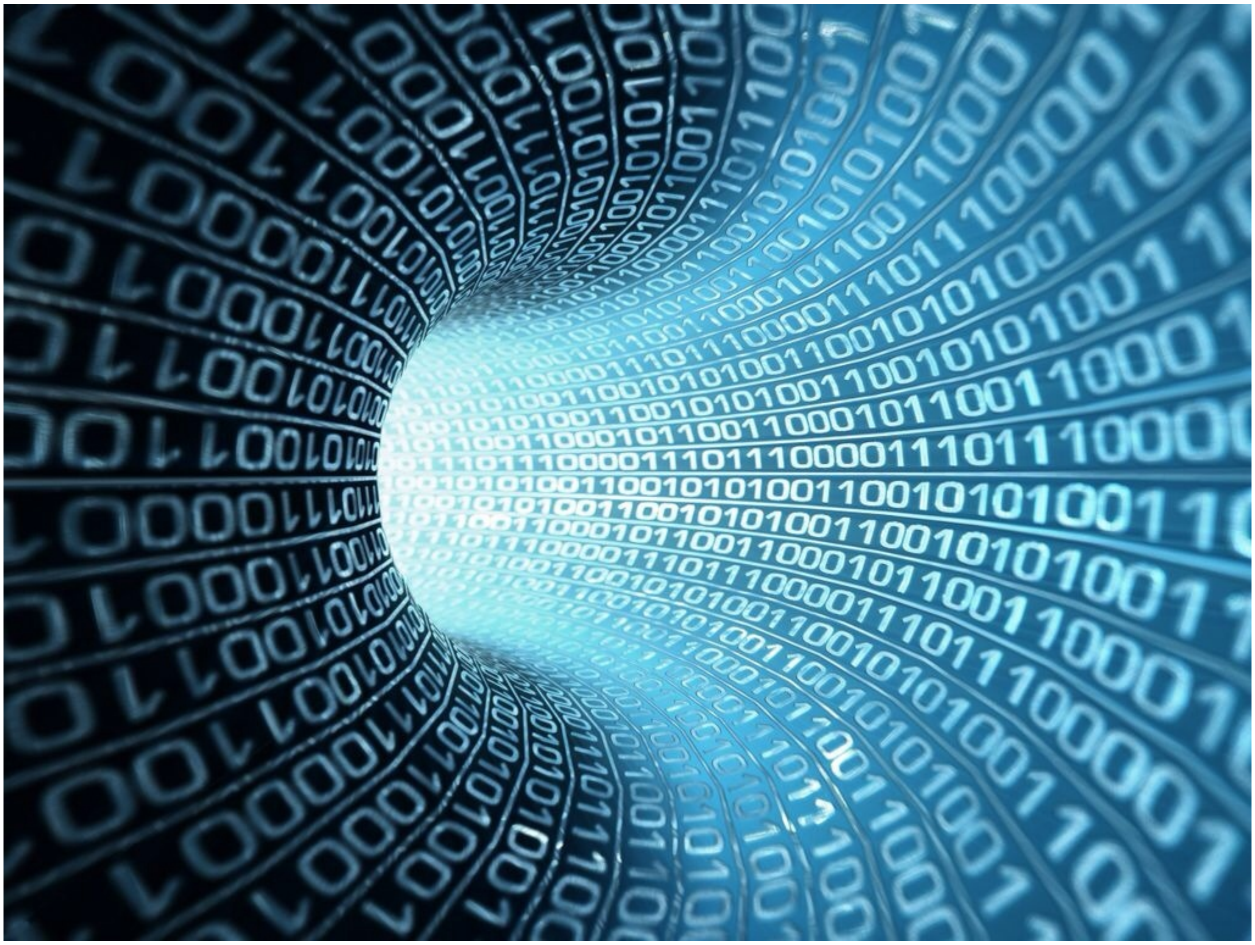
Report: DIAML Assignment 7



I, the undersigned, have read the entire contents of the syllabus for course 18-785 (Data Inference and Applied Machine Learning) and agree with the terms and conditions of participating in this course, including adherence to CMU's AIV policy.

Signature: Tunga Tessema

Andrew ID: tchamiss

Full Name: Tunga Tessema

Librairies

- **Pandas:** to read excel and csv files, convert them to dataframes and perform operations on them
- **numpy:** to create an array, calculate square roots and calculate correlation coefficients
- **matplotlib.pyplot:** to plot graphs
- **statsmodels.api**: to calculate linear and logistic regression models
- **sklearn**: to build decision tree, lasso model, random forest classifier and regressors and to calculate metrics like accuracy_score, mean_squared_error
- **Scipy**: to condense distance matrix
- **Searborn**: to plot heatmap
- **YFinance**: to download dow jones index data

Question 1

1.1 Give a qualitative description of Principal Component Analysis (PCA) and its applications in machine learning. Why might it be useful to consider PCA to transform a set of explanatory variables?

The Principal Component Analysis is a popular unsupervised learning technique for reducing the dimensionality of data. It increases interpretability yet, at the same time, it minimizes information loss.
The following are applications of PCA in machine learning:

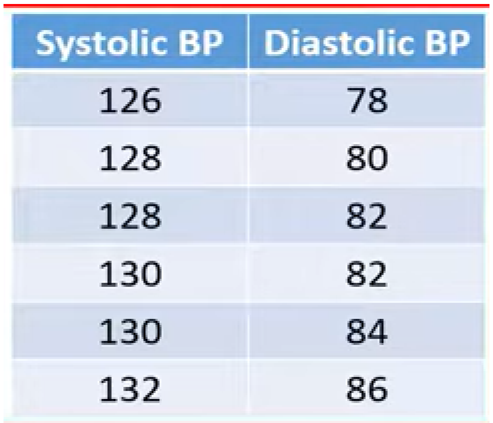
- PCA is used to visualize multidimensional data.
- It is used to reduce the number of dimensions in healthcare data.
- PCA can help resize an image.
- It can be used in finance to analyze stock data and forecast returns.
- PCA helps to find patterns in the high-dimensional datasets.

It is useful to consider PCA because:

- Less dimensions for a given datasets means less computation or training time
- Redundancy is removed after removing similar entries from the dataset
- Space required to store the data is reduced
- Makes the data easy for plotting in 2D and 3D plots
- It helps to find out the most significant features and skip the rest
- Leads to better human interpretations

1.2 Write down the mathematical equations for PCA explaining how one transforms the raw input data matrix X into a new set of variables. Give an interpretation of each matrix.

Example data: two types of blood pressure

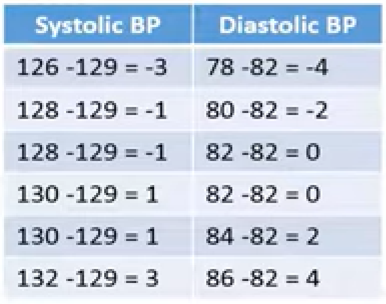


Systolic BP	Diastolic BP
126	78
128	80
128	82
130	82
130	84
132	86

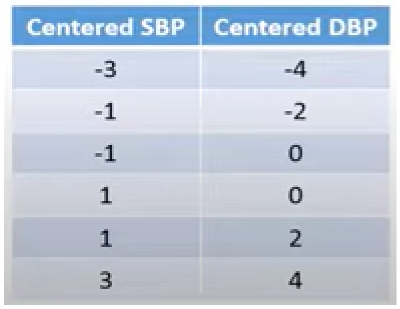
Steps

1. Center the data
2. Calculate the covariance matrix
3. Calculate the eigen values of the covariance matrix
4. Calculate the eigen vectors of the covariance matrix
5. Order the eigen vectors
6. Calculate the principal components

1. We center the data by subtracting the mean from each observation

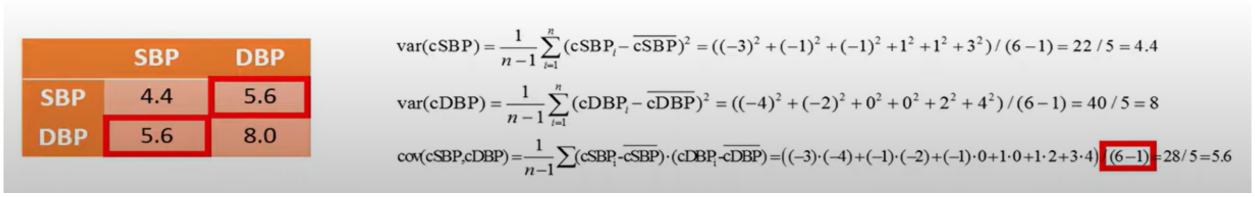


Systolic BP	Diastolic BP
126 -129 = -3	78 -82 = -4
128 -129 = -1	80 -82 = -2
128 -129 = -1	82 -82 = 0
130 -129 = 1	82 -82 = 0
130 -129 = 1	84 -82 = 2
132 -129 = 3	86 -82 = 4



Centered SBP	Centered DBP
-3	-4
-1	-2
-1	0
1	0
1	2
3	4

2. Calculate the covariance matrix = The main diagonal of the covariance matrix includes the variance of each variable. To calculate the variance of the centered data, we simply sum the squared values (because the mean of the centered data is always zero). Finally we calculate the covariance which is a measure of how much the two variables spread together. The sample covariance is calculated by multiplying the centered values of the two variables.



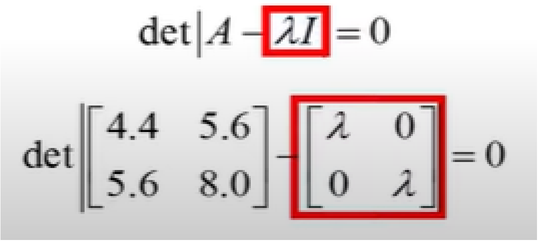
	SBP	DBP
SBP	4.4	5.6
DBP	5.6	8.0

$$\text{var(cSBP)} = \frac{1}{n-1}\sum_{i=1}^{n}(\text{cSBP}_i - \overline{\text{cSBP}})^2 = ((-3)^2 + (-1)^2 + (-1)^2 + 1^2 + 1^2 + 3^2)/(6-1) = 22/5 = 4.4$$

$$\text{var(cDBP)} = \frac{1}{n-1}\sum_{i=1}^{n}(\text{cDBP}_i - \overline{\text{cDBP}})^2 = ((-4)^2 + (-2)^2 + 0^2 + 0^2 + 2^2 + 4^2)/(6-1) = 40/5 = 8$$

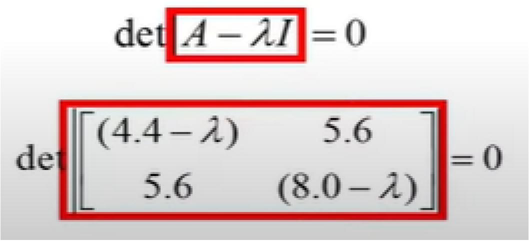
$$\text{cov(cSBP,cDBP)} = \frac{1}{n-1}\sum(\text{cSBP}_i - \overline{\text{cSBP}})\cdot(\text{cDBP}_i - \overline{\text{cDBP}}) = ((-3)\cdot(-4) + (-1)\cdot(-2) + (-1)\cdot 0 + 1\cdot 0 + 1\cdot 2 + 3\cdot 4)/\boxed{(6-1)} = 28/5 = 5.6$$

3. Calculate the eigen values of the covariance matrix. A is the covariance matrix, lambda is what we are looking for and I is an identity matrix which has the same number of rows and columns as the covariance matrix. Subtracting them gives the second matrix on the image.



$$\det|A - \boxed{\lambda I}| = 0$$

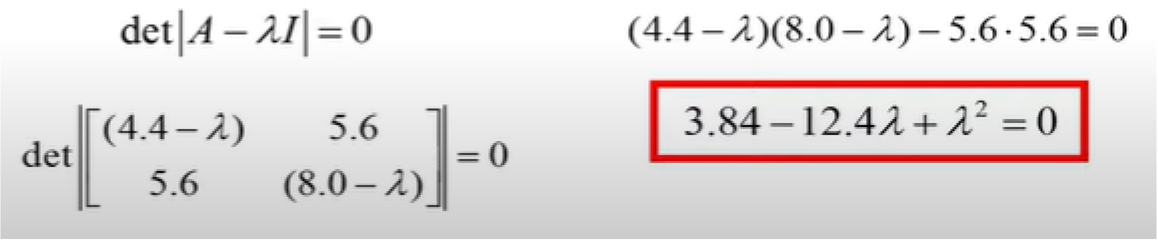
$$\det\left[\begin{bmatrix} 4.4 & 5.6 \\ 5.6 & 8.0 \end{bmatrix} - \boxed{\begin{bmatrix} \lambda & 0 \\ 0 & \lambda \end{bmatrix}}\right] = 0$$



$$\det\boxed{|A - \lambda I|} = 0$$

$$\det\boxed{\begin{bmatrix} (4.4-\lambda) & 5.6 \\ 5.6 & (8.0-\lambda) \end{bmatrix}} = 0$$

Next we calculate the determinant matrix, which is the product of the main diagonal minus the second diagonal.



$$\det|A - \lambda I| = 0$$

$$\det\begin{bmatrix} (4.4-\lambda) & 5.6 \\ 5.6 & (8.0-\lambda) \end{bmatrix} = 0$$

$$(4.4-\lambda)(8.0-\lambda) - 5.6\cdot 5.6 = 0$$

$$\boxed{3.84 - 12.4\lambda + \lambda^2 = 0}$$

We solve the quadratic equation and find our two eigen values.

$$\boxed{3.84 - 12.4\lambda + \lambda^2} = 0$$

$$\lambda_1 = 0.32 \qquad \lambda_2 = 12.08$$

4. Calculate the eigen vectors of the eigen values:

Let's calculate for the first eigen value 12.08

$$A \cdot v = \lambda \cdot v$$

$$\begin{bmatrix} 4.4 & 5.6 \\ 5.6 & 8.0 \end{bmatrix} \cdot \begin{bmatrix} x \\ y \end{bmatrix} = 12.08 \cdot \begin{bmatrix} x \\ y \end{bmatrix}$$

$$4.4x + 5.6y = 12.08x$$
$$5.6x + 8.0y = 12.08y$$

$$5.6y = 7.68x$$
$$5.6x = 4.08y$$
$$y = 1.37\boxed{x}$$
$$1.37x = y$$

We replace x by 1 and find the following eigen vector. Then we normalize the vector to get a length of 1, and get the vector on the second image

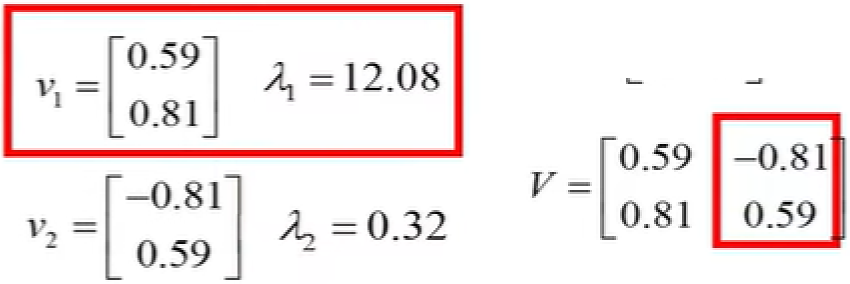
$$v_2 = \begin{bmatrix} 1 \\ 1.37 \end{bmatrix} \qquad v_2 = \begin{bmatrix} 0.59 \\ 0.81 \end{bmatrix}$$

We do the same thing for the second eigen value: 0.32. Our final results are

$$v_2 = \begin{bmatrix} 0.59 \\ 0.81 \end{bmatrix} \quad \lambda_2 = 12.08$$

$$v_1 = \begin{bmatrix} -0.81 \\ 0.59 \end{bmatrix} \quad \lambda_1 = 0.32$$

5. We order the eigen vectors: The eigen vector with the largest eigen value becomes our first eigen vector, so we called it v1 instead of v2. We then put these two vectors together into a matrix called V. We put the first vector as the first column and the second vector as the second column.

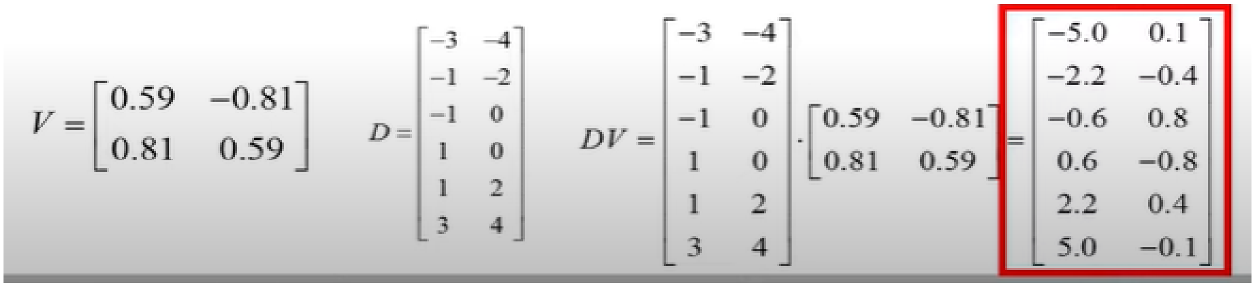


$$v_1 = \begin{bmatrix} 0.59 \\ 0.81 \end{bmatrix} \quad \lambda_1 = 12.08$$

$$v_2 = \begin{bmatrix} -0.81 \\ 0.59 \end{bmatrix} \quad \lambda_2 = 0.32$$

$$V = \begin{bmatrix} 0.59 & -0.81 \\ 0.81 & 0.59 \end{bmatrix}$$

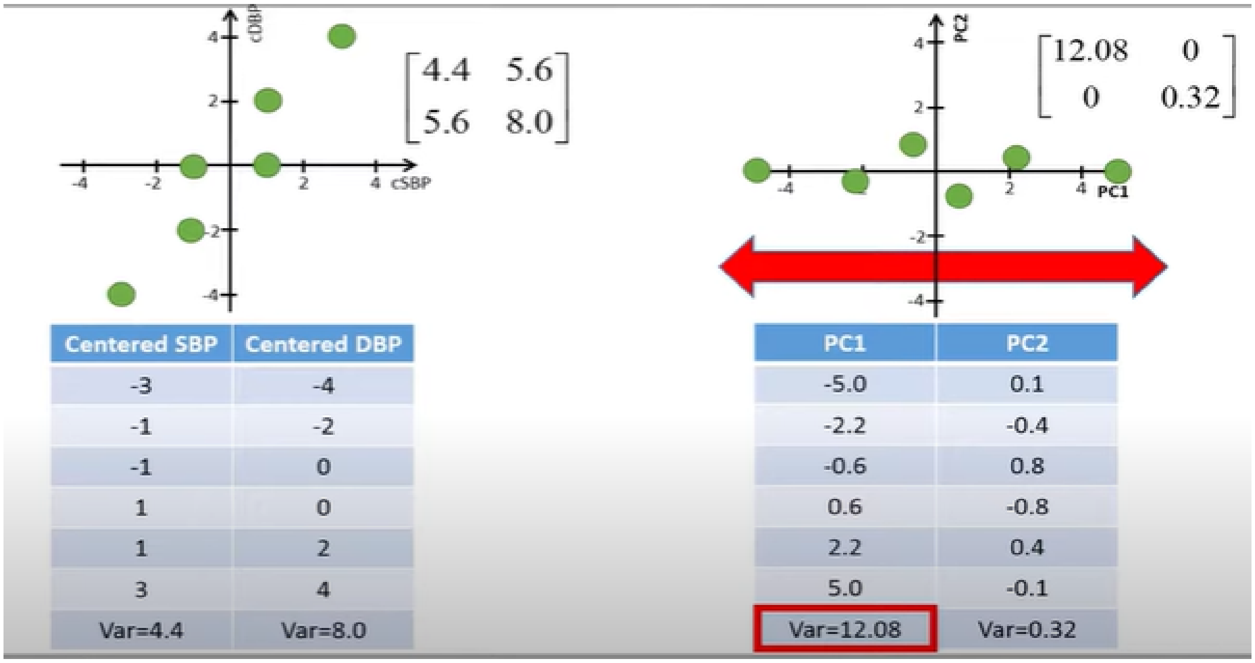
6. We calculate the principal components: we multiply the vector V by the matrix of our data D.



$$V = \begin{bmatrix} 0.59 & -0.81 \\ 0.81 & 0.59 \end{bmatrix} \qquad D = \begin{bmatrix} -3 & -4 \\ -1 & -2 \\ -1 & 0 \\ 1 & 0 \\ 1 & 2 \\ 3 & 4 \end{bmatrix}$$

$$DV = \begin{bmatrix} -3 & -4 \\ -1 & -2 \\ -1 & 0 \\ 1 & 0 \\ 1 & 2 \\ 3 & 4 \end{bmatrix} \cdot \begin{bmatrix} 0.59 & -0.81 \\ 0.81 & 0.59 \end{bmatrix} = \begin{bmatrix} -5.0 & 0.1 \\ -2.2 & -0.4 \\ -0.6 & 0.8 \\ 0.6 & -0.8 \\ 2.2 & 0.4 \\ 5.0 & -0.1 \end{bmatrix}$$

7. Interpret the PCA

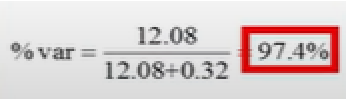


$$\begin{bmatrix} 4.4 & 5.6 \\ 5.6 & 8.0 \end{bmatrix} \qquad \begin{bmatrix} 12.08 & 0 \\ 0 & 0.32 \end{bmatrix}$$

Centered SBP	Centered DBP
-3	-4
-1	-2
-1	0
1	0
1	2
3	4
Var=4.4	Var=8.0

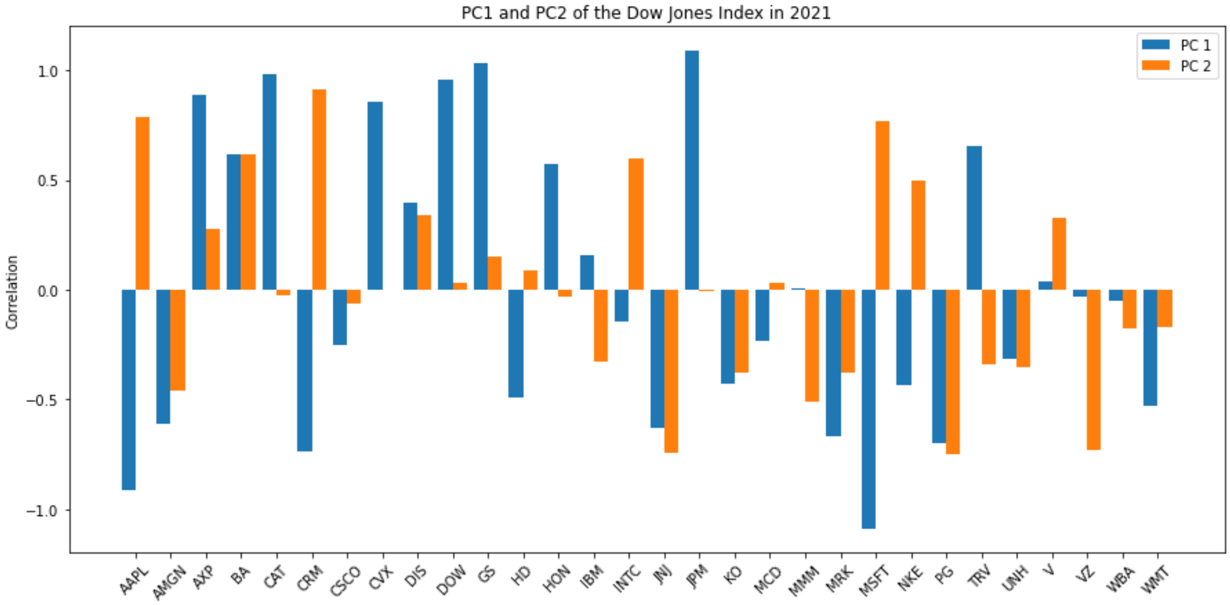
PC1	PC2
-5.0	0.1
-2.2	-0.4
-0.6	0.8
0.6	-0.8
2.2	0.4
5.0	-0.1
Var=12.08	Var=0.32

On the first covariance matrix, we see that the variance of SBP is 4.4 and the variance of DBP is 8.0. Their covariance is 5.6 which shows that there is a positive correlation between the two variables. When we transform the data using PCA, the first variable called PC1 has a variance of 12.8 and the second variable called PC2 has a variance of 0.32. This means that almost all variance is kept in the first principal component. If we divide the variance of PC1 by the total variance, we see that it accounts for 97.4% of the total variance. In addition, we see that the covariance of PC1 and PC2 is 0, which means they are totally uncorrelated. Hence, we can simply delete the second principal component.



$$\%\,\text{var} = \frac{12.08}{12.08+0.32} = \boxed{97.4\%}$$

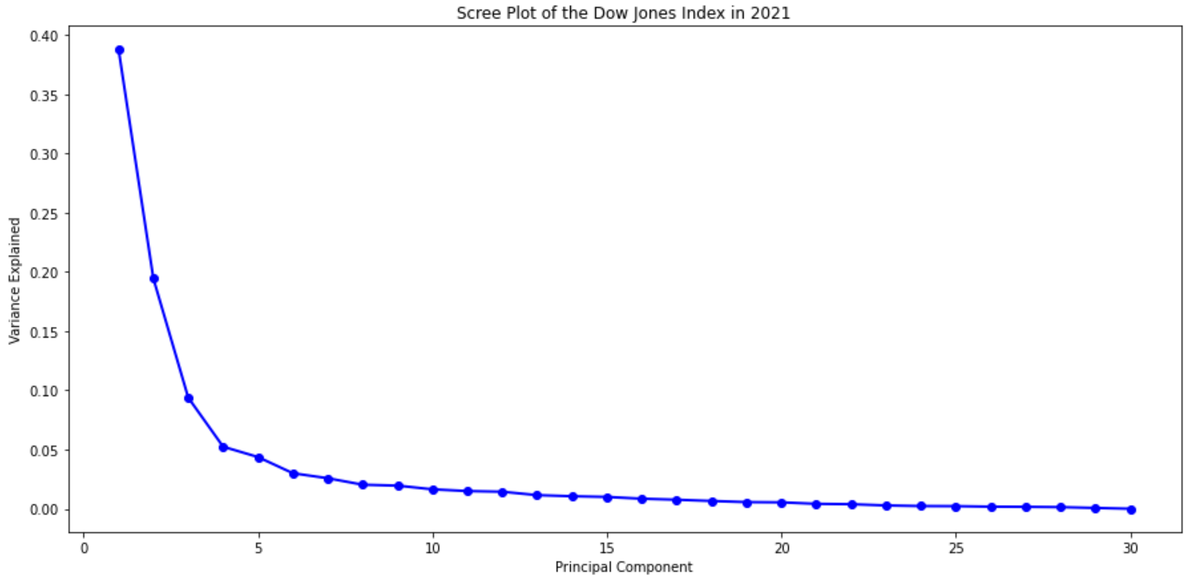
1.3 Use at least one year of daily returns to calculate the correlation matrix for the 30 stocks that are constituents of the Dow Jones Index. MATLAB's "BlueChipStockMoments" can be used to calculate the correlation matrix. Use this correlation matrix for PCA and construct bar graphs to show the weight of each stock for the first and second principal components. Is the first or second principal component similar to the market (equal weight on each stock)? Discuss why?



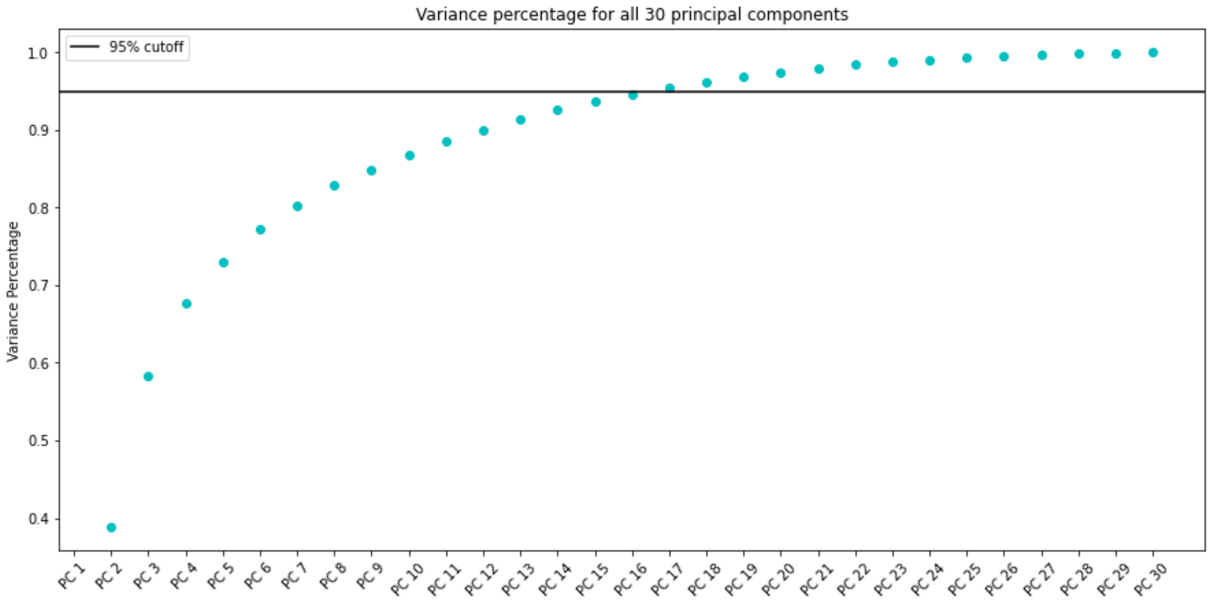
I first imported the Dow Jones Index of 2021(from Jan 1, 2021 to Jan 1, 2022) from Yahoo Finance. I then selected the Adj Close columns and dropped the other columns. I then calculated the daily returns by using the pct_change function. I then calculated the dataframes correlation. The above graph shows the weight that PC1 and PC2 give to each stock.

The weight of each component in the current market and what is shown on the graph doesn't match. On wikipedia, we can see that UNH has the highest weight, however on this graph MSFT and JPM have the highest weight on PC1 and CRM has the highest weight on PC2.

1.4 Calculate the amount of variance explained by each principal component and make a 'Scree' plot. How many principal components are required to explain 95% of the variance?

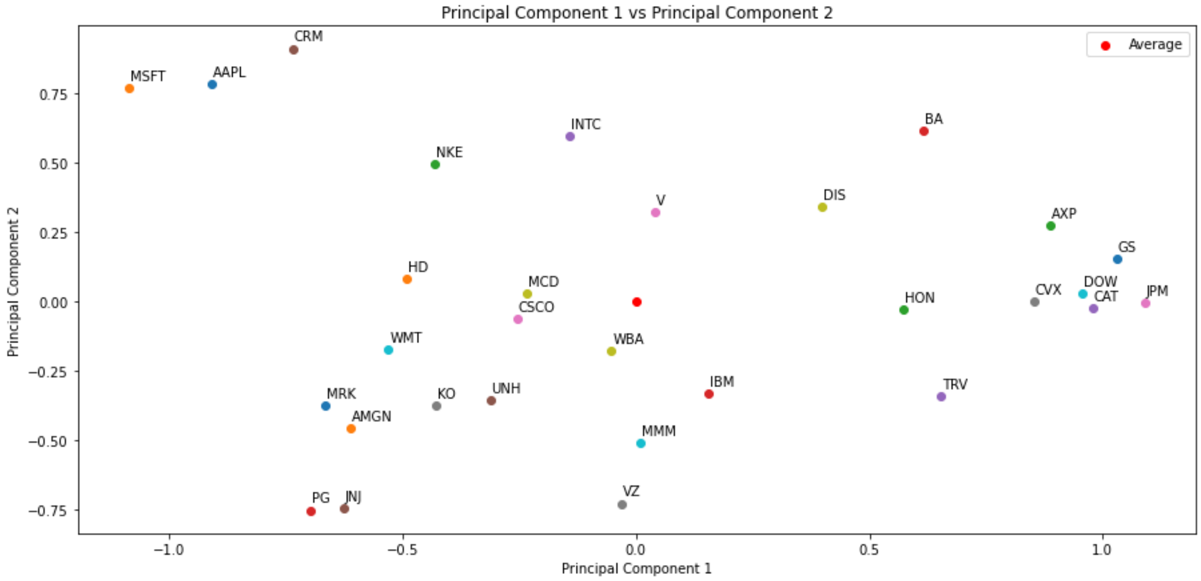


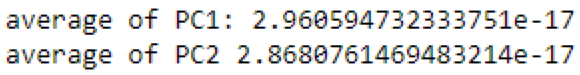
I used the PCA function of sklearn to create 30 principal components and fit them to the correlation matrix. I then got the variance explained by each principal component by using the explained_variance_ratio_ property of the PCA function. I plotted the Variance explained against each principal component in the above graph.



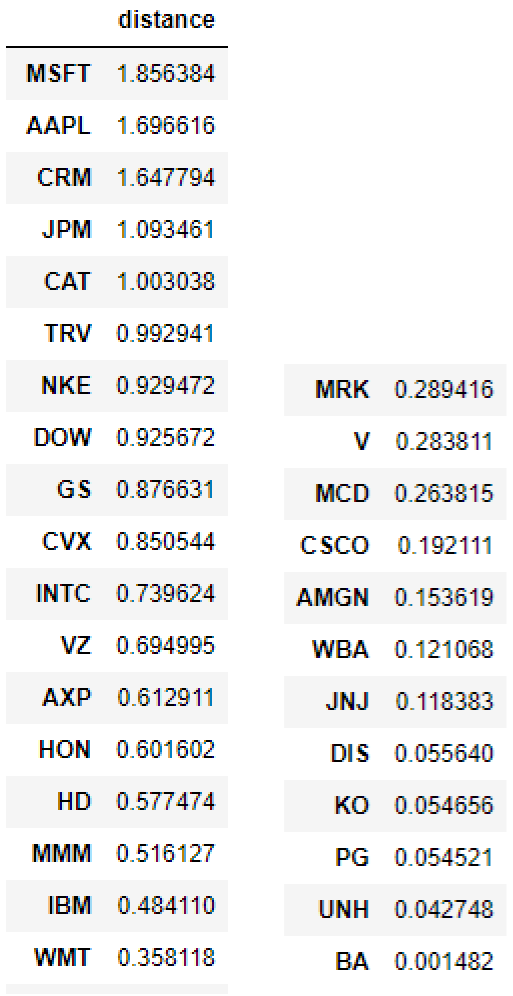
I then calculated the variance percentage of each principal component by dividing the variance of a principal component by the total variance. I also plotted another graph that shows the cumulative variance percentage. We see from the above graph that the 95% of the variance is explained by 16 stocks.

1.5 Investigate the scatter plot of the first two principal components and calculate the average of all 30 stocks. Based on Euclidean distances away from this average, identify the three most distant stocks. Can you explain why these stocks are unusual?





```
average of PC1: 2.960594732333751e-17
average of PC2 2.8680761469483214e-17
```



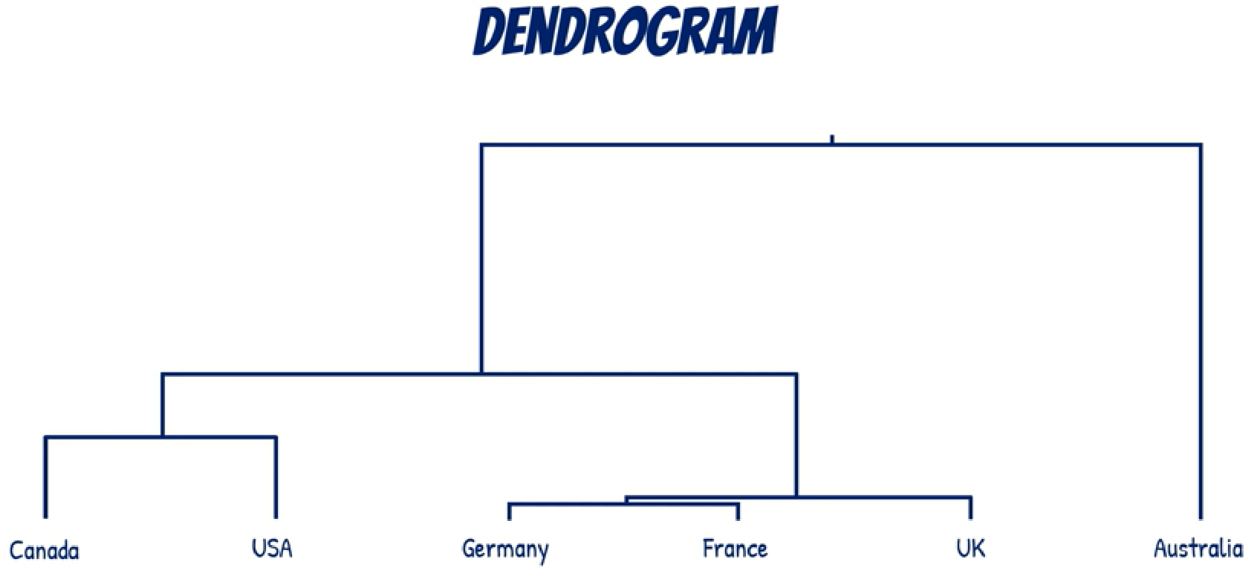
	distance
MSFT	1.856384
AAPL	1.696616
CRM	1.647794
JPM	1.093461
CAT	1.003038
TRV	0.992941
NKE	0.929472
DOW	0.925672
GS	0.876631
CVX	0.850544
INTC	0.739624
VZ	0.694995
AXP	0.612911
HON	0.601602
HD	0.577474
MMM	0.516127
IBM	0.484110
WMT	0.358118
MRK	0.289416
V	0.283811
MCD	0.263815
CSCO	0.192111
AMGN	0.153619
WBA	0.121068
JNJ	0.118383
DIS	0.055640
KO	0.054656
PG	0.054521
UNH	0.042748
BA	0.001482

I plotted the PC2 against PC1. I then calculated the average of each PC using the mean function. I got the averages shown above and plotted it on the graph using a red dot. While plotting I also calculated the euclidean distance of each stock from the average value and stored it in the above dataframe. As can be seen in the above dataframe, Microsoft, Apple and Salesforce are the three most distant stocks. All of them are technology companies and in the year 2021, all of them have a high weight on both PC1 and PC2. This may be explained by the fact that after covid19, the stock of these companies has increased.

Question 2

2.1. Describe the components of a dendrogram, how it is constructed and how it is interpreted.

A dendrogram is a type of tree diagram showing hierarchical clustering relationships between similar sets of data. We start with each case being its own cluster. There are a total of N clusters. Second, using some similarity measure like euclidean distance, we group the two closest clusters together, reaching an ‘n minus 1’ cluster solution. Then we repeat this procedure until all observations are in a single cluster.

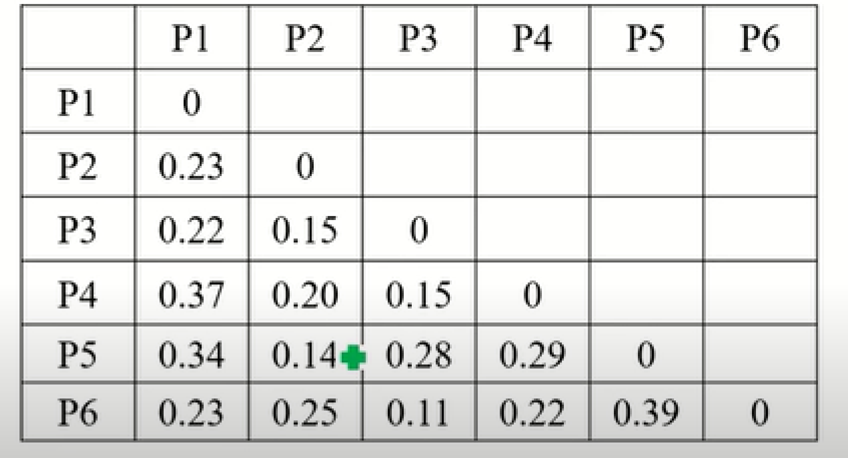


To see how we can interpret a dendrogram, let’s see the above example created on a country cluster.

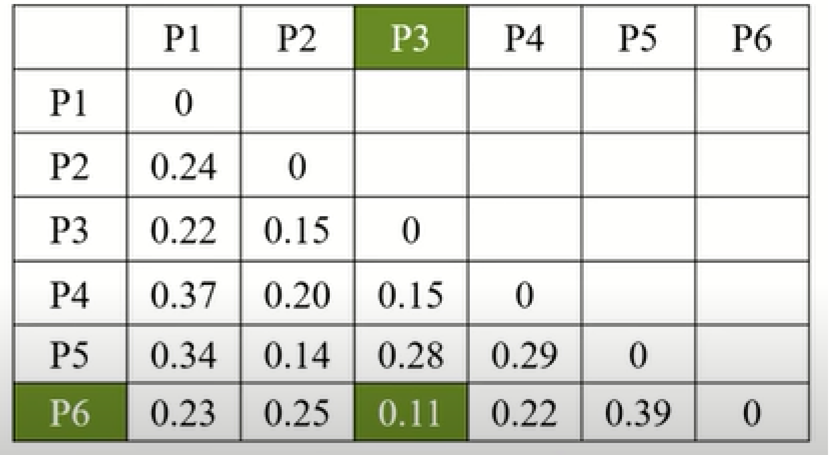
- The first two lines that merge are those of Germany and France. According to the dendrogram, these two countries are the closest in terms of the features considered.
- Well, the bigger the distance between two links, the bigger the difference in terms of the chosen features. As you can see, Germany, France and the UK merged into 1 cluster very quickly. This shows us that they are very similar in terms of ‘longitude’ and ‘latitude’. Moreover, Germany and France are closer than Germany and UK, or France and UK. The USA and Canada came together not long after. However, it took half of the dendrogram to join these 5 countries together. This indicates the Europe cluster and the North America cluster are not so alike. Finally, the distance needed for Australia to join the other 5 countries was the other half of the dendrogram, meaning it is extremely different from them. To sum up, the distance between the links shows similarity, or better: dissimilarity
- Choice of the number of clusters: when you draw a straight line, you should count the number of links that have been broken. When the distance between two stages is too big, it is probably a good idea to stop there. For our case, I would draw the line at 3 clusters and remain with North America, Europe, and Australia.

2.2. Given a collection of pairwise dissimilarity values, describe the steps involved in constructing a dendrogram.

Let’s take an example:
We have the following dissimilarity matrix:

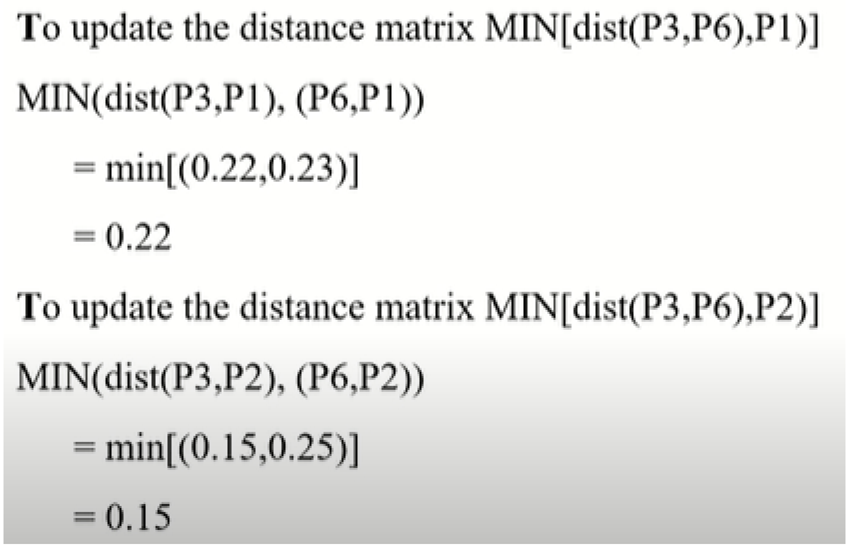


	P1	P2	P3	P4	P5	P6
P1	0					
P2	0.23	0				
P3	0.22	0.15	0			
P4	0.37	0.20	0.15	0		
P5	0.34	0.14	0.28	0.29	0	
P6	0.23	0.25	0.11	0.22	0.39	0



	P1	P2	P3	P4	P5	P6
P1	0					
P2	0.24	0				
P3	0.22	0.15	0			
P4	0.37	0.20	0.15	0		
P5	0.34	0.14	0.28	0.29	0	
P6	0.23	0.25	0.11	0.22	0.39	0

We first take the minimum value: 0.11, which is between P3 and P6. We form the first cluster. We then recalculate the values by grouping P3 and P6 together. In this case since the dissimilarity values are distances, we take the minimum of the distance between P3 and P1 and P6 and P1. That is 0.22. We do the same for P2, P4 and P5.



To update the distance matrix MIN[dist(P3,P6),P1)]

MIN(dist(P3,P1), (P6,P1))

= min[(0.22,0.23)]

= 0.22

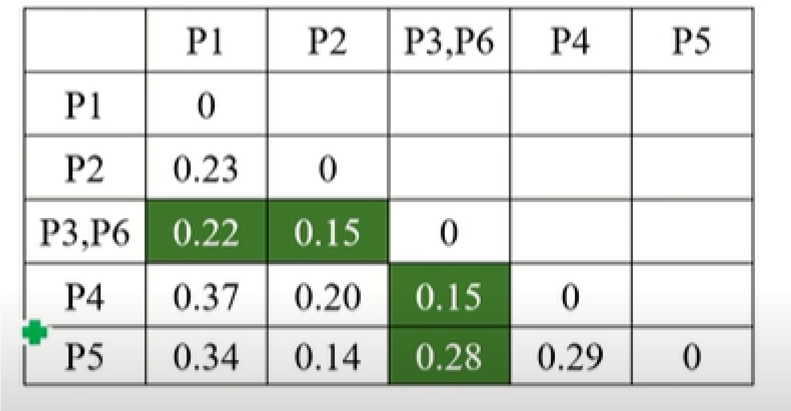
To update the distance matrix MIN[dist(P3,P6),P2)]

MIN(dist(P3,P2), (P6,P2))

= min[(0.15,0.25)]

= 0.15

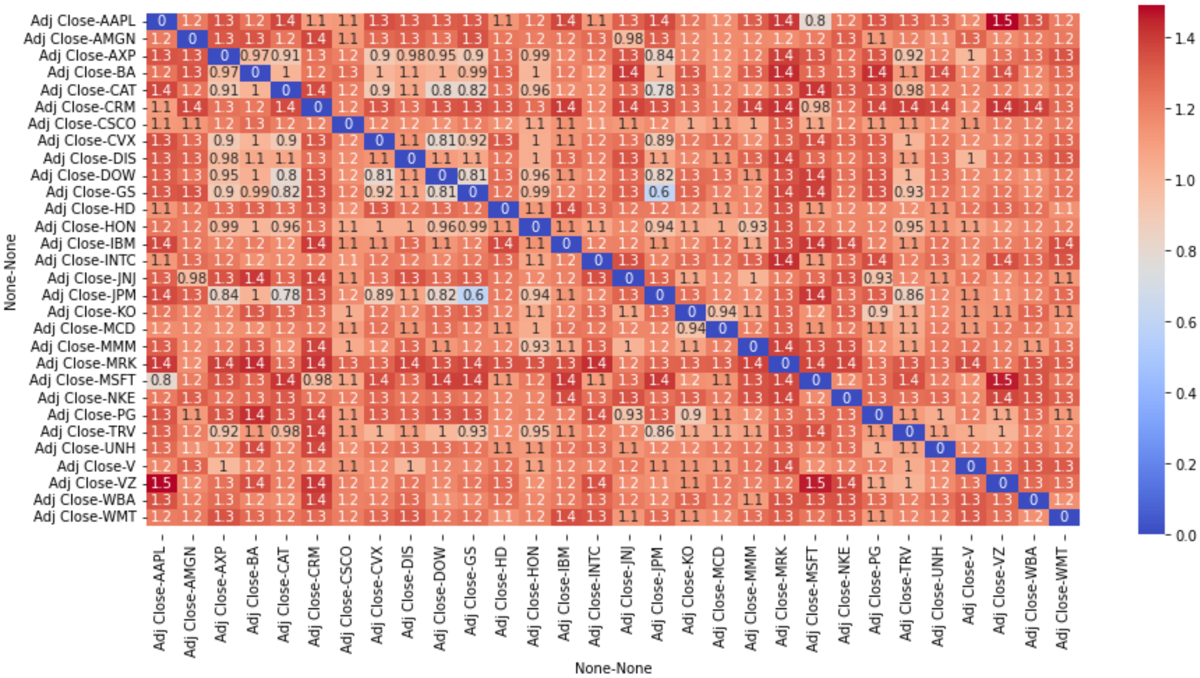
We then get the following matrix:



	P1	P2	P3,P6	P4	P5
P1	0				
P2	0.23	0			
P3,P6	0.22	0.15	0		
P4	0.37	0.20	0.15	0	
P5	0.34	0.14	0.28	0.29	0

We again repeat the previous steps, take the minimum and recalculate the distance based on the new cluster.

2.3. Use the correlation matrix from question (1.3) above to provide pairwise distances between the 30 stocks. Give the formula for this rescaled distance and provide an interpretation of small and large distances.



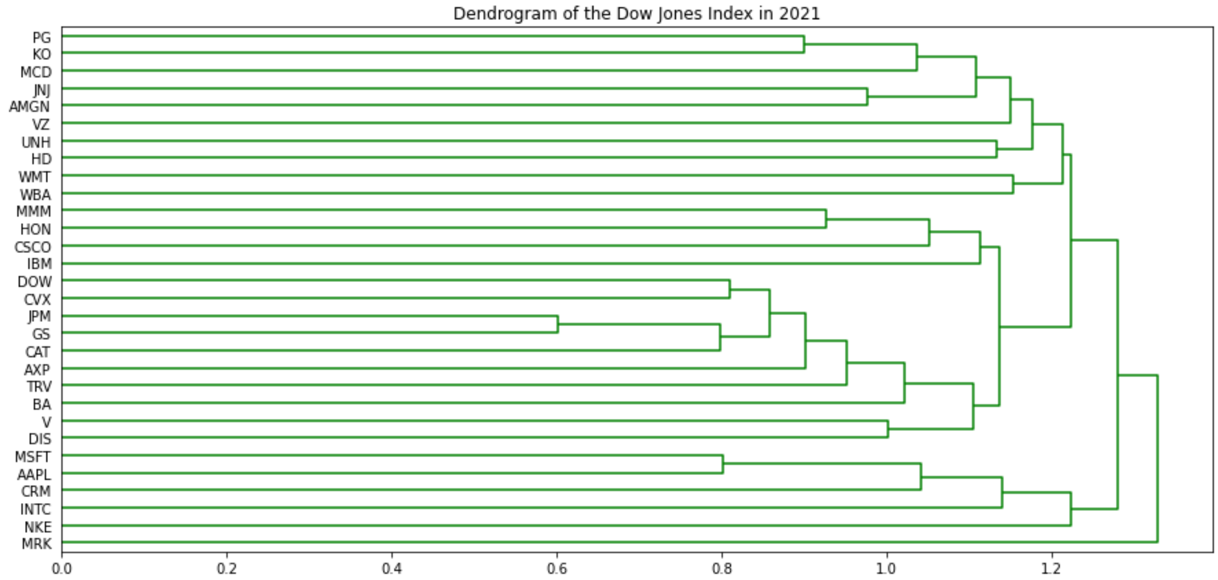
To calculate the pairwise distance, I used the following formula:

$$d_{ij} = (2(1 - \rho_{ij}))^{1/2}.$$

where Pij is their Pearson correlation coefficient.

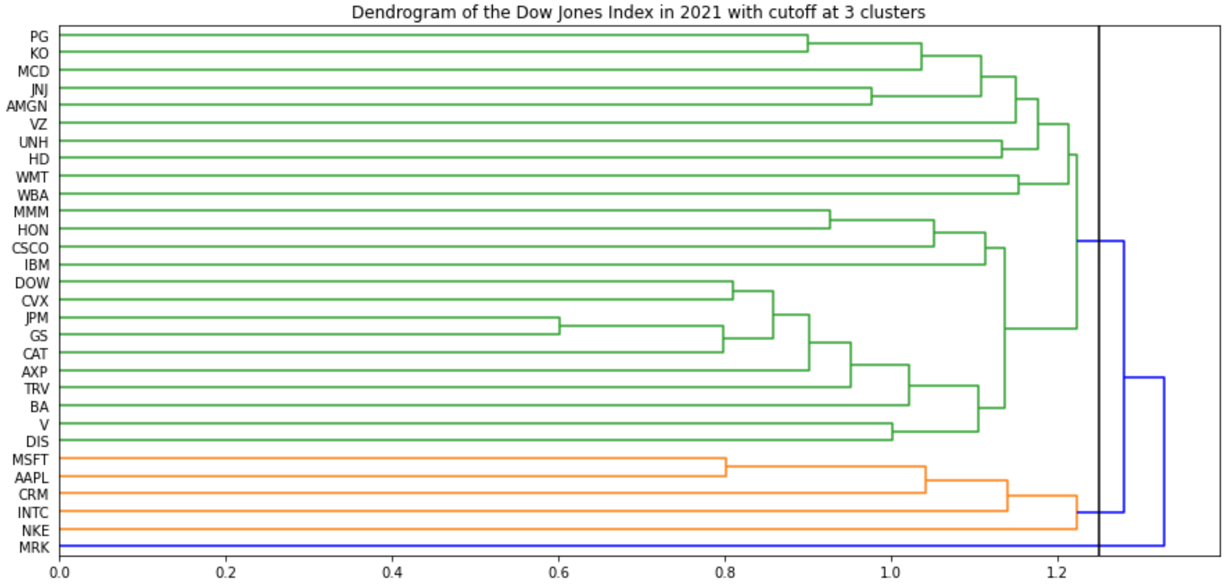
From the above heatmap, we can see that small distances are closer to 0 and large distances are closer to 1.5.

2.4. Construct a horizontal dendrogram using the average linkage approach, carefully labeling the graphic with the names of the 30 stocks.



To construct a dendrogram, I first condensed the pairwise matrix using the squareform function from scipy library. Condensing means removing the upper or lower part of the matrix that has the same symmetric values. I then passed the condensed matrix to the scipy.cluster.hierarchy.linkage method and chose the average linkage options. I then passed the linkage to the dendrogram function of scipy and plotted the dendrogram using matplotlib.

2.5. Use the dendrogram to provide a few clusters of stocks and list the stocks that are members of each cluster. Can you provide a description of each cluster and relate it to industrial sectors such as Financials, Energy etc?



To get three clusters, I identified the cutoff point by looking at the dendrogram. I identified point 1.1. I then drew a vertical line at 1.1 and colored the 3 clusters differently.
The first cluster (in green) is composed of:

- PG, KO, MCD, JNJ, AMGN, VZ, UNH, HD, WMT, WBA, MMM, HON, CSCO, IBM, DOW, CVX, JPM, GS, CAT, AXP, TRV, BA, V, DIS

The second cluster (in orange) is composed of:

- MSFT, AAPL, CRM, INTC, NKE

The third cluster (in blue) is composed of:

- MRK.

The first cluster is composed of companies that are providing consumer goods. It includes companies like Coca Cola, McDonald's, Johnson & Johnson …etc. Even though the companies provide consumer goods in different sectors like fast food, pharmaceuticals, finance, aerospace…etc, they all still provide consumer goods.
The second cluster is mainly composed of technology companies like Microsoft (MSFT), Apple (AAPL), Salesforce (CRM) and Intel (INTC). We also have Nike which is engaged in the design, development, manufacturing, and worldwide marketing and sales of footwear, apparel, equipment, accessories, and services. Nike is the most different from other companies in the cluster, we can see on the dendrogram that it has joined the cluster towards the end.
The third cluster contains MRK. MRK refers to Merck & Co. which is an american pharmaceuticals company.

3. Ensembles for classification
3.1 Name three sources of uncertainty and explain how they impact on the modeling process when using machine learning approaches.

Real world systems are immensely complex, and models that attempt to simulate them are essentially simplified mathematical representations of physical phenomena.

There are three sources of uncertainty.

1. Observational uncertainty describes our ability to obtain a precise data measurement. This is a common source of uncertainty where precision of repeated surveys may be affected by several factors, such as skill level of the field crew, precision of sampling devices, and location of survey points.

2. Parametric uncertainty is uncertainty regarding the distribution of possible parameter values. Sources include data quality and data completeness, so it is reduced as more data becomes available.
3. Structural uncertainty is whether the model appropriately represents the physical phenomenon. This type of uncertainty can be accounted for by considering alternate models, or using the weighted average of several models.

3.2 What is the concept behind model averaging and give some examples of how this technique can be implemented in practice when generating predictions?

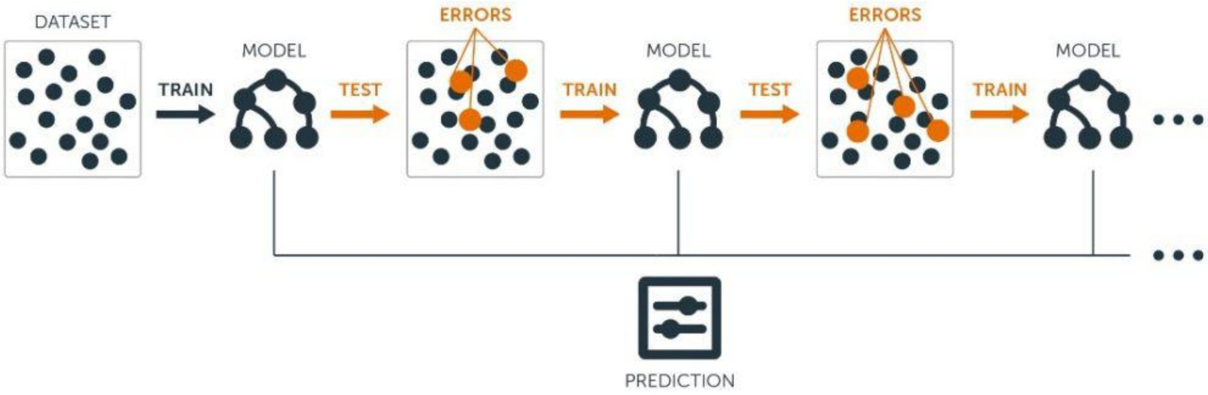
One way to tackle uncertainty is by forming a consensus between lots of models. The idea is when we are trying to make predictive models some models will be just right for the prediction point while some will overestimate or underestimate. By averaging over all the models, we can even out the overestimation and underestimation. Especially in the limit of a large number of models, we can apply the law of central tendency which states that with an increasingly large number of values the probability distribution approaches a central mean. So, if we can form a lot of models and take the average over them, we can expect that the resulting prediction is more robust than the individual prediction.

3.3 What kind of ensemble methods can be used to reduce the effects of uncertainty and improve on individual models? How do they achieve this goal?

There are various ensemble learning types:

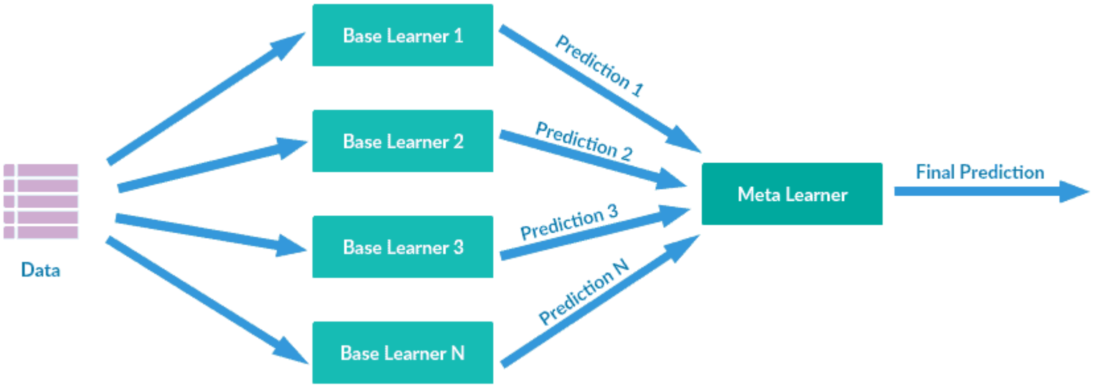
1. Sequential Ensemble Learning (Boosting)
2. Parallel Ensemble Learning (Bootstrap Aggregating => Bagging)
3. Stacking

1. Boosting uses the sequential approach. The key idea of the boosting algorithm is incrementally building an ensemble by training each new model instance to emphasize the training instances that previous models misclassified. So, one model is learning from the mistakes of another which boosts the learning.



There are many boosting algorithms, for example, AdaBoost, Stochastic Gradient Boosting, XGBoost, CatBoost, and others.

2. Stacking involves training a model (called the Meta Learner) to combine predictions of multiple other Machine learning algorithms (Base Learners). First, Base Learners are trained using the available data. Second, the Meta Learner is trained to make a final prediction using the Base Learners' predictions as the input data.



Stacking obtains better performance results than any of the individual algorithms. It can be used to successfully solve both supervised and unsupervised ML problems.

3. Bagging follows the following algorithm:
   3.1. To start with, let's assume you have some original data that you want to use as your training set (dataset D). You want to have K base models in our ensemble.

3.2. In order to promote model variance, Bagging requires training each model in the ensemble on a randomly drawn subset of the training set. The number of samples in each subset is usually as in the original dataset (for example, N), although it can be smaller.

3.3. To create each subset you need to use a bootstrapping technique:

3.3.1. First, randomly pull a sample from your original dataset D and put it to your subset

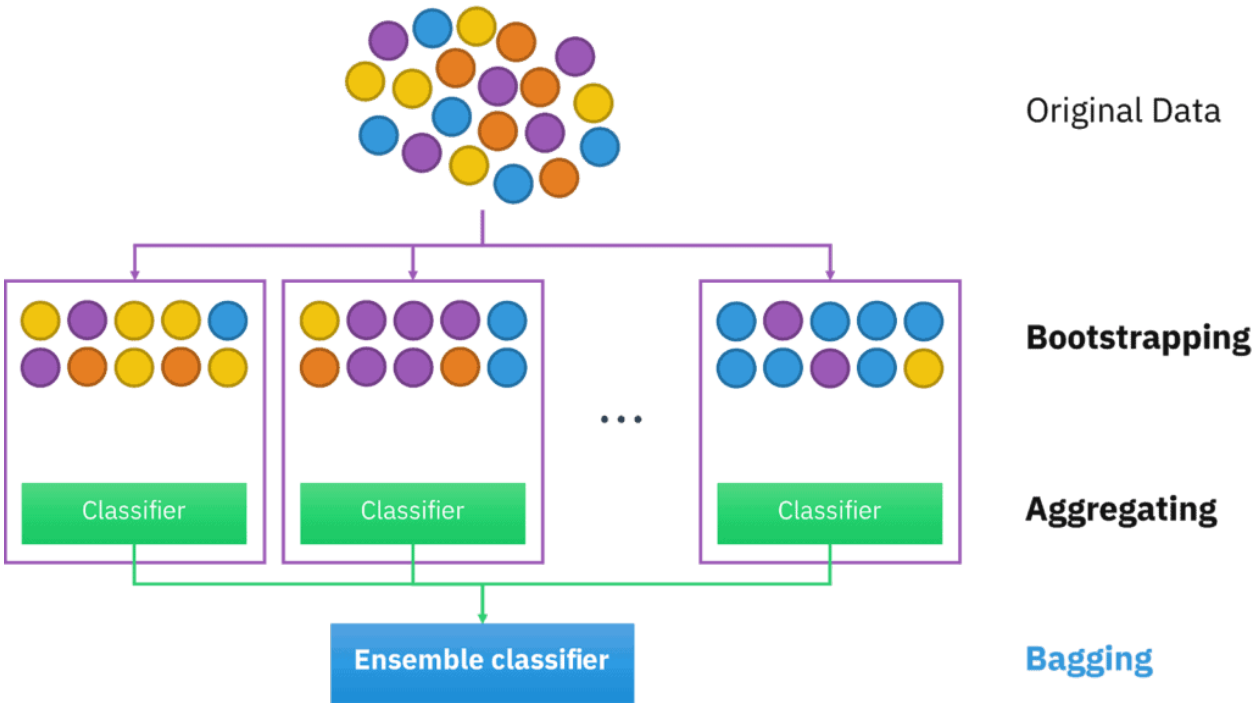
3.3.2. Second, return the sample to D (this technique is called sampling with replacement)

3.3.3. Third, perform steps 3.3.1 and 3.3.2 N (or less) times to fill your subset

3.3.4. Then perform steps 3.3.1, 3.3.2, and 3.3.3 K – 1 time to have K subsets for each of your K base models

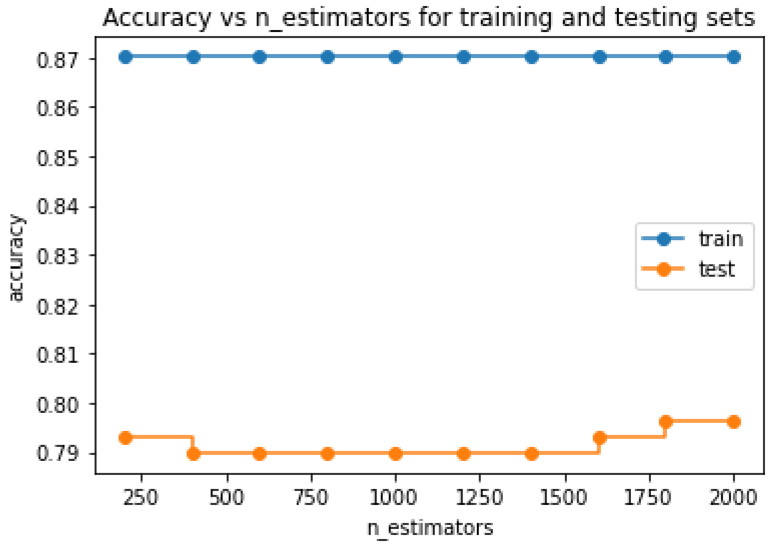
3.3.5. Build each of K base models on its subset

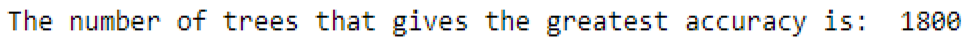
3.3.6. Combine your models and make the final prediction



If you are solving a Classification problem, you should use a voting process to determine the final result. The result is usually the most frequent class among K model predictions. In the case of Regression, you should just take the average of the K model predictions.

3.4 Construct a random forest (RF) model and apply this to the Titanic dataset. Explain how you selected the optimal number of trees and support your choice using a graph.

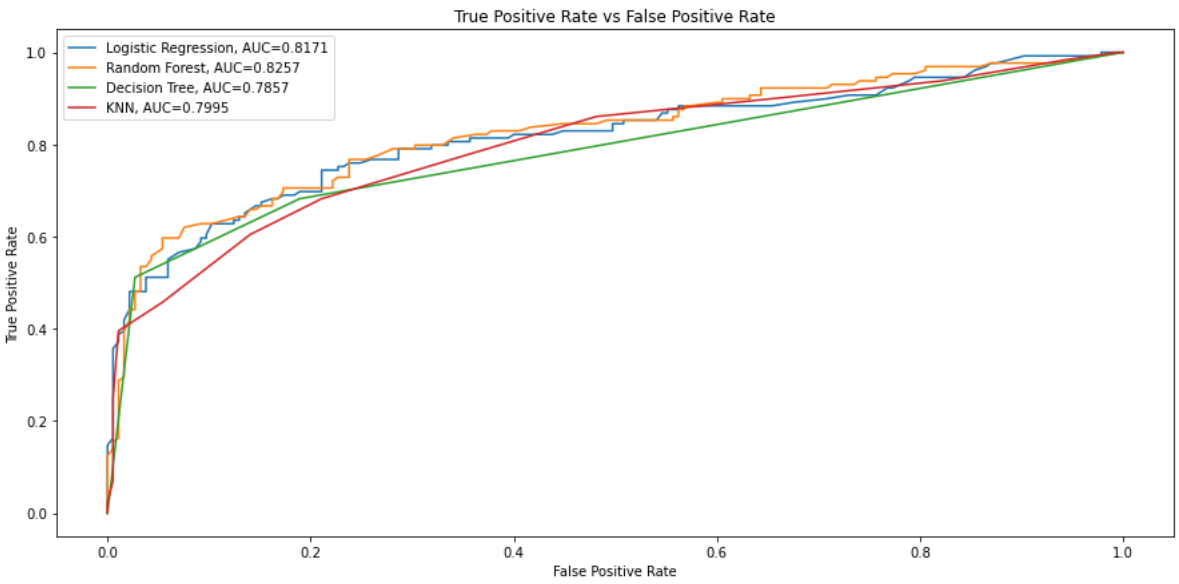




```
 The number of trees that gives the greatest accuracy is:  1800
```

I first imported the titanic dataset into a pandas dataframe. I then selected the variables sex, age and pclass as independent variables and survived as dependent variable. I then split the dataset into 70% training set and 30% testing set.
I then used the RandomForestClassifier function to create a random forest classifier model. Using a for loop, I created multiple models having 200 to 2000 trees (specified by the n_estimators property). I retrieved the accuracy score from each model and plotted it in the above graph. As you can see in the above graph, the model with 1800 trees gives the maximum accuracy.

3.5 Undertake a ROC analysis and show how the RF performs relative to the previous models (logistic regression, classification tree and KNN). Provide evidence to show as clearly as possible which model is best for classifying survival on the Titanic.



I first constructed the optimized model for each classification model type. I then used the roc_curve function to get the true positive rate and the false positive rate of each function. I plotted the true positive rate against the false positive rate. The more the curve hugs the top left corner of the plot, the better the model does at classifying the data. To quantify this, we can calculate the AUC – area under the curve – which tells us how much of the plot is located under the curve. The closer AUC is to 1, the better the model. To calculate the AUC, I used the roc_auc_score function.
As we can see from the above graph, random forest classification gives us the highest AUC (0.82), followed by logistic regression (0.81), then KNN classifier (0.79) and finally decision tree (0.78).
The best model for the classification of the titanic dataset is the random forest model.

4. Ensembles for regression

4.1 Describe the concept of a random forest (RF) regression model.

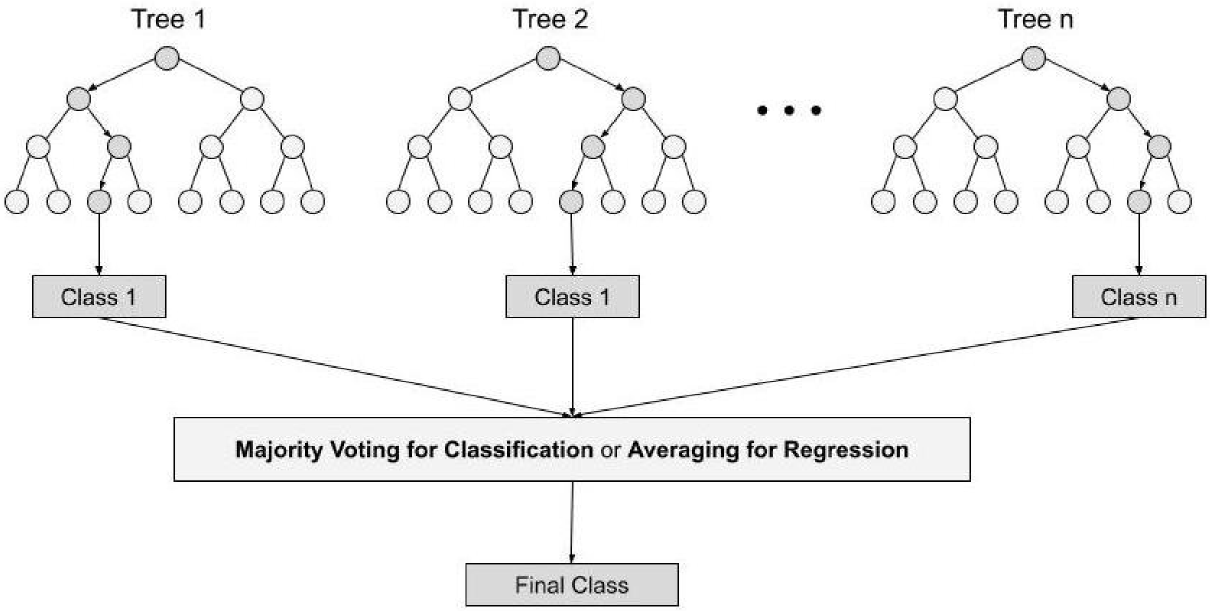
Steps involved in random forest algorithm:

Step 1: In Random forest n number of random records are taken from the data set having k number of records.

Step 2: Individual decision trees are constructed for each sample.

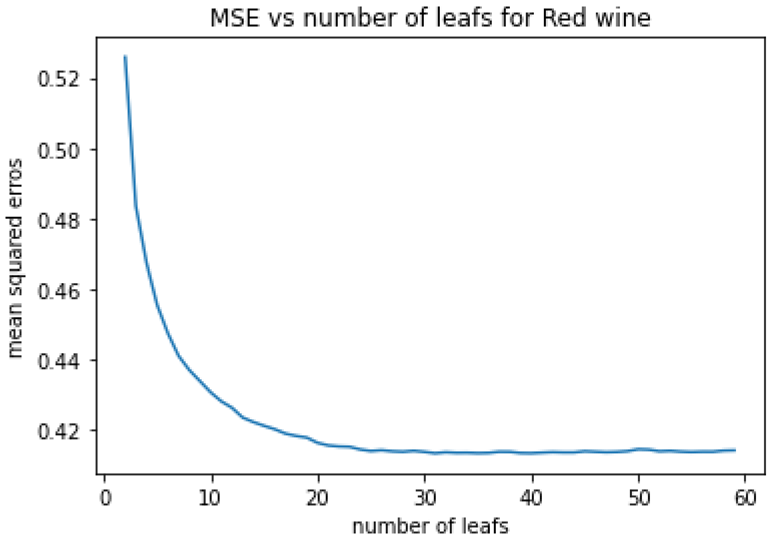
Step 3: Each decision tree will generate an output.

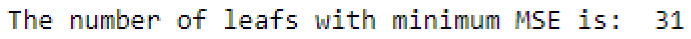
Step 4: Final output is considered based on Averaging for regression.


Tree 1
Tree 2
• • •
Tree n
Class 1
Class 1
Class n
Majority Voting for Classification or Averaging for Regression
Final Class

4.2 Construct a random forest (RF) model for the red wine dataset and show how the optimal number of leaves was estimated.

To construct the random forest regression model for the red wine dataset, I first searched for the optimal number of leaves and trees. Below, I have explained the steps used.

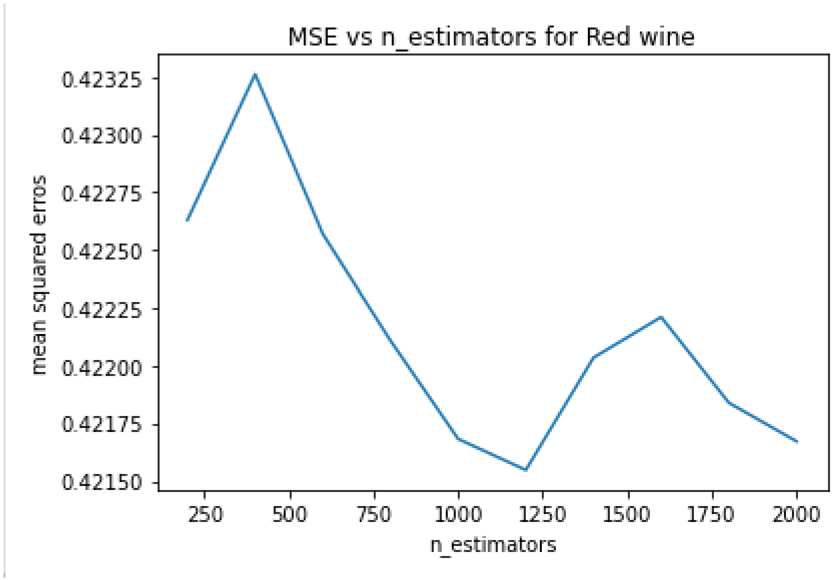


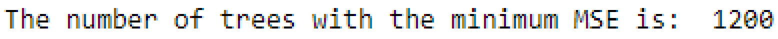


```
The number of leafs with minimum MSE is:  31
```

I first loaded the red wine dataset into a pandas dataframe. I then split the dataset into 75% training dataset and 25% testing dataset. I then used the RandomForestRegressor function to create a model and cross_val_score to get the MSE (mean squared error) of the model. In the RandomForestRegressor function, I used the max_leaf_nodes property to create models which had 2 to 60 maximum leaf nodes. I then plotted the MSE of each model. From the above graph we can see that the number of leaves with the minimum MSE is 31.

4.3 Explain and show how the optimal number of trees was computed.



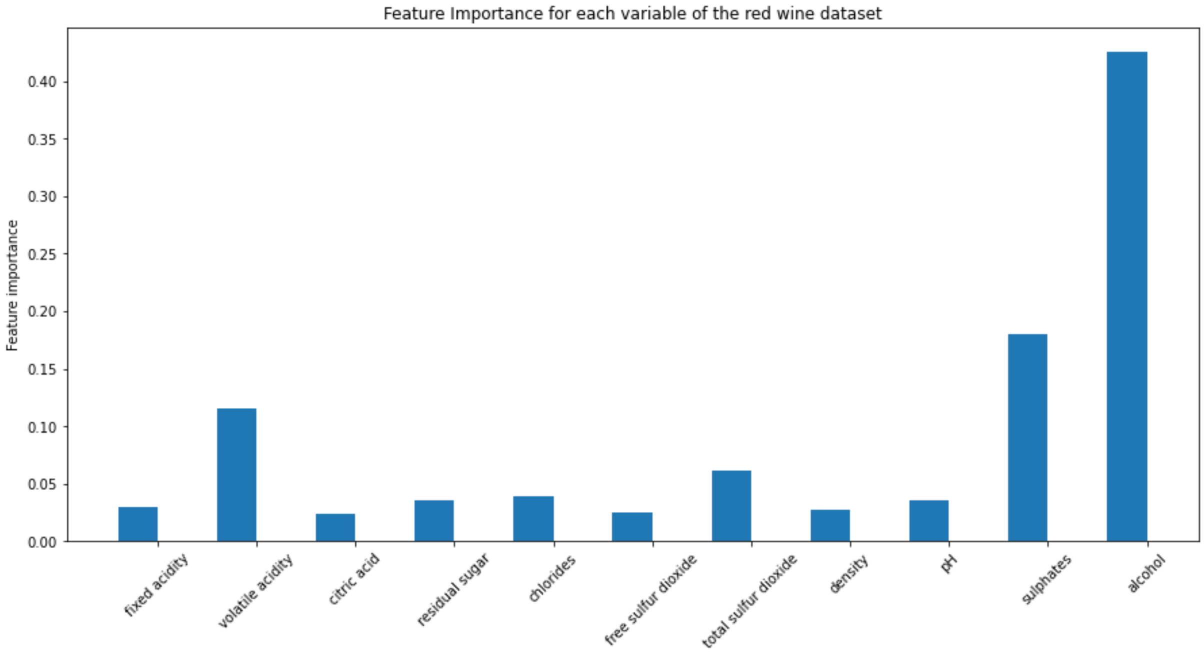


```
The number of trees with the minimum MSE is:  1200
```

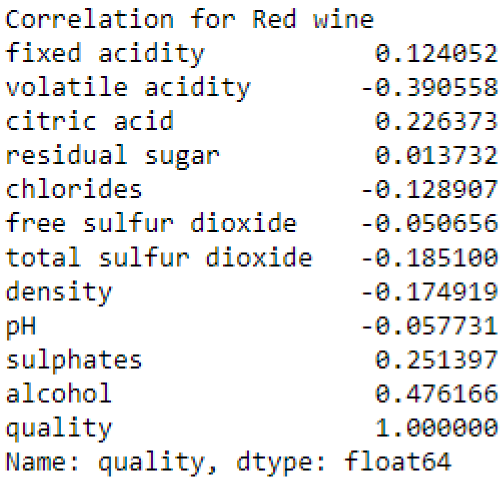
To compute the optimal number of trees, I used the RandomForestRegressor. Using a for loop, I created multiple models having trees from 200 to 2000 by incrementing 10 at the tree number for each iteration. I used the cross_val_score to get the MSE of each model. I then plotted the above graph. As you can see, on the above graph, the number of trees with the minimum MSE is 1200.

I then created the optimized RandomForestRegressor with 1200 trees and a maximum of 31 leaf nodes.

4.4 Provide a bar graph showing the importance of each feature and compare this with the results from Assignment 6 (using correlation and LASSO).

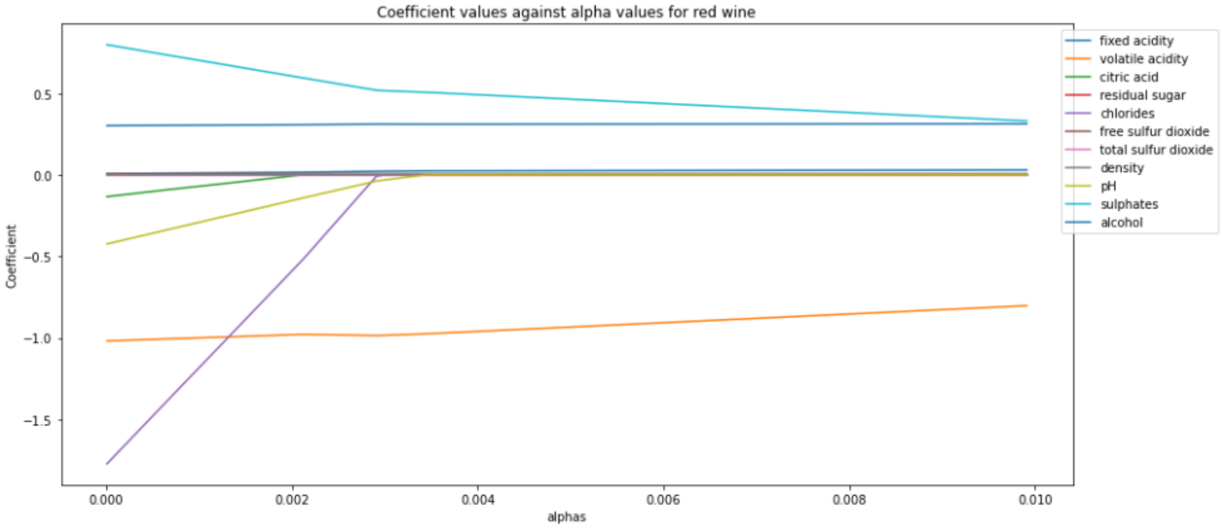


I used the optimized random forest regression model to get the importance of each feature using its feature_importances_ property. I then plotted the above bar graph. The graph shows that the 3 most important features are alcohol (0.42), sulphates (0.18) and volatile acidity (0.11).



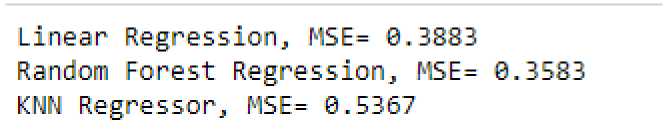
```
Correlation for Red wine
fixed acidity           0.124052
volatile acidity       -0.390558
citric acid             0.226373
residual sugar          0.013732
chlorides              -0.128907
free sulfur dioxide    -0.050656
total sulfur dioxide   -0.185100
density                -0.174919
pH                     -0.057731
sulphates               0.251397
alcohol                 0.476166
quality                 1.000000
Name: quality, dtype: float64
```

The above table is the correlation for the red wine dataset from assignment 6. We can see that alcohol has the highest correlation with 0.47 then comes sulphates with 0.25 correlation and then we have volatile acidity with -0.39 correlation.



The above graph shows the coefficient values against alpha for red wine's lasso regression. We can see that as alpha increases, the coefficient of more and more features gets close to 0. At the end, the three features that are left are alcohol (in dark blue), sulphates (in light blue) and volatile acidity (in orange).

4.5 What is the performance of the RF model and compare it with the linear regression and KNN models constructed during Assignment 6. Present sufficient information to support your conclusion about the best model for the red wine dataset.



```
Linear Regression, MSE= 0.3883
Random Forest Regression, MSE= 0.3583
KNN Regressor, MSE= 0.5367
```

To compare the different models, I first created each model, fitted it using the training dataset and predicted using the testing dataset. I then calculated the MSE of each model using the mean_squared_error function. I obtained the above values. We can see that the random forest regression model has the least MSE (0.35) then comes the linear regression model with 0.38 and finally the KNN regression model with 0.53. The best model for the red wine dataset is the random forest regression model.

References

1. Principal Component Analysis in Machine Learning | Simplilearn
2. PCA : the math - step-by-step with a simple example - YouTube

3. Flat and Hierarchical Clustering | The Dendrogram Explained - YouTube
4. Hierarchical Agglomerative Clustering [HAC - Single Link] - YouTube
5. A Quick Overview of Model Averaging Approaches | by Francois Chabannes | The Startup | Medium
6. Random Forest | Introduction to Random Forest Algorithm (analyticsvidhya.com)
7. Using Principal Component Analysis (PCA) for Machine Learning | by Wei-Meng Lee | Towards Data Science
8. ML | Principal Component Analysis(PCA) - GeeksforGeeks
9. 5-Minute Machine Learning. Principal component analysis (PCA) is… | by Andre Violante | Towards Data Science
10. Maths behind PCA | Maths behind Principal Component Analysis (analyticsvidhya.com)
11. Mathematical Approach to PCA - GeeksforGeeks
12. The Math of Principal Component Analysis (PCA) | by adam dhalla | Analytics Vidhya | Medium
13. How does PCA work? Maths behind PCA | Analytics Vidhya (medium.com)
14. Principal Component Analysis(PCA) | Guide to PCA (analyticsvidhya.com)

15. What is Dendrogram? (visual-paradigm.com)
16. What is a Dendrogram? - Hierarchical Cluster Analysis - Displayr
17. How to vectorize pairwise (dis)similarity metrics | by Ben Cook | Towards Data Science
18. python - Dendrogram through scipy given a similarity matrix - Stack Overflow

19. Modeling Fundamentals: Understanding Uncertainty | AIR Worldwide (air-worldwide.com)
20. Model Averaging: A Robust Way to Deal with Model Uncertainty | by Osman Mamun | Towards Data Science
21. Model averaging methods: how and why to build ensemble models | by Michael Mahoney | Towards Data Science
22. A Quick Overview of Model Averaging Approaches | by Francois Chabannes | The Startup | Medium

23. Model Averaging: A Robust Way to Deal with Model Uncertainty | by Osman Mamun | Towards Data Science
24. Model averaging methods: how and why to build ensemble models | by Michael Mahoney | Towards Data Science
25. A Quick Overview of Model Averaging Approaches | by Francois Chabannes | The Startup | Medium

26. Uncertainty Quantification Part 1: Ensemble Methods — Nicholas A. Rossi (rossidata.com)
27. 12. Using model ensembles to reduce uncertainty – Geophysical Fluid Dynamics Laboratory (noaa.gov)

28. What is Random Forest? | IBM
29. Random Forest | Introduction to Random Forest Algorithm (analyticsvidhya.com)
30. Random Forest Regression in Python - GeeksforGeeks

31. How to Use ROC Curves and Precision-Recall Curves for Classification in Python - MachineLearningMastery.com
32. How to Plot a ROC Curve in Python (Step-by-Step) - Statology

33. Random Forest Regression in Python - GeeksforGeeks

34. How to Plot a ROC Curve in Python (Step-by-Step) - Statology
35. How to Plot Multiple ROC Curves in Python (With Example) - Statology
36. Sklearn Random Forest Classifiers in Python Tutorial | DataCamp

37. Random Forest Regression in Python - GeeksforGeeks

38. Random Forest Regression - The Definitive Guide | cnvrg.io