Carnegie Mellon University Africa

COURSE 18-785: DATA, INFERENCE & APPLIED MACHINE LEARNING

ASSIGNMENT 6

Nchofon Tagha Ghogomu

ntaghagh

November 11, 2024

LIBRARIES:

The following libraries were used:

- Pandas[1]
- Numpy[2]
- Pyplot from Matplotlib [3]
- Stats from Scipy
- Seaborn
- Linear model from Sklearn
- Linear Regression from Sklearn Linear model
- Logistic Regression from Sklearn Linear model

- import pandas as pd # Pandas
- import numpy as np # Numpy
- from scipy import stats # Scipy - for statistics
- import matplotlib.pyplot as plt # Matplotlib - for ploting
- import seaborn as sns # Seaborn - for ploting
- from sklearn import linear_model
- from sklearn.linear_model import LinearRegression
- from sklearn.linear_model import LinearRegression
- from sklearn.linear_model import LogisticRegression
- from sklearn.metrics import confusion_matrix, classification_report
- from sklearn.model_selection import train_test_split
- import statsmodels.api as sm
- from tabulate import tabulate
- from sklearn.tree import DecisionTreeClassifier
- from sklearn.metrics import classification_report, accuracy_score
- from sklearn.model_selection import cross_val_score
- from sklearn.preprocessing import StandardScaler
- from sklearn.neighbors import KNeighborsClassifier
- from sklearn.linear_model import LassoCV
- from sklearn.neighbors import KNeighborsRegressor

Programming Language:

- Python

INTRODUCTION

This is a collection of exercises to help us familiarise ourselves with machine learning concepts and models like principal component analysis, uncertainties, random forest, ensembles, to name a few. We will be applying these concepts and models to real world scenarios and for the models, we will compare their accuracies with that of previous models. At the end of this exercise, we should have garnered the necessary skills that prepares us for Kaggle-like competitions.

SOLUTIONS

QUESTION 1: PCA.

1. PCA and Application:

It is commonplace today to deal with high dimensional dataset with increasing number of features. In the concept of machine learning, this will mean threat to the accuracy of the model, and most often, overfitting. Principal component analysis (PCA) is employed today when dealing with high dimensional datasets with large number of features.

The task of PCA is to reduce the dimensionality of some high-dimensional data points by linearly projecting them onto a lower-dimensional space in such a way that the reconstruction error made by this projection is minimal[4]. It can also be defined as a versatile statistical method for reducing a cases-by-variables data table to its essential features, called principal components[5]. In essence, it is a process that summarises high dimensional data in a way that captures the greatest variance or information found in the data points. The first principal component account for the greatest percentage of variation in the data set followed by the second principal component and so on.

Two typical applications of PCA in Machine Learning include:

i) Quantitative Finance[6]: In a financial setting where it become imperative to analyse a huge portfolio of stocks, principal component analysis can come in handy to reduce the complexity of analysis. In a situation where a model will be trained on this data, just the significant stocks will be kept.

ii) Image compression: During image comparison, we try to take out features that will not greatly distort the overall view of the image. PCA can help us achieve this.

Considering PCA to transform a set of explanatory variables is essential for the following reasons:

i) Dimensionality reduction: PCA helps us extract the most useful features in a data set: feature extraction[7].

ii) Noise Filtering: If we focus on the principal components of our given datasets, we eliminate redundant information or noise[8].

2. Mathematical Equations:

To perform PCA, we go through a succession of steps[9,10]. The input metrics to this series of equation is an N × M matrix, where N is the number of entries and M the number of features or dimension of the input matrix

i) *Computation of covariance matrix:* We compute the covariance matrix S, and M×M matrix, the covariance between every feature in the input matrix:

$$S = (\sigma j, k) j, k = 1, \ldots, d, where\ \sigma j, k = cov(Xj, Xk).$$

ii) *Eigen Value Decomposition:* We perform the decomposition

$$S = PDP^T \text{ where,}$$

$P = (v1, \ldots, vd)$ is an orthogonal matrix $(PP^T = I)$

$D = Diag(\lambda 1, \ldots, \lambda d), with\ \lambda 1 \geq \lambda 2 \geq \ldots \geq \lambda d$

At this point, we can choose values of lambda that explain the greatest variance: the lambda values with the greatest weights.

iii) *Computation of the principal component:* The principal component analysis Y, of a point X can be given as:

$$Y_i = X_i V \in Rk, i = 1, \ldots, n$$

3. Correlation matrix and stock weights on the first 2 principal components:

To construct the covariance matrix:

i) Using Yahoo finance and tickers for the 30 stocks, the 'Adj Close' stock price for 2023 is extracted.

ii) Daily returns is calculated using pct.change.

iii) These values are scaled using standard scaler.

iv) Using the scaled stock returns, the correlation matrix was calculated

v) PCA is performed using the correlation matrix.

The following correlation matrix was obtained:

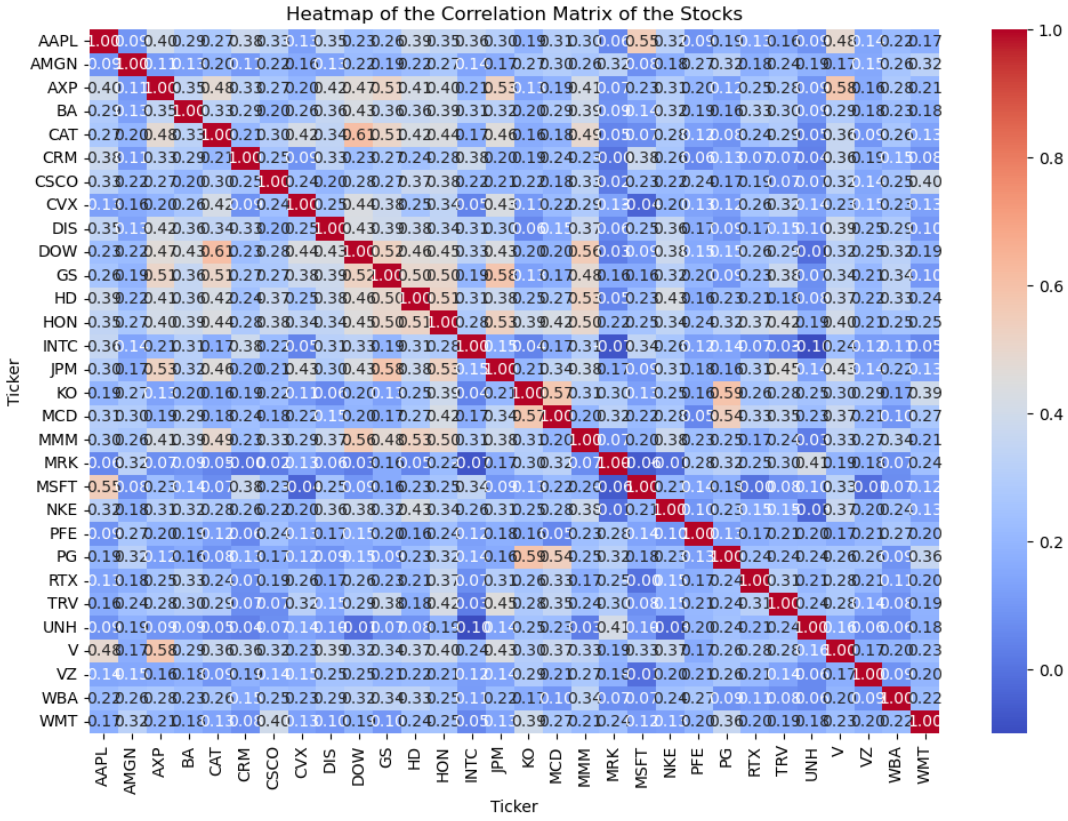
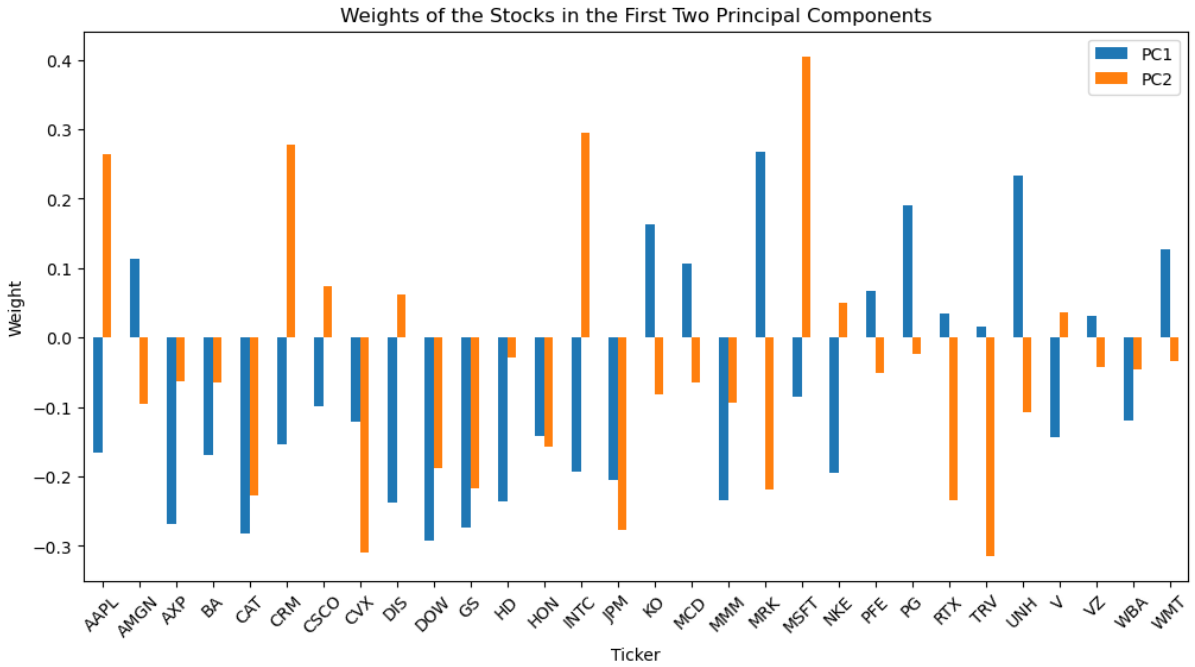


Figure 1 Heat Map of Correlation Matrix

Using the stock returns and PCA, the stock weights on first 2 principal component yield the following bar graph plot.



We observe that the weight of each stock is not the same on both principal components. The top 3 stocks whose weights were close (but not similar) in both principal components were Honeywell International Inc. (HON), Goldman Sachs (GS), and Caterpillar Inc. (CAT). On a large scale, the stocks did not portray equal weight on both principal components.

4. Variance explained by PCs and scree plots

To compute the explained variance by the principal components:

i) The explained variance ration method of PCA is employed.
ii) The variance is plot and the cumulative variance is computed.
iii) Now the number of components that cumulate to a wight of 95% is calculated.

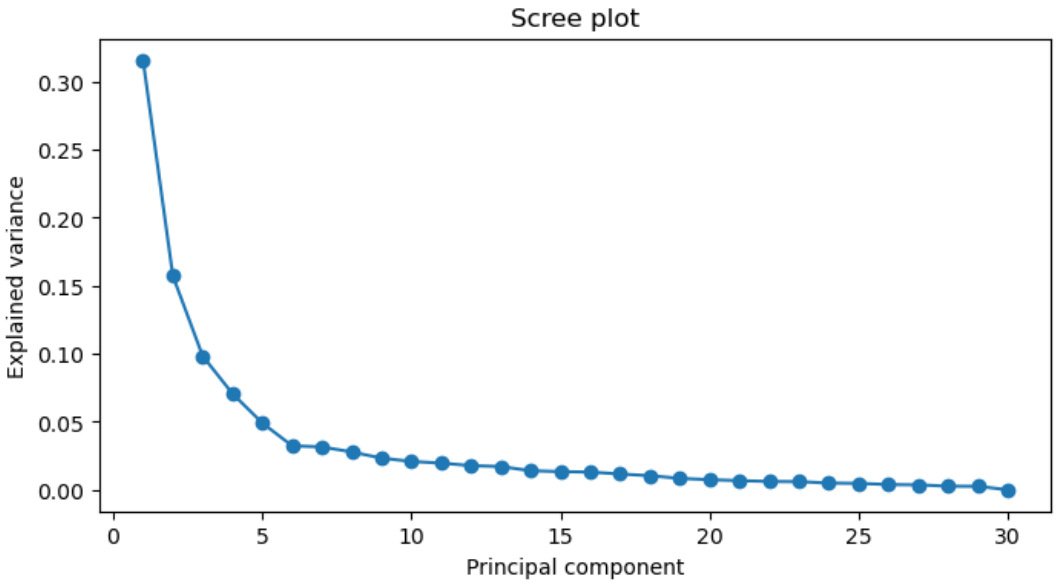
The result of the following result was obtained:

- Weight of principal components:

PC1	PC2	PC3	PC4	PC5	PC6	PC7	PC8	PC9	PC10	PC11	PC12	PC13	PC14	PC15
31.58%	15.71%	9.83%	7.06%	4.91%	3.23%	3.15%	2.78%	2.32%	2.08%	1.96%	1.77%	1.71%	1.40%	1.34%

PC16	PC17	PC18	PC19	PC20	PC21	PC22	PC23	PC24	PC25	PC26	PC27	PC28	PC29	PC30
1.30%	1.16%	1.04%	0.84%	0.74%	0.67%	0.61%	0.60%	0.49%	0.47%	0.38%	0.36%	0.26%	0.25%	0.00%

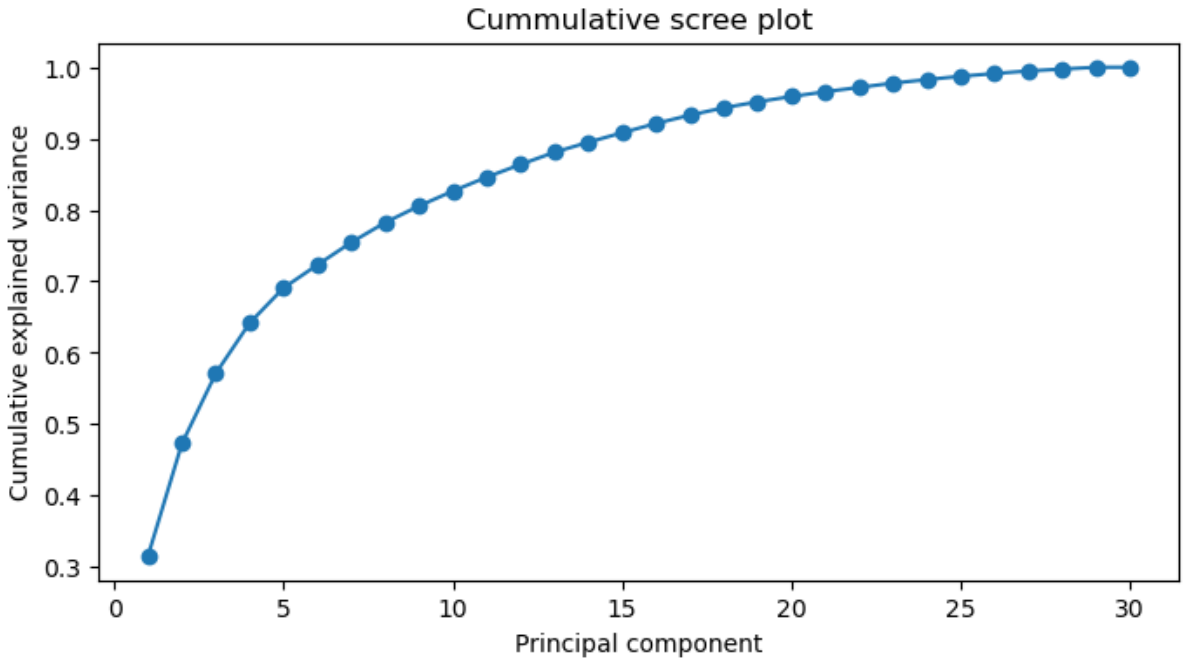
- Scree plot:



- Number of principal components that explain 95% of variance:

```
Number of principal components that explain at least 95% of the variance: 19
```

The cumulative scree plot was obtained as seen bellow.



5. Investigation of scatter plot of the first two principal components

To arrive at the results:

i) A scatter plot of the stocks against the first two principal components is done
ii) Get the average of stock weight on the first two principal components.
iii) Using Euclidian distance, we substance stock distance from the average.

The following scatter plot was achieved:

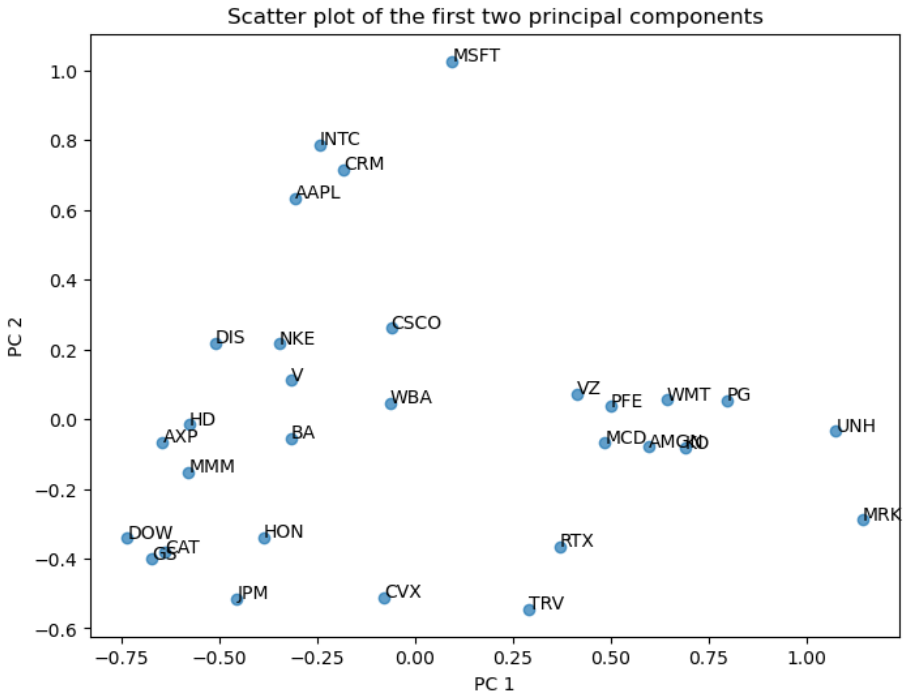


Figure 2 Scatter plot against first two principal components

The most distance stocks were obtained to be:

```
Three most distant stocks:
Ticker
MSFT    0.454151
MRK     0.383247
INTC    0.363497
```

Some of the potential reasons to why these stocks are furthest apart could be:

- Activity sector: They have clearly distinct areas of activities: MSFT is into software, MRT (Merck & Co) is into the health sector and INTC (Intel Inc.) is into hardware manufacturing. The diversity of their area of activity will certainly make them portray divers characteristics
- Market: These companies all have different market audience which will contribute to the difference of their stream of revenue. This also potentially contributes to their distances.

These stocks can be considered unusual as they sit high up in the market chain of their respective activity sector and because they have a focussed area of interest.

QUESTION 2: Dendrogram

1. Definition, Components, Construction and Interpretation:

Dendrograms are tree-like hierarchical representations of relationships between features based on the distances between these features[11]. Dendrograms are usually generated as an output to visualise hierarchical clustering. Dendrograms are made up of branches and leaves.

- A leaf is basically a feature of a simple observation.
- Identical leaves (distance wise), fuse into branches, moving towards the root of the dendrogram.
- Likewise, bigger branches are form as identical branches fuse together.
- The corresponding distance at which the leaves fuse matches the distance between the observations.

It is worth nothing that features that fuse at a lower distance are more similar to each other than those that fuse further away. The difference in the point where leaves and branches fuse portray different relationships between the leaves and branches. Therefore, we construct a dendrogram starting with pairwise comparison of features by evaluating the distance between them. The closes features are merges in increasing magnitude of distance.

To read a dendrogram, we begin from the leaves to the root. Features or observations with close characteristics merge closet to the leaves. Therefor, features that merge further away from away from the leaves are not as closely related as those who merged earlier. Also, dendrograms gives us an idea of how to allocate clusters. The hierarchical analysis can serve as a guide in understanding how the features can be clustered together.

2. Constructing dendrograms from pairwise dissimilarity values.

A dendrogram is built from the bottom up. Similarities are grouped up the leaves to the root contain the dataset. This follows the hierarchical clustering algorithm, as follows:

i) Begin with all features representing its own cluster.
ii) Check all possible pairwise similarity between clusters, know as inter-cluster similarities.
iii) Fuse the most similar cluster pair. The height of the dendrogram at which this fusion occurs explains the dissimilarity between the clusters.
iv) Treating the grouped cluster as a single cluster, we repeat the procedure for all remaining cluster. We keep grouping the most similar clusters.

Linkage describes the pairwise dissimilarities between clusters. There are multiple approaches to calculating linkage which include including[12]:

- Complete linkage: greatest distance between two points in clusters
- Single linkage: least distance between two points in clusters
- Average linkage: The arithmetic mean of distances every possible pair of point between clusters.
- Centroid linkage: Distances between the centres of both clusters.

They all differ in the way the distance between the clusters is calculated. At this point, we can construct our dendrogram starting with the leaves. As we move to the root, the pairs are fused at a distance matching their level of dissimilarities.

3. Pairwise distances between stocks and interpretation.

To compute pair wise distances between stokes using the earlier calculated covariance matrix:

i) The Euclidian matric was employed in calculating distance.
ii) Scipy's spatial distance class function was called with the covariance matrix and the Euclidian distant matric as input.
iii) The distance between cluster pair was obtained, plotted and printed.

The average linkage is used in this exercise and is given by the formular:

$$L(R,S) = \frac{1}{n_r + n_s}\sum_{i=1}^{n_s}\sum_{j=1}^{n_r} D(r_i, s_j),\ r_i \in R, s_i \in S$$

Where:

- R and S are two separate clusters.
- $n_r\ and\ n_s$ are the number of elements in cluster R and S respectively,
- $r_i\ and\ s_j$ are points in cluster r and s respectively
- $D(r_i, s_j)$ is the Euclidian distance between the points in cluster R and S

Small distances portray observations, features, or clusters that present similar characteristics, or are close together. Clusters fused further portray fewer similarities between them. The top 5 similar pairs were obtained to be:

```
Stock pair: KO - PG, Distance   : 0.6332
Stock pair: CAT - DOW, Distance   : 0.6487
Stock pair: GS - JPM, Distance   : 0.7054
Stock pair: HD - MMM, Distance   : 0.7149
Stock pair: MCD - Cluster 0, Distance   : 0.7574
```

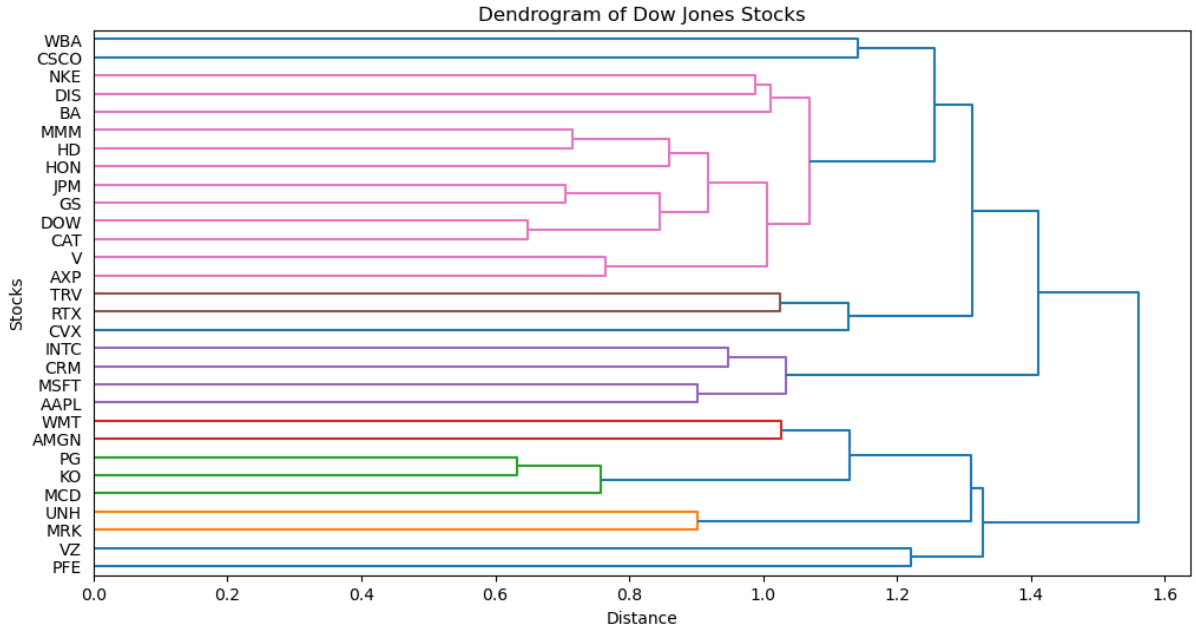
We observe that Procter & Gamble (PG) and Coca-Cola (KO) are into the production of stapple consumer goods. As such, they certainly will present much similar behaviours in the stock market.

4. Constructing a dendrogram:

To construct the dendrogram:

i) Construct the linkage matrix using linkage from Scipy hierarchical clustering[13]. The covariance matrix and average linkage is used.
ii) With dendrogram from Scipy, the dendrogram matrix is plotted against stock names.

We obtain the following.



5. Exploring clusters from the dendrogram.

In this part of the exercise, the stocks were split into 6 clusters and observed. Cluster was used from Scipy hierarchical cluster to select stocks found in this cluster. The result is shown bellow:

```
Cluster 1: PFE, VZ
Cluster 2: MRK, UNH
Cluster 3: AMGN, KO, MCD, PG, WMT
Cluster 4: AAPL, CRM, INTC, MSFT
Cluster 5: CVX, RTX, TRV
Cluster 6: AXP, BA, CAT, CSCO, DIS, DOW, GS, HD, HON, JPM, MMM, NKE, V, WBA
```

Let us examine clusters with two or more stocks:

i) Cluster 1 – PFE and VZ

Though into different activity sectors, PFE (Pfizer) and VZ (Verizon) are leading companies in pharmaceutical and telecommunication respectively.

ii) Cluster 2 – MRK, and UNH:

We can infare that Merck & Co. (MRK) and UnitedHealth Group (UNH) are of the health industry. For over 130, MRK has been into the development of important medicines and vaccines[14]. UnitedHealth Group is a health care and well-being company with a mission to help people live healthier lives and help make the health system work better for everyone[15]. They are both leading in their domains and have strong international presence. This accounts for their similarities. They can be classified under healthcare.

iii) Cluster 3 – AMGN, KO, MCD, PG, and WMT:

Coca-Cola (KO), McDonald's (MCD), and Procter & Gamble (PG) and Walmart (WMT) are companies into daily consumer goods production. Their product include food, beverages, sanitary, home and selfcare products, to name a few. They can be classified as companies that provide necessary basic consumer goods: staple companies.

iv) Cluster 4 - AAPL, CRM, INTC, MSFT:

Apple (AAPL), Salesforce (CRM), Intel (INTC), and Microsoft (MSFT) are clearly leading tech giants. They are into technology and the production of electronics. This certainly is the reason for the similarities. This therefor can be closely related to the technological sector.

v) Cluster 5 - CVX, RTX, and TRV:

Chevron (CVX) is into energy exploitation and production while Raytheon Technologies (RTX) is into the Aerospace and Defence Sector, and Travelers (TRV) is an insurance company into high-risk insurance. These are certainly companies related to high energy production or usage. Closely related to the energy sector.

vi) Cluster 6 - AXP, BA, CAT, CSCO, DIS, DOW, GS, HD, HON, JPM, MMM, NKE, V, WBA:

The elements of this cluster have some similarities but are companies with diverse areas of interests. We have for example, financial and investment service providers like V, JPM, GS, and AXP. We also have conglomerate industries and consumer good companies. The similarities between them. IT is closely related to the financial sector.

It can be concluded that companies in the same clusters clearly have a range of similarities. These similarities can fade when many companies are in each cluster. The greater the number of clusters, the clearer the similarities are. Likewise, the fewer the clusters, the broader the similarities capture become. The clustering can be seen in the plot bellow.

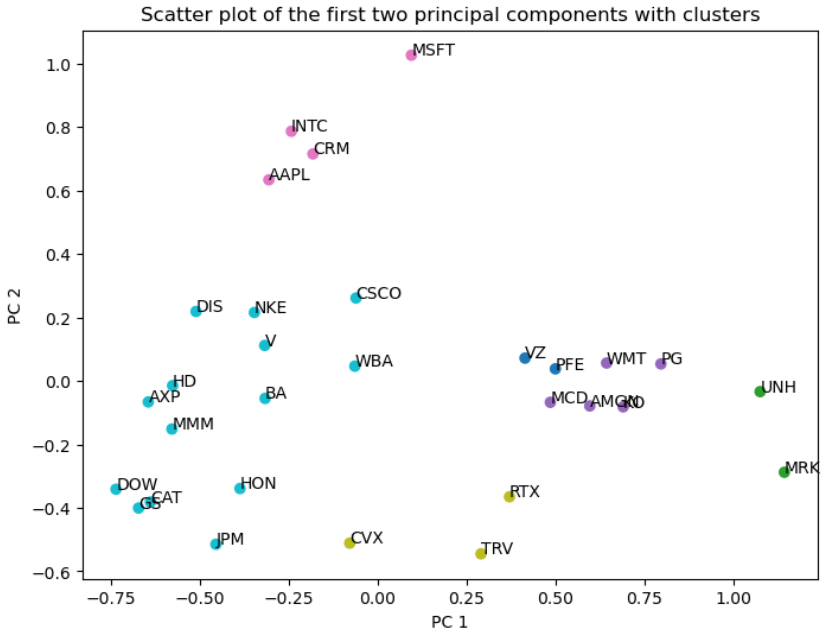


Figure 3 Clusters of the Dow Jones stocks

QUESTION 3: Ensembles for classification

1. Uncertainty:

As inferred from our class notes[16], three sources of uncertainties, with there implications include:

i) Observational uncertainty: These refers to measurement variability as a result of the limitations of the measuring process[17]. The implication of this uncertainty is the reliability of the data we collect (observed data). This can also lead to under/over-estimation of forecast performance. For high precision systems, this can lead to wrong predictions. For example, if a robot poorly locates a point, his movement to that point will not be accurate.

ii) Parametrical uncertainty: This is uncertainty that arises due to the lack of precise knowledge about the parameters of a system[18]. This can also lead to a wide divergence of prediction from actual value. This is obvious since all parameters were certainly not taken into consideration.

iii) Structural uncertainty: This is uncertainty that comes from the model chosen. This can lead to wrong conclusions about the system even is the parameters and data collection is right.

In general, these uncertainties can perturb a system, causing poor forecast, performance, and even the representation of a given system.

2. Model averaging

The concept of model averaging[19] comes into play when we want to create an optimal model by averaging out a collection of other models. During model training, we a bound to have models that overestimate, underestimate or perform fairly. By averaging out these models, we can even out the overestimation and underestimation that will could yield a more robust model. Some practical application of weather of model averaging can be seen in:

- Multimodal AI: In multimodal learning, model averaging is employed to increase accuracy, robustness and, reduced variance when integrating information from varied modalities.
- Weather forecasting: In domains that have varies number of parameters to observe, some models will perform better on some parameters, others will perform well short or long term. Model averaging helps bring these varies models to reduce uncertainty and enhance robustness.

3. Ensemble methods

Ensemble methods in machine learning are technics that combines results from a variety of machine learning models to create a prediction with greater accuracy and prediction stability[20]. In other words, they help reduce uncertainties while improving predictive performance. Some methods and how they achieve their goals include:

i) Bagging (Bootstrap Aggregating): In this method, multiple random samples from the training dataset extracted (bootstrapped), and used to create several individual models. Now, these predictions are aggregated to form the most efficient predictions.

ii) Boosting: This is a technique based on sequentially training a model, where the current model attempts to learn from the errors committed by the previous models.

iii) Stacking: In this technic, models are successfully being trained, and the output from the current model is fed into the subsequent higher-level model as and input.

iv) Hybrid approach: Some technics are developed as a hybrid of other technics like bagged boosting, which combines the bagging and boosting methods.

4. Constructing a random forest

To construct the random forest:

i) The titanic data is loaded, and the features of interest are selected: ('age', 'sex', 'pclass', 'survived'). 'Survived' is the dependent variable and the rest independent.

ii) Null rows for survived were dropped and dummy variables was created for 'sex' and 'pclass'.

iii) 2% of data is used for testing and the rest for training

iv) A range from 50 to 1000 with a step of 50 is created. This will later help ust use the optimal number of leaves.

v) For every number of trees in the range, the model is created, and the accuracy is recorded.

vi) The best number of trees corresponds to that which produces the best model accuracy.

The following graph of number of leaves to accuracy was obtained.

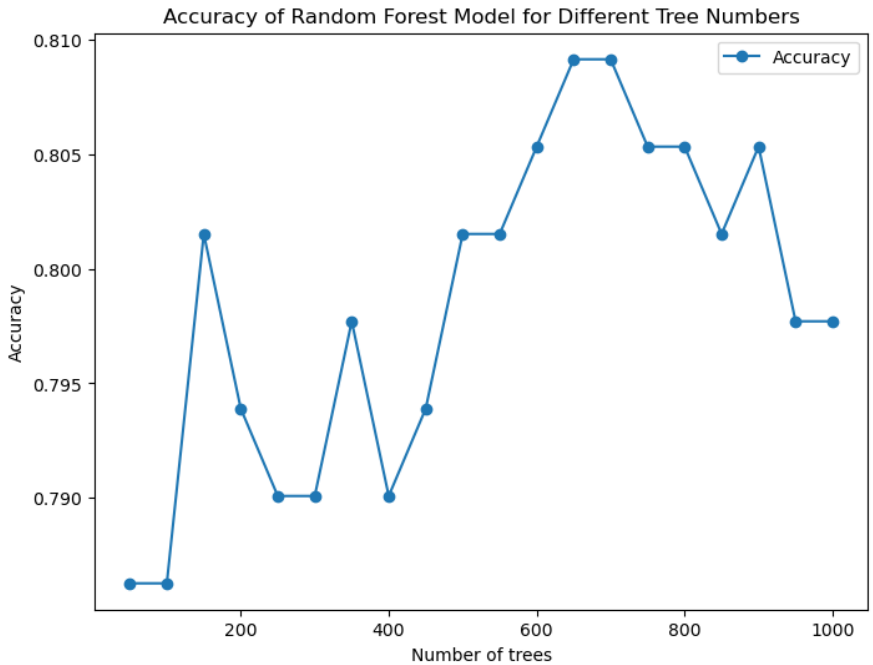


Figure 4 Number of leaves vs accuracy

The best number of leaves was obtained to be 650 with an accuracy of 0.81

```
Best number of trees: 650
Best accuracy: 0.81
```

5. ROC analysis:

The Receiver Operating Characteristic analysis is a useful way to assess the accuracy of model predictions by plotting sensitivity versus (1-specificity) of a classification test (as the threshold varies over an entire range of diagnostic test results)[21]. The Area Under the Curve (AUC) gives us an idea of the extent to which a predicted value will be of the right order. Generally, an AUC value close to 1 indicates better performance. To perform the ROC analysis:

i) For every model, the probability estimates of the positive class is calculated by applying the method predict_proba on the regression model.

ii) The false positive and true positive for every model is obtained using Sklearn's, roc_curve method on the test sample and the probability estimate.

iii) These values are plotted and the area under the curve – AUC for every model is calculated.

It is worth noting that these models were generated using the best parameters obtained in this exercise and the previous one.

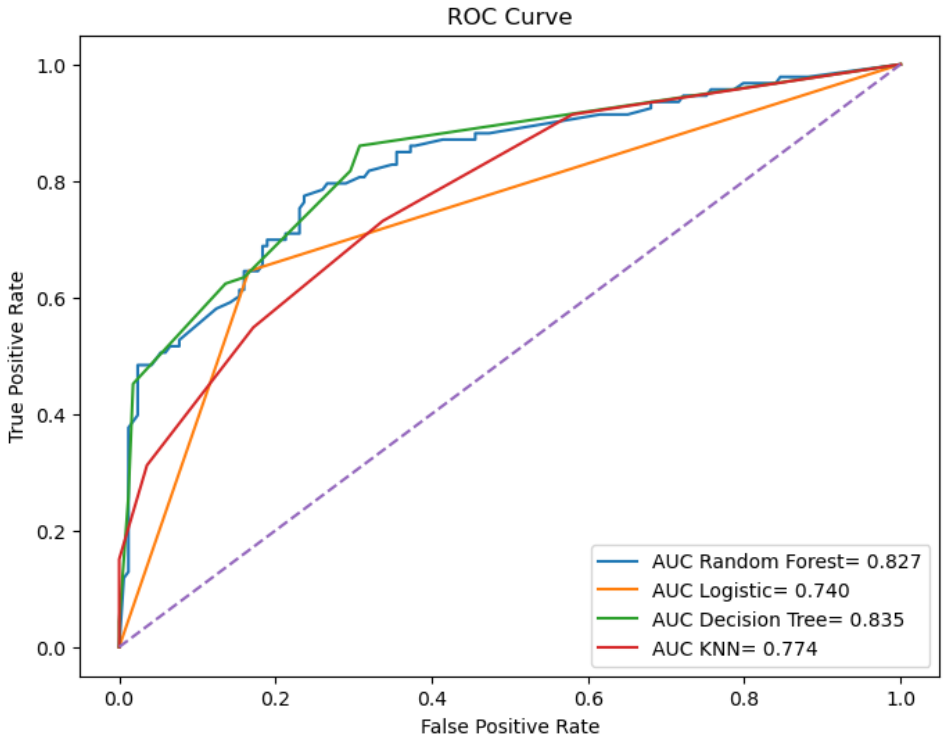


Figure 5 The result of this analysis is presented in Model analysis using the ROC curve

Figure 5. We notice that the Decision Tree model had the greatest AUC score of 0.835. It was followed closely by the Random Forest. Therefore, based on the results here obtained, the Ideal model for the Kagle competition will be the linear regression model.

QUESTION 4: Ensembles for regression

1. Random forest (RF) regression model

Random Forest Regression, is a machine learning ensemble technique, that leverages multiple decision trees and the bagging method (Bootstrap Aggregation) to effectively handle both regression and classification problems[20]. Some key features of random forest include:

i) Robustness: It robust to outliers, noisy data since it is and reduces the likelihood of overfitting
ii) Accuracy boosting: Ensemble models are based on several models. By sourcing the strength of every model, they tend to outperform individual models.
iii) Better generalisation: Ensemble methods tend to reduce variance.

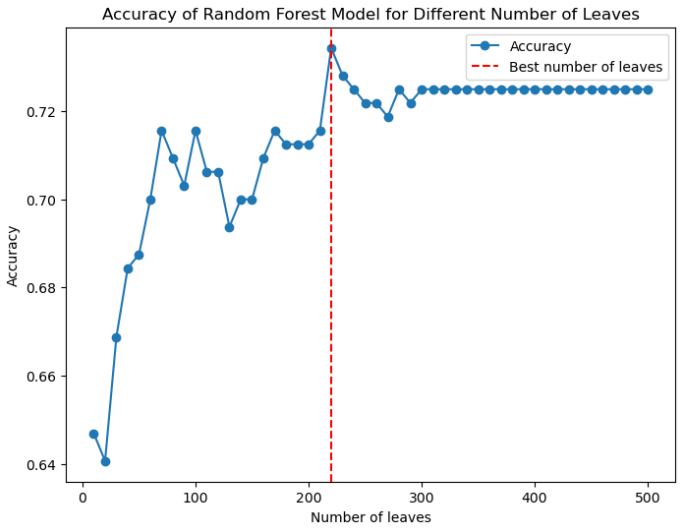
However, they demand more computational power as would demand a single model. Also, Interpretability of the model becomes difficult. It becomes hard to tell with certainty what the model is doing – the Blackbox effect.

2. Random forest model

To construct a random forest:

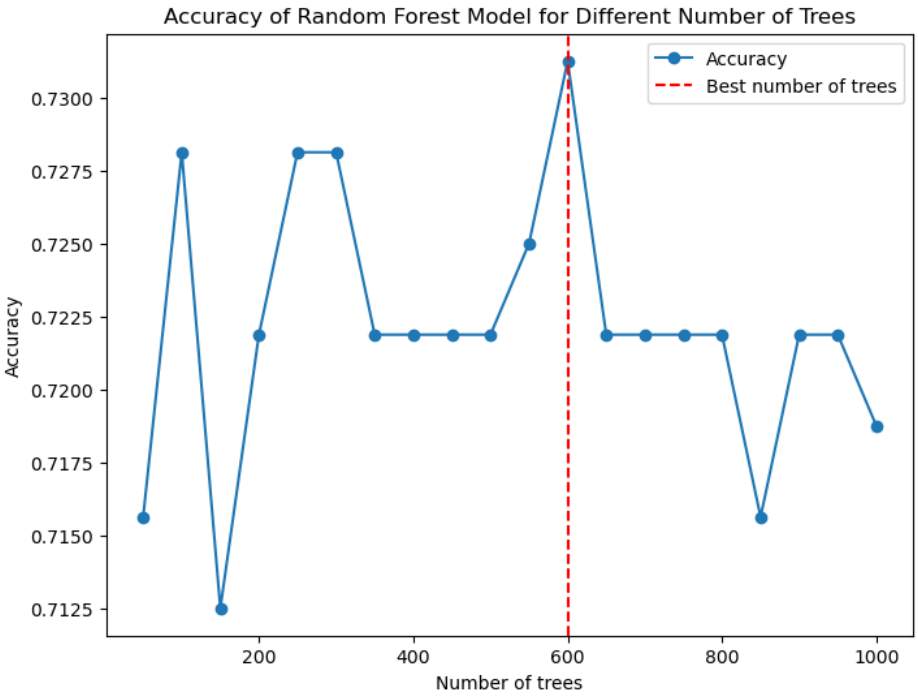
1. The wine data is loaded with all features. 'Wine quality' is set as the dependent variable and the rest independent. 2% of data is used for testing and the rest for training
2. A range from 10 to 500 with a step of 10 is created. This will later help calculate the optimal number of leaves.
3. For every number of leaves in the range, the model is created, and the accuracy is recorded.
4. The best number of leaves corresponds to that which produces the best model accuracy.

The following was obtained, with the best number of leaves being 220.



3. Optimal number of trees.

Likewise, the obtain the optimal number of trees, a range of trees from 50 to 1000 with a step of 50 was created and the model accuracy was tested on the number of trees. The following results were obtained:



Finally, using the optimal number of leaves and trees, the following model scores were obtained:

```
Best number of leaves for red wine: 220
Best number of trees for red wine: 600
Best accuracy for red wine: 0.73
```

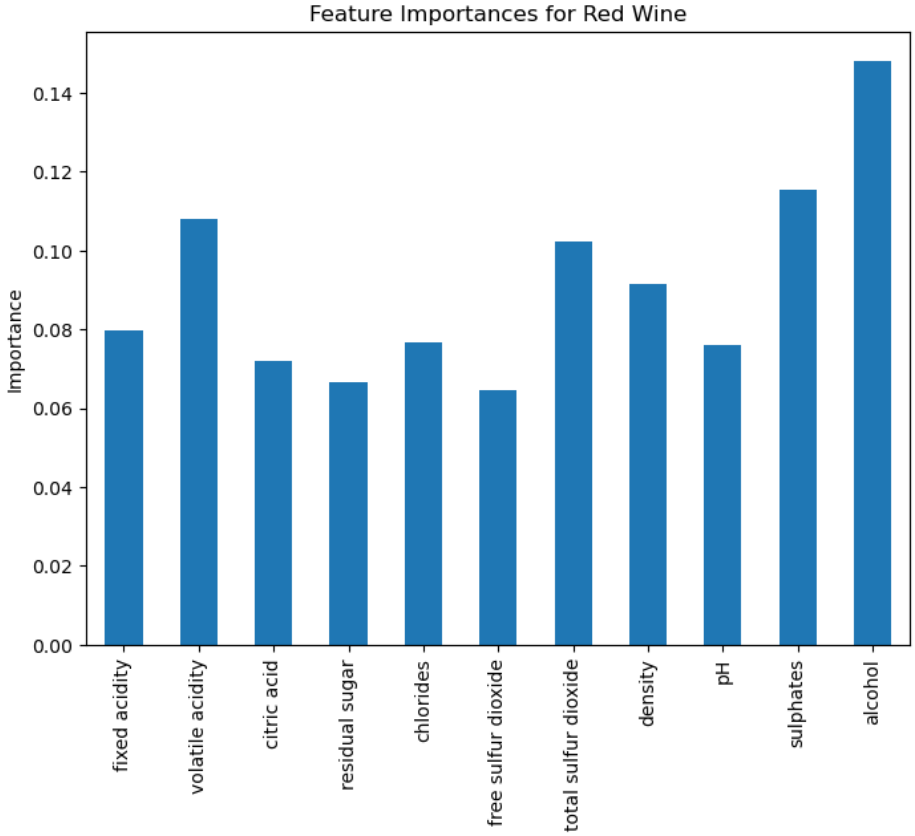
4. Bar graph of features and comparison.

To plot the bar graph of the various features:

i) The feature importance of Red Wine is calculated by applying feature_importances_ on the random forest model.
ii) Now, these values are plotted using a bar graph.

The figure bellow is a bar plot of the importance of each feature. The top 3 features with the most important features were obtained to be:

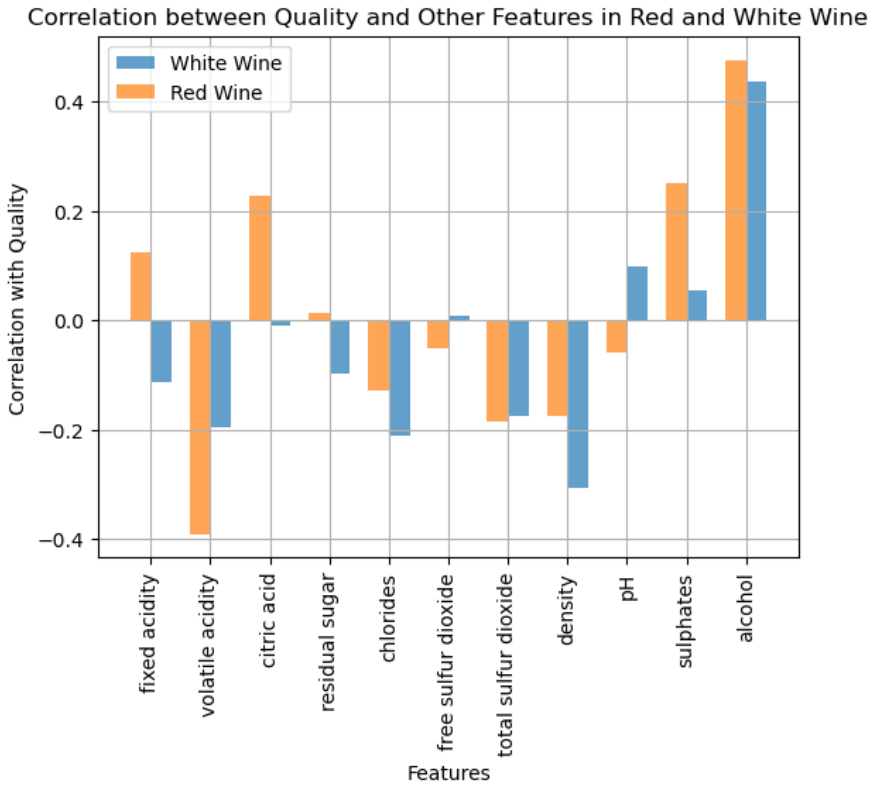
- Alcohol.
- Sulphate.
- Volatile Acidity



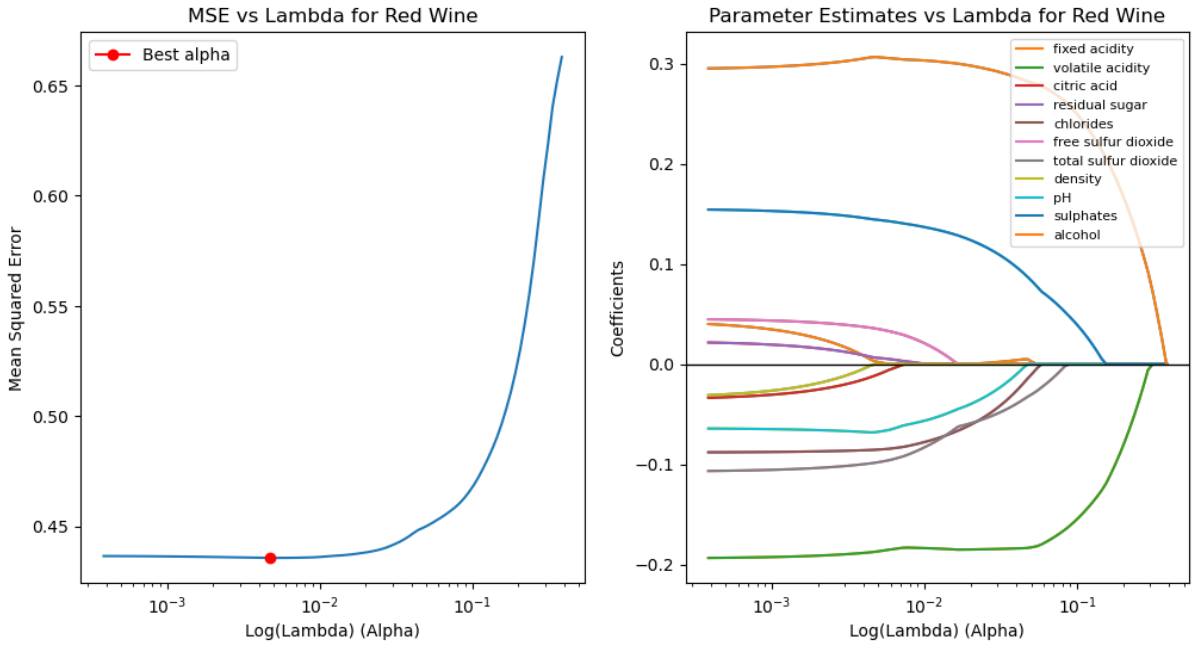
Comparison.

The following results for correlation and LASSO was obtained in Assignment 6.

i) Correlation:



ii) LASSO



Comparing the feature importance:

We observed a very close feature importance similarity in the 3 models. The 3 top features of greatest importance in all 3 models are:

- Alcohol
- Sulphate
- Volatile acidity.

All three models perform similarly in identifying feature importance. Therefor, we can conclude on the essential features needed to make a good wine. However, to ascertain the performance of in of the model, we will need to compare accuracy related values. This will be seen in the next response.

5. Performance comparison: Random Forest against Linear Regression and KNN

To compare the performance of these models, we will be looking at the R-squared, Mean Squared Error and Accuracy. The result is presented in the table bellow:

Model	**R-squared**	**MSE**	**Accuracy**
Random Forest Regression	0.72	0.38	0.72
Linear Regression	0.36	0.41	0.36
KNN	0.35	0.42	0.51

Based on the provided metrics, Random Forest Regression appears to be the best-performing model. It exhibits the highest R-squared value, the lowest Mean Squared Error, and a high

accuracy, indicating a strong fit to the data. While Linear Regression and KNN offer simpler interpretations, they fall short in terms of predictive accuracy. Other parameters that can be considered when choosing a model includes data and complexity, computational resources, and the importance of interpretability. But in the scope of this assignment, we can safely conclude that the random forest was more robust and accurate.

```
Performance Metrics for Red Wine Linear Regression
Accuracy for red wine linear regression: 0.3605517030386881
MSE for red wine linear regression: 0.41676716722140805
R-squared for red wine linear regression: 0.3605517030386881
```

```
Random forest regession model performance for red wine:

MSE for Red Wine: 0.38
Accuracy for Red Wine: 0.72
R-squared for Red Wine: 0.72
```

```
Average cross-validation MSE for KNN Red Wine: 0.5153021159874607
Average cross-validation MSE for KNN White Wine: 0.6485926664026181
Red Wine:
Mean Squared Error: 0.42099999999999993
White Wine:
Mean Squared Error: 0.48506122448979594
Red Wine:
R squared value: 0.3557823637531944
White Wine:
R squared value: 0.37368876747100044
```

Course Conclusion and Reflection.

Over the course of this course, we have been able to explore a multitude of ways and technics we can use to infer data from a given data set. We have been able to manipulate data in a way that gives us information and insights on the data. Finally, we were introduced to the fundamentals of machine learning. We have explored fundamental models, there area of application and their working principles. We have also learned through practice how to benchmark these models, and technics that will render these models more robust and accurate. On a personal note, this has given me greater confidence in approaching data exploration, analysis and the application of machine learning.

References

[1] “pandas documentation — pandas 2.2.2 documentation.” Accessed: Sep. 02, 2024. [Online]. Available: https://pandas.pydata.org/pandas-docs/stable/index.html

[2] “NumPy -.” Accessed: Sep. 02, 2024. [Online]. Available: https://numpy.org/

[3] “Matplotlib documentation — Matplotlib 3.9.2 documentation.” Accessed: Sep. 02, 2024. [Online]. Available: https://matplotlib.org/stable/

[4] L. Wiskott, “Lecture Notes on Principal Component Analysis”.

[5] “Principal component analysis | Nature Reviews Methods Primers.” Accessed: Nov. 24, 2024. [Online]. Available: https://www.nature.com/articles/s43586-022-00184-w

[6] “Applications of Principal Component Analysis (PCA),” OpenGenus IQ: Learn Algorithms, DL, System Design. Accessed: Dec. 04, 2024. [Online]. Available: https://iq.opengenus.org/applications-of-pca/

[7] “Introduction to Dimensionality Reduction,” GeeksforGeeks. Accessed: Nov. 24, 2024. [Online]. Available: https://www.geeksforgeeks.org/dimensionality-reduction/

[8] “What Is Principal Component Analysis (PCA)? | IBM.” Accessed: Nov. 24, 2024. [Online]. Available: https://www.ibm.com/topics/principal-component-analysis

[9] A. Alhallag, “Principal Component Analysis (PCA) — A Step-by-Step Practical Tutorial (w/ Numeric Examples),” Medium. Accessed: Nov. 24, 2024. [Online]. Available: https://medium.com/@aalhallag/principal-component-analysis-pca-a-step-by-step-practical-tutorial-w-numeric-examples-01c7b1412b53

[10] “18.650 (F16) Lecture 9: Principal Component Analysis (PCA) | Statistics for Applications | Mathematics,” MIT OpenCourseWare. Accessed: Nov. 28, 2024. [Online]. Available: https://ocw.mit.edu/courses/18-650-statistics-for-applications-fall-2016/resources/mit18_650f16_pca/

[11] T. Bock, “What is a Dendrogram?,” Displayr. Accessed: Nov. 30, 2024. [Online]. Available: https://www.displayr.com/what-is-dendrogram/

[12] “ML | Types of Linkages in Clustering,” GeeksforGeeks. Accessed: Nov. 30, 2024. [Online]. Available: https://www.geeksforgeeks.org/ml-types-of-linkages-in-clustering/

[13] “Hierarchical clustering (scipy.cluster.hierarchy) — SciPy v1.14.1 Manual.” Accessed: Nov. 30, 2024. [Online]. Available: https://docs.scipy.org/doc/scipy/reference/cluster.hierarchy.html

[14] “Who we are,” Merck.com. Accessed: Dec. 01, 2024. [Online]. Available: https://www.merck.com/company-overview/

[15] “Mission & Values.” Accessed: Dec. 01, 2024. [Online]. Available: https://www.unitedhealthgroup.com/uhg/mission-values.html

[16] “Files.” Accessed: Dec. 01, 2024. [Online]. Available: https://canvas.cmu.edu/courses/42065/files/folder/Lectures/Week%2012?preview=11993920

[17] B. Brown, “Observation uncertainty”.

[18] “https://library.fiveable.me/key-terms/control-theory/parametric-uncertainty.” Accessed: Dec. 02, 2024. [Online]. Available: https://library.fiveable.me/key-terms/control-theory/parametric-uncertainty

[19] “Model Averaging: A Robust Way to Deal with Model Uncertainty | by Osman Mamun | Towards Data Science.” Accessed: Dec. 04, 2024. [Online]. Available: https://towardsdatascience.com/model-averaging-a-robust-way-to-deal-with-model-uncertainty-a604c4ab2050

[20] “Random Forest Regression in Python,” GeeksforGeeks. Accessed: Dec. 05, 2024. [Online]. Available: https://www.geeksforgeeks.org/random-forest-regression-in-python/

[21] “SPSS Statistics Subscription - Early Access.” Accessed: Dec. 03, 2024. [Online]. Available: https://www.ibm.com/docs/en/spss-statistics/beta?topic=features-roc-analysis

[22] “Decision Tree,” GeeksforGeeks. Accessed: Nov. 16, 2024. [Online]. Available: https://www.geeksforgeeks.org/decision-tree/
[23] “Pruning decision trees,” GeeksforGeeks. Accessed: Nov. 19, 2024. [Online]. Available: https://www.geeksforgeeks.org/pruning-decision-trees/
[24] “Building and Implementing Decision Tree Classifiers with Scikit-Learn: A Comprehensive Guide,” GeeksforGeeks. Accessed: Nov. 19, 2024. [Online]. Available: https://www.geeksforgeeks.org/building-and-implementing-decision-tree-classifiers-with-scikit-learn-a-comprehensive-guide/
[25] “3.1. Cross-validation: evaluating estimator performance,” scikit-learn. Accessed: Nov. 19, 2024. [Online]. Available: https://scikit-learn/stable/modules/cross_validation.html
[26] “What is the k-nearest neighbors algorithm? | IBM.” Accessed: Nov. 19, 2024. [Online]. Available: https://www.ibm.com/topics/knn
[27] “K-Nearest Neighbor(KNN) Algorithm,” GeeksforGeeks. Accessed: Nov. 19, 2024. [Online]. Available: https://www.geeksforgeeks.org/k-nearest-neighbours/
[28] C. Story, “KNN vs. Linear Regression: How To Choose The Right ML Algorithm,” Medium. Accessed: Nov. 19, 2024. [Online]. Available: https://medium.com/@skytoinds/knn-vs-linear-regression-how-to-choose-the-right-ml-algorithm-4f6bf01a4202