Uncertainty

This refers to the variability or noise that is present in the input data.

Impact: Propagating the input uncertainty through the model can stabilize the decision boundaries, and enhance prediction reliability even under noisy conditions [7] Input uncertainty is particularly profitable when the quantity of the input uncertainty is known, although high-quality datasets are necessary for effective modeling.

2. Data Uncertainty

Data Uncertainty encompasses imperfections in datasets, such as noise, outliers, and missing values. To mitigate the effects of these imperfections, machine learning techniques like Capped Extended Support Vector Regression are useful, which helps to produce more accurate structural performance estimates [8]

Impact: The presence of noise can limit observed model performance, hiding the true potentials of the model [9]

3. Model Uncertainty

Model uncertainty is as a result of the limitations of the model itself, including bias and variance.

Impact: Addressing model bias and variance is essential for improving predictions, especially in complex domains like chemistry [9]. In addition, ensembling methods can effectively quantify and reduce model uncertainty, thereby enhancing overall model robustness [9]

While these uncertainties can hinder model performance, they also present opportunities for improvement through advanced modeling techniques and better data

There are 4 Questions that are documented in total. Each Question is started on a new page. Please, scroll down even when you see a blank page.

management. Understanding and addressing these uncertainties is vital for developing more reliable machine learning applications.

3.2 Concept of Model averaging and Examples of implementation in practice

Model averaging is a statistical technique that is designed to measure the inherent uncertainty in the model selection process. Most times, the traditional statistical analysis neglects the uncertainty in model structures. Model averaging has been applied successfully to many statistical model classes because it is successful at improving predictive performance. Model averaging combines predictions from multiple models to improve overall accuracy and robustness.

Implementation in practice...

1. Bayesian Model Averaging

This assigns probabilities to different models on the basis of their posterior probabilities given the data. It averages the weights of predictions by the probabilities. Bayesian Model Averaging provides direct model selection, combined estimation and prediction.

Strength: It explicitly captures model uncertainty and it offer probabilistic insights. Also, it provides better predictive performance when models are well-calibrated.

Challenges: It is computationally intensive for large datasets or complex models, and it requires careful prior specification for models.

Use Case: It is commonly used in settings where the objective is robust uncertainty quantification, such as in medical diagnostics.

2. Federated Learning Applications

In federated learning, model averaging aggregates client models trained on different datasets to create a global model, using local datasets. It enhances performance even with heterogenous data distributions. [10] Iterative Moving Average (IMA) can be employed to refine the global model during late training, thereby reducing the prediction error and improving accuracy [11]

Strength: It ensures data privacy of local data on devices, and it enhances scalability to large, distributed datasets.

There are 4 Questions that are documented in total. Each Question is started on a new page. Please, scroll down even when you see a blank page.

Challenges: It is very sensitive to data heterogeneity across devices.

Use Case: Particularly useful in privacy-sensitive domains like healthcare diagnostics, finance systems for fraud detection

3. Data Subset Selection

Model Averaging can also be applied in data subset selection, where different models are compared across various data subsets. With this approach, we are able to manage model uncertainty and improve predictive performance [12]

Strength: It reduces the variance by combining models trained on wide range of data samples. It is also simple to implement, especially in ensemble methods like bagging.

Challenges: Its performance largely depends on the quality and diversity of subsets, and it may not handle systematic biases that are present in all subsets.

Use Case: It is widely used in random forests

3.3 Ensemble Methods Description and Reduction of Uncertainty Effects

Ensemble Methods in machine learning are powerful techniques that combine multiple models to enhance predictive performance and to mitigate uncertainty. By aggregation the prediction of several models, the methods can reduce variance and bias, leading to more robust outcomes. The following are key ensemble methods and their approach to improving model accuracy.

1. Bagging

This technique entails training several models independently on random subsets of data and then finding the average of their results. It is effective in minimizing the variance, and it is especially useful for high-variance models such as decision trees [13]

2. Boosting

Boosting trains models sequentially. Each model focuses on the errors made by the previous one. This method reduces bias and can significantly improve accuracy as seen in the diagnosis of diabetes. [14]

3. Stacking

Stacking combines different models such as Random Forest and Support Vector Machines, to create a meta-model that aggregates their predictions. Stacking can

There are 4 Questions that are documented in total. Each Question is started on a new page. Please, scroll down even when you see a blank page.

leverage the strengths of diverse algorithms, thereby improving overall performance. [14]

Application and Benefits

1. Disease Prediction: Ensemble methods have been very useful in the prediction of various infectious diseases, diabetes, cancer etc. They help to improve diagnostic accuracy and reduce false positives and negatives. [15]
2. Forecasting: In nonstationary contexts, ensembles of metamodels can effectively capture complex patterns leading to improved forecasting accuracy when compared to individual models.[16]

3.4 Construction of Random Forest Model

Step 1: Loading the Titanic Dataset

I loaded the *titanic3.csv* file into the Google Colaboratory tool using *pandas.csv()* function. The dataset contained 1309 entries. It also has NaN values. I have used the *head()* and *info()* functions to understand the dataset.

Step 2: Data Preprocessing

I selected 'pclass', 'sex', 'age' and 'fare' columns as the *predictor* variables while I selected 'survived' as the *target* variable, which is the dependent variable to be predicted. I handled the missing values in the 'age' and the 'fare' column, replacing them with the median value because the median is more robust to outliers. Also, I transformed the categorical feature, 'sex' column to numerical values, using 0 to represent male, and 1 to represent female. Lastly, I selected the X and y variables to be used in the next step.

Step 3: Split Dataset into Training and Testing set

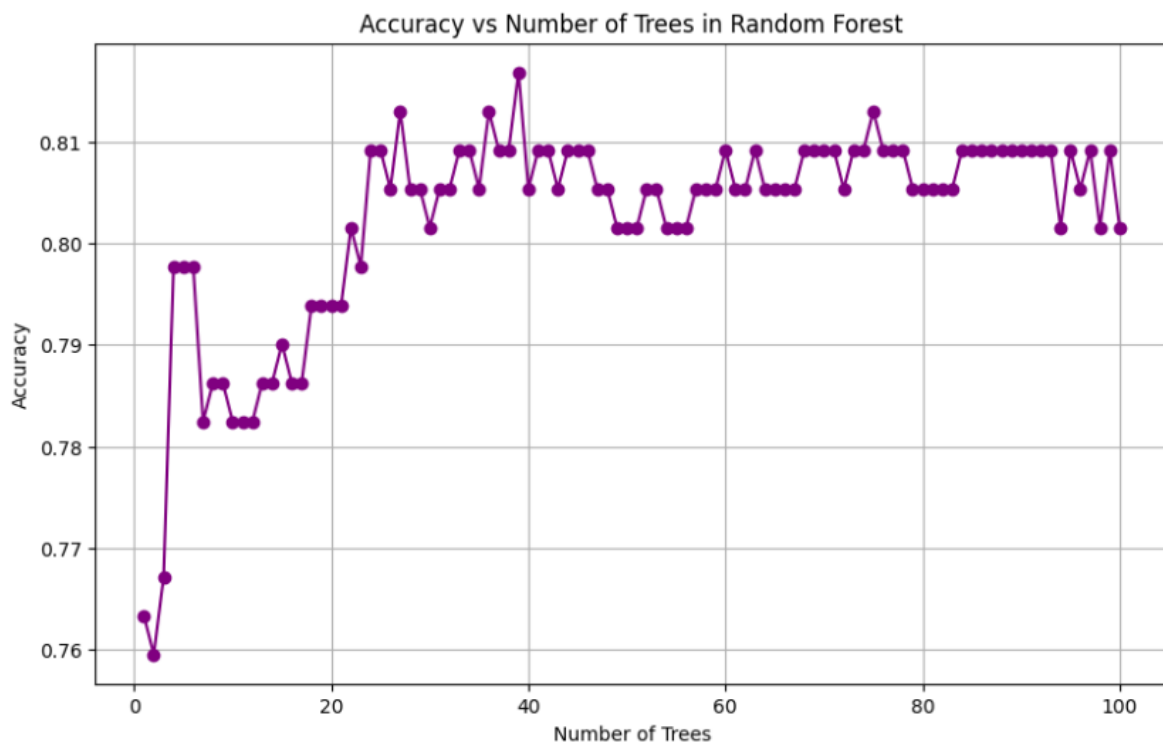
I split my dataset into the training and testing set using the *train_test_split()* function. The training set is 80% while the testing set is 20%.

Step 4: Train the Random Forest Model

I instantiated the model using *RandomForestClassifier()* function and trained with varying number of trees, *n_estimators* to find the optimal count.

There are 4 Questions that are documented in total. Each Question is started on a new page. Please, scroll down even when you see a blank page.

I plotted accuracy versus the number of trees.



I also calculated the **Optimal Number of Trees to be 39**

3.5 Perform ROC Analysis for Logistic Regression, Decision Tree, Random Forest and KNN

Another reason why I had to drop Null values was because of Logistic Regression.

Step 1: Fitting the Models

I trained the Logistic Regression model, Decision Tree, Random Forest, and KNN model on the preprocessed Titanic training dataset.

Step 2: Generate Predictions

I obtained the predicted probabilities for the test dataset.

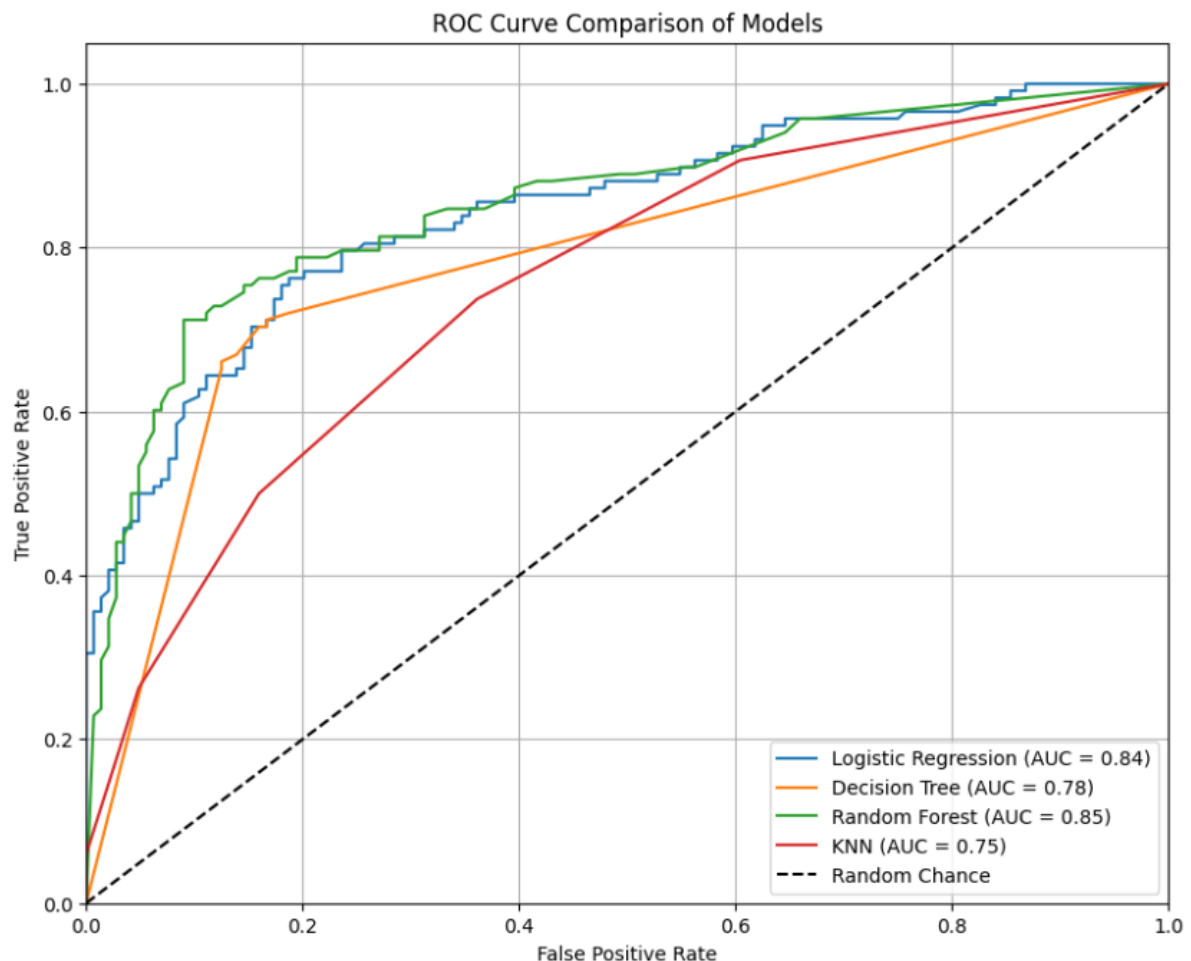
Step 3: Compute ROC Curve

I computed the True Positive Rate (TPR) and False Positive Rate (FPR) using the ROC Curve `roc_curve()` from the Scikit-Learn library.

There are 4 Questions that are documented in total. Each Question is started on a new page. Please, scroll down even when you see a blank page.

Step 4: Plot the ROC Curve

I plotted the ROC Curve of True Positive Rate (TPR) versus False Positive Rate (FPR) for each of the models. I also added a diagonal for comparison. The plot has a legend showing the Area Under the Curve (AUC) for each of the models.



Step 5: Compare the Models

I compared each of the model's accuracy and displayed the values

```
Logistic Regression Accuracy: 0.76
Decision Tree Accuracy: 0.77
Random Forest Accuracy: 0.81
KNN Accuracy: 0.67
```

Logistic Regression Accuracy is **0.76**

Decision Tree Accuracy is **0.77**

There are 4 Questions that are documented in total. Each Question is started on a new page. Please, scroll down even when you see a blank page.

Random Forest Accuracy is **0.81**

KNN Accuracy is **0.67**

From the AUC values, Random Forest has performed better than every other models considered, with an AUC of **0.84**. KNN has the least Area Under the Curve value of **0.75**.

QUESTION 4 – ENSEMBLES FOR CLASSIFICATION

There are 4 Questions that are documented in total. Each Question is started on a new page. Please, scroll down even when you see a blank page.

4.1 Random Forest Regression Model

Random Forest (RF) regression models are powerful machine learning tools that utilize an ensemble of decision trees to predict continuous outcomes. They excel in handling complex datasets by capturing intricate relationships between features and target variables. RF regression operates by aggregating the predictions from multiple trees, which enhances accuracy and robustness against overfitting. This model is particularly effective in scenarios with high-dimensional data and noise, as demonstrated in various applications, including climate impact analysis and material fatigue strength prediction.

Key Features of Random Forest Regression

- **Ensemble Learning:** Random Forest combines predictions from numerous decision trees, improving overall model accuracy and stability [17]
- **Feature Importance:** RF evaluates the significance of each feature, allowing researchers to identify critical predictors in the dataset [18]
- **Noise Resilience:** The Bagging method used in RF helps to mitigate the effects of data inaccuracies and noise, making it suitable for complex problems [19]

Applications of Random Forest Regression

- **Climate Impact Pathways:** Random Forest Regression has been employed to trace and rank the impacts of climate disturbances, showcasing its usefulness in environmental science [20]
- **Material Science:** In the prediction of fatigue strength of ferrous alloys, Random Forest regression outperformed traditional methods, significantly reducing analysis time and costs [21]

4.2 Construction Random Forest Trees with different number of leaves

Step 1: Loading Dataset and Preprocessing

I loaded the red wine dataset and separated each column with comma (,). Thereafter, I split the dataset into features – X, and Target variable – y.

Step 2: Splitting of Dataset into Training and Testing Sets

There are 4 Questions that are documented in total. Each Question is started on a new page. Please, scroll down even when you see a blank page.

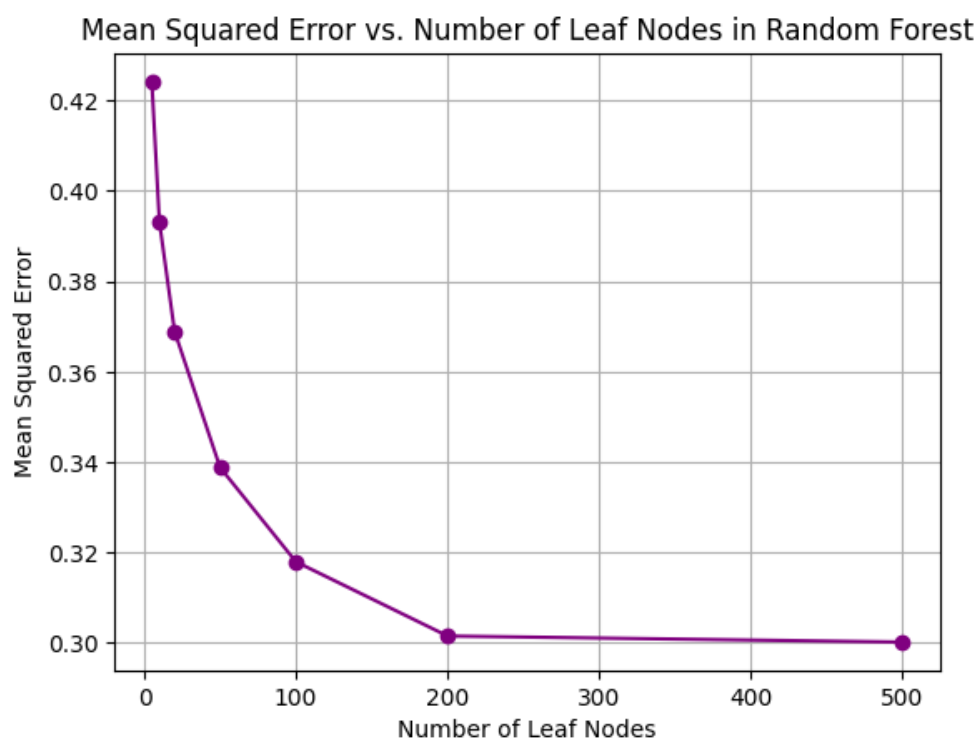
I split the dataset into training and testing sets, having 80% of training and 20% of testing set.

Step 3: Training of Random Forest Model with different number of leaves

I constructed the *RandomForestRegressor* with 100 n estimators and `max_leaf_nodes`. I calculated the mean squared error on the test set and plotted a graph to visualize the relationship between the number of leaves and model performance. From the graph, I obtained the optimal number of leaves, **20**, the lowest MSE which is subsequently used.

Step 4: Construct Random Forest model with the optimal number of leaves

The model is built with the optimal number of leaves identified from the previous plot, and the MSE is printed to evaluate the model's performance

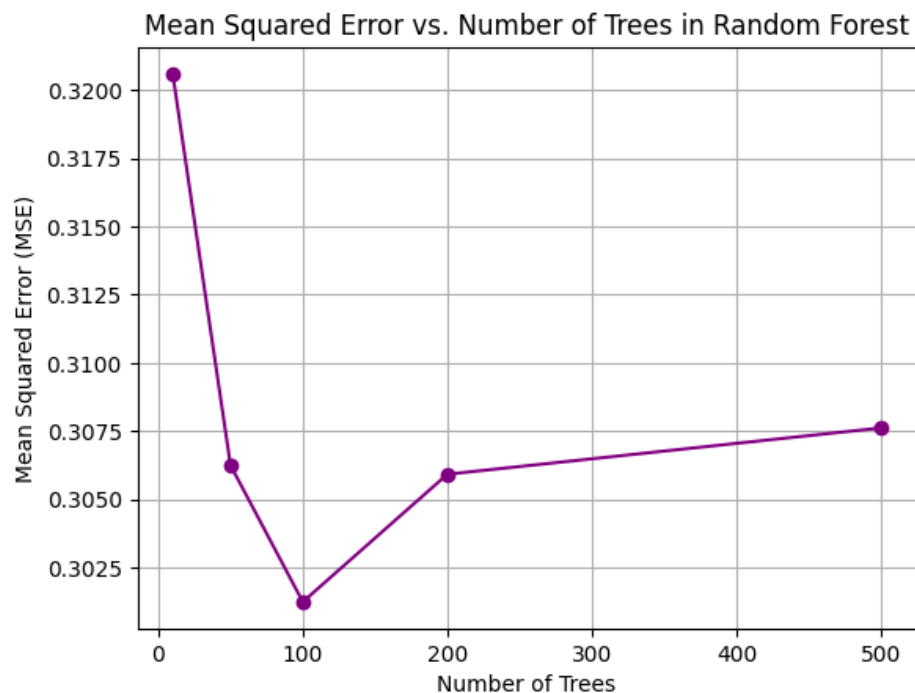


4.3 Give the optimal number of trees and Explain how it was computed

Model Training: The model is trained with different numbers of trees, keeping the `max_leaf_nodes` constant.

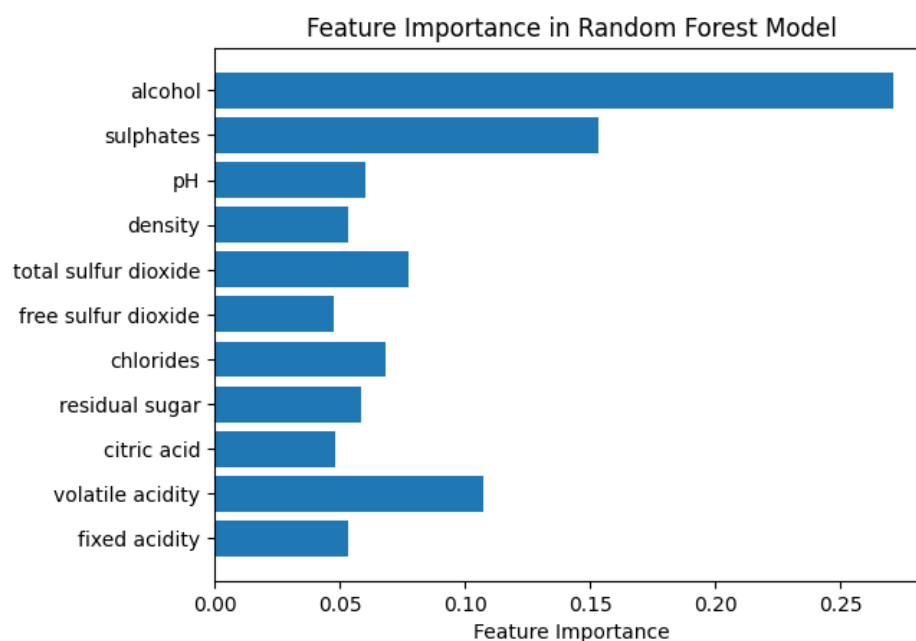
There are 4 Questions that are documented in total. Each Question is started on a new page. Please, scroll down even when you see a blank page.

Error Calculation and Plotting: The MSE is calculated for each model and plotted against the number of trees.



Mean Squared Error with Optimal Number of Trees: 0.31796846624071373

4.4 Bar Graph of the Features showing importance of each



There are 4 Questions that are documented in total. Each Question is started on a new page. Please, scroll down even when you see a blank page.

4.5 Calculate the Performance of Random Forest Model

Mean Squared Error (MSE) for Linear Regression: 0.3900251439639545

Mean Squared Error (MSE) for KNN Regression: 0.5319999999999999

Mean Squared Error (MSE) for Random Forest: 0.3689440605612798

References

- [1] H. Abdi and L. J. Williams, "Principal component analysis," *WIREs Comput. Stat.*, vol. 2, no. 4, pp. 433–459, 2010, doi: 10.1002/wics.101.
- [2] Alexander, "Hierarchical Clustering / Dendrogram: Simple Definition, Examples," Statistics How To. Accessed: Dec. 04, 2024. [Online]. Available: <https://www.statisticshowto.com/hierarchical-clustering/>
- [3] "Application of a dendrogram seriation algorithm to extract pattern from plant breeding data," *Euphytica*, vol. 213, no. 4, pp. 1–11, Mar. 2017, doi: 10.1007/S10681-017-1870-Z.
- [4] "Hierarchical clustering algorithm for dendrogram construction and cluster counting," *Informatika Ta Mat. Metodi V Model.*, vol. 13, no. 1–2, pp. 5–15, Apr. 2023, doi: 10.15276/imms.v13.no1-2.5.
- [5] "Hierarchical Clustering with OWA-based Linkages, the Lance-Williams Formula, and Dendrogram Inversions," *arXiv.org*, vol. abs/2303.05683, Mar. 2023, doi: 10.48550/arXiv.2303.05683.
- [6] "A Simple and Efficient Method to Compute a Single Linkage Dendrogram.," *ArXiv Data Struct. Algorithms*, Nov. 2019, Accessed: Dec. 04, 2024. [Online]. Available: <https://typeset.io/papers/a-simple-and-efficient-method-to-compute-a-single-linkage-2tgmp3woa5>
- [7] "Unified Uncertainties: Combining Input, Data and Model Uncertainty into a Single Formulation," *SciSpace - Paper*. Accessed: Dec. 05, 2024. [Online]. Available: <https://typeset.io/papers/unified-uncertainties-combining-input-data-and-model-4vkzv8bf3i>
- [8] "Machine learning aided uncertainty quantification for engineering structures involving material-geometric randomness and data imperfection," *Comput. Methods Appl. Mech. Eng.*, Apr. 2024, doi: 10.1016/j.cma.2024.116868.
- [9] "Characterizing Uncertainty in Machine Learning for Chemistry," *J. Chem. Inf. Model.*, vol. 63, pp. 4012–4029, Jun. 2023, doi: 10.1021/acs.jcim.3c00373.
- [10] "Understanding Model Averaging in Federated Learning on Heterogeneous Data," *arXiv.org*, vol. abs/2305.07845, May 2023, doi: 10.48550/arXiv.2305.07845.
- [11] "Understanding and Improving Model Averaging in Federated Learning on Heterogeneous Data," *IEEE Trans. Mob. Comput.*, pp. 1–16, Jan. 2024, doi: 10.1109/tmc.2024.3406554.
- [12] "Model averaging approaches to data subset selection," *SciSpace - Paper*. Accessed: Dec. 05, 2024. [Online]. Available: <https://typeset.io/papers/model-averaging-approaches-to-data-subset-selection-2ynm3h0m>
- [13] "Tree-Based Ensemble Models, Algorithms and Performance Measures for Classification," *Adv. Sci. Technol. Eng. Syst. J.*, Nov. 2023, doi: 10.25046/aj080603.
- [14] "Ensemble Machine Learning Approach for Detecting and Predicting Diabetes Mellitus Using Bagging and Stacking," *SciSpace - Paper*. Accessed: Dec. 05, 2024. [Online]. Available: <https://typeset.io/papers/ensemble-machine-learning-approach-for-detecting-and-23wh9axhv0>
- [15] "Application of ensemble machine learning methods for diabetes diagnosis," *BIO Web Conf.*, vol. 121, pp. 01002–01002, Jan. 2024, doi: 10.1051/bioconf/202412101002.

There are 4 Questions that are documented in total. Each Question is started on a new page. Please, scroll down even when you see a blank page.

- [16] "Ensemble of adaptive predictors for multivariate nonstationary sequences and its online learning," *Radio Electron. Comput. Sci. Control*, no. 4, pp. 91–91, Jan. 2024, doi: 10.15588/1607-3274-2023-4-9.
- [17] "Random forests with parametric entropy-based information gains for classification and regression problems," *PeerJ*, vol. 10, Jan. 2024, doi: 10.7717/peerj-cs.1775.
- [18] "Random Forest-Based Analysis of Variability in Feature Impacts," SciSpace - Paper. Accessed: Dec. 05, 2024. [Online]. Available: <https://typeset.io/papers/random-forest-based-analysis-of-variability-in-feature-3k225x3xq17r>
- [19] "Random Forest Regression-based Model for Fitting Multilayer Isotropic Medium Scattering Problems," SciSpace - Paper. Accessed: Dec. 05, 2024. [Online]. Available: <https://typeset.io/papers/random-forest-regression-based-model-for-fitting-multilayer-31og0w34ljoe>
- [20] "Random Forest Regression Feature Importance for Climate Impact Pathway Detection," SciSpace - Paper. Accessed: Dec. 05, 2024. [Online]. Available: <https://typeset.io/papers/random-forest-regression-feature-importance-for-climate-41p8yckcc68l>
- [21] "A random forest regression with Bayesian optimization-based method for fatigue strength prediction of ferrous alloys," *Eng. Fract. Mech.*, Nov. 2023, doi: 10.1016/j.engfracmech.2023.109714.

There are 4 Questions that are documented in total. Each Question is started on a new page. Please, scroll down even when you see a blank page.