CARNEGIE MELLON UNIVERSITY - AFRICA

DATA, INFERENCE & APPLIED MACHINE LEARNING

(COURSE 18-785)

Professor Patrick McSharry

Assignment 7

MUNYANEZA Kenny Roger

04th December 2023

Question 1

1. PCA

1.1.1.

Principal Component Analysis (PCA) is an unsupervised learning algorithm that uses an orthogonal transformation to break the correlation between variables to form a set of linearly uncorrelated variables called principal components(Pcs) with the goal of reducing dimensionality of a dataset but also preserving the most important relationships between variables.
Pcs are arranged in terms of the amount of variance captured, whereby the pc in first position indicates the maximum variance.

1.1.2. Application of PCA in machine learning:

- **Feature Selection:** Principal Component Analysis is utilized for feature selection, a critical process in identifying essential variables within a dataset. In machine learning scenarios with many variables, discerning the most important ones can be challenging. PCA streamlines this process, aiding in the identification of key features crucial for model performance.

- **Data Visualization:** PCA serves as a powerful tool for data visualization by reducing the dimensionality of the dataset. High-dimensional data can be effectively represented in two or three dimensions, simplifying interpretation. This capability is especially valuable in gaining insights from complex datasets, making patterns and relationships more accessible.

- **Noise Reduction:** Principal Component Analysis contributes to noise reduction in datasets. By excluding principal components with low variance, assumed to represent noise, PCA enhances the signal-to-noise ratio. This reduction in noise makes it easier to discern the true underlying structure within the data, improving overall data quality and analysis outcomes.

1.1.3. Advantages of PCA:

- Since PCA is based on linear algebra, it is computationally easy to solve.
- Principal Components-trained machine learning algorithms converge faster than machine learning algorithms trained on the original dataset.
- By using PCA beforehand to transform explanatory variables lowers the dimensions of the training dataset, hence preventing over-fitting.
  [1] [2]

1.2

Equation 1: Standardization equation

Z = $\frac{X-u}{\sigma}$, Where σ is the standard deviation, u is the mean and X is the original dataset.

This equation removes any bias from the distribution by ensuring that each variable has a mean of 0 and a standard deviation of 1.

Equation 2: Covariance calculation equation

$C = \frac{1}{n-1} X.X^T$, where X is the Dataset Matrix , X^T is the transpose of X, and n is the number of elements.

The resultant covariance matrix is square matrix of d x d dimensions, and along the diagonal of each covariance matrix are variance scores for each variable.

Equation 3: Eigenvalues calculation equation

$|C - \lambda I| = 0$, where λ is the eigenvalue and I is the Identity Matrix, and | - | is the determinant.

This equation's roots give the eigenvalues λ_1 to λ_n .

Equation 4: Equation to find eigenvectors from eigenvalues

For each eigenvalue λ, we determine eigenvectors using the equation:

$C . X = \lambda . X$, where C is the covariance matrix, X is a matrix of the variables in the dataset and λ is the lambda value.

The eigenvector with the highest eigenvalue is the Principal Component of the dataset. The top k eigenvectors, where k is the desired dimensionality of the reduced data, are selected to form the projection matrix **W**. The matrix W is used to transform the original data into the new subspace.

Equation 5: Transforming the Samples onto the new subspace

To transform our samples onto the new subspace, we use the k dimensional matrix W in the equation:

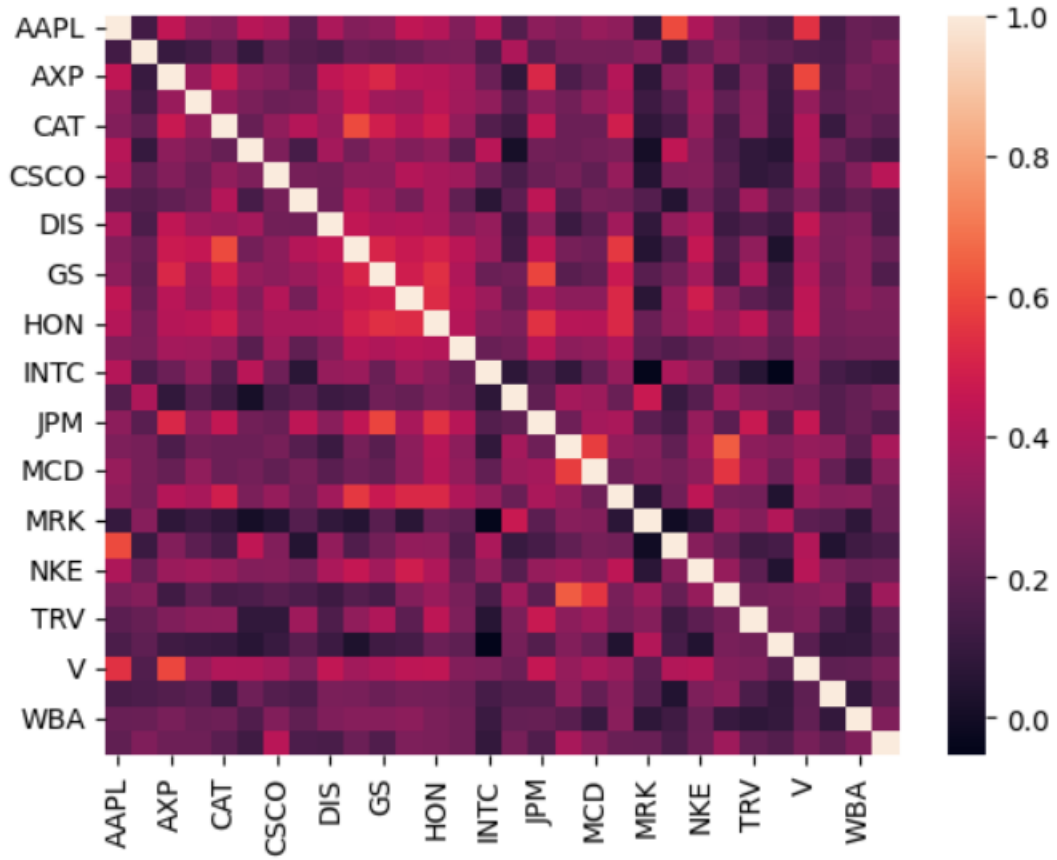
Z = W' . x, where W' is the transpose of matrix W.

Z is the transformed matrix, where each column represents a principal component, and each row represents an observation. The columns are uncorrelated, and the data is represented in a reduced-dimensional space. [3] [4]

1.3.

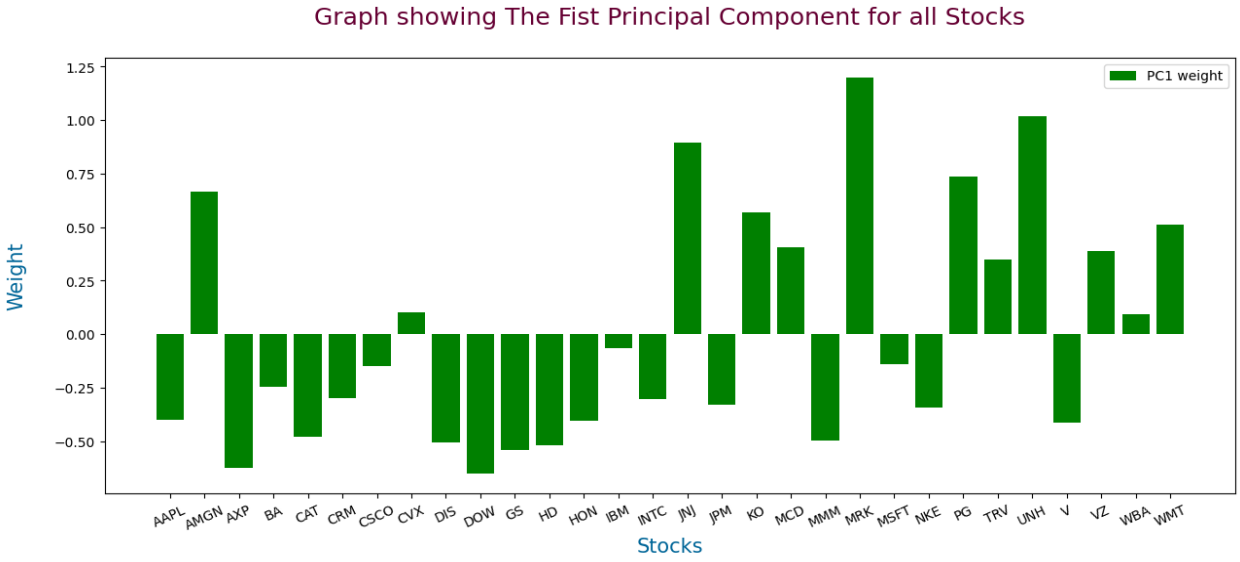
Using yahoo finance module, I Loaded the Dow Jones Index dataset with 30 stocks, for a year recorded daily. Then, I calculated the daily returns for each stock and computed the correlation matrix of the returns amongst the 30 stocks.

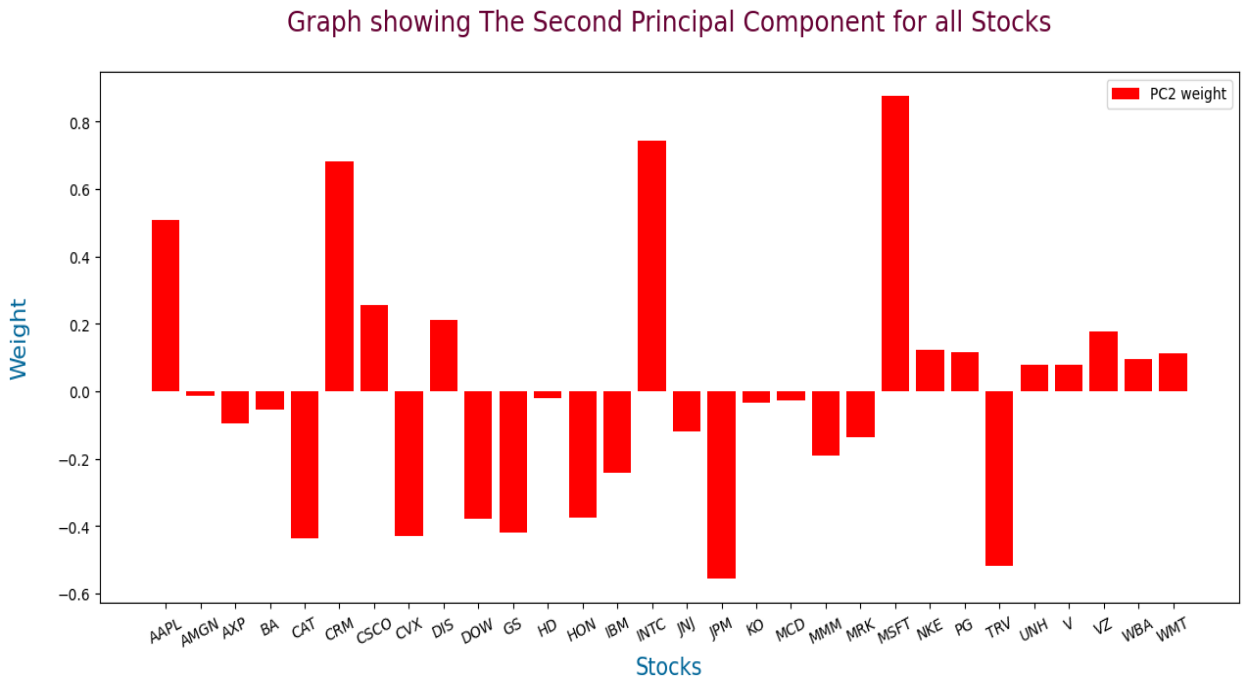
Correlation Matrix Obtained:



Then, I used the above matrix for Principal Component Analysis, by setting the output principal components to 2. This resulted in the first and second principal components.

Bar graphs obtained from PCA





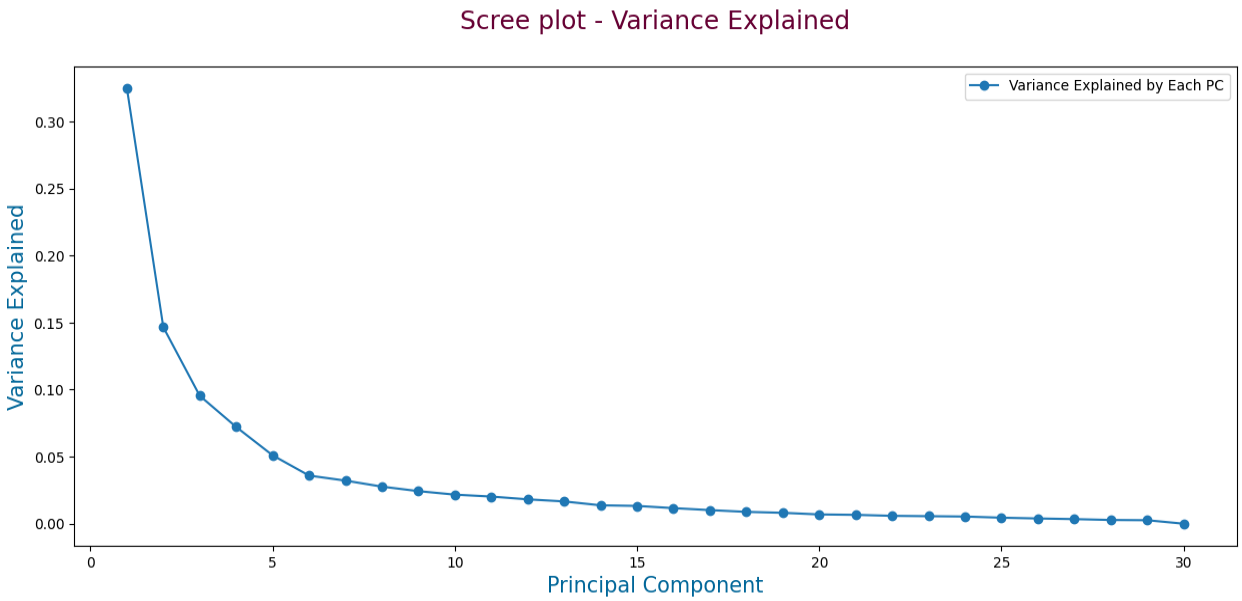
Similarity between the first or second principal component and the market

The cumulative variance represented by all principal components is equivalent to the overall variance present in the original dataset. The initial principal component captures the greatest variability in the data, while the subsequent principal components capture the maximum variance that is Orthogonal to the preceding principal components. Therefore, PC1 and PC2 have different weights as it can be inferred from the two graphs, but the overall variance is consistent with the variance in the original dataset. [5]

1.4.

To determine the percentage of total variance explained by each principal component, I used matplotlib to create a scree plot. The explained variance ration is an attribute of the PCA model fitted (explained_variance_ratio_).

Scree-plot obtained:

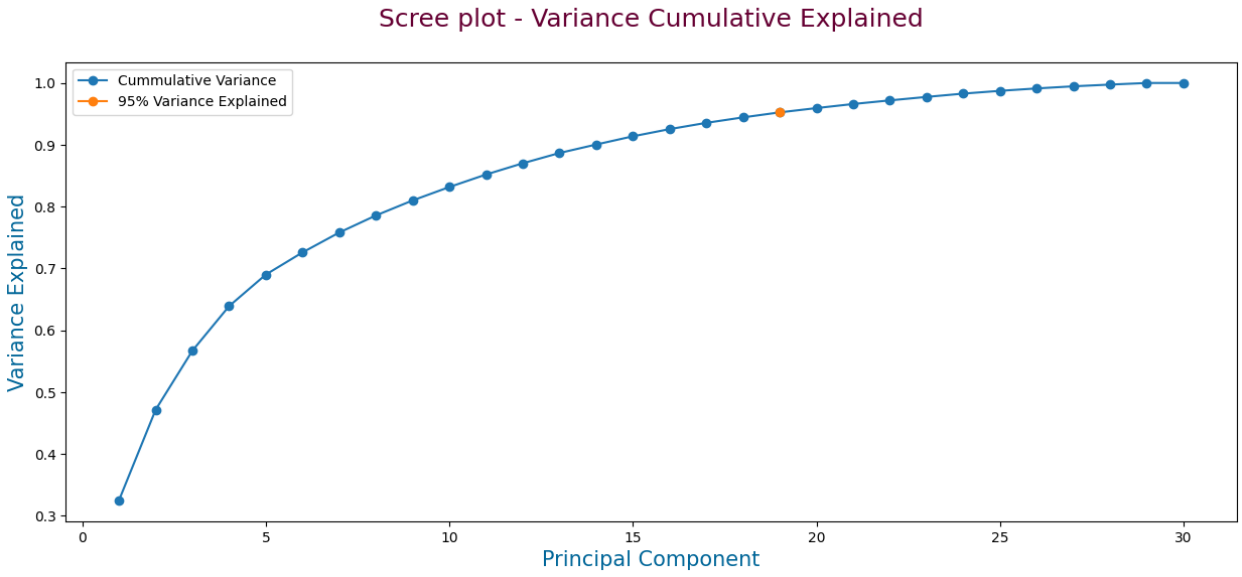


Insights:

- Variance explained by Principal Components is highest at the first Principal component.
- Variance explained by Principal Components decrease for each principal component.
- The relationship between variance and Principal Components is not constant.

I proceeded with determining the principal component the explains 95% of the variance using the cumulative variance explained by principal components.

Cumulative variance explained by Principal Components:

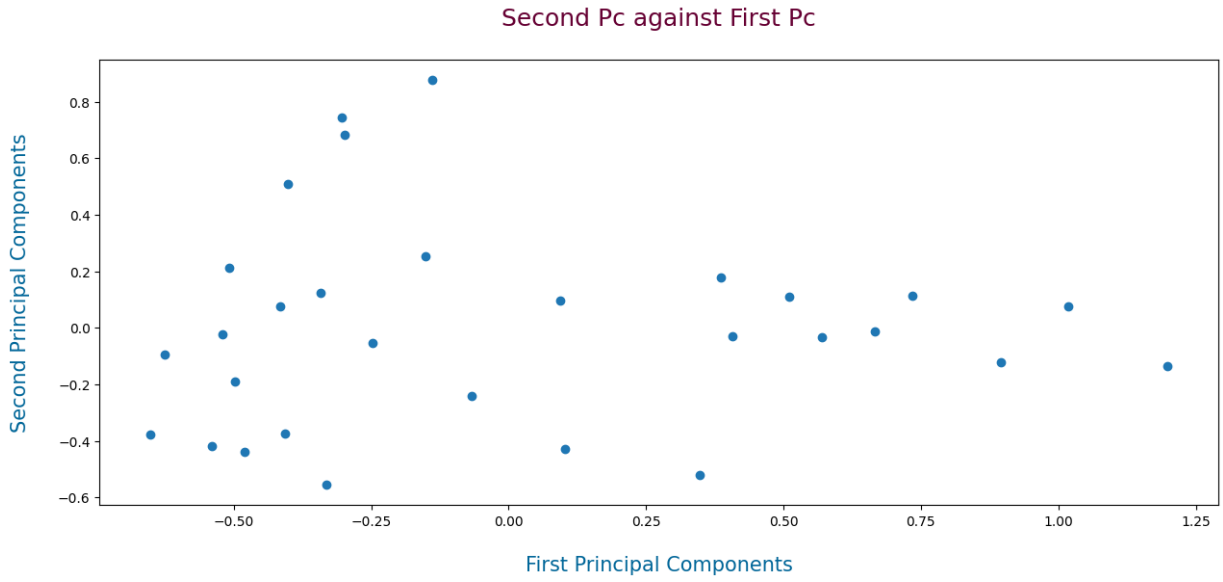


Insights:

- 19 Principal components are required to cover 95% of variance.
- All variance in the data is covered by 30 principal components.

1.5.

Using Matplotlib, I plotted the Second Principal component against the first principal component.



Insights:

- The first and second principal components are not strongly correlated.

Then, I computed the averages for PC1 and PC2 and the three stocks with the farthest Euclidean distance from the means:

Results Obtained:

```
The stocks farthest from PC1 and Euclidean Distance
[('MRK', 1.198163876236158), ('UNH', 1.0180475101917439), ('JNJ', 0.8954681008161441)]

The most distant stocks in PC1 are ['MRK', 'UNH', 'JNJ']



The stocks farthest from PC2 and Euclidean Distance
[('MSFT', 0.8756493953663066), ('INTC', 0.7428716871874939), ('CRM', 0.6819076395099254)]

The most distant stocks in PC2 are ['MSFT', 'INTC', 'CRM']
```

- Pharmaceutical industry, and Managed Health care's stocks are the farthest from the mean of PC1.

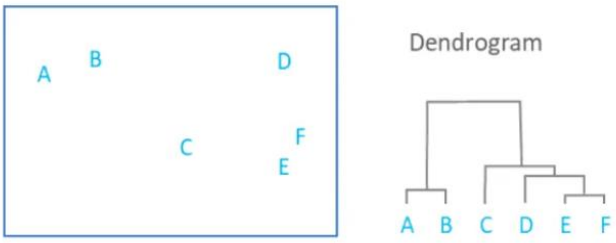
- Information Technology, and Semi-Conductor industry's stocks are the farthest from the mean of PC2.

Question 2

2. Dendrogram

2.1.1.

A dendragon is composed of clusters separated by a distance, whereby at the beginning each datapoint is considered as a cluster.
A dendrogram is constructed from a collection of dissimilar pairwise values through a hierarchical clustering process. [6]



Understanding a dendrogram mostly relies on examining the height at which two entities are connected. In the given instance, the connection height between E and F is the smallest, indicating their high similarity. Following this, the next pair of entities with the closest resemblance are A and B. The dendrogram indicates that the main distinction between clusters lies in the comparison between the cluster containing A and B and the one containing C, D, E, and F.[7]

2.2.

Starting with a collection of dissimilarity values for each pair of objects or data points in your dataset,

- Initially, each object is considered as an individual cluster leading to a total of K clusters.
- Identify the pair of objects with the smallest dissimilarity value. Merge these objects into a single cluster, creating a new level in the dendrogram make of k-1 clusters.
- Update the dissimilarity matrix to reflect the dissimilarity between the newly formed cluster and the remaining individual clusters.
- Repeat Steps 2-3by identifying the next closest pair of either individual clusters or previously formed clusters. Continue merging and updating the dissimilarity matrix until all objects are part of a single cluster.
- The dendrogram is constructed based on the order of merging. Objects that are joined earlier in the process are considered more similar, as the height of the linkage represents the dissimilarity.[8]

2.3.

Using the pdist() function from scipy.spatial.distance module, I computed the pairwise distance of the previously calculated correlation matrix of daily returns for 30 stocks.

Pairwise distances list obtained

The list of pairwise distances is of size: N(N-1)/2. Whereby, N is the size of the input matrix, which is 30 in this case.

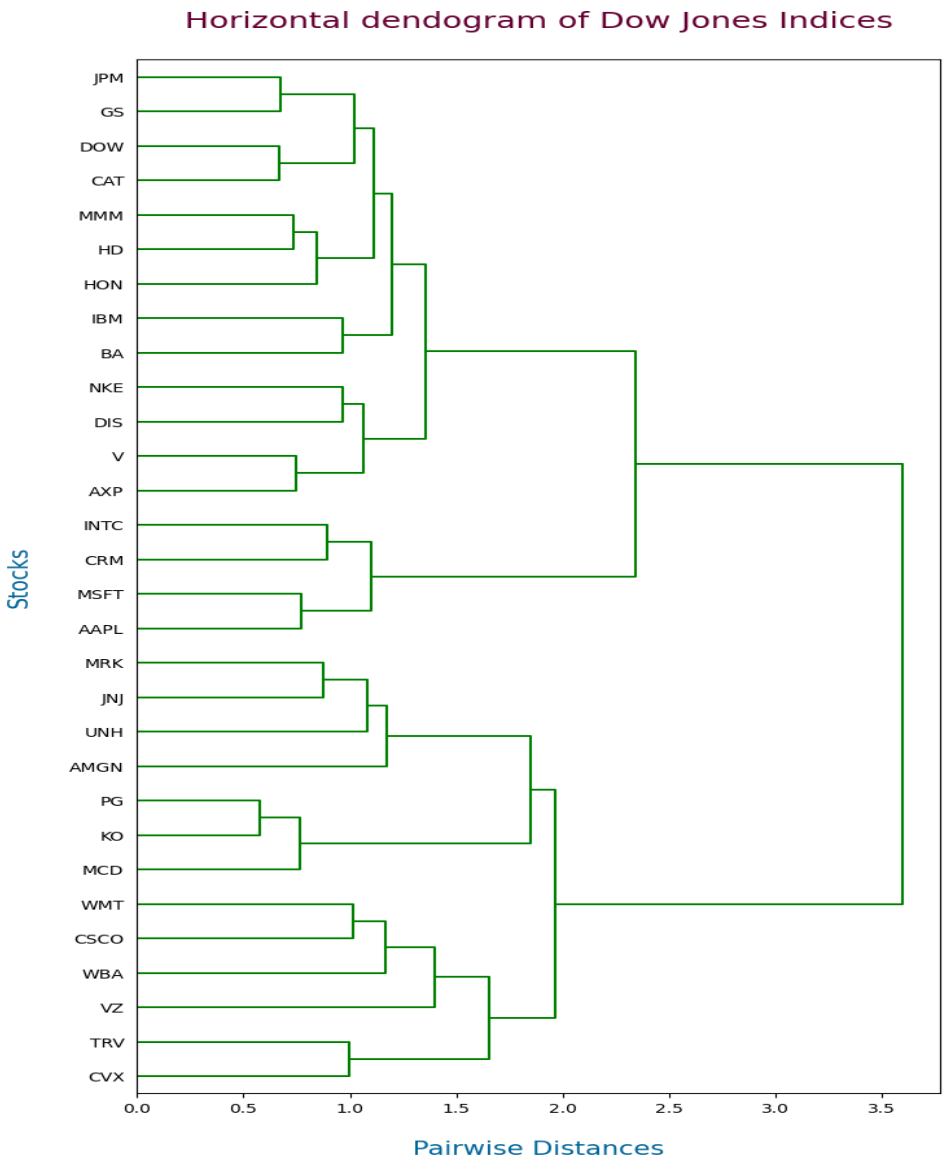
```
The pairwise distance between stock sis:
[1.59728911 0.99882102 1.12515682 1.27123316 0.95940364 1.04590408
 1.45729592 0.9975986  1.27906207 1.21586862 0.96546697 1.11370694
 1.22525307 1.07463062 1.64707596 1.26049489 1.40778217 1.24879196
 1.17130904 1.89914846 0.77275395 0.9913263  1.46446892 1.50311563
 1.75427267 0.76547678 1.52462701 1.39874827 1.43513863 1.65794127
 1.42961339 1.48877755 1.56531114 1.32655999 1.30886073 1.53570551
 1.58118911 1.55681093 1.51203389 1.53917039 1.24893571 1.51219647
 0.91938341 1.5137748  1.23493451 1.25263918 1.45297195 1.14309359
 1.55714567 1.41725157 1.16609979 1.25508003 1.24997946 1.58054655
 1.26934856 1.21567005 1.08023829 1.04814376 0.88755655 1.19164238
 1.17637139 1.3233518  0.88727167 0.92020686 0.80573396 0.9503596
 1.03838471 1.07500933 1.34909276 1.7629985  0.86483615 1.6094911
 1.46399268 0.98657321 1.94869869 1.32251181 1.06674191 1.71205376
 1.35228925 1.83835056 0.74664466 1.52714869 1.32382538 1.48577418
 1.03684417 1.18555151 1.16837041 1.18846751 0.9888102  0.93116641
 1.03769642 1.02395887 1.01452068 0.96662605 1.16653018 1.47559883
 1.08247634 1.32987358 1.18641325 0.96571766 1.70866038 1.33868555
 0.96544616 1.4175945  1.17070645 1.66094395 1.08016563 1.31255539
 1.23937098 1.29264436 1.37465074 1.15530042 1.01180738 1.07040859
 0.66751295 0.82078581 0.98536117 0.95746991 1.06179118 1.45255744
 1.66173317 0.87200365 1.51193177 1.42592376 0.8507591  1.87263969
 1.5498307  1.07932914 1.65264198 1.21794855 1.8020746  1.06735641
 1.53204509 1.29657057 1.46406422 1.16038207 1.4592942  1.00648892
 1.37798821 1.30715215 1.24948857 1.38650061 1.35823004 0.89224367
...
 1.3020654  1.35016647 1.36158049 1.4545143  1.30365734 1.75819822
 1.44409072 1.50865729 1.34052864 1.50157813 1.43809322 1.40004547
 1.35888708 1.21768831 1.14804484]
```

The small distances indicate grater similarity between daily returns of the 30 stocks and longer distances indicate higher dissimilarity between daily returns of the 30 stocks.

2.4.

Using average linkage, I constructed a horizontal dendrogram that uses Euclidean distance as metric distance between points.

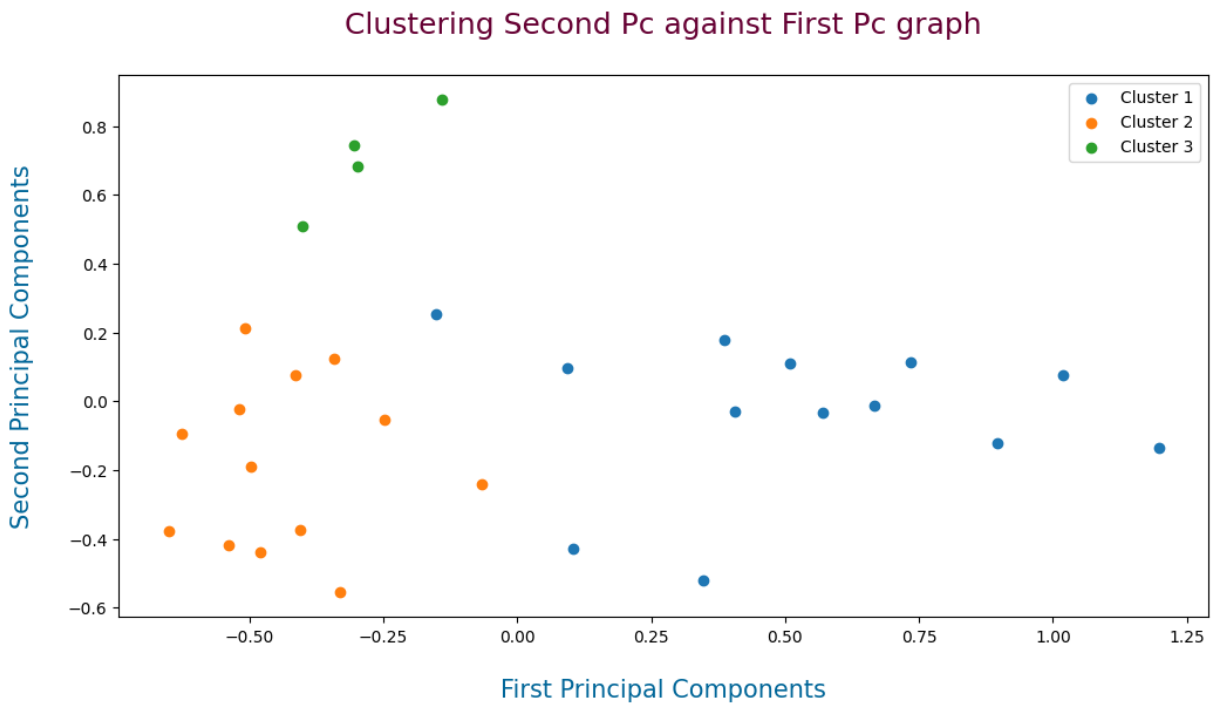
Horizontal dendrogram obtained:



2.5.1.

With this linkage, I clustered the daily returns correlation matrix into three clusters using AgglomerativeClustering function from Sklearn.cluster module in python. Then Visually segregated a graph of the second principal component against the first principal components into the computed clusters.

Graph showing the clustered principal components

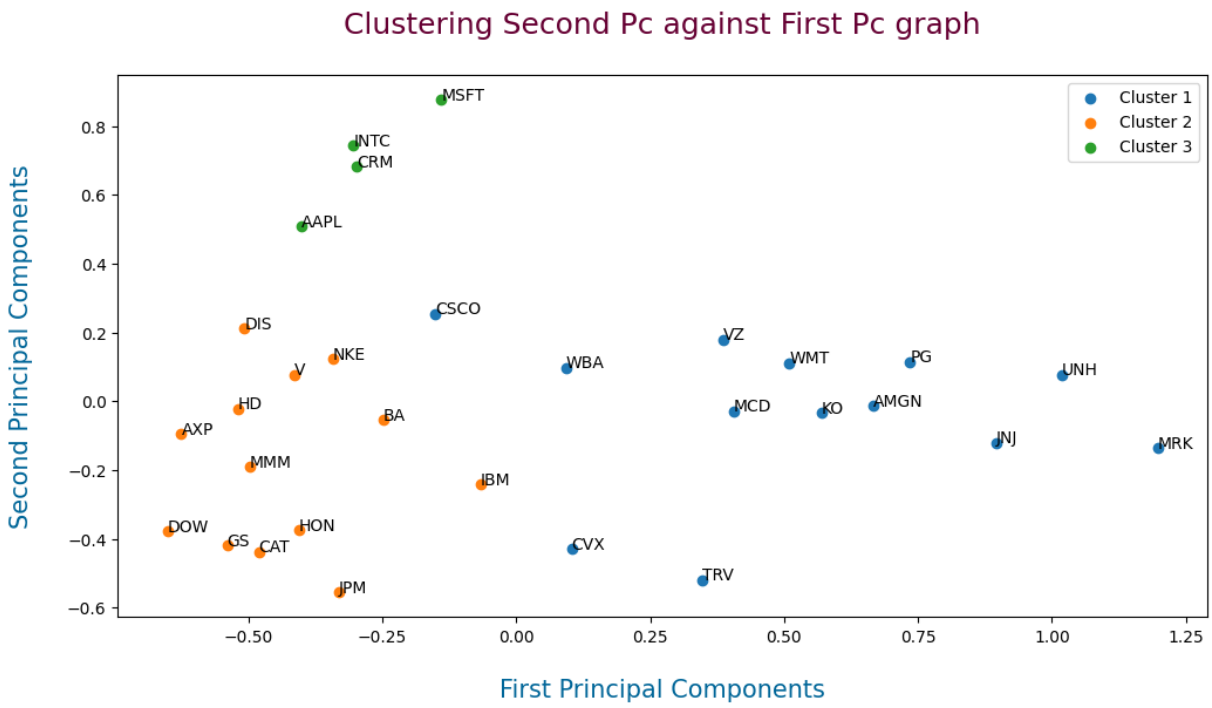


Insights

- The stocks are equal distributed between cluster 1 and cluster 2
- Cluster 3 has the lowest number of stocks, which is equal to 4.

2.5.2.

Qualitative description of each cluster



Conclusion:

- Cluster 1 containing stocks such as JNJ, MRK, and AMGN can be related to the Pharmaceutical industry.
- Cluster 2 containing stocks such as AXP, GS, V, and JPM can be related to Financial Services industry.
- Cluster 3 containing stocks such as AAPL, CRM, and MSFT can be related to the Information Technology industry.

Question 3

3. Ensembles for classification

3.1.1.

3 sources of uncertainty are:

- Observational uncertainty
- Parametrical uncertainty
- Structural uncertainty

3.3.2.

The main sources of uncertainty in machine learning are:

- **Noise**
  Noise refers to variability in observation. The variability not only influences the inputs or measurements but also extends to the outputs. For instance, an observation might carry an inaccurate class label. Consequently, even though we possess observations within the domain, it is essential to anticipate a degree of variability or unpredictability.

- **Incomplete Coverage of the Domain**
  Data points employed to train a model, originating from a specific domain, are inherently a sample and therefore incomplete. Meaning that obtaining all observations is an impractical goal; otherwise, there would be no necessity for a predictive model. This implies the perpetual existence of unobserved instances. Certain aspects of the problem domain will inevitably lack coverage, regardless of how effectively we promote our models to generalize. Our aspiration is to encompass the cases present in the training dataset along with the significant instances that are not.
- **Imperfect Model of the Problem**

  In the realm of machine learning, there will invariably be some degree of error present. This extends beyond the model itself, encompassing the entire process involved in its creation, including the selection and preparation of data, the choice of training hyperparameters, and the interpretation of model predictions. The occurrence of model error implies the potential for imperfect predictions, such as discrepancies between the predicted quantity in a regression scenario and the anticipated value, or the assignment of a class label that deviates from the expected classification.[9]

3.2. Model Averaging:

3.2.1.

Model Averaging is a technique designed to address the inherent uncertainty associated with the process of selecting a model. By taking the average across a variety of competing models, one can include model uncertainty in drawing conclusions about parameters and predictions. In many cases, model averaging is successful at improving predictive performance.

3.2.2.

Methods of implementing Model Averaging:

- **Equal Weights:** One straightforward approach to averaging models is to calculate the simple arithmetic mean of their predictions. However, this method assumes that each model equally represents the underlying data, which may not be the case in this context, potentially limiting its effectiveness in significantly reducing overall error. Despite this limitation, an advantage of this method is its quick and straightforward implementation.

- **Fit-Based Weights:** This approach involves assigning weights to models based on a performance metric. Virtually any metric with a standardized definition across component models can be employed, such as AIC or BIC for nested models or metrics like MSE and MAPE.

- **Bayesian Model Averaging:** Bayesian Model Averaging (BMA) incorporates parameter uncertainty through the prior distribution and addresses model uncertainty by deriving posterior parameter and model posteriors using Bayes' theorem. BMA offers direct model selection, as well as combined estimation and prediction.

3.3.

Ensemble methods used to the reduce the effect of uncertainty:

Basic Ensemble methods

- **Averaging method:** Primarily employed in regression tasks, this method involves constructing multiple models independently and returning the average prediction across all models. Overall, the aggregated output tends to be superior to individual predictions due to a reduction in variance.
- **Max voting:** Primarily employed in classification tasks, this method involves constructing multiple independent models and aggregating their individual outputs, referred to as 'votes.' The output is determined by the class with the highest number of votes.

[10]

Advanced ensemble methods [11]

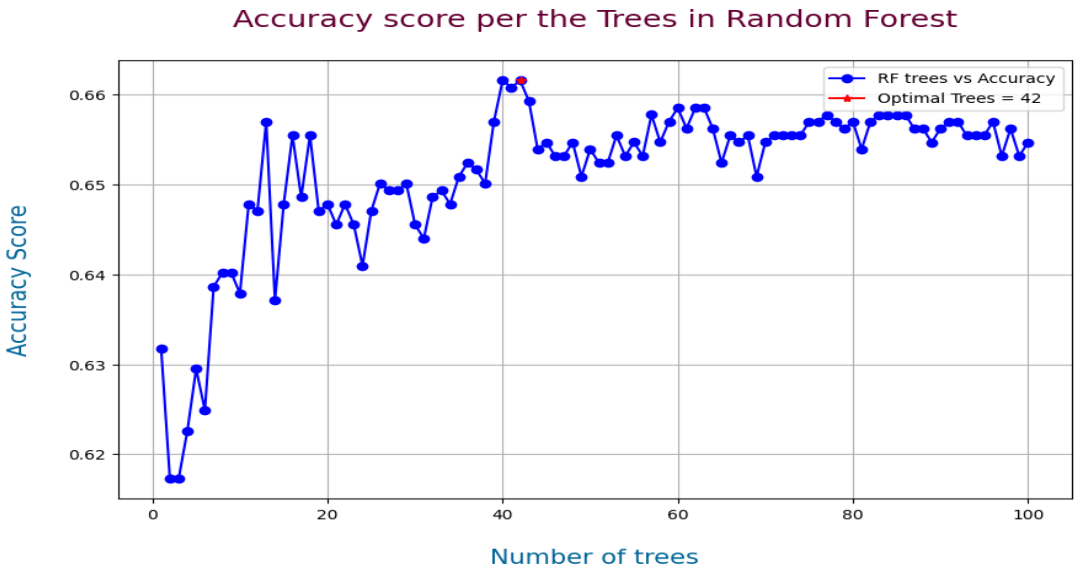
- **Stacking:** Stacking is a machine learning technique that brings together several models (used for classification or regression) through a meta-model (meta-classifier or meta-regression). First, the base models are trained using the entire dataset. Then, the meta-model is trained using the features produced by these base models as output. The base models in stacking are usually diverse, and the meta-model works to identify the most important features from the base models to achieve the highest accuracy.
- **Blending:** It resembles the previously mentioned stacking method, but instead of using the entire dataset to train the base models, a separate validation dataset is maintained to generate predictions.
- **Bagging:** Bagging, also referred to as a bootstrapping method, involves running base models on bags to ensure a representative distribution of the entire dataset. A bag is a subset of the dataset that includes replacements, maintaining the same size as the complete dataset. The ultimate result is created by aggregating the outputs of all base models.
- **Boosting:** Boosting is a sequential approach in machine learning that aims to mitigate the impact of an incorrect base model on the final output. Rather than merging base models, this method concentrates on constructing a new model that relies on the preceding one. The new model endeavors to correct the mistakes of its predecessor. Each of these models is termed

weak learners. The ultimate model, often referred to as the strong learner, is crafted by computing the weighted mean of all the weak learners.

3.4.

I fitted Random Forest Classifier models, for different number of trees ranging from 1 tree to 100 trees. After, I used cross validation to determine the accuracy of all those models, which led me to identifying the optimal number of trees to fit the Random Forest Classifier model.

Results Obtained:



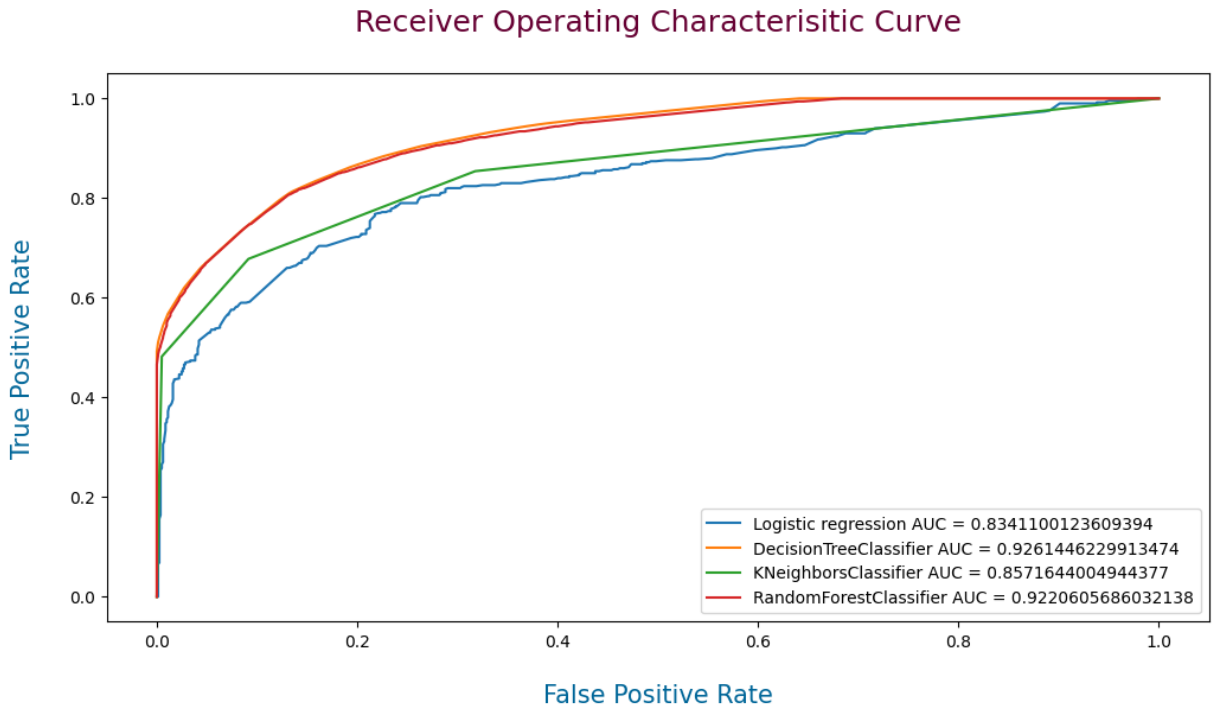
Inference:

The model with the highest accuracy uses 42 trees; therefore, the optimal number of trees is 42.

3.5.

To determine clearly how well the models such as Logistic Regression, Decision Tree Classifier, K Nearest Neighbors, and Random Forest Classifier fits the titanic dataset, I used Receiver Operating Characteristics Analysis (ROC). This analysis determines the sensitivity and specificity of the models depending on the **True Positive** and **False Positive rates** curve of each model. To quantify the results of ROC analysis, I used Area Under the Curve(AUC), which tells how much area is under the ROC curve of each model. [12]

Results Obtained:



Conclusion:

From the AUC values of each model, the best model for classifying survival on the Titanic is **Decision Tree Classifier model**.

Question 4

4. Ensembles for regression

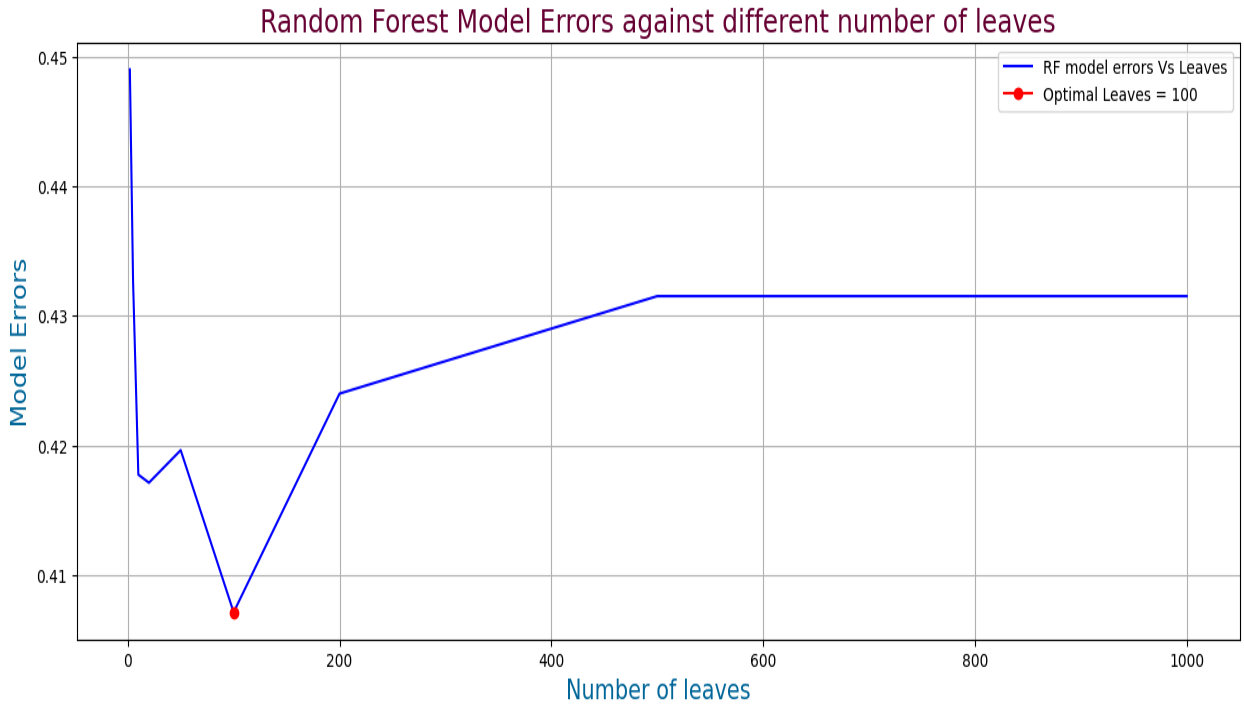
4.1.

The Random Forest regression algorithm leverages the principle of the 'wisdom of the crowds' by employing multiple diverse regression decision trees and aggregating their outputs through a voting mechanism. It achieves this by averaging predictions across numerous deep decision trees, each trained on different subsets of the same training set. The primary objective is to mitigate variance, even though there is a slight increase in bias and a trade-off in interpretability. In essence, the Random Forest model is an ensemble technique designed to enhance the overall performance of the final model. Random Forest operates through a combination of multiple decision trees, employing a technique known as Bootstrap and Aggregation, or bagging. This methodology involves random sampling of both rows and features from the dataset, creating distinct sample datasets for each individual model. This process, known as Bootstrap, contributes to the diversity of the decision trees within the Random Forest ensemble, enhancing its predictive power. [13] [14] [15]

4.2.

I use different number of leaves in the range [2, 5, 10, 20, 50, 100, 200, 500, 1000] to fit Random Forest models and determined the optimal number of leaves as the number of leaves for the model with the lowest errors.

Results obtained

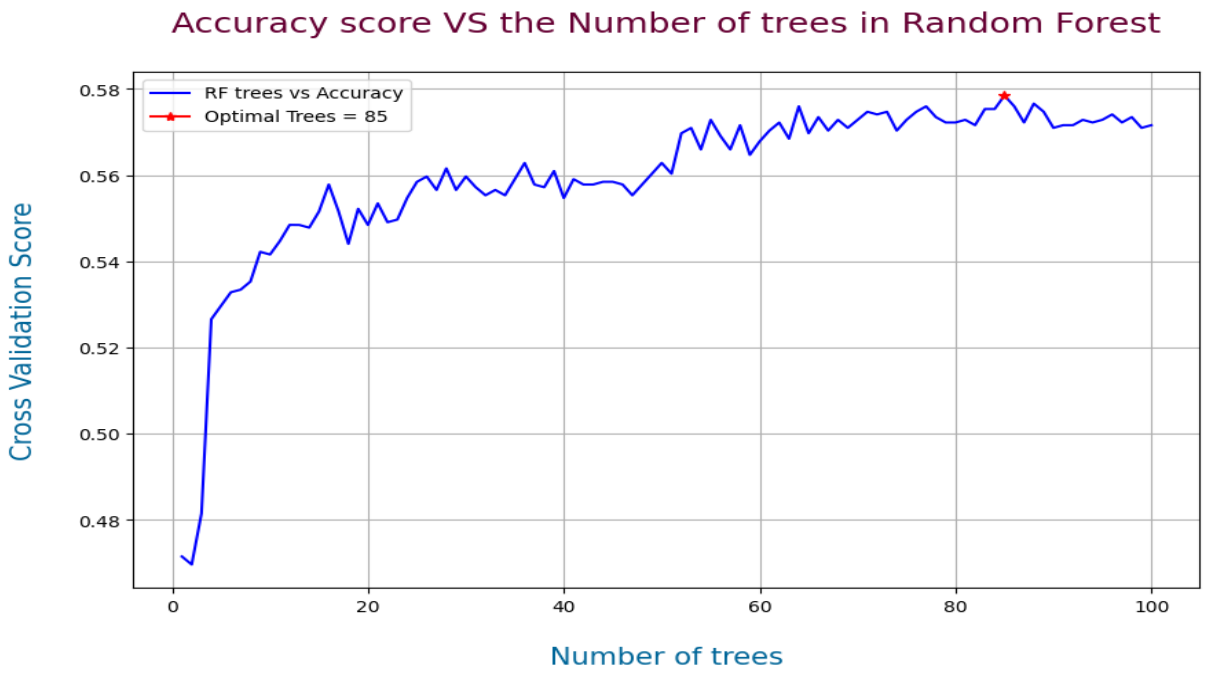


Conclusion:

The optimal number of leaves that lead to lowest errors of the model is 100 leaves.

4.3.

I fitted a Random Forest model using a wide range of trees ranging from 1 to 100 and determined the accuracy of all the 100 models using cross validation. From the cross-validation score, the optimal number of trees to fit the red wine dataset is displayed in the graph below:



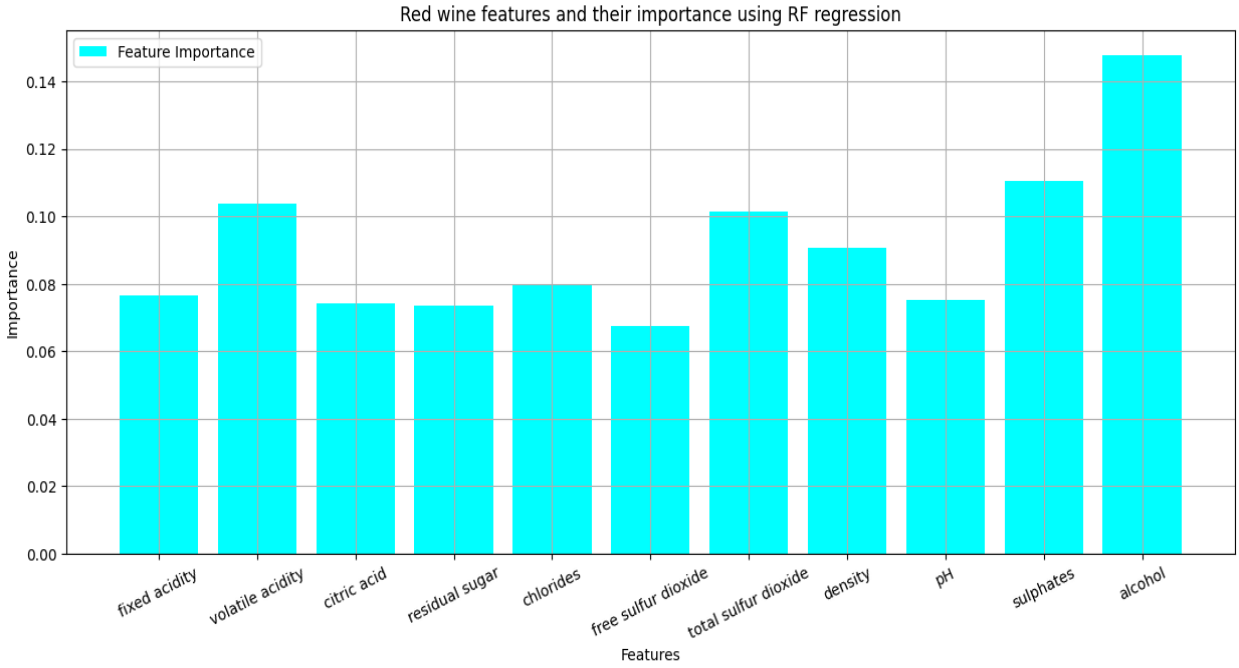
Conclusion :

The optimal number of trees to fit a Random Forest model with maximum accuracy is 85 trees.

4.4.

After fitting a Random Forest model with the red wine dataset, the fitted model's attribute called "feature_importances_" provided me the order of feature importance. Then, I plotted this distribution of feature importance using a bar graph.

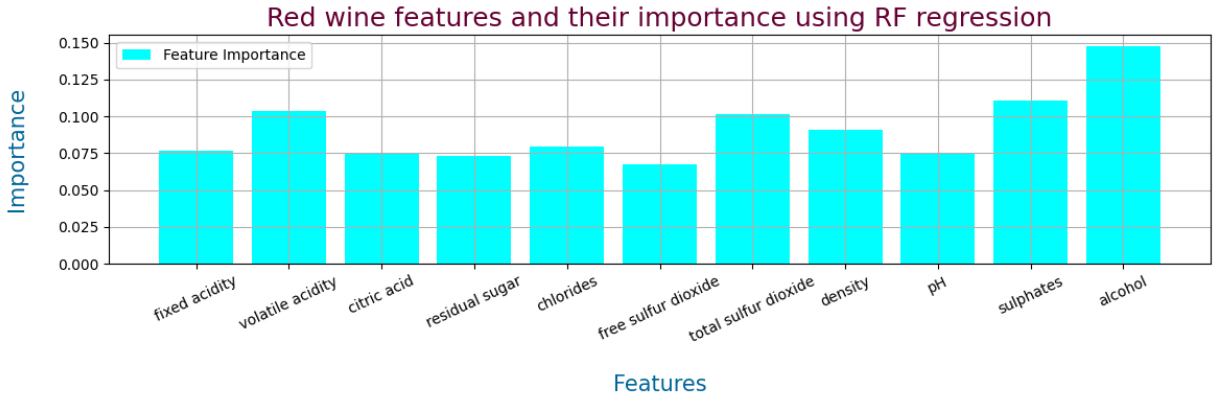
Results Obtained:

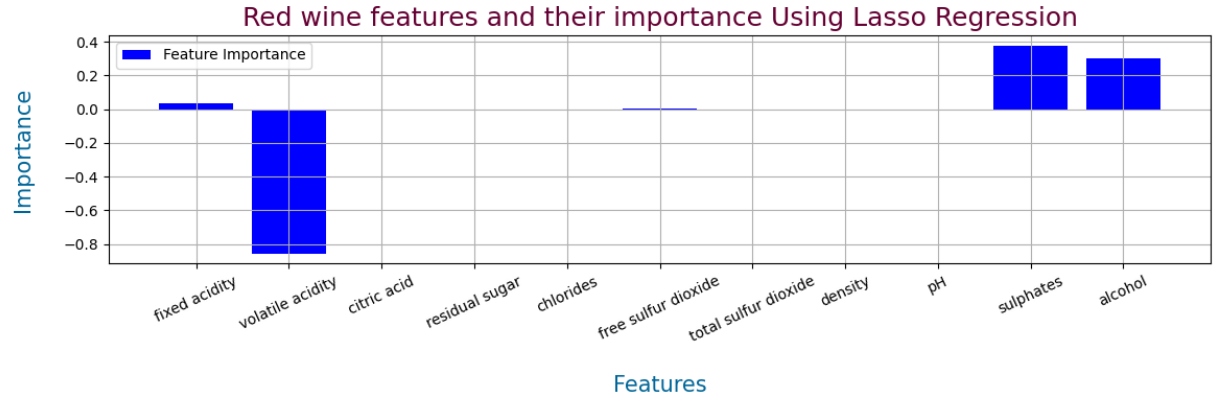


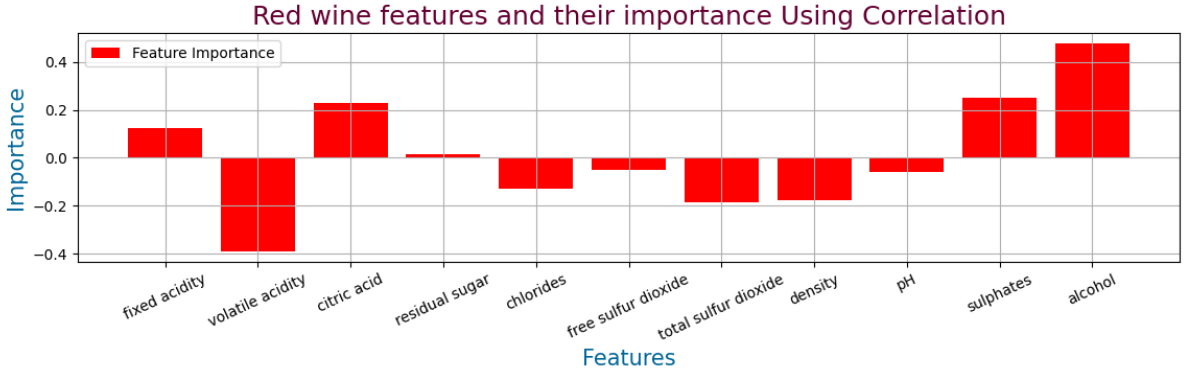
Interpretation:

- Alcohol is the most important feature for the Random Forest Model, followed by Sulphates.
- Free sulfur dioxide is the least important feature for the Random Forest Model.

Comparison with the results from assignment 6







Interpretation:

- Both correlation and Random Forest model use alcohol as the most important feature.
- Different from the other methods of feature importance, the feature importance metric of Random Forest model shows that all features have a positive importance.
- Lasso Regression has the lowest number of important features.

4.5.

By splitting the red wine dataset into train and test set, I was able to fit three models using the train dataset namely the Random Forest model, the K Nearest Neighbors model and the Linear Regression model. Then I used the test data set to examine the performance of the three models to determine the best model. Specifically, I computed the Mean Squared Error of each of the three models.

Results obtained

```
The MSE of the KNeighborsClassifier model is 0.925
The MSE of the Linear regression model is 0.390025143963954
The MSE of the RF model is 0.396875
```

Conclusion:

The K- Nearest Neighbors model would perform best on the red wine dataset.

[1] ‘Principal Component Analysis(PCA)’, GeeksforGeeks. Accessed: Nov. 27, 2023. [Online]. Available: https://www.geeksforgeeks.org/principal-component-analysis-pca/
[2] ‘CMUdiaml11.pdf: Data, Inference, and Applied Machine Learning - DIAML’. Accessed: Nov. 27, 2023. [Online]. Available: https://canvas.cmu.edu/courses/36029/files/10410047?module_item_id=5647659
[3] ‘Mathematical Approach to PCA’, GeeksforGeeks. Accessed: Nov. 27, 2023. [Online]. Available: https://www.geeksforgeeks.org/mathematical-approach-to-pca/
[4] A. Dubey, ‘The Mathematics Behind Principal Component Analysis’, Medium. Accessed: Nov. 27, 2023. [Online]. Available: https://towardsdatascience.com/the-mathematics-behind-principal-component-analysis-fff2d7f4b643
[5] ‘Principal Component Analysis(PCA) - GeeksforGeeks’. Accessed: Nov. 30, 2023. [Online]. Available: https://www.geeksforgeeks.org/principal-component-analysis-pca/
[6] ‘CMUdiaml11.pdf: Data, Inference, and Applied Machine Learning - DIAML’. Accessed: Nov. 28, 2023. [Online]. Available: https://canvas.cmu.edu/courses/36029/files/10410047?module_item_id=5647659
[7] T. Bock, ‘What is a Dendrogram?’, Displayr. Accessed: Nov. 28, 2023. [Online]. Available: https://www.displayr.com/what-is-dendrogram/
[8] ‘SciPy - Cluster Hierarchy Dendrogram’, GeeksforGeeks. Accessed: Nov. 28, 2023. [Online]. Available: https://www.geeksforgeeks.org/scipy-cluster-hierarchy-dendrogram/
[9] J. Brownlee, ‘A Gentle Introduction to Uncertainty in Machine Learning’, MachineLearningMastery.com. Accessed: Nov. 29, 2023. [Online]. Available: https://machinelearningmastery.com/uncertainty-in-machine-learning/
[10] M. Mahoney, ‘Model averaging methods: how and why to build ensemble models’, Medium. Accessed: Nov. 29, 2023. [Online]. Available: https://towardsdatascience.com/model-averaging-methods-how-and-why-to-build-ensemble-models-b4e5d97cbb4
[11] ‘Ensemble Methods in Python’, GeeksforGeeks. Accessed: Nov. 29, 2023. [Online]. Available: https://www.geeksforgeeks.org/ensemble-methods-in-python/
[12] Zach, ‘How to Plot a ROC Curve in Python (Step-by-Step)’, Statology. Accessed: Dec. 02, 2023. [Online]. Available: https://www.statology.org/plot-roc-curve-python/
[13] ‘The Ultimate Guide to Random Forest Regression’. Accessed: Nov. 30, 2023. [Online]. Available: https://www.keboola.com/blog/random-forest-regression
[14] ‘Random Forest Regression in Python’, GeeksforGeeks. Accessed: Nov. 30, 2023. [Online]. Available: https://www.geeksforgeeks.org/random-forest-regression-in-python/
[15] ‘CMUdiaml12.pdf: Data, Inference, and Applied Machine Learning - DIAML’. Accessed: Nov. 30, 2023. [Online]. Available: https://canvas.cmu.edu/courses/36029/files/10426822?module_item_id=5649624