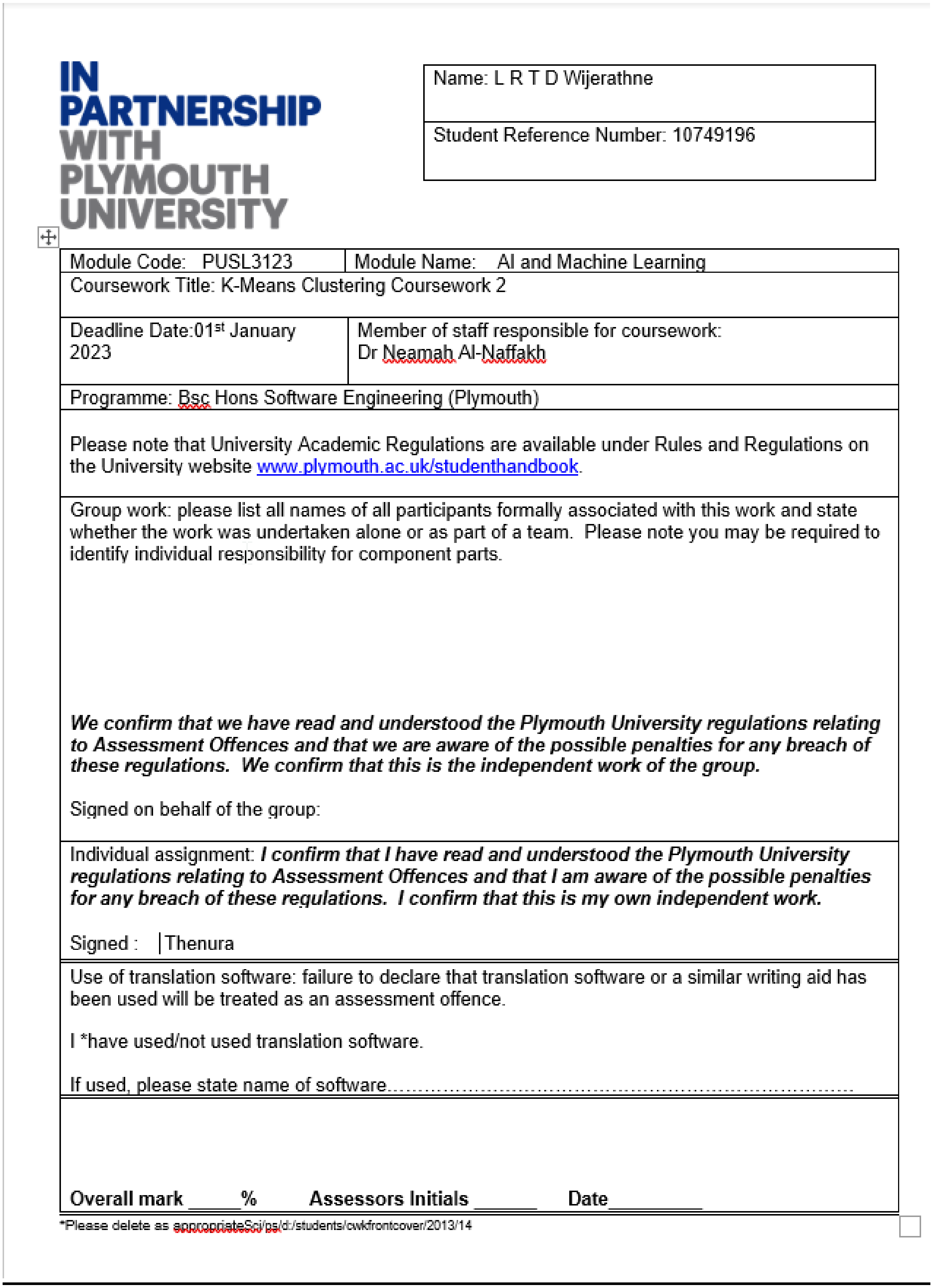


IN PARTNERSHIP WITH PLYMOUTH UNIVERSITY

Name: L R T D Wijerathne
Student Reference Number: 10749196

Module Code: PUSL3123	Module Name: AI and Machine Learning
Coursework Title: K-Means Clustering Coursework 2	
Deadline Date:01st January 2023	Member of staff responsible for coursework: Dr Neamah Al-Naffakh
Programme: Bsc Hons Software Engineering (Plymouth)	

Please note that University Academic Regulations are available under Rules and Regulations on the University website www.plymouth.ac.uk/studenthandbook.

Group work: please list all names of all participants formally associated with this work and state whether the work was undertaken alone or as part of a team. Please note you may be required to identify individual responsibility for component parts.

We confirm that we have read and understood the Plymouth University regulations relating to Assessment Offences and that we are aware of the possible penalties for any breach of these regulations. We confirm that this is the independent work of the group.

Signed on behalf of the group:

Individual assignment: ***I confirm that I have read and understood the Plymouth University regulations relating to Assessment Offences and that I am aware of the possible penalties for any breach of these regulations. I confirm that this is my own independent work.***

Signed : Thenura

Use of translation software: failure to declare that translation software or a similar writing aid has been used will be treated as an assessment offence.

I *have used/not used translation software.

If used, please state name of software...

Overall mark _____% Assessors Initials ______ Date_________

*Please delete as appropriateSci/ps/d:/students/cwkfrontcover/2013/14

Contents

Introduction

Machine learning is a method of teaching computers to learn from data, without being explicitly programmed. It is a subset of artificial intelligence that uses statistical techniques to enable computers to learn from data and make predictions or decisions without human intervention. (*What is Machine Learning? Concepts & Examples - Data Analytics*, no date)

Unsupervised learning is a type of machine learning model where an algorithm is trained on an unlabeled dataset, where the desired output is not known. The goal of unsupervised learning is to discover hidden patterns or structure in the data. Unlike supervised learning, unsupervised learning does not use labeled data, and instead the algorithm must find a way to infer the underlying structure of the data on its own.

K-means clustering is a method of unsupervised learning, which is a technique for grouping similar data points together. The final deliverable of this coursework is the implementation of K-means clustering algorithm on the student's unique id to generate a personal data matrix with different results depending on the id.(*KMeans Explained*, no date)

Along with it is an overview of machine learning, unsupervised learning and K-means clustering technique with the explanation of the implementation of this technique in MATLAB.

Overview

Unsupervised Machine Learning

Unsupervised learning is a type of machine learning where an algorithm is trained on an unlabeled dataset, where the desired output is not known. The goal of unsupervised learning is to discover hidden patterns or structure in the data. Unlike supervised learning, unsupervised learning does not use labeled data, and instead the algorithm must find a way to infer the underlying structure of the data on its own.(*What is Machine Learning? Definition, Types, Applications*, no date)

There are several types of unsupervised learning, such as clustering, dimensionality reduction, and anomaly detection.

Clustering is the task of grouping similar data points together. Common clustering algorithms include k-means and hierarchical clustering. Clustering can be used for tasks such as market segmentation and image segmentation.(*Clustering in Machine Learning - Javatpoint*, no date)

Dimensionality reduction techniques are used to reduce the number of features in the dataset while preserving as much information as possible. Common dimensionality reduction techniques include principal component analysis (PCA) and t-Distributed Stochastic Neighbor Embedding (t-SNE).(*Exploring Clustering Algorithms: Explanation and Use Cases - neptune.ai*, no date a)

Anomaly detection is the task of identifying data points that are unusual or do not conform to the general pattern in the dataset. This can be useful for detecting fraud or identifying defective items in a manufacturing process.

Unsupervised learning is useful in a variety of applications, such as natural language processing market segmentation, image compression and anomaly detection in network intrusion detection system. Unsupervised learning algorithms are typically more complex than supervised learning algorithms and can require more computational resources, but they can be more powerful when it comes to discovering hidden patterns in the data. Often, unsupervised learning models are evaluated using visualization or quantitative measures such as silhouette score, or Inertia.

K-means Algorithm

K-means clustering is a method of clustering, which is a technique for grouping similar data points together. The goal of the algorithm is to partition a set of data points into K clusters, where each cluster is defined as a group of points that are similar to each other and dissimilar to points in other clusters.(*Exploring Clustering Algorithms: Explanation and Use Cases - neptune.ai*, no date b)

The algorithm works by first initializing K cluster centroids, where K is the number of clusters desired. This is typically done by randomly selecting K data points to serve as the centroids. Then, each data point is assigned to the cluster associated with the nearest centroid. Once all data points have been assigned, the cluster centroid is recomputed as the mean of all the data points assigned to that cluster. This process is repeated until the cluster assignments no longer change or a maximum number of iterations is reached.

K-means is a popular and widely used algorithm due to its simplicity and efficiency. However, it has some limitations, one of the main disadvantages is that it's sensitive to initial centroid position and it can stuck in local minimum. It also assumes that the clusters are spherical and equally sized, which may not be the case for all datasets. Additionally, the algorithm requires the number of clusters to be specified in advance, which can be difficult to determine.

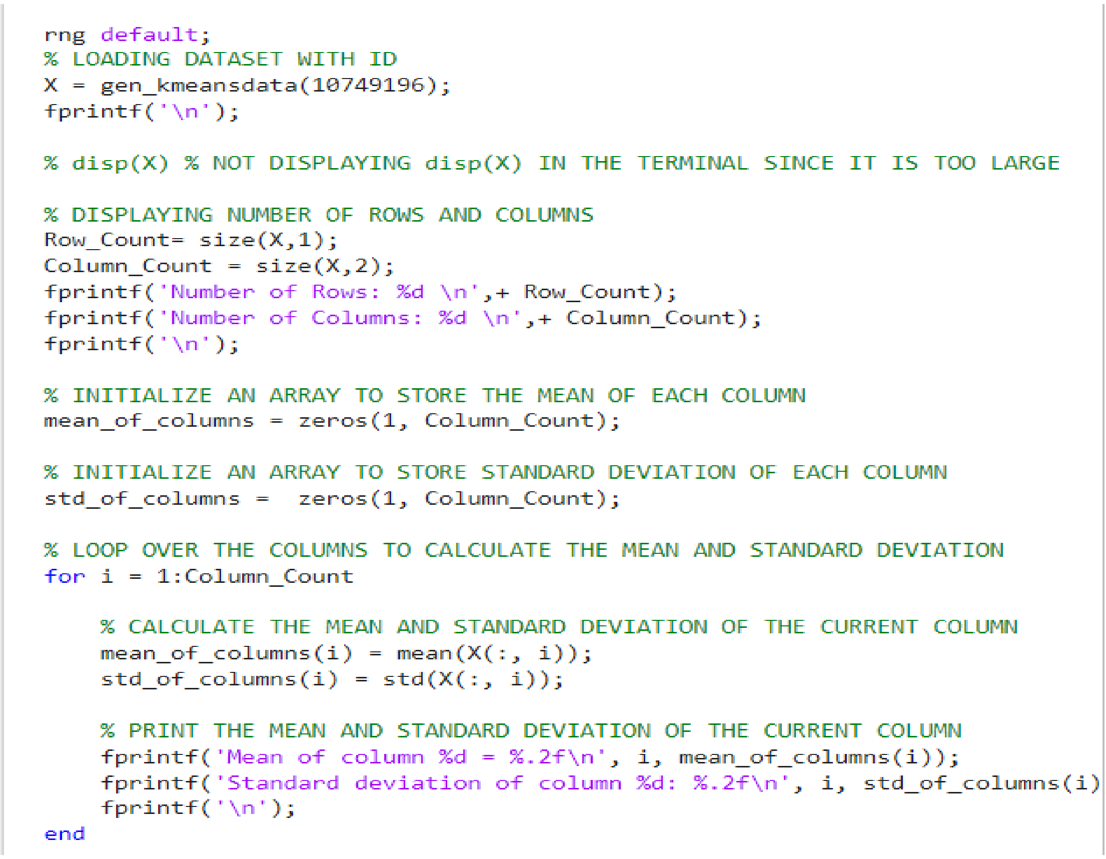
There are also variations of K-means, such as K-means++ which is a more sophisticated initialization method that addresses the issue of getting stuck in local minima. Another variation is the fuzzy K-means algorithm, which allows data points to belong to more than one cluster with varying degrees of membership.(*Hierarchical Clustering in Machine Learning - Javatpoint*, no date)

K-means is widely used in many applications, such as image compression, image segmentation, and customer segmentation in marketing. It also can be used as a preprocessing step for other machine learning algorithms, such as dimensionality reduction and anomaly detection.

Implementation

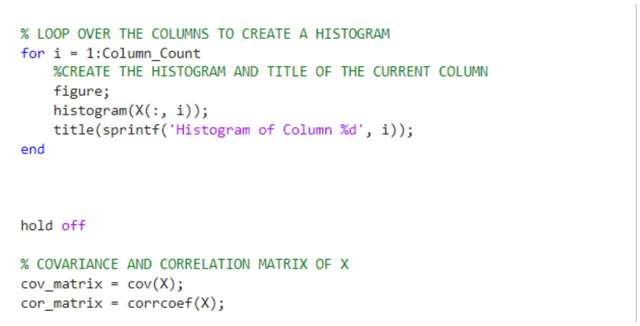
K-means clustering is a method of vector quantization, originally from signal processing, that aims to partition n observations into k clusters in which each observation belongs to the cluster with the nearest mean.

To perform K-means Clustering an initial K centroids should be selected for a K number of clusters randomly from the data points. Next each data point should be assigned to the cluster whose centroid is closest. Then it is required to compute a new centroid for each cluster by taking the mean of all the points assigned to that specific cluster. The second and third step should be repeated for a desired number of times or until the centroids can no longer move. Finally, the final assignments of the data points should be returned to the clusters. The explanation of how this process was achieved in the coursework and the how the other requirements were fulfilled will be explained briefly below with code samples.



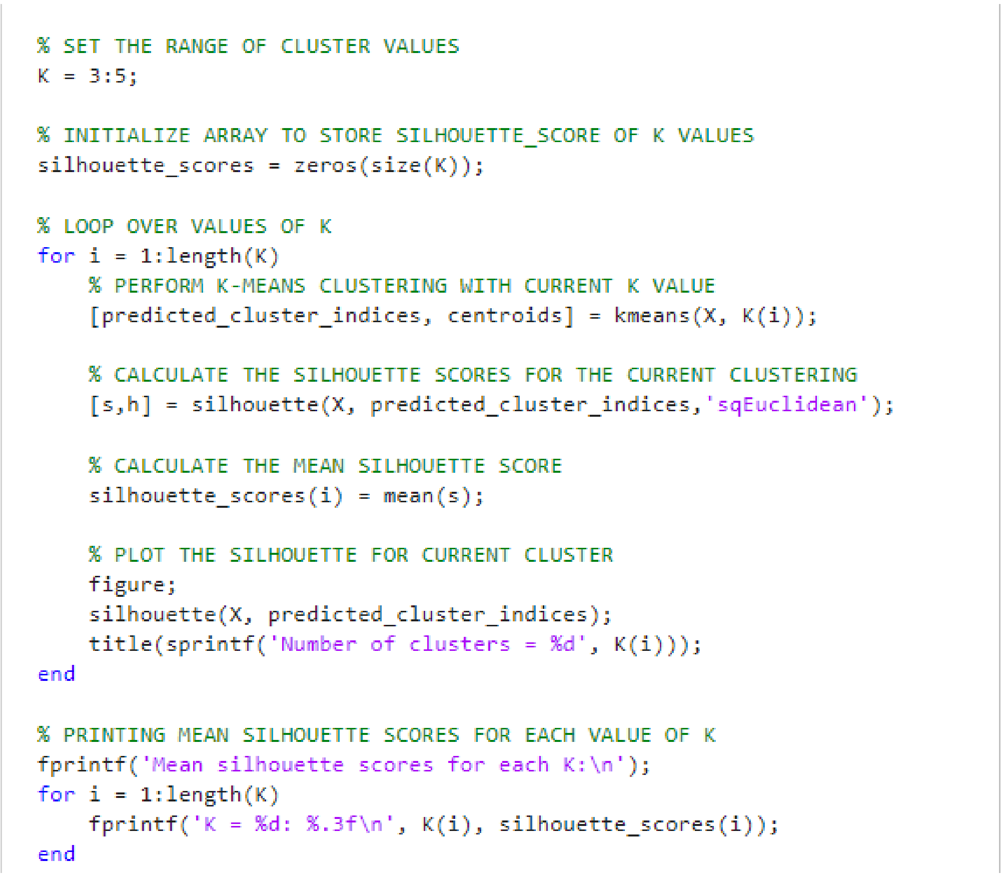
```
rng default;
% LOADING DATASET WITH ID
X = gen_kmeansdata(10749196);
fprintf('\n');

% disp(X) % NOT DISPLAYING disp(X) IN THE TERMINAL SINCE IT IS TOO LARGE

% DISPLAYING NUMBER OF ROWS AND COLUMNS
Row_Count= size(X,1);
Column_Count = size(X,2);
fprintf('Number of Rows: %d \n',+ Row_Count);
fprintf('Number of Columns: %d \n',+ Column_Count);
fprintf('\n');

% INITIALIZE AN ARRAY TO STORE THE MEAN OF EACH COLUMN
mean_of_columns = zeros(1, Column_Count);

% INITIALIZE AN ARRAY TO STORE STANDARD DEVIATION OF EACH COLUMN
std_of_columns =  zeros(1, Column_Count);

% LOOP OVER THE COLUMNS TO CALCULATE THE MEAN AND STANDARD DEVIATION
for i = 1:Column_Count

    % CALCULATE THE MEAN AND STANDARD DEVIATION OF THE CURRENT COLUMN
    mean_of_columns(i) = mean(X(:, i));
    std_of_columns(i) = std(X(:, i));

    % PRINT THE MEAN AND STANDARD DEVIATION OF THE CURRENT COLUMN
    fprintf('Mean of column %d = %.2f\n', i, mean_of_columns(i));
    fprintf('Standard deviation of column %d: %.2f\n', i, std_of_columns(i));
    fprintf('\n');
end
```

Initially the dataset is loaded with the student ID and assigned to X and then the Row count and Column count is displayed. Then two arrays are initialized to store the mean of each column and standard deviation of each column. Finally mean, standard deviation displayed in terminal respectively using a for loop.



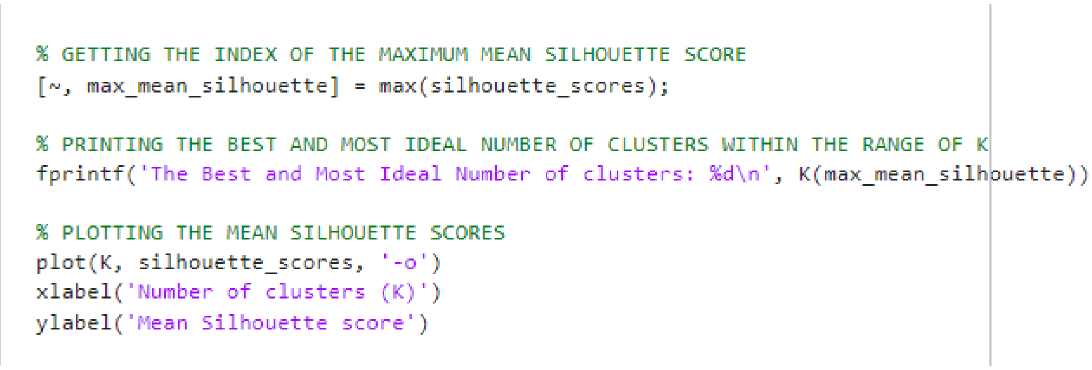
```
% LOOP OVER THE COLUMNS TO CREATE A HISTOGRAM
for i = 1:Column_Count
    %CREATE THE HISTOGRAM AND TITLE OF THE CURRENT COLUMN
    figure;
    histogram(X(:, i));
    title(sprintf('Histogram of Column %d', i));
end


hold off

% COVARIANCE AND CORRELATION MATRIX OF X
cov_matrix = cov(X);
cor_matrix = corrcoef(X);
```

Next the histogram of each column is displayed using a loop, Covariance and the Correlation Matrices of the data set X is set as cov_matrix and cor_matrix respectively to be stored in the workspace and the variable K which is the range of number of Clusters that we desire is also defined. The covariance and correlation matrices are stored in the workspace and can be viewed if required.



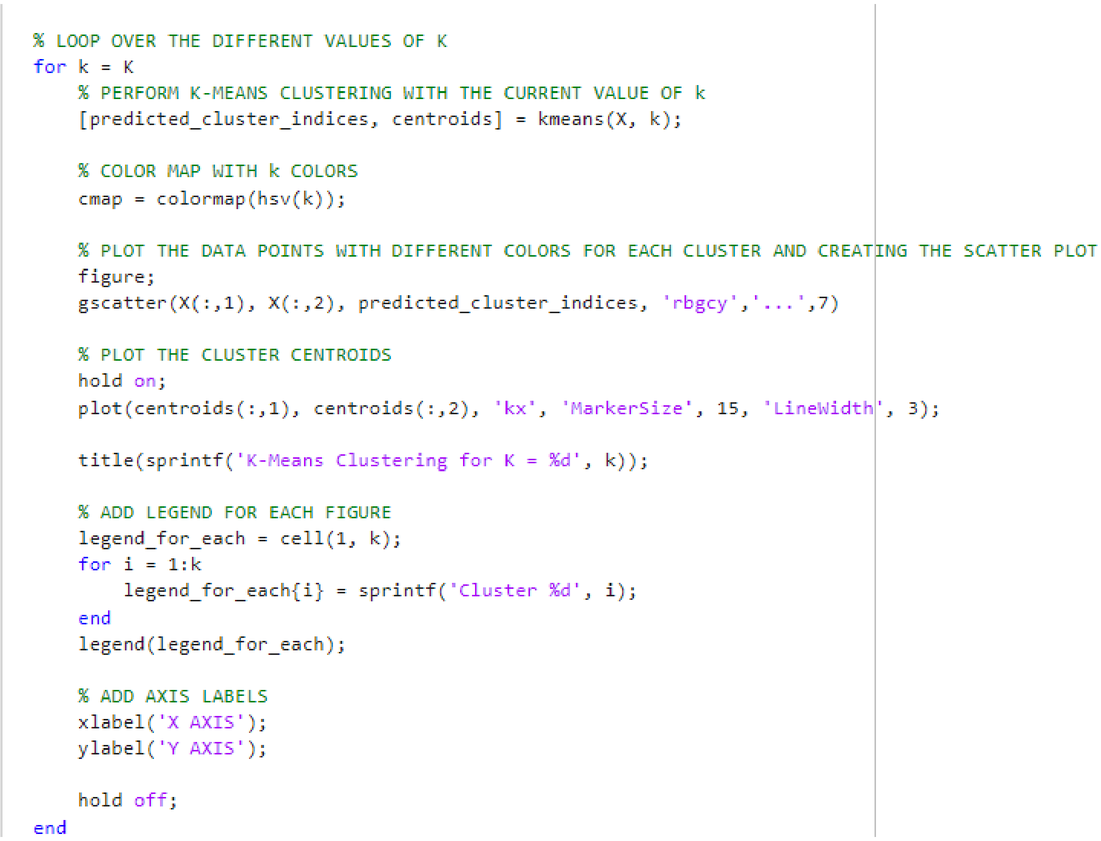
```
% SET THE RANGE OF CLUSTER VALUES
K = 3:5;

% INITIALIZE ARRAY TO STORE SILHOUETTE_SCORE OF K VALUES
silhouette_scores = zeros(size(K));

% LOOP OVER VALUES OF K
for i = 1:length(K)
    % PERFORM K-MEANS CLUSTERING WITH CURRENT K VALUE
    [predicted_cluster_indices, centroids] = kmeans(X, K(i));

    % CALCULATE THE SILHOUETTE SCORES FOR THE CURRENT CLUSTERING
    [s,h] = silhouette(X, predicted_cluster_indices,'sqEuclidean');

    % CALCULATE THE MEAN SILHOUETTE SCORE
    silhouette_scores(i) = mean(s);

    % PLOT THE SILHOUETTE FOR CURRENT CLUSTER
    figure;
    silhouette(X, predicted_cluster_indices);
    title(sprintf('Number of clusters = %d', K(i)));
end

% PRINTING MEAN SILHOUETTE SCORES FOR EACH VALUE OF K
fprintf('Mean silhouette scores for each K:\n');
for i = 1:length(K)
    fprintf('K = %d: %.3f\n', K(i), silhouette_scores(i));
end
```

In this segment a range of values (clusters 3:5) is defined as K and K-means clustering is done with the defined K, the mean silhouette score of the K Cluster is entered into an array and the silhouette plot is generated. The silhouette will be generated from the predicted_cluster_indices which was produced as a result due to the K-means clustering and the data set. This block will repeat itself with the above for each instance K.



```
% GETTING THE INDEX OF THE MAXIMUM MEAN SILHOUETTE SCORE
[~, max_mean_silhouette] = max(silhouette_scores);

% PRINTING THE BEST AND MOST IDEAL NUMBER OF CLUSTERS WITHIN THE RANGE OF K
fprintf('The Best and Most Ideal Number of clusters: %d\n', K(max_mean_silhouette))

% PLOTTING THE MEAN SILHOUETTE SCORES
plot(K, silhouette_scores, '-o')
xlabel('Number of clusters (K)')
ylabel('Mean Silhouette score')
```

Next the max or the best mean silhouette score of the mean silhouette scores array generated by different K values (Cluster numbers) is obtained and will be printed on the terminal along with the plot of Mean Silhouette Scores of all K values.



```
% LOOP OVER THE DIFFERENT VALUES OF K
for k = K
    % PERFORM K-MEANS CLUSTERING WITH THE CURRENT VALUE OF k
    [predicted_cluster_indices, centroids] = kmeans(X, k);

    % COLOR MAP WITH k COLORS
    cmap = colormap(hsv(k));

    % PLOT THE DATA POINTS WITH DIFFERENT COLORS FOR EACH CLUSTER AND CREATING THE SCATTER PLOT
    figure;
    gscatter(X(:,1), X(:,2), predicted_cluster_indices, 'rbgcy','...',7)

    % PLOT THE CLUSTER CENTROIDS
    hold on;
    plot(centroids(:,1), centroids(:,2), 'kx', 'MarkerSize', 15, 'LineWidth', 3);

    title(sprintf('K-Means Clustering for K = %d', k));

    % ADD LEGEND FOR EACH FIGURE
    legend_for_each = cell(1, k);
    for i = 1:k
        legend_for_each{i} = sprintf('Cluster %d', i);
    end
    legend(legend_for_each);

    % ADD AXIS LABELS
    xlabel('X AXIS');
    ylabel('Y AXIS');

    hold off;
end
```

Finally, a for loop is initiated to perform K-means clustering and obtain color maps to generate the scatter plot with its title, legend and axis for each value of K.

The result will include eleven figures with four histograms for each column , three mean silhouette score figures for each of K , one silhouette score plot plotting the three mean silhouette scores of the range of K and three scatter plots for each cluster. Furthermore mean and standard deviation of each column, Mean silhouette score for each of K and the max index of mean silhouette scores will be displayed on the terminal. Along with the covariance and correlation matrices being stored in the workspace. All of this will be displayed on the appendix.

Discussions

- Stopping Criteria for K-means Clustering

  The stopping criteria for k-means clustering are the conditions that determine when the algorithm should stop iterating. There are several ways to define the stopping criteria:

  Maximum number of iterations: This is the most straightforward method. The algorithm is set to run for a fixed number of iterations, and stops after that, regardless of the cluster assignments. This method is simple to implement but may not be optimal as the algorithm may stop before the cluster assignments have stabilized.

  Convergence: This method is based on the idea that the algorithm has converged when the cluster assignments no longer change. The algorithm stops iterating when the cluster assignments for all data points remain the same for a given number of consecutive iterations, or when the changes in cluster assignments are below a certain threshold. This method is more sensitive to the specific characteristics of the dataset and can lead to more accurate cluster assignments.

  Improvement in cost function: This method is based on the improvement of the cost function, such as the sum of squared distances between points and their cluster centroid. The algorithm stops when the cost function is not improving enough or when it reaches a certain threshold.

  It is important to note that the choice of stopping criteria will depend on the specific requirements of the application and the characteristics of the dataset. In some cases, a combination of these criteria may be used for more robust results. In addition, stopping criteria must be chosen carefully, as it can impact the performance of the algorithm and the outcome.

- Drawbacks of K-means clustering

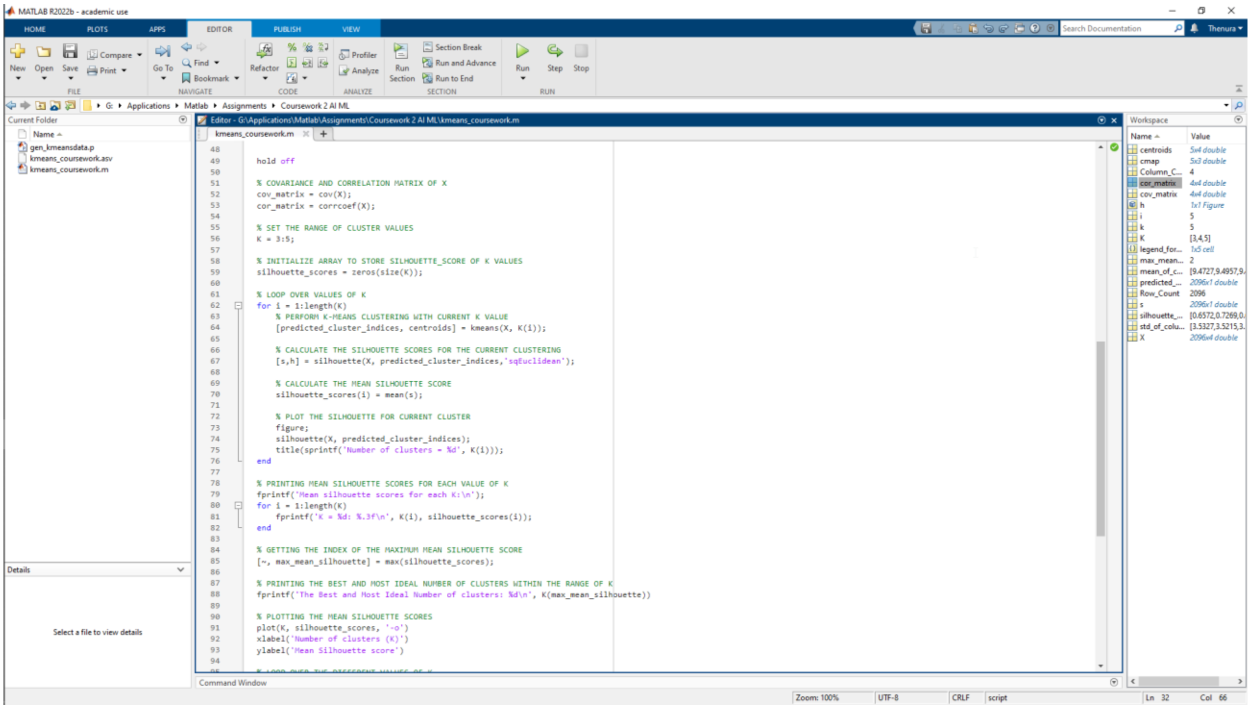
  There are quite a few drawbacks of clustering which makes this algorithm at a look. One of the major drawbacks is the lack of optimal set of clusters and needs to be decided before.

  Another problem is the lack of consistency which could be seen due to the choice of the order in which data is read which caused the final results to change.

  Furthermore K-means handles only numerical data which can be seen why sprint() was used in quite a lot of places in the implementation part. It is also quite difficult to predict the k-values for the algorithm.

Appendix

Code

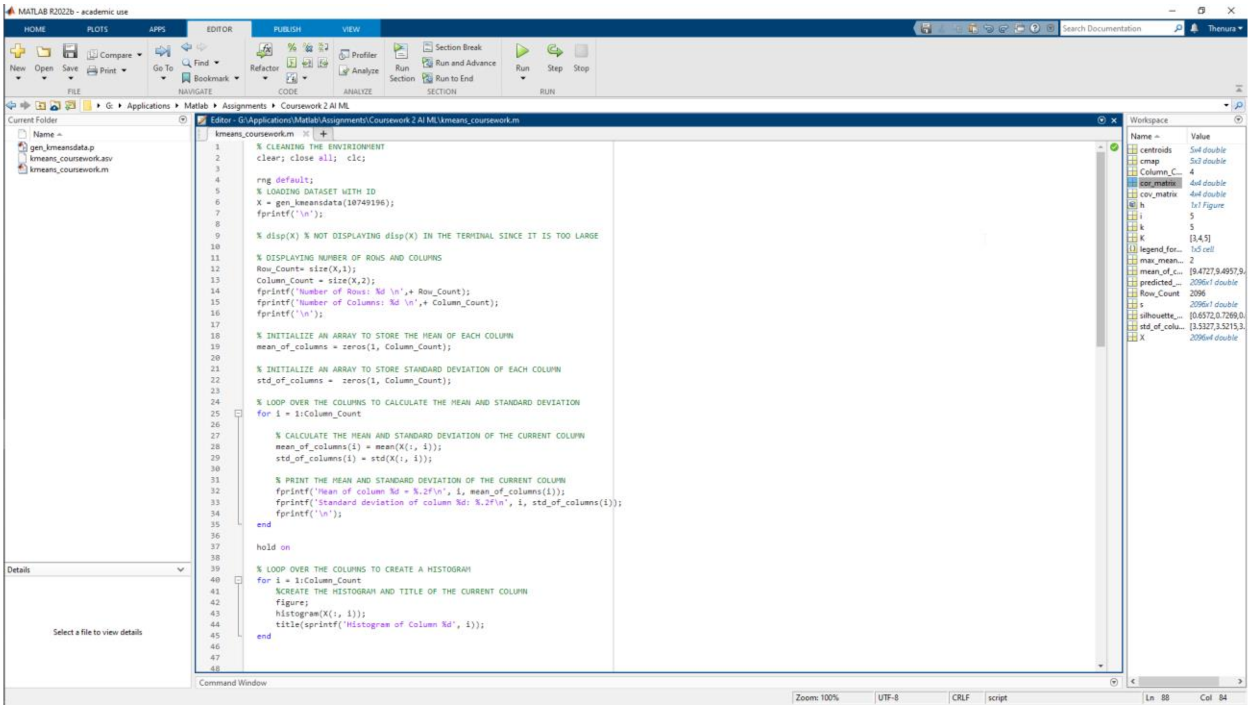


```matlab

hold off

% COVARIANCE AND CORRELATION MATRIX OF X
cov_matrix = cov(X);
cor_matrix = corrcoef(X);

% SET THE RANGE OF CLUSTER VALUES
K = 3:5;

% INITIALIZE ARRAY TO STORE SILHOUETTE_SCORE OF K VALUES
silhouette_scores = zeros(size(K));

% LOOP OVER VALUES OF K
for i = 1:length(K)
    % PERFORM K-MEANS CLUSTERING WITH CURRENT K VALUE
    [predicted_cluster_indices, centroids] = kmeans(X, K(i));

    % CALCULATE THE SILHOUETTE SCORES FOR THE CURRENT CLUSTERING
    [s,h] = silhouette(X, predicted_cluster_indices,'sqEuclidean');

    % CALCULATE THE MEAN SILHOUETTE SCORE
    silhouette_scores(i) = mean(s);

    % PLOT THE SILHOUETTE FOR CURRENT CLUSTER
    figure;
    silhouette(X, predicted_cluster_indices);
    title(sprintf('Number of clusters = %d', K(i)));
end

% PRINTING MEAN SILHOUETTE SCORES FOR EACH VALUE OF K
fprintf('Mean silhouette scores for each K:\n');
for i = 1:length(K)
    fprintf('K = %d: %.3f\n', K(i), silhouette_scores(i));
end

% GETTING THE INDEX OF THE MAXIMUM MEAN SILHOUETTE SCORE
[~, max_mean_silhouette] = max(silhouette_scores);

% PRINTING THE BEST AND MOST IDEAL NUMBER OF CLUSTERS WITHIN THE RANGE OF K
fprintf('The Best and Most Ideal Number of clusters: %d\n', K(max_mean_silhouette))

% PLOTTING THE MEAN SILHOUETTE SCORES
plot(K, silhouette_scores, '-o')
xlabel('Number of clusters (K)')
ylabel('Mean Silhouette score')

```

Workspace

Name	Value
centroids	5x4 double
cmap	5x3 double
Column_C...	4
cor_matrix	4x4 double
cov_matrix	4x4 double
h	1x1 Figure
i	5
k	5
K	[3,4,5]
legend_for...	1x5 cell
max_mean...	2
mean_of_c...	[9.4727,9.4957,9.
predicted_...	2096x1 double
Row_Count	2096
s	2096x1 double
silhouette_...	[0.6572,0.7269,0.
std_of_colu...	[3.5327,3.5215,3.
X	2096x4 double

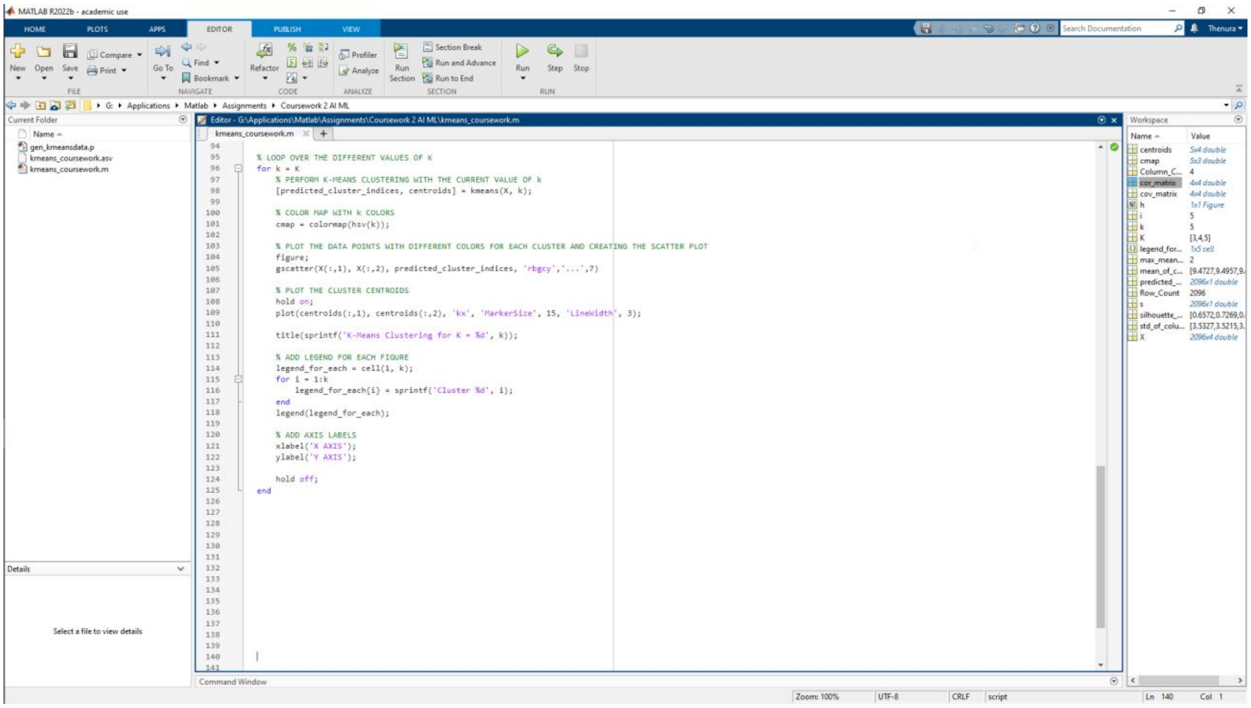
Notice that I have commented the disp(X) just for the screenshots since it displays a long result.



```matlab
% CLEANING THE ENVIRIONMENT
clear; close all;  clc;

rng default;
% LOADING DATASET WITH ID
X = gen_kmeansdata(10749196);
fprintf('\n');

% disp(X) % NOT DISPLAYING disp(X) IN THE TERMINAL SINCE IT IS TOO LARGE

% DISPLAYING NUMBER OF ROWS AND COLUMNS
Row_Count= size(X,1);
Column_Count = size(X,2);
fprintf('Number of Rows: %d \n',+ Row_Count);
fprintf('Number of Columns: %d \n',+ Column_Count);
fprintf('\n');

% INITIALIZE AN ARRAY TO STORE THE MEAN OF EACH COLUMN
mean_of_columns = zeros(1, Column_Count);

% INITIALIZE AN ARRAY TO STORE STANDARD DEVIATION OF EACH COLUMN
std_of_columns =  zeros(1, Column_Count);

% LOOP OVER THE COLUMNS TO CALCULATE THE MEAN AND STANDARD DEVIATION
for i = 1:Column_Count

    % CALCULATE THE MEAN AND STANDARD DEVIATION OF THE CURRENT COLUMN
    mean_of_columns(i) = mean(X(:, i));
    std_of_columns(i) = std(X(:, i));

    % PRINT THE MEAN AND STANDARD DEVIATION OF THE CURRENT COLUMN
    fprintf('Mean of column %d = %.2f\n', i, mean_of_columns(i));
    fprintf('Standard deviation of column %d: %.2f\n', i, std_of_columns(i));
    fprintf('\n');
end

hold on

% LOOP OVER THE COLUMNS TO CREATE A HISTOGRAM
for i = 1:Column_Count
    %CREATE THE HISTOGRAM AND TITLE OF THE CURRENT COLUMN
    figure;
    histogram(X(:, i));
    title(sprintf('Histogram of Column %d', i));
end



```

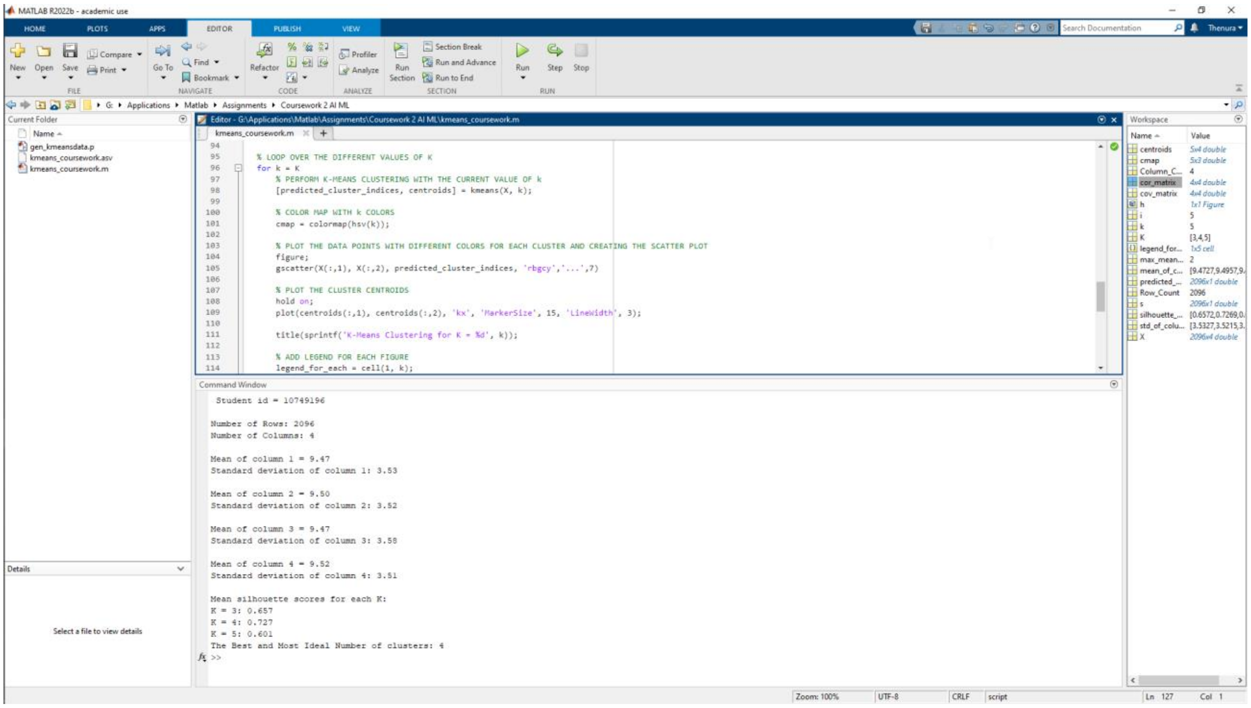
Workspace

Name	Value
centroids	5x4 double
cmap	5x3 double
Column_C...	4
cor_matrix	4x4 double
cov_matrix	4x4 double
h	1x1 Figure
i	5
k	5
K	[3,4,5]
legend_for...	1x5 cell
max_mean...	2
mean_of_c...	[9.4727,9.4957,9.
predicted_...	2096x1 double
Row_Count	2096
s	2096x1 double
silhouette_...	[0.6572,0.7269,0.
std_of_colu...	[3.5327,3.5215,3.
X	2096x4 double



```
% LOOP OVER THE DIFFERENT VALUES OF K
for k = K
    % PERFORM K-MEANS CLUSTERING WITH THE CURRENT VALUE OF k
    [predicted_cluster_indices, centroids] = kmeans(X, k);

    % COLOR MAP WITH k COLORS
    cmap = colormap(hsv(k));

    % PLOT THE DATA POINTS WITH DIFFERENT COLORS FOR EACH CLUSTER AND CREATING THE SCATTER PLOT
    figure;
    gscatter(X(:,1), X(:,2), predicted_cluster_indices, 'rbgcy','...',7)

    % PLOT THE CLUSTER CENTROIDS
    hold on;
    plot(centroids(:,1), centroids(:,2), 'kx', 'MarkerSize', 15, 'LineWidth', 3);

    title(sprintf('K-Means Clustering for K = %d', k));

    % ADD LEGEND FOR EACH FIGURE
    legend_for_each = cell(1, k);
    for i = 1:k
        legend_for_each{i} = sprintf('Cluster %d', i);
    end
    legend(legend_for_each);

    % ADD AXIS LABELS
    xlabel('X AXIS');
    ylabel('Y AXIS');

    hold off;
end
```

Runtime Terminal

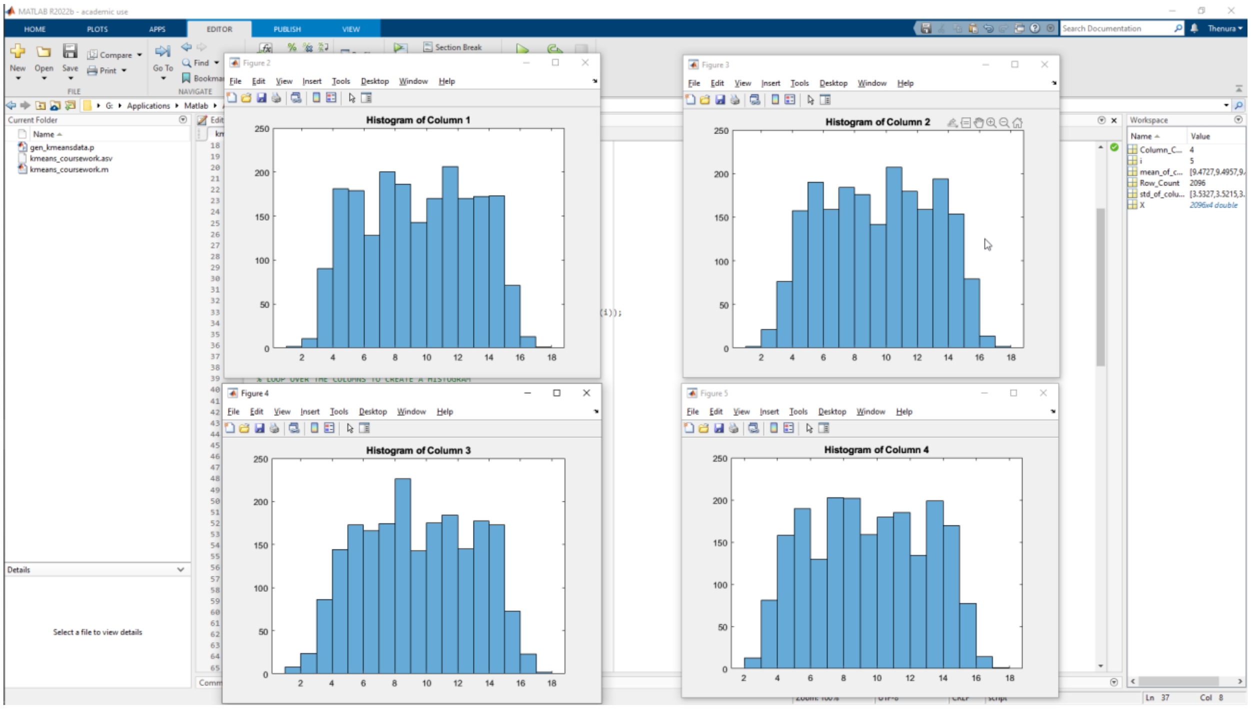


```
% LOOP OVER THE DIFFERENT VALUES OF K
for k = K
    % PERFORM K-MEANS CLUSTERING WITH THE CURRENT VALUE OF k
    [predicted_cluster_indices, centroids] = kmeans(X, k);

    % COLOR MAP WITH k COLORS
    cmap = colormap(hsv(k));

    % PLOT THE DATA POINTS WITH DIFFERENT COLORS FOR EACH CLUSTER AND CREATING THE SCATTER PLOT
    figure;
    gscatter(X(:,1), X(:,2), predicted_cluster_indices, 'rbgcy','...',7)

    % PLOT THE CLUSTER CENTROIDS
    hold on;
    plot(centroids(:,1), centroids(:,2), 'kx', 'MarkerSize', 15, 'LineWidth', 3);

    title(sprintf('K-Means Clustering for K = %d', k));

    % ADD LEGEND FOR EACH FIGURE
    legend_for_each = cell(1, k);
```

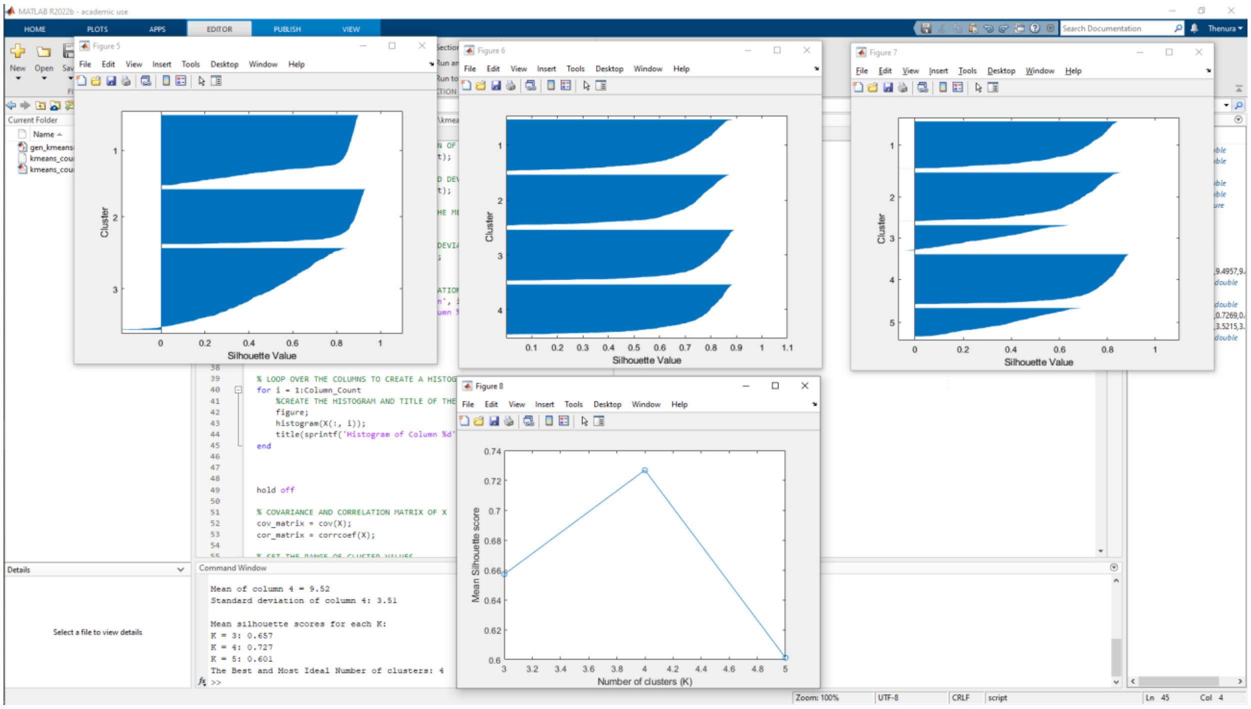
```
 Student id = 10749196

Number of Rows: 2096
Number of Columns: 4

Mean of column 1 = 9.47
Standard deviation of column 1: 3.53

Mean of column 2 = 9.50
Standard deviation of column 2: 3.52

Mean of column 3 = 9.47
Standard deviation of column 3: 3.58

Mean of column 4 = 9.52
Standard deviation of column 4: 3.51

Mean silhouette scores for each K:
K = 3: 0.657
K = 4: 0.727
K = 5: 0.601
The Best and Most Ideal Number of clusters: 4
>>
```

Figures

- Histogram



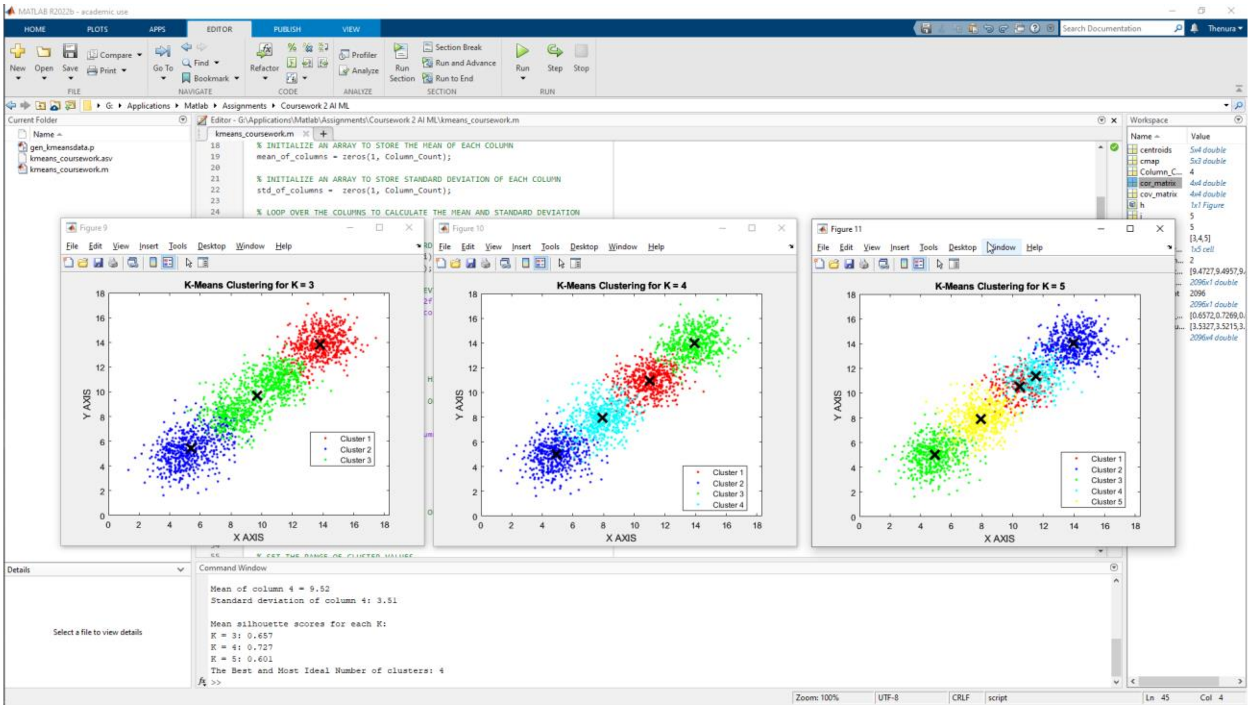
- Silhouette and Mean Silhouette Plotting



```
% LOOP OVER THE COLUMNS TO CREATE A HISTOG
for i = 1:Column_Count
    %CREATE THE HISTOGRAM AND TITLE OF THE
    figure;
    histogram(X(:, i));
    title(sprintf('Histogram of Column %d'
end



hold off

% COVARIANCE AND CORRELATION MATRIX OF X
cov_matrix = cov(X);
cor_matrix = corrcoef(X);
```

```
Mean of column 4 = 9.52
Standard deviation of column 4: 3.51

Mean silhouette scores for each K:
K = 3: 0.657
K = 4: 0.727
K = 5: 0.601
The Best and Most Ideal Number of clusters: 4
>>
```

- Scatter Plot



```
% INITIALIZE AN ARRAY TO STORE THE MEAN OF EACH COLUMN
mean_of_columns = zeros(1, Column_Count);

% INITIALIZE AN ARRAY TO STORE STANDARD DEVIATION OF EACH COLUMN
std_of_columns =  zeros(1, Column_Count);

% LOOP OVER THE COLUMNS TO CALCULATE THE MEAN AND STANDARD DEVIATION
```

```
Mean of column 4 = 9.52
Standard deviation of column 4: 3.51

Mean silhouette scores for each K:
K = 3: 0.657
K = 4: 0.727
K = 5: 0.601
The Best and Most Ideal Number of clusters: 4
>>
```

Conclusions

Therefore we can conclude that even though K-means is one of most simple algorithms of unsupervised learning there are still a lot of cons such as less consistency, and having to be able pre determine the k value etc. This might be suitable for some projects where ass unsuitable for others therefore depends on the type of project and scale of it.

References

Clustering in Machine Learning - Javatpoint (no date). Available at: https://www.javatpoint.com/clustering-in-machine-learning (Accessed: 23 January 2023).

Exploring Clustering Algorithms: Explanation and Use Cases - neptune.ai (no date a). Available at: https://neptune.ai/blog/clustering-algorithms (Accessed: 23 January 2023).

Exploring Clustering Algorithms: Explanation and Use Cases - neptune.ai (no date b). Available at: https://neptune.ai/blog/clustering-algorithms (Accessed: 23 January 2023).

Hierarchical Clustering in Machine Learning - Javatpoint (no date). Available at: https://www.javatpoint.com/hierarchical-clustering-in-machine-learning (Accessed: 23 January 2023).

KMeans Explained (no date). Available at: https://ml-explained.com/blog/kmeans-explained (Accessed: 23 January 2023).

What is Machine Learning? Concepts & Examples - Data Analytics (no date). Available at: https://vitalflux.com/what-is-machine-learning-concepts-examples/ (Accessed: 23 January 2023).

What is Machine Learning? Definition, Types, Applications (no date). Available at: https://www.mygreatlearning.com/blog/what-is-machine-learning/ (Accessed: 23 January 2023).