NANYANG TECHNOLOGICAL UNIVERSITY
SINGAPORE

SC4020 : Data Analytics & Mining

Team Members:
WONG ZI JUN (U2221712A)
Bhupathiraju Mihir Varma (U223870K)
Lim Kuan Hwee Sean (U2121484D)
Batra Parth (U2222378E)

Table of contents

Abstract

In the field of data science, clustering plays a crucial role in unsupervised learning. Numerous clustering algorithms exist, each offering unique advantages. This report delves into various clustering algorithms, detailing their strengths and weaknesses, and examines the impact of adjusting their hyperparameters on performance. Multiple datasets from diverse domains are utilised to assess the effectiveness and efficiency of these algorithms.

Introduction

1) Analysis of students performance in exams

Dataset from https://www.kaggle.com/datasets/spscientist/students-performance-in-exams
The dataset, sourced from Kaggle, is linked to the education domain. With 5 categorical variables on the student's background, and 3 numerical score values, clustering is employed in this dataset to possibly identify recurring characteristics linked to the performance of students.

2) Mall Customer Segmentation Data

Dataset from
https://www.kaggle.com/datasets/vjchoudhary7/customer-segmentation-tutorial-in-python/data
The dataset, sourced from Kaggle, pertains to the commerce domain and encompasses details of 200 customers, including age, gender, income, and spending levels. Clustering is employed in this dataset to delineate distinct customer segments for targeted advertising campaigns and promotions based on spending levels.

3) World Happiness Report 2023

Dataset from
https://www.kaggle.com/datasets/simonaasm/world-happiness-index-by-reports-2013-2023
The dataset sourced from Kaggle comprises data from the World Happiness Index for the year 2023. With 7 key factors: economy, family, health, freedom, trust, generosity and dystopia. We want to identify clusters of countries based on happiness indicators to understand common characteristics among high, medium, and low-happiness countries.

4) Netflix movies and tv shows

Dataset from
https://www.kaggle.com/datasets/shivamb/netflix-shows/versions/2?resource=download
The dataset sourced from Kaggle contains information about movies and TV shows available on Netflix, including attributes such as title, genre, rating, release year, and duration. This dataset provides insights into the type of content offered by Netflix across different years and regions. Our goal is to identify patterns in content production and preferences across various genres, ratings, and release years.

Methods

K Means

The advantages of Kmeans:

- Efficiency: K-Means is very efficient and fast. Its linear time complexity allows it to handle large datasets effectively. For large, unlabeled data, K-Means provides valuable insights as an unsupervised clustering method.
- Simplicity: Due to its simplicity, Kmeans is relatively straightforward to implement and helps uncover hidden data groups in complex datasets. The results can be easily interpreted even if the dataset size is large.
- Flexibility: K-Means is a flexible algorithm that adapts easily to changes. This flexibility allows us to use it for tasks like data segmentation, data analysis, dimensionality reduction, feature engineering, anomaly detection, semi-supervised learning, search engines, and image segmentation.

The drawbacks of Kmeans:

- We need to specify the number of clusters, which can be a hassle. The solution for this problem is to use the elbow method to find the optimal number of clusters. We run Kmeans for a number of clusters (k) and find the Silhouette Coefficient for each number of clusters and find the elbow point. We explored this method below and elaborated on it more in the experiments section.
- It is necessary to run the Kmeans algorithm several times to avoid suboptimal solutions. Since Kmeans is very sensitive to the centroid initialization and different initializations of centroids result in different clusters.
- It doesn't behave well when the clusters have varying sizes, different densities, or nonspherical shapes. Optimizing for inertia as a metric can be challenging because it assumes that clusters are globular. This can result in the algorithm struggling to create elongated or irregularly shaped clusters, and it tends to perform better with clusters that are spherical or circular in shape.

DBSCAN (Density-Based Spatial Clustering of Applications with Noise)

DBSCAN is an unsupervised machine learning algorithm primarily used for clustering tasks. DBSCAN identifies clusters based on the density of data points in a region, making it effective for discovering arbitrarily shaped clusters. It does not rely on a predetermined number of clusters. The algorithm groups points that are closely packed together, while points in low-density regions are classified as noise. A key feature of DBSCAN is its ability to handle datasets with varying densities and its robustness to outliers. Its two main parameters, the neighbourhood radius (epsilon) and the minimum number of points required to form a dense region (minPts), determine how clusters are formed. DBSCAN is widely used in applications such as geospatial analysis, image recognition, and anomaly detection.

Two important parameters: (1) Minimum samples (min_samples or minPts) are the least number of points required to form a cluster. (2) Epsilon (eps) is the maximum distance where two points can still be in the same cluster.

For each instance, the algorithm counts how many instances are located within a small distance epsilon from it. This region is called the instances of epsilon-neighbourhood. If an instance has at least min_samples instances in its epsilon-neighbourhood, then it is considered a core instance, which are basically those that are located in dense regions. All instances in the neighbourhood of a core instance belong to the same cluster. The neighbourhood may include other core instances; therefore, a long sequence of neighbouring core instances forms a single cluster. Any instance that isn't a core instance and doesn't have one in its neighbourhood is considered an anomaly.

The value of min_samples tells us how the algorithm handles noise. For large or noisy datasets, setting a higher minPts ensures that more meaningful points form part of a cluster. The eps parameter is Crucial, as it determines the size of the search region, which then determines how many points are included in a cluster. If eps is set too large and distinct, nearby clusters may merge. If it's set too small, it could cause fragmentation within a single cluster.

The advantages of DBSCAN:

- DBSCAN is a very simple and powerful algorithm that can identify any number of clusters, and we don't need to specify the number of clusters in the beginning.
- DBSCAN can identify clusters of any shape and performs well with arbitrary cluster shapes.
- It is robust to outliers and can detect them easily.
- It has just two hyperparameters, eps and min_samples.

The drawbacks of DBSCAN:

- If there is no sufficient low-density region around some clusters, DBSCAN can struggle to capture all clusters.
- DBSCAN is sensitive to changes in epsilon value.
- The computational complexity is O(m^2n), so it doesn't scale well to large datasets.

Agglomerative Clustering

Agglomerative clustering is a hierarchical, unsupervised machine learning algorithm used for clustering tasks. It starts by treating each data point as its own cluster and progressively merges the closest pairs of clusters based on a chosen similarity measure, continuing until only one cluster remains or a termination condition is met. This “bottom-up” approach builds a hierarchy of clusters, which can be visualised using a dendrogram. Unlike partition-based algorithms like KMeans, agglomerative clustering does not require specifying the number of clusters in advance, as the optimal number can be determined by cutting the dendrogram at the appropriate level.

Agglomerative clustering connects the nearest pair of clusters and captures clusters of various shapes. It is a flexible and informative cluster tree. It can scale to large numbers of instances if we provide a connectivity matrix, but without a connective matrix, the algorithm doesn't scale well to large datasets.

Three important parameters: (1) The linkage criterion (single, complete, average linkage) determines how distances between clusters are measured and affects the merging process. (2) The distance metric (Euclidean, Manhattan, cosine similarity) defines how similarity is calculated between individual data points. (3) Number of clusters (n_cluster) allows the user to early stop the algorithm by specifying the number of clusters to retain.

The advantages of agglomerative clustering:

- No need to specify the number of clusters upfront, as the hierarchical structure offers flexibility.
- Works well with small datasets and can reveal nested data structures through dendrograms.
- Can handle non-spherical and arbitrarily shaped clusters effectively.
- Connectivity constraints (using a connectivity matrix) allow it to scale to larger datasets by restricting merges to nearby points.

The drawbacks of agglomerative clustering:

- It is computationally expensive; with a time complexity of $O(n^2 \log n)$, it struggles with very large datasets without a connectivity matrix.
- Memory-intensive, as it requires storing distance matrices for all points.
- Sensitive to noise and outliers, which can affect the merging process.
- Choice of linkage and distance metric can have a significant impact on the final clusters.
- Without a connectivity matrix, it does not scale well, limiting its application on massive datasets.

Mean Shift

Mean shift clustering is centroid-based clustering algorithm and it starts by placing a circle centred on each instance, then for each circle it computes the mean of all the instances

located within it, and it shifts the circle so that it is centred on the mean. Then iterate the mean shifting step until each of them is centred on the mean of the instances it contains. Mean shift shifts circle in the direction of higher density until each of them has found a local density maximum. Then all the instances whose circles have settled close enough are assigned to the same cluster.

The advantages of Mean shift:

- Mean shift can find any number of clusters of any shape.
- It has only one hyperparameter called bandwidth, and it relies on local density estimation.
- It determines the number of clusters based on bandwidth, which is set by us.
- Outliers don't cause any issues to the algorithm.

The drawbacks of Mean shift:

- Unlike DBSCAN, mean shift tends to chop clusters into pieces when they have internal density variations.
- Its computational complexity is O(m^2n), so it is not suited for large datasets.
- Bandwidth value selection is a nontrivial task, and the result is dependent on bandwidth. The value of bandwidth can cause independent clusters to be combined or data points to be missed out.

Experiments

1) Students performance in exams dataset

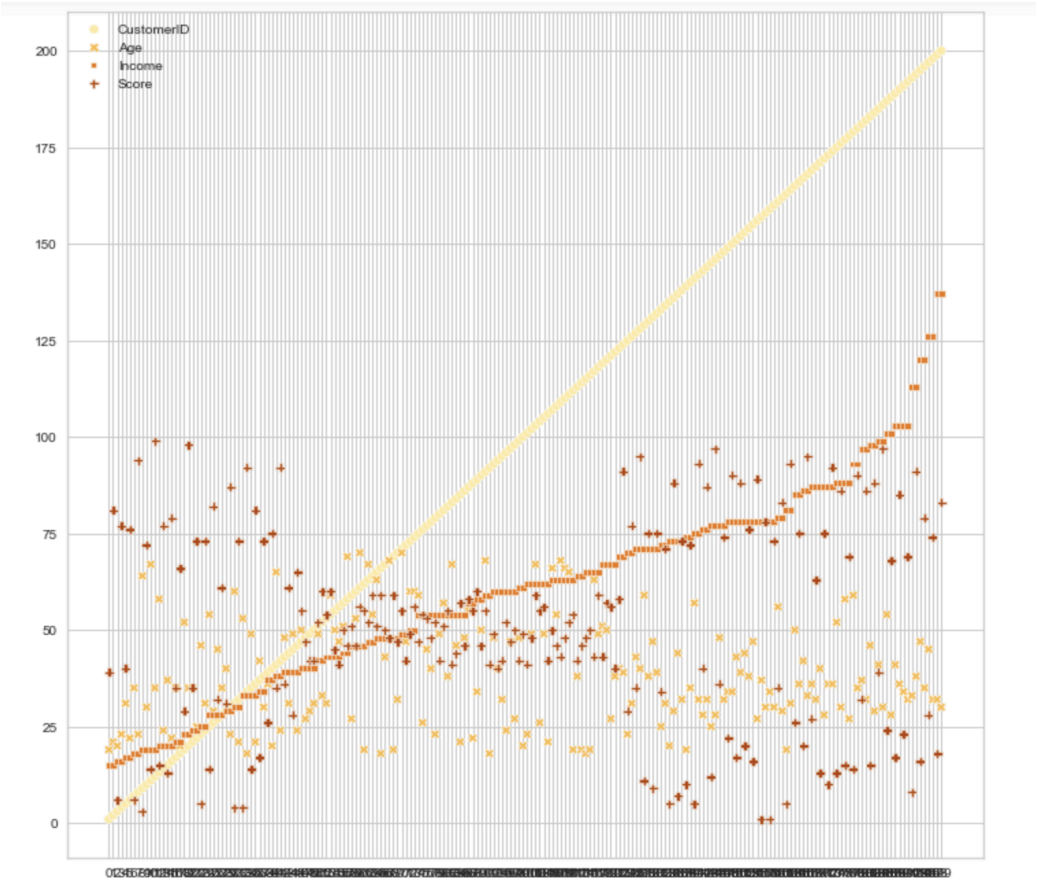
Firstly, data preprocessing is performed to ensure no erroneous data.

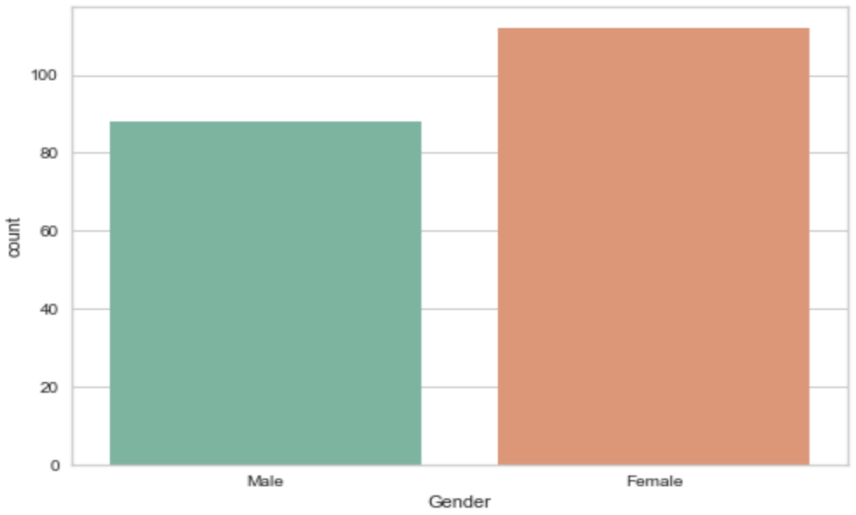
Secondly, EDA (exploratory data analysis) is performed to visualise and identify possible trends. There are multiple categorical variables and 3 numerical variables: math score, reading score and writing score. We want to find out what affects student performance, so we will use clustering to cluster the students based on their grades and discover potential patterns linking to exam performance

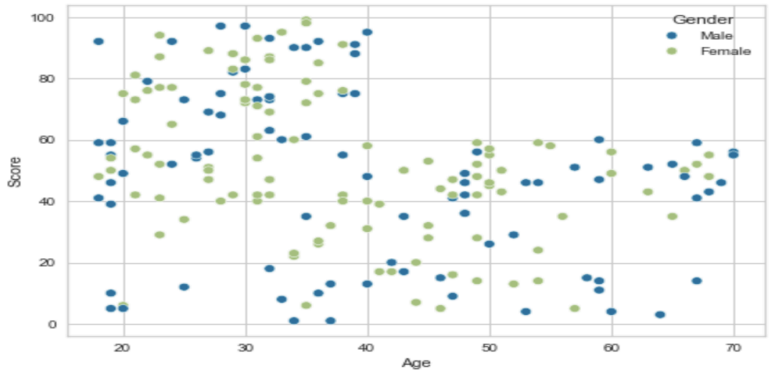
Thirdly, we applied min-max scaling to ensure all features contribute equally to the calculations. However, since the scores already range from 0 to 100, scaling may be less critical in this case, as the features are already on a comparable scale. Further details and graph plot visualisations can be found in the notebook SC4020 Student Performance.ipynb.

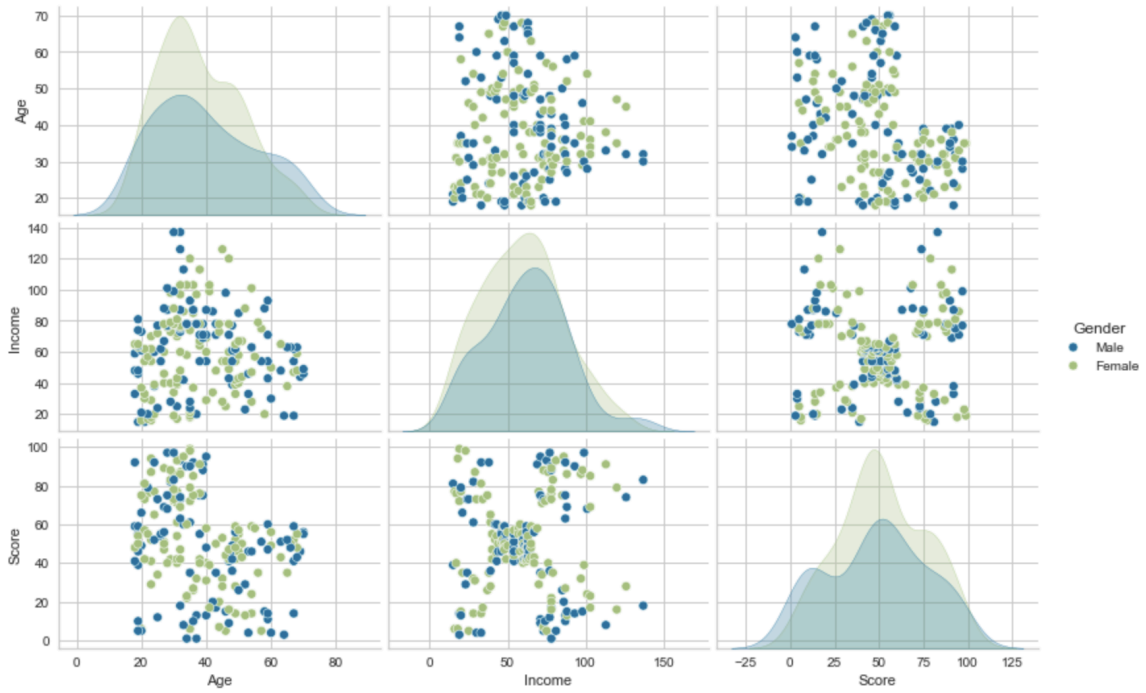
2) Mall Customer Segmentation Data

We performed EDA and found there are more female customers than male customers. Females are 56% of total customers, and the mean and median income of male customers is higher than female customers. In terms of standard deviation, it is almost similar for both groups, but we can see an outlier from male where the annual income is close to 140K. We found that spending scores are higher for women than men. We found that most wealthy customers are from age's around 20-45, where younger men from age 25-40 are richer than the women.



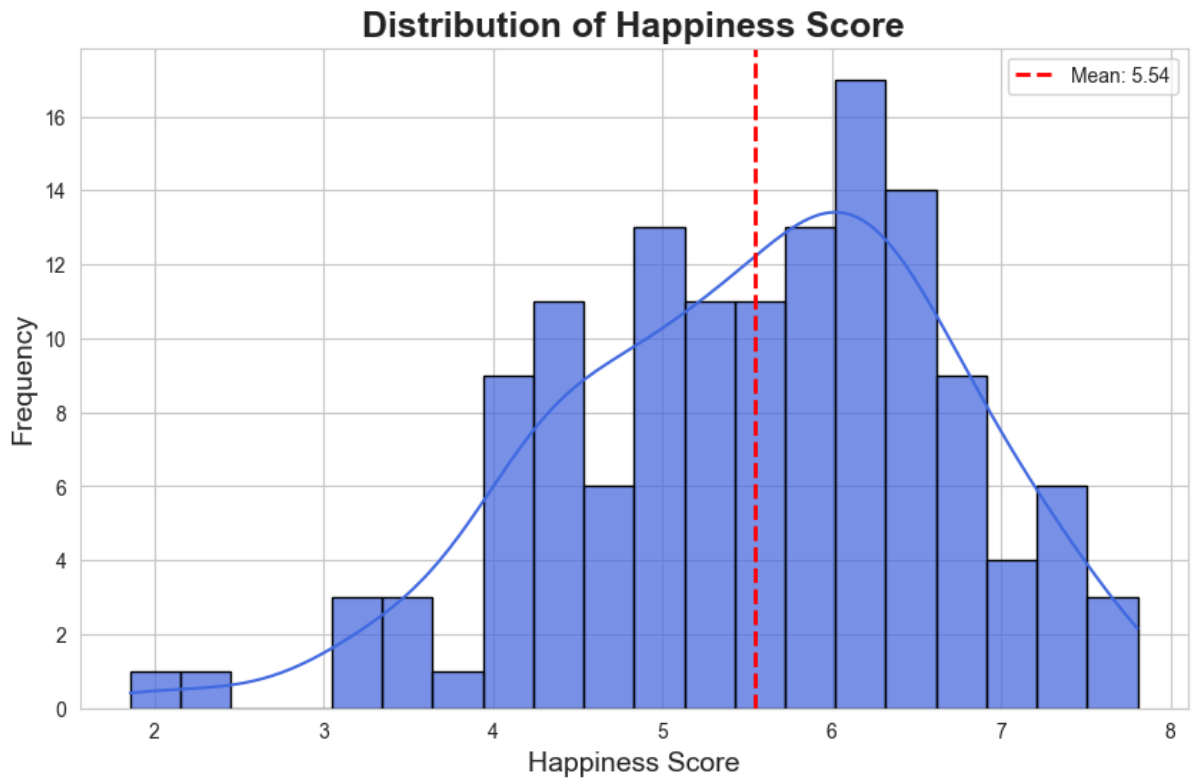

count
100
80
60
40
20
0
Male
Female
Gender

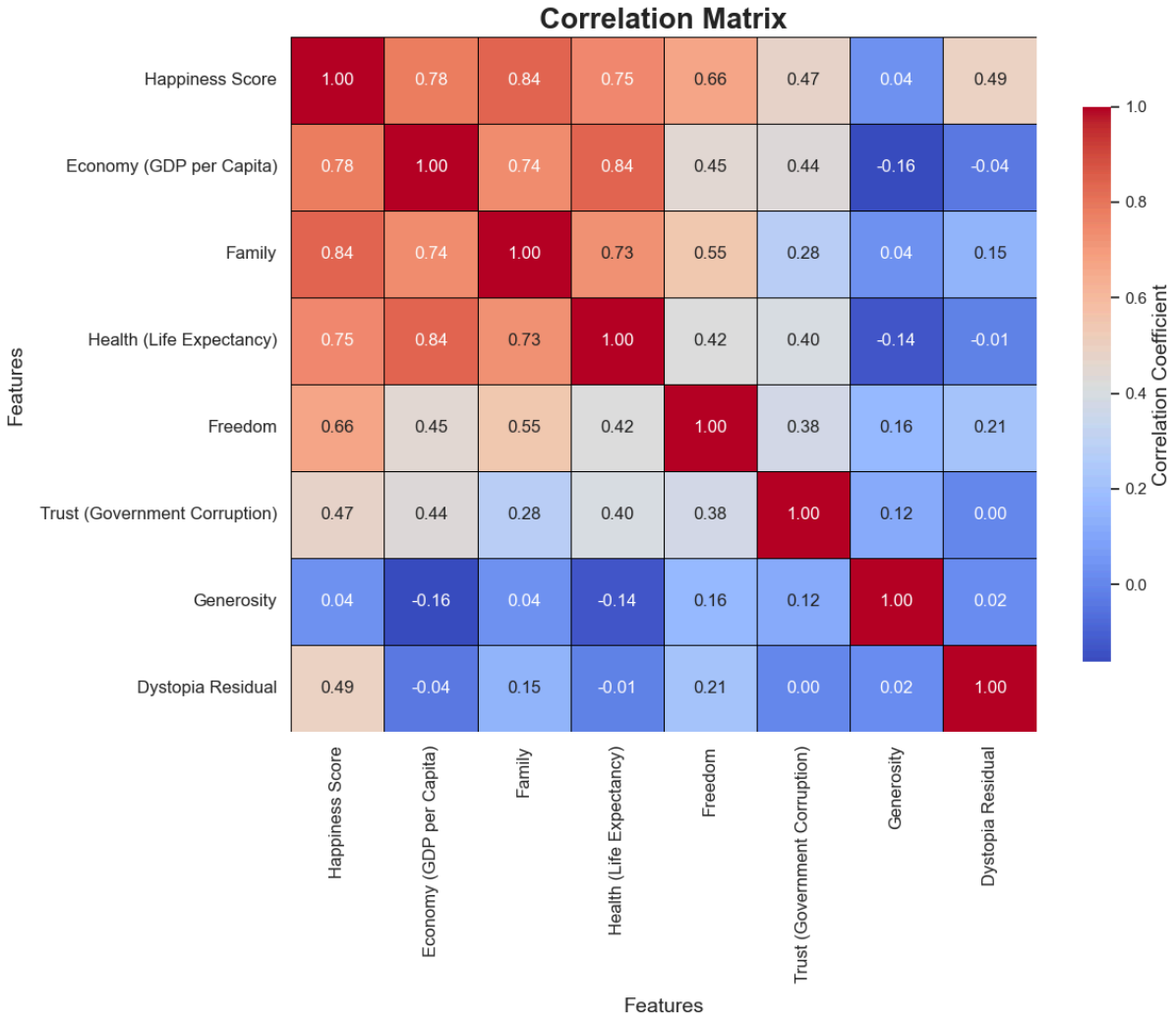

Gender
Male
Female
Score
100
80
60
40
20
0
20
30
40
50
60
70
Age


Age
Income
Score
Gender
Male
Female
0
20
40
60
80
Age
0
50
100
150
Income
−25
0
25
50
75
100
125
Score

3) World Happiness Report 2023

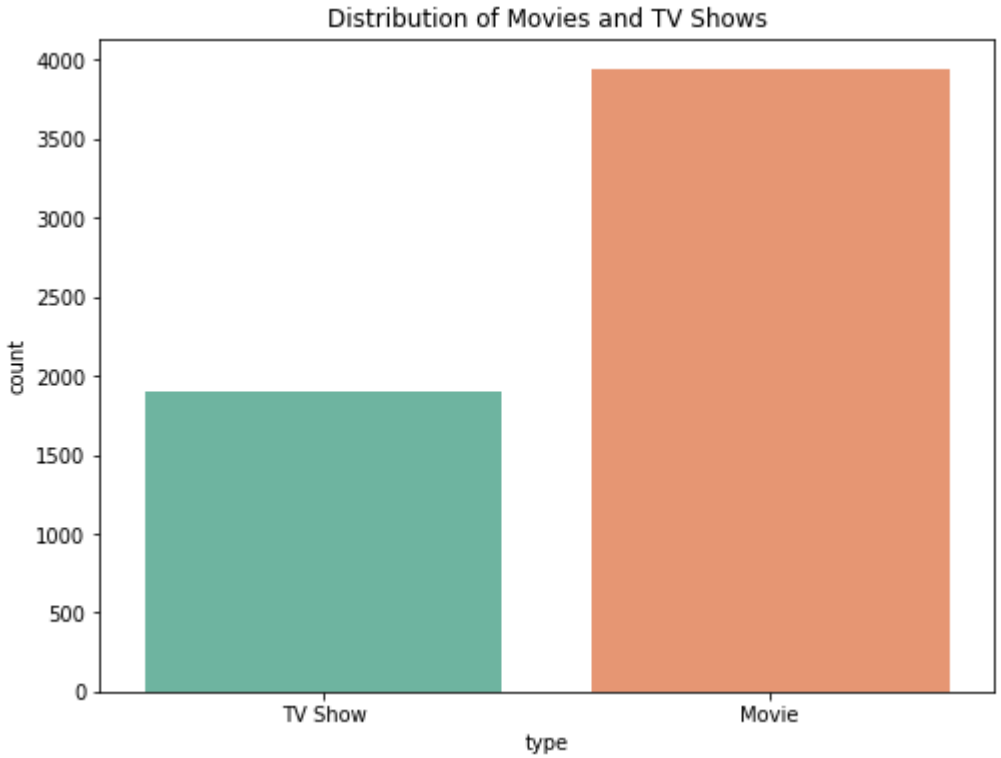
We performed EDA and found that economy, family, and health have a strong correlation to the happiness score. Interestingly, economy and health also have a strong correlation to each other. The region with the highest happiness score is Australia and New Zealand, while the region with the lowest is Southern Asia. Singapore ranks 25th with a happiness score of 6.58.

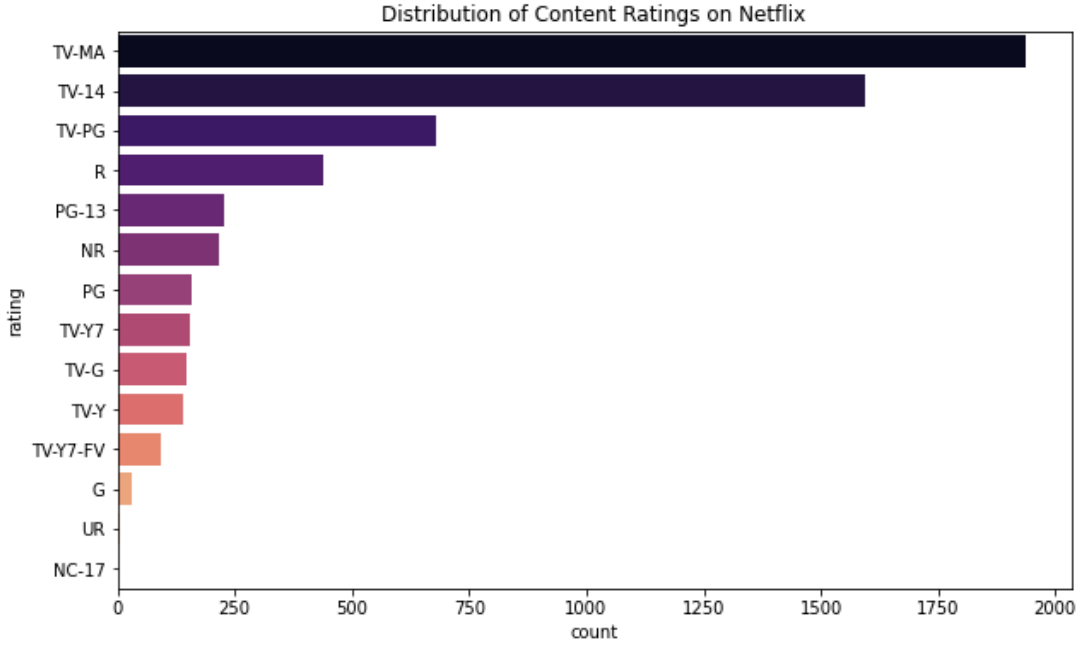


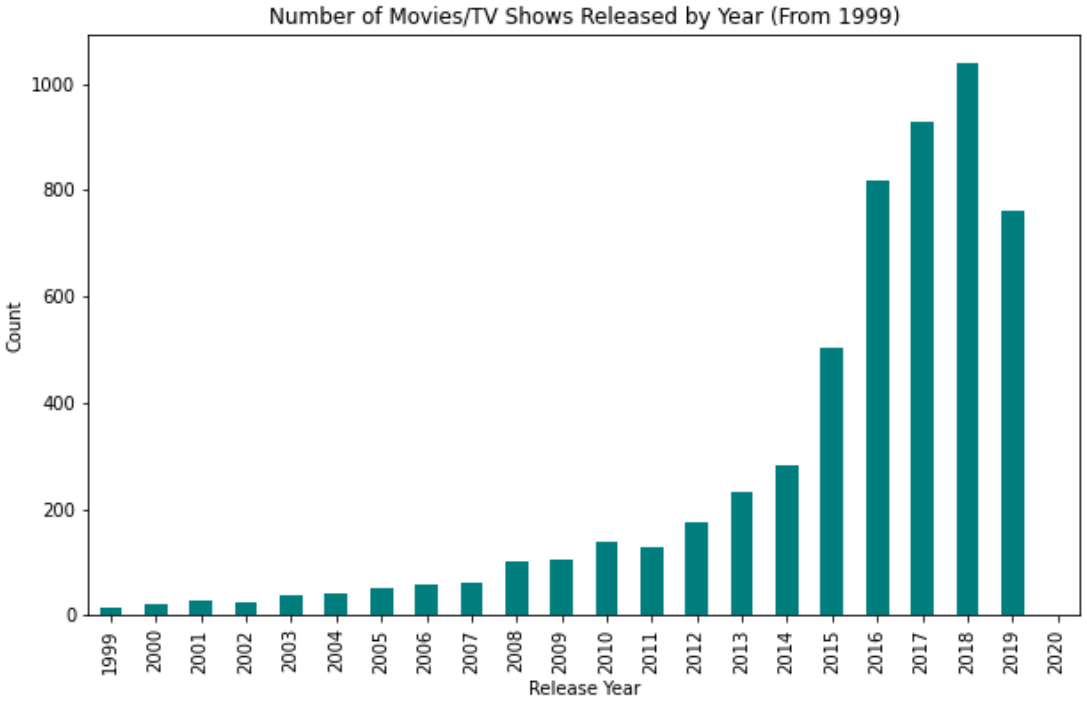


4) Netflix Movies and TV Shows

In our EDA, we discovered that Netflix produces more movies than TV shows, with a notable increase in content production after 2010. Drama, International Movies, and Comedies are the most common genres on the platform. Additionally, we observed that the distribution of content ratings leans towards TV-MA, indicating a focus on mature content.

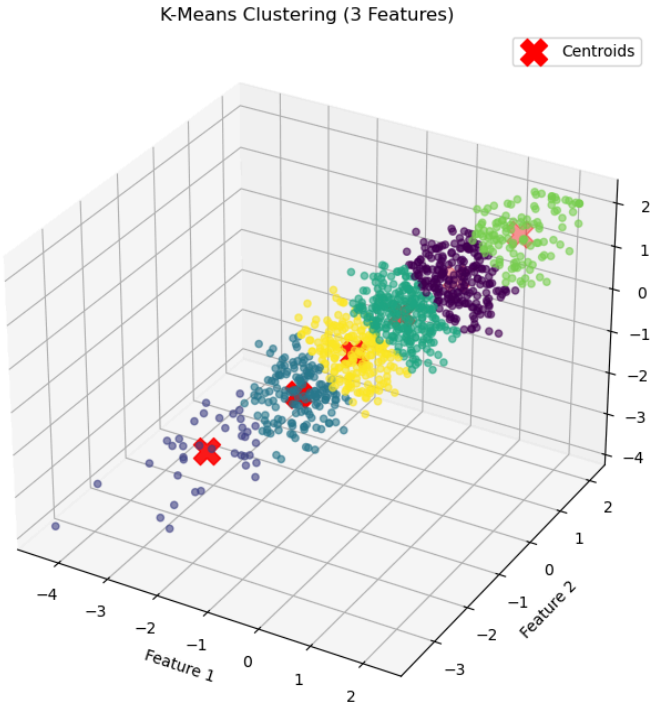

Distribution of Movies and TV Shows
4000
3500
3000
2500
2000
1500
1000
500
0
count
TV Show
Movie
type


Distribution of Content Ratings on Netflix
TV-MA
TV-14
TV-PG
R
PG-13
NR
PG
TV-Y7
TV-G
TV-Y
TV-Y7-FV
G
UR
NC-17
rating
0
250
500
750
1000
1250
1500
1750
2000
count


Number of Movies/TV Shows Released by Year (From 1999)
Count
0
200
400
600
800
1000
1999
2000
2001
2002
2003
2004
2005
2006
2007
2008
2009
2010
2011
2012
2013
2014
2015
2016
2017
2018
2019
2020
Release Year

K-means

1) **Students Performance Dataset :** We used the elbow method to determine an optimal k-number of centroids and then performed K-means using the optimal k derived. We then repeated the process with different K values to observe the differences. Since the process was straightforward with no issues, further technical details and graph visualisations can be found in the SC4020 Student Performance.ipynb notebook. The performance of the k-means clustering was optimal, separating cleanly into 6 clusters based around their scores.



K-means clustering, 6 centroids
Features - 1: math score, 2: reading score, 3: writing score

Students Performance Clustering Results

Analysis:

(Cluster 0 has highest average score; Cluster 5 has lowest.)

Gender: females tend to perform better than males at higher scores (with consideration of ~1.1x ratio of F to M)

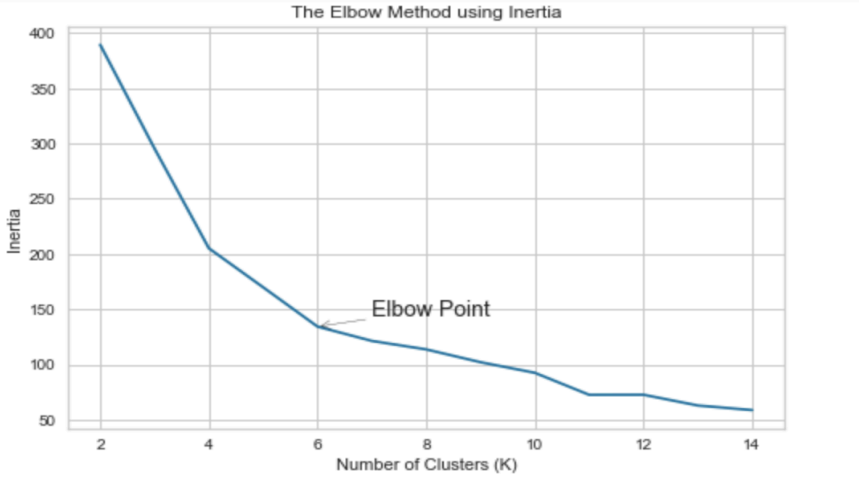
Race: group E tends to perform better, group A worse, others have no significant pattern

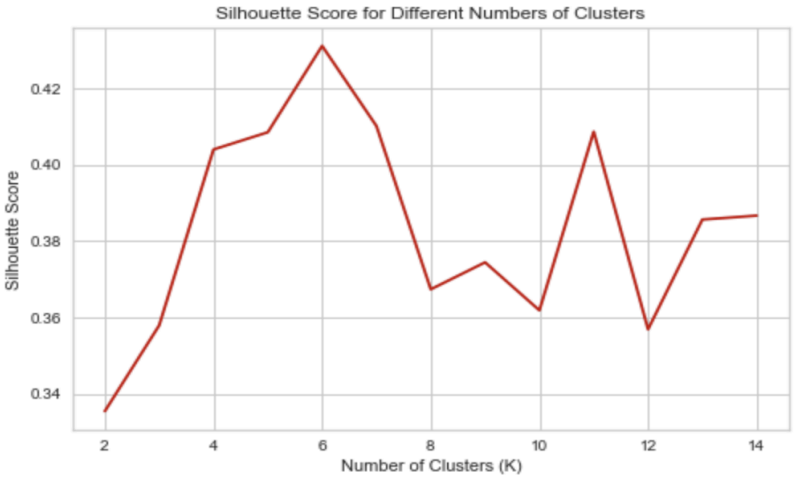
Parental Education: those who perform well tend to have more educated parents, and vice versa for those who perform worse.

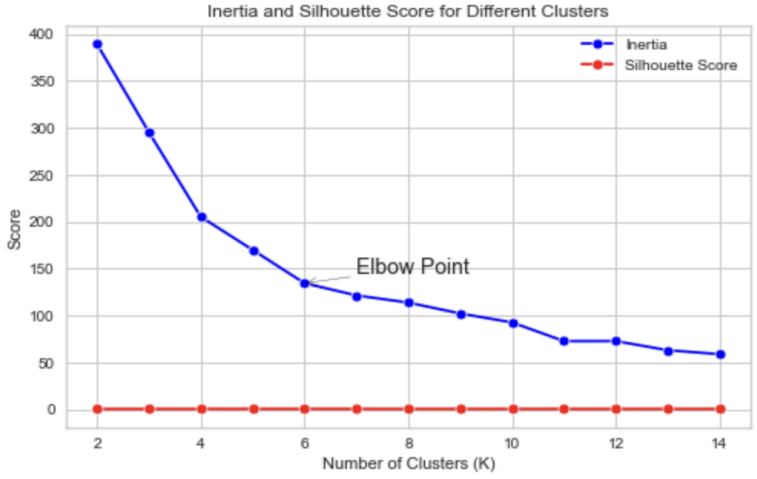
Lunch Ratio: those who take standard lunch tend to perform better than those who take free lunch (socio-economic)

Test Prep Course Ratio: Those who do better tend to take the test prep

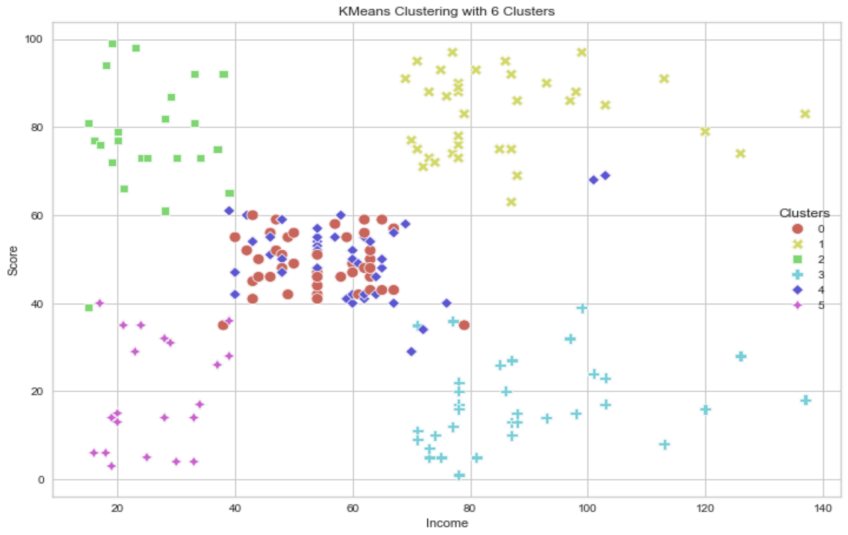
2) **Mall segmentation dataset :** We need to find the optimal number of clusters before doing K-means, which is considered one of the disadvantages because it isn't needed for other algorithms like DBSCAN. The Elbow method can be used to find the number of clusters. In the elbow method, we measure within-cluster sum of squares (WCSS) while iterating over a number of clusters (k) and choose the best one. The best number of clusters can be identified by looking for the elbow point where the graph starts to look like a straight line.

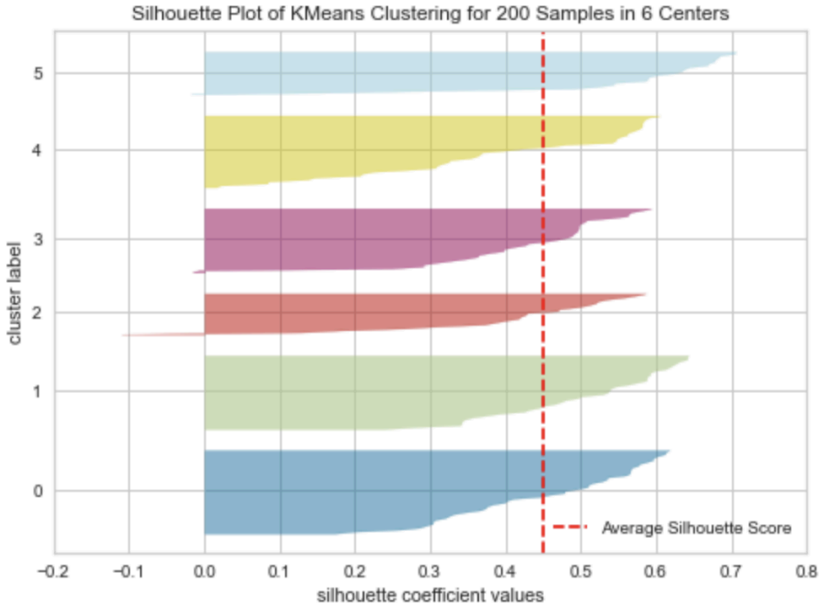






```
KMeans Clustering for 6 clusters:
Number of clusters: 6
Clustering took 0.01 s
Silhouette Coefficient: 0.43950556159154563
```

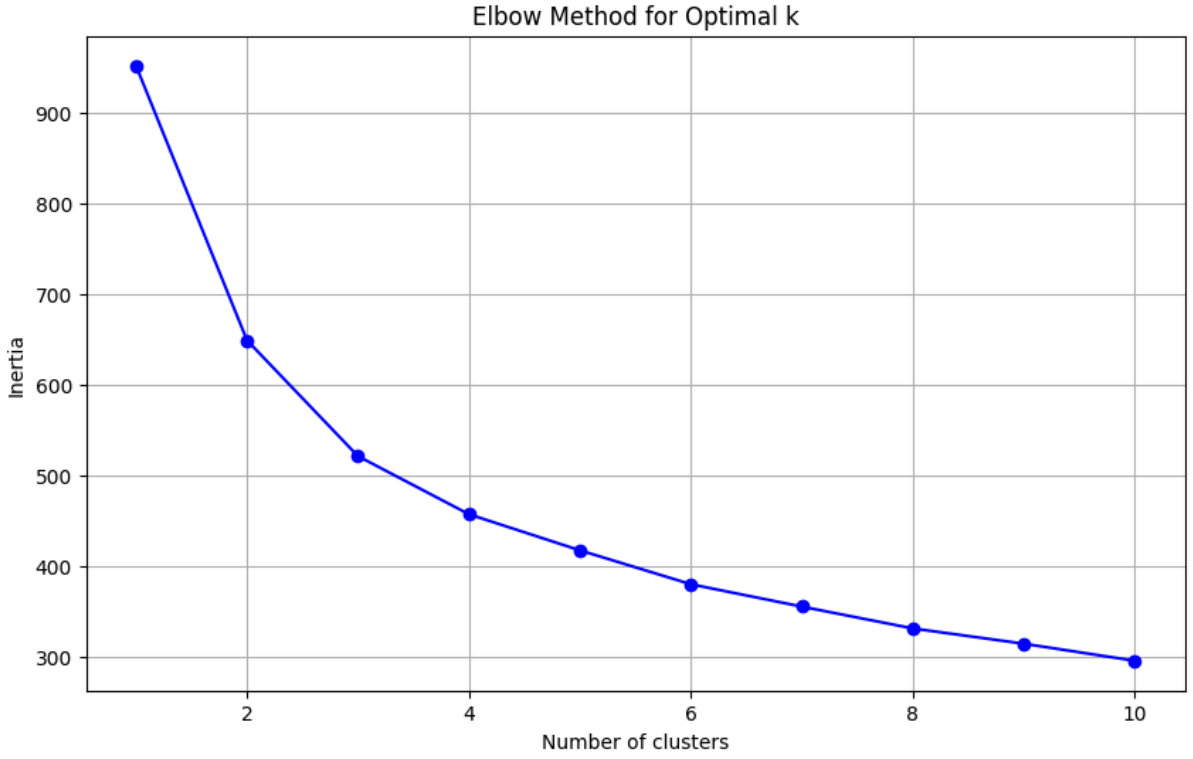


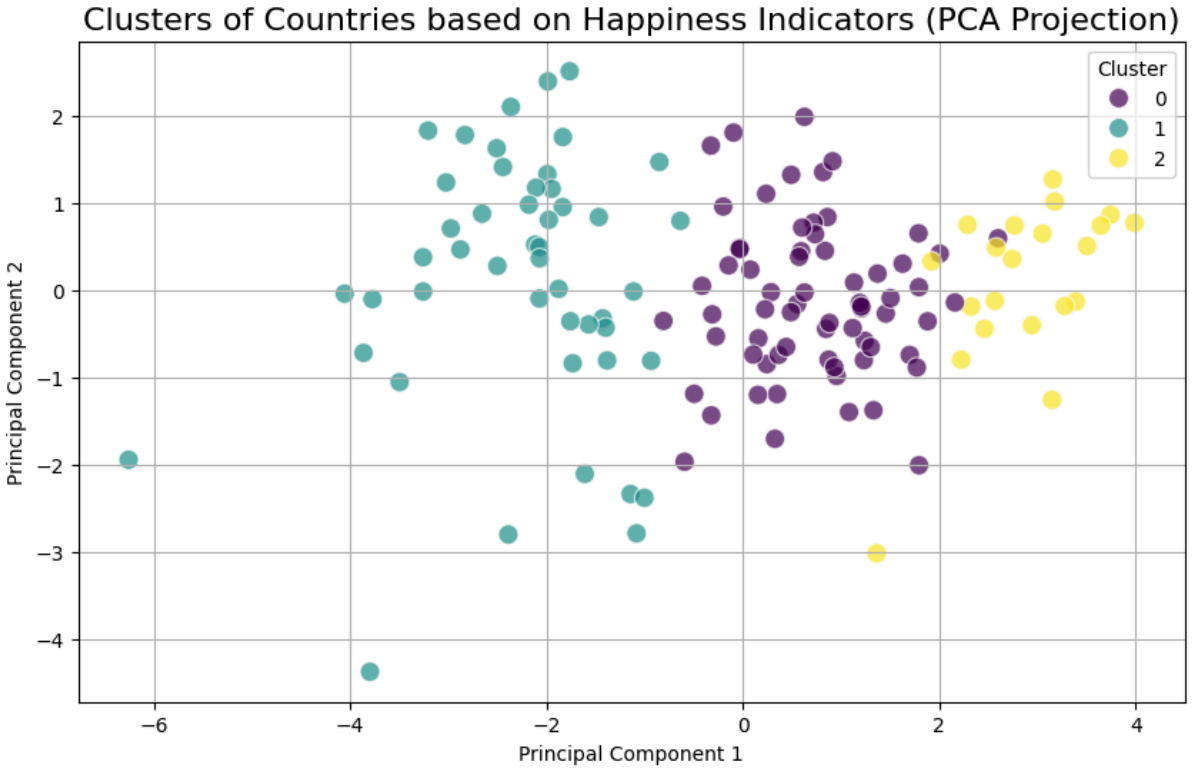


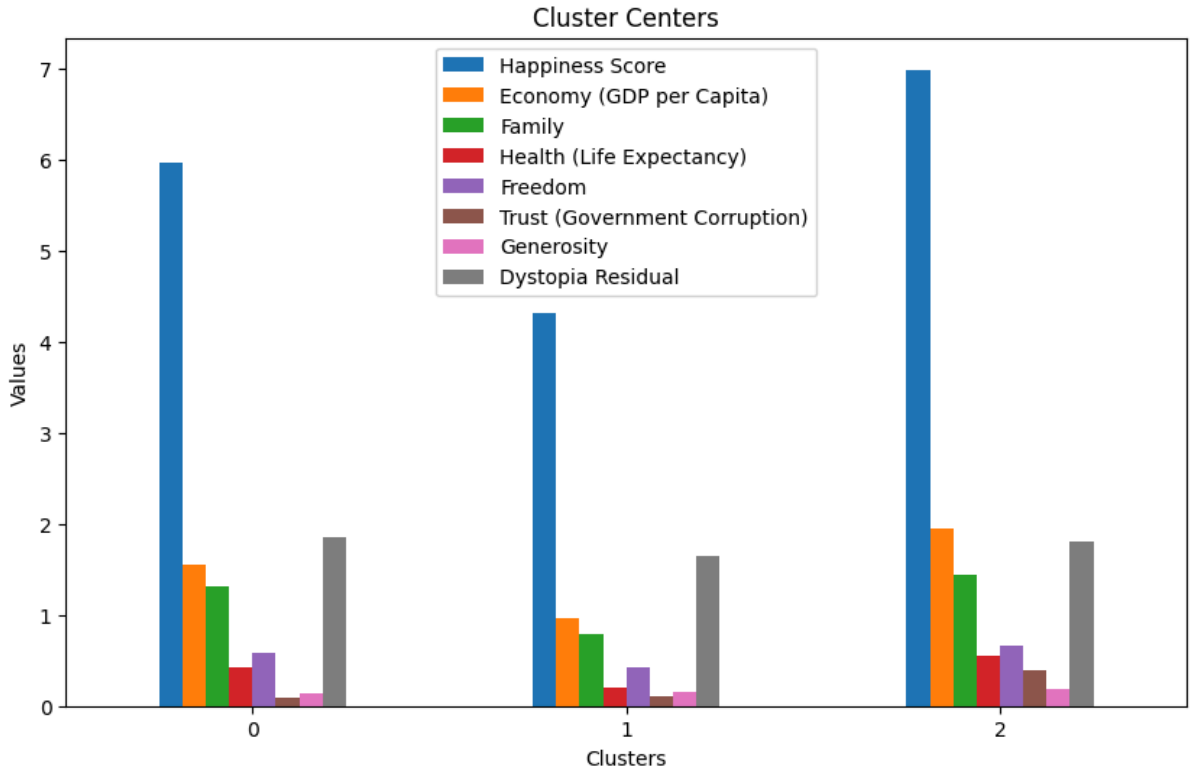
Analysis: The Kmeans algorithm shows that it is able to group 6 clusters, but there are quite a few data points that are misclassified, as we can clearly see in clusters 0 (red) and 4 (blue).

From the plots produced above from the elbow method and Kmeans, we can conclude that the optimal number of clusters is 6. It is accurate for this dataset because we observed the elbow point using an inertia graph, which gave 6 as well as we found that the silhouette coefficient is the highest for Kmeans clustering for 6 clusters. We know that the silhouette score is 0.43950556159154563, which is considered moderately good clustering as few data points do not fit into their particular clusters.

3) **World Happiness Index 2023:** The K-means algorithm was applied to the World Happiness dataset based on the Economy (GDP per Capita), Family, Health (Life Expectancy), Freedo', Trust (Government Corruption), Generosity and Dystopia Residual. The first plot shows the elbow point where we determined to use the optimal number of clusters k, k = 3.







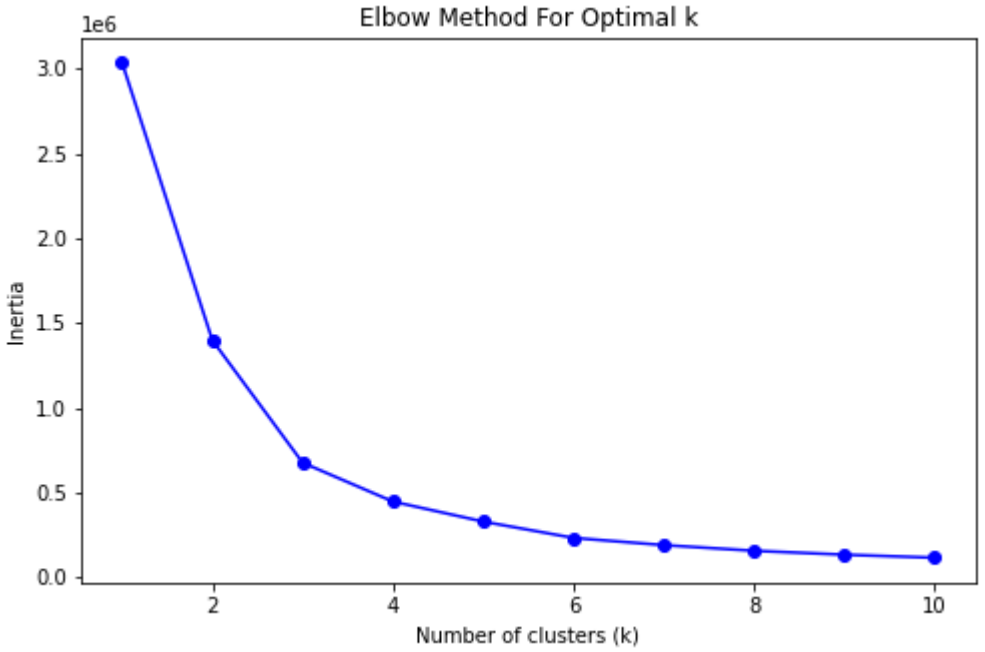
Analysis: In the second plot, we decided to use Principal Component Analysis (PCA) to plot the scatter plot. This is because it is hard to visualise clusters in multiple dimensions, and PCA enables us to reduce it to 2 dimensions. Each point represents a country. Each principal component is a linear combination of the original features (e.g. Economy, Family, Trust) where PC1 explains the maximum amount of variance and PC2 explains the next most variance. It is important to note PC2 is uncorrelated to PC1.

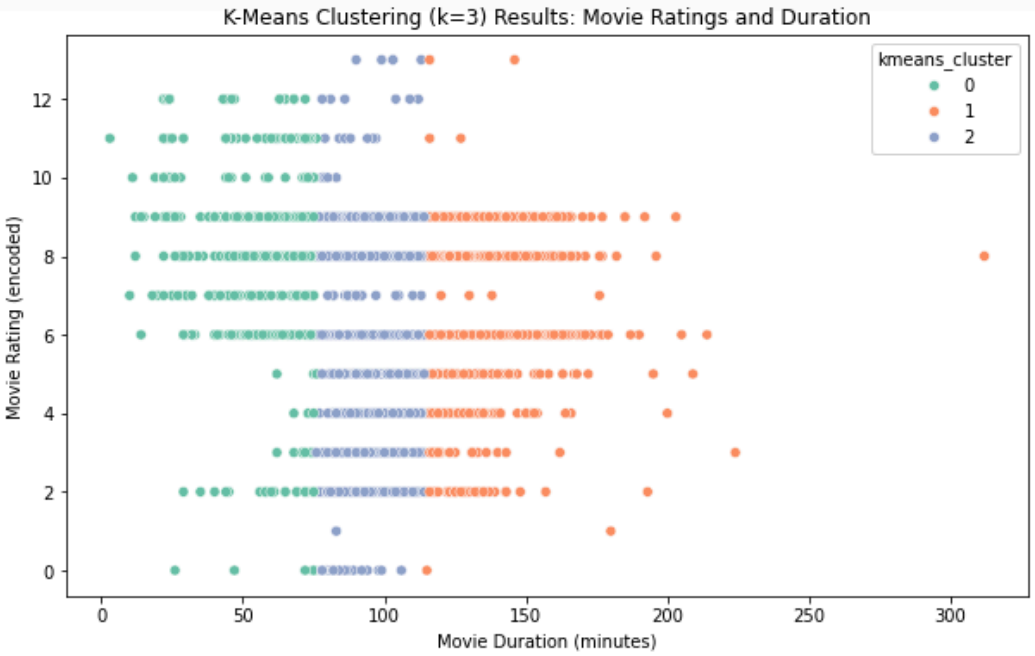
In the third plot, it shows the cluster centres plot. Based on the KMeans clustering, it shows a detailed overview of the average values for each happiness indicator within each cluster.

This helps us understand the second plot better; cluster 2 belongs to the countries with high happiness scores, while cluster 1 belongs to the countries with the middle happiness score, and cluster 0 belongs to the countries with the lowest happiness scores. We can see that between clusters 0 and 2, it is very clearly defined with no overlaps. However, there are some overlaps between clusters 1 and 2, which shows that some countries may not be easily distinguishable, or there are some countries that are considered middle in ranking but capable of transitioning to the higher ranks. This could be said for the countries in cluster 1 that are close to cluster 0, they may be at risk of losing their rankings.

The third plot could also be an indicator of which indicators a country needs to improve to increase in ranking. For example, a country in cluster 1 needs to increase its GDP, or a country in cluster 2 needs to improve its health-related policies to improve their rankings.

4) **Netflix and Movies Dataset:** The K-means algorithm was applied to the Netflix dataset to group movies based on their duration and rating. Using the Elbow Method, as shown in the first plot, we determined that the optimal number of clusters was 3, as indicated by the "elbow point," where the inertia starts to flatten after k=3. This suggests that increasing the number of clusters beyond this point does not significantly improve clustering quality.





	duration	rating	kmeans_cluster
1	67	TV-G	0
2	135	TV-14	1
3	106	TV-14	2
5	107	TV-14	2
6	81	TV-MA	2

Analysis: In the second plot, the results of the K-means clustering (with 3 clusters) are displayed, showing how movies were grouped based on their duration and encoded rating. We can see that cluster 0 (green) represents shorter movies with lower ratings, while cluster 2 (orange) groups higher-rated movies with longer durations. Cluster 1 (blue) consists of movies with middle-range durations and ratings. This clustering suggests that movies with longer durations tend to have higher ratings, while shorter ones generally receive lower ratings.

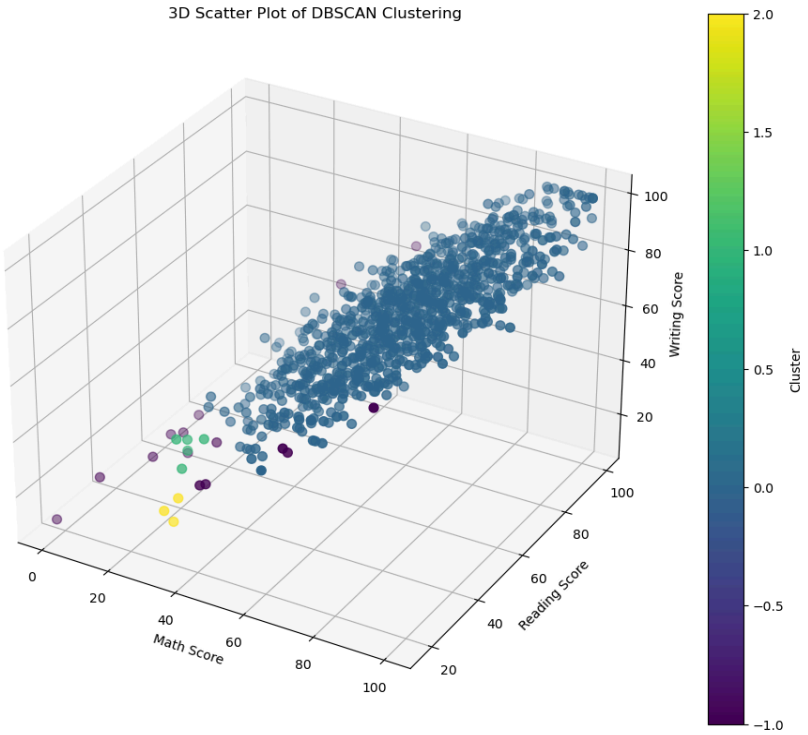
From these plots, we can conclude that K-means effectively grouped the movies into distinct clusters, revealing meaningful patterns related to the duration and rating of movies on Netflix. The separation of these clusters indicates some level of differentiation among movies in terms of their length and audience reception.

Evaluation of the K Means Algorithm:

- Performance: Kmeans is very efficient, especially even if the dataset is large, as it is very efficient.
- Parameter Tuning: We can tune the number of clusters to find the optimal number of clusters for the elbow method but may not be useful for large datasets.
- Correctness: Kmeans is very efficient for particular types of datasets; it works well when the clusters are spherical in shape, but elongated or non-spherical clusters will not be formed. Therefore, K-Means performs poorly in regard to correctness.
- Deterministic: Kmeans is not deterministic due to random initialization, as random initialization can result in different clusterings.

DBSCAN

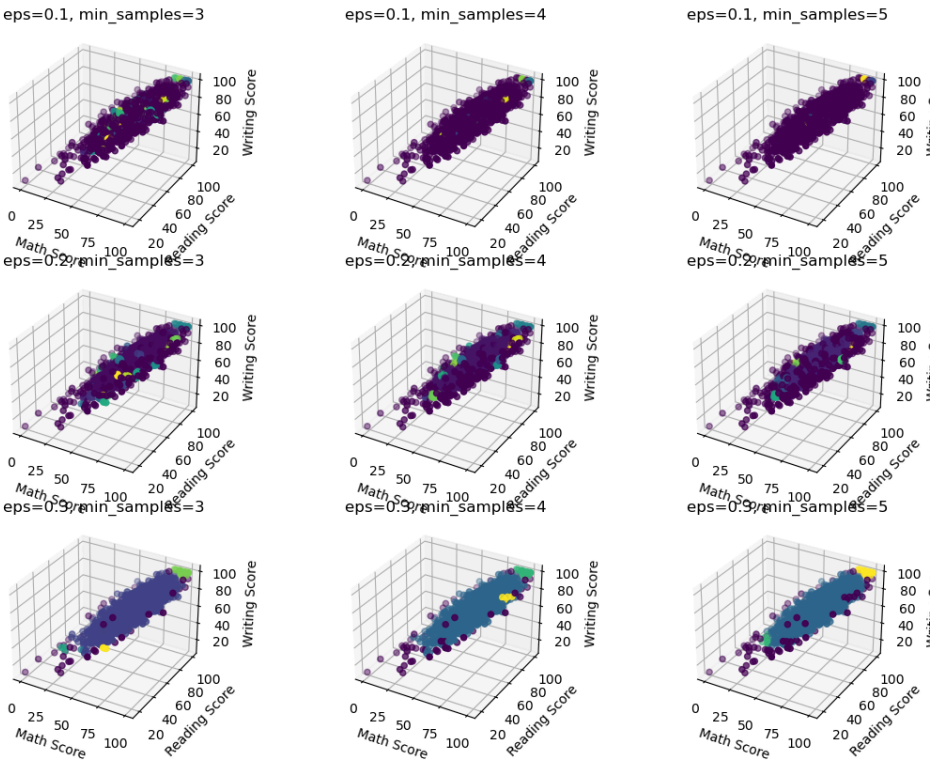
1) **Students Performance dataset:** We have two main parameters, the epsilon and minimum points. For epsilon, we used a k-distance plot and applied the elbow method to the graph at the maximum bend obtained to get a good estimate for epsilon. This point suggests a good value because the distances increase sharply afterwards, indicating the transition from a dense region to a sparse one. For minimum points, we used 4, as it is low-dimension data.



Epsilon: 0.4 | Min points: 4

Cluster	Count
0	977
-1	15
1	5
2	3

The results obtained show that almost all points are considered as cluster 0, with 15 points considered as noise. This is ineffective clustering.



Parameter Tuning

We attempt to vary the parameters to tune it for a better result; however, the results are similarly ineffective.

Analysis:

Why did DBScan fail for the student performance dataset?

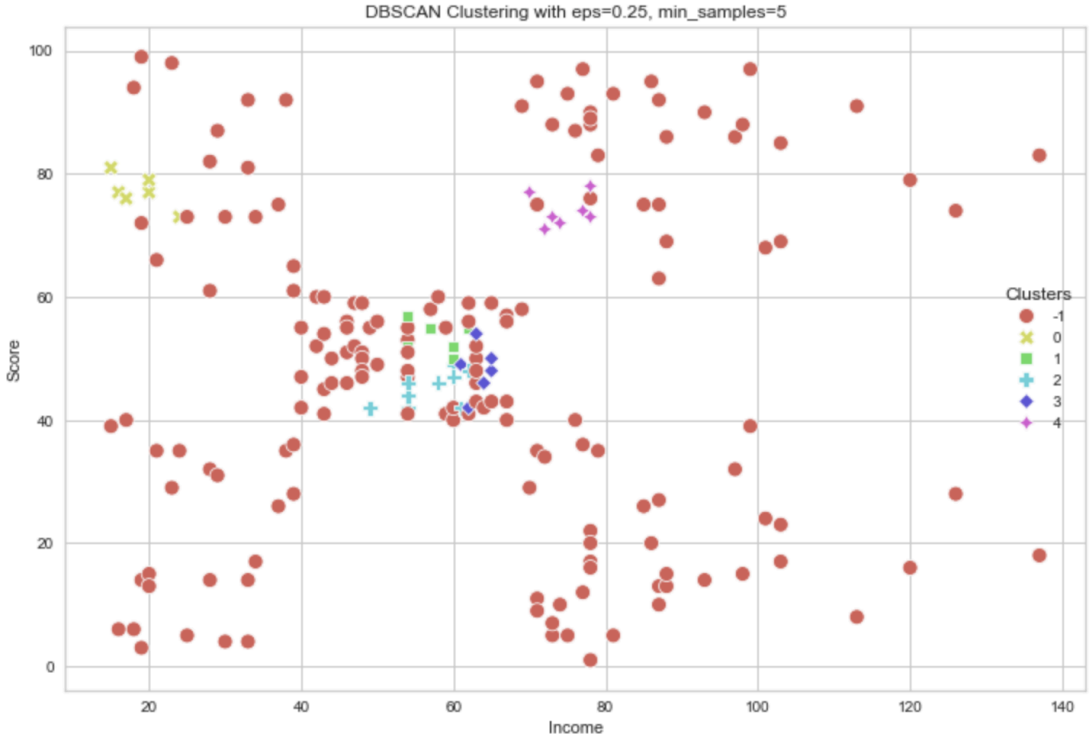
A reason would be that there are no clear, dense regions in the data. DBSCAN relies on finding dense regions in the data. If the data lacks such density (or has overlapping clusters), DBSCAN may fail to separate them effectively. In our data, we do not have clear density separations, thus DBSCAN was ineffective as all data points are considered together as one huge cluster, thus highlighting a weakness of DBSCAN.

2) **Mall Segmentation dataset:** By analysing the dataset, we found that density-based clustering algorithms such as DBSCAN yield inconclusive results in the visualisations.

   This difference in performance to k-means can be attributed to the feature space exhibiting relatively uniform density with indistinct cluster boundaries. DBSCAN link points from different clusters within the same neighbourhood region, accommodating non-spherical clusters.

Analysis: Before Parameter tuning: Default eps=0.25 and min_samples=5

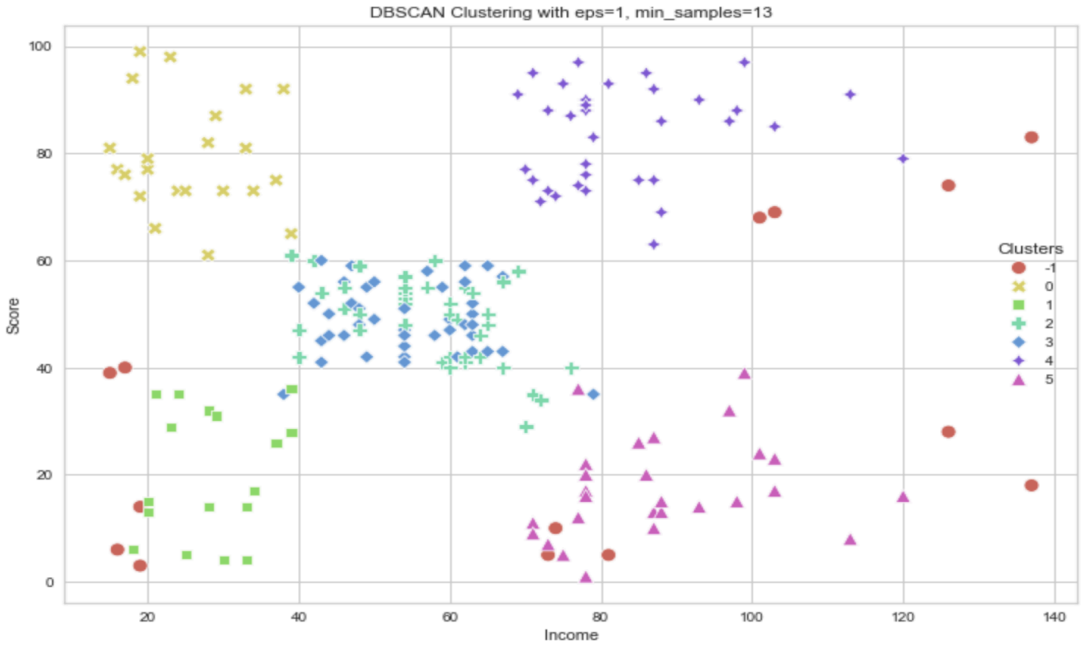
```
DBSCAN Clustering with eps=0.25, min_samples=5
Number of clusters: 6
Clustering took 0.00 s
Silhouette Coefficient: -0.3074851897869123
```



This negative silhouette score of -0.30748 suggests that DBSCAN clustering with eps=0.25 and min_samples=5, resulting in 6 clusters has poorly defined cluster boundaries. Many points are either misclassified, poorly clustered, or assigned as noise. A negative silhouette score indicates that points are often closer to points in neighbouring clusters than to points within their own clusters, leading to weak cohesion and poor separation between clusters.

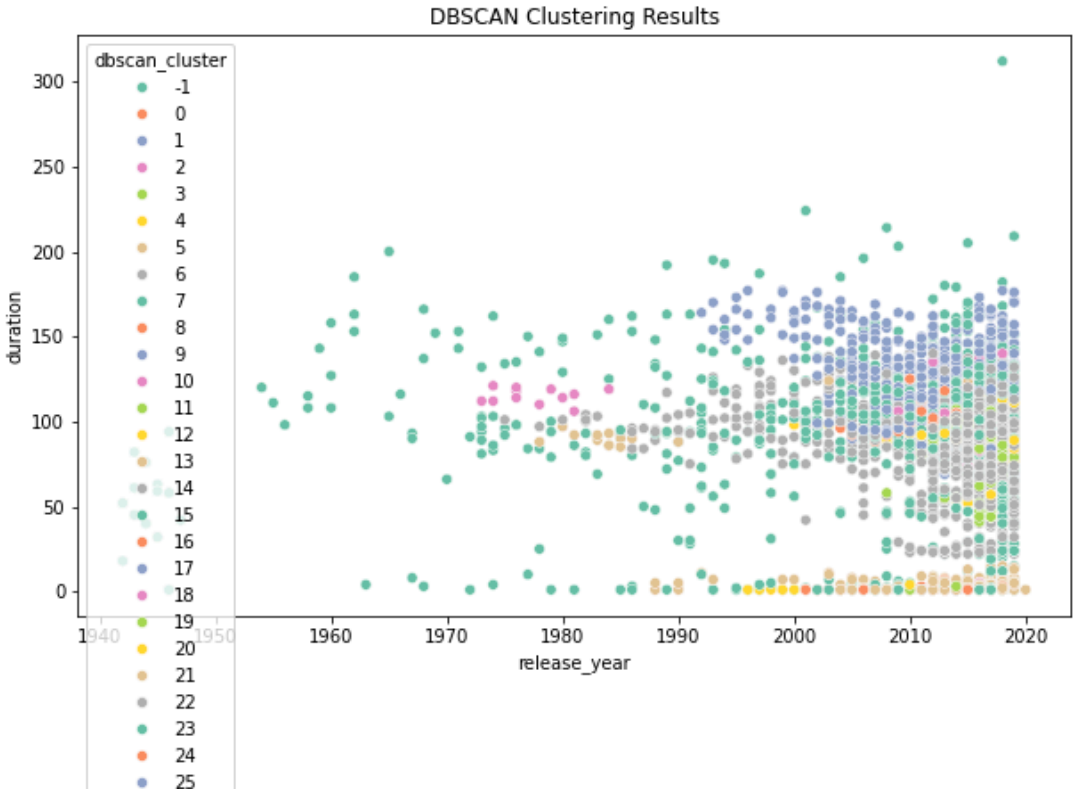
After Parameter tuning: (1) The best silhouette score with the best eps and best min samples iterating over a number of epsilon in the range of (1,200) and min samples in the range of (1,30).

```
New best score: 0.4278476609498341 with eps=1, min_samples=1
New best score: 0.45478711456509 with eps=1, min_samples=2
New best score: 0.5179571560534026 with eps=1, min_samples=3
New best score: 0.5202401032073163 with eps=1, min_samples=5
New best score: 0.5428719285191996 with eps=1, min_samples=13
Best Silhouette Score: 0.5428719285191996 with eps=1, min_samples=13
```

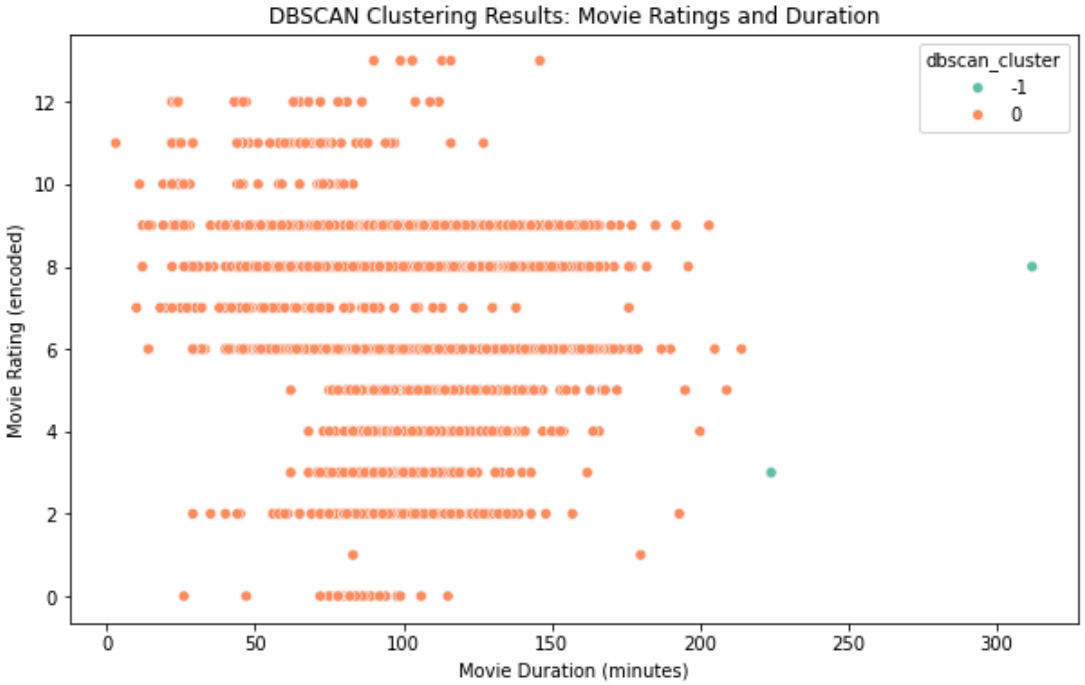


(2) Varying Epsilon: The min_samples is set to 5 and epsilons eps = [0.0175, 0.025, 0.05, 0.1, 0.2, 0.5, 1, 1.5], where we found that testing DBSCAN with eps=1.5 gave number of clusters = 4, silhouette coefficient = 0.33395.
(3) Varying min_samples: The min_samples_values = [1, 5, 10, 15, 20] and Epsilon is set to eps=1.5 where we got the Best Silhouette Score: 0.2694145893687977 with eps=1.5, min_samples=20
By performing a grid search over eps and min_samples, we were able to achieve significant improvements in the performance of the DBSCAN algorithm.

4) Netflix and Movies Dataset:
By analyzing the Netflix dataset, we observed that the density-based clustering algorithm DBSCAN yielded less clear results compared to K-means. This difference can be attributed to the dataset's characteristics, where the density of points is relatively uniform, and there are no sharply defined cluster boundaries. DBSCAN typically excels when identifying clusters of varying shapes and densities, but in this case, it struggled to separate distinct groups effectively.
In contrast to K-means, DBSCAN does not assume spherical clusters, making it well-suited for datasets with irregular shapes. However, due to the relatively homogeneous distribution of data points in this Netflix dataset, DBSCAN linked points from different clusters within the same neighborhood, leading to overlapping clusters and inconclusive visualizations. Therefore, K-means performed better in identifying distinct clusters based on movie ratings and durations.


DBSCAN Clustering Results
dbscan_cluster
-1
0
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
duration
300
250
200
150
100
50
0
1940
1950
1960
1970
1980
1990
2000
2010
2020
release_year

	release_year	duration	dbscan_cluster
0	2019	1.0	0
2	2019	135.0	1
3	2019	106.0	-1
4	2019	2.0	2
5	2018	107.0	3



	duration	rating	dbscan_cluster
1	67.0	TV-G	0
2	135.0	TV-14	0
3	106.0	TV-14	0
5	107.0	TV-14	0
6	81.0	TV-MA	0

Analysis: The images show the results of applying DBSCAN clustering on a movie dataset based on attributes like release year, duration, and ratings. In the first plot, DBSCAN clustered movies by release year and duration. Clusters are colored differently, and outliers are marked as -1. Most movies before 1980 vary in duration, but from 1980 onward, movie durations become more consistent, with many clustering between 80 and 150 minutes. The turquoise outliers indicate movies that deviate significantly in duration.

The second table provides a snapshot of how DBSCAN classified movies based on release year and duration. For instance, movies from 2019 with varying durations (e.g., 1 minute, 135 minutes) were placed into different clusters, while one with a duration of 106 minutes was flagged as noise (cluster -1). This indicates DBSCAN's ability to detect movies that don't fit typical patterns, classifying them as outliers.

In the third plot, clustering was done based on movie ratings and duration. Most movies, regardless of their ratings (like TV-G, TV-14, etc.), fall into cluster 0, with durations ranging from 60 to 150 minutes. However, movies longer than 200 minutes were marked as outliers (cluster -1). The ratings don't significantly affect clustering, but longer films are uncommon, with DBSCAN identifying them as exceptions. This showcases DBSCAN's strength in detecting movies with unusual characteristics based on both their duration and ratings.

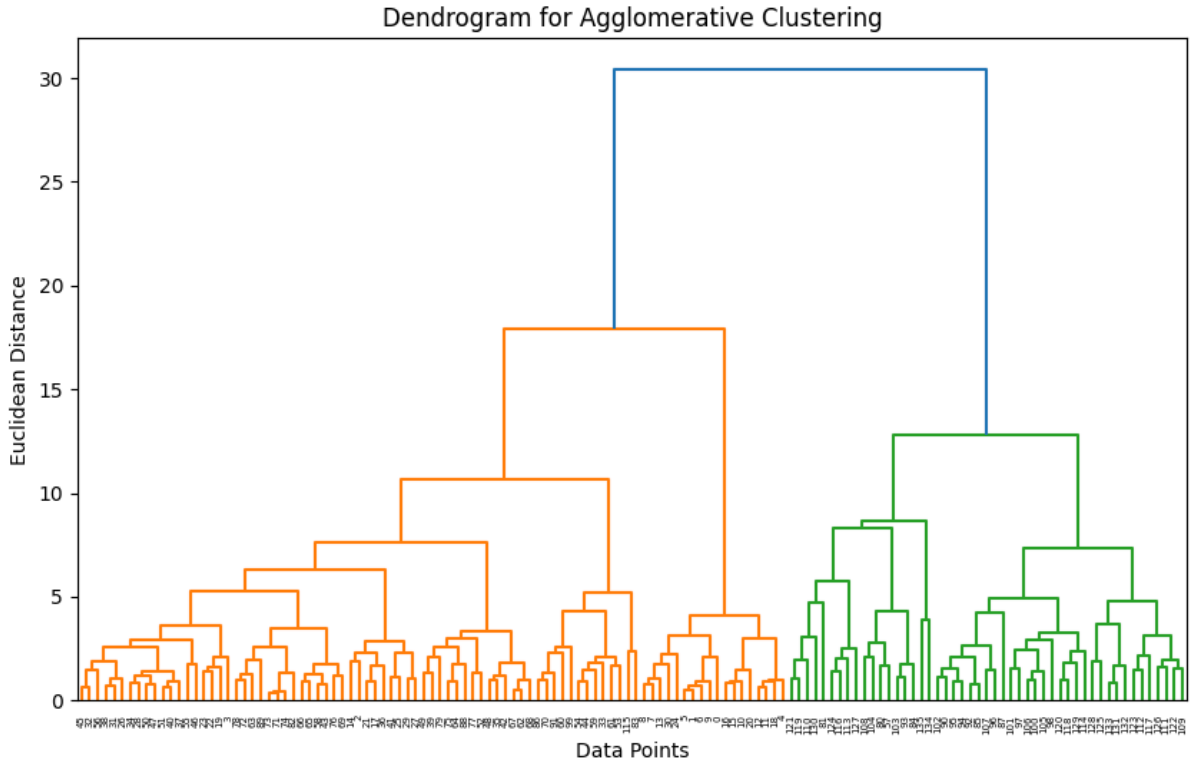
Evaluation of DBSCAN Algorithm

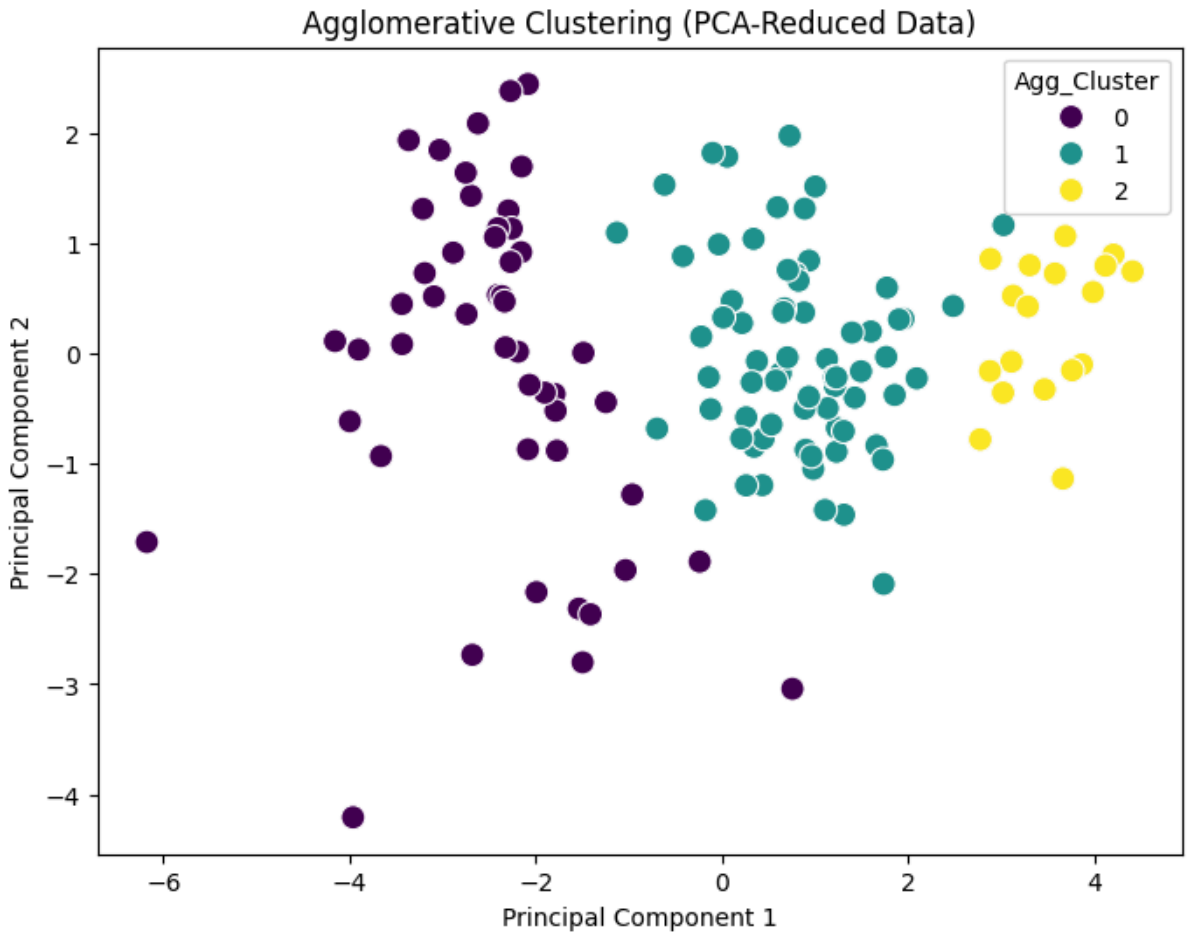
- Performance: DBSCAN is a very simple, powerful, and fast algorithm.
- Parameter Tuning: epsilon and min_samples are crucial for the algorithm's success. Where epsilon is the maximum distance between two points where they are neighbours, we must estimate a good value for epsilon because if set too high, it causes clusters to merge, or when set too low, it can fragment clusters, but its a nontrivial task to find a good epsilon value. Similarly, for min_samples, its non-trivial to find a good value.
- Correctness: DBSCAN is capable of assigning the correct cluster if clusters are similar densities, as it is capable of detecting clusters in datasets with noise and if noise is ignored. It also depends on epsilon and min_samples, where incorrect values give incorrect merging of different clusters.
- Deterministic: DBSCAN produces the same and consistent clustering results as long as the epsilon and min_samples values are not changed.

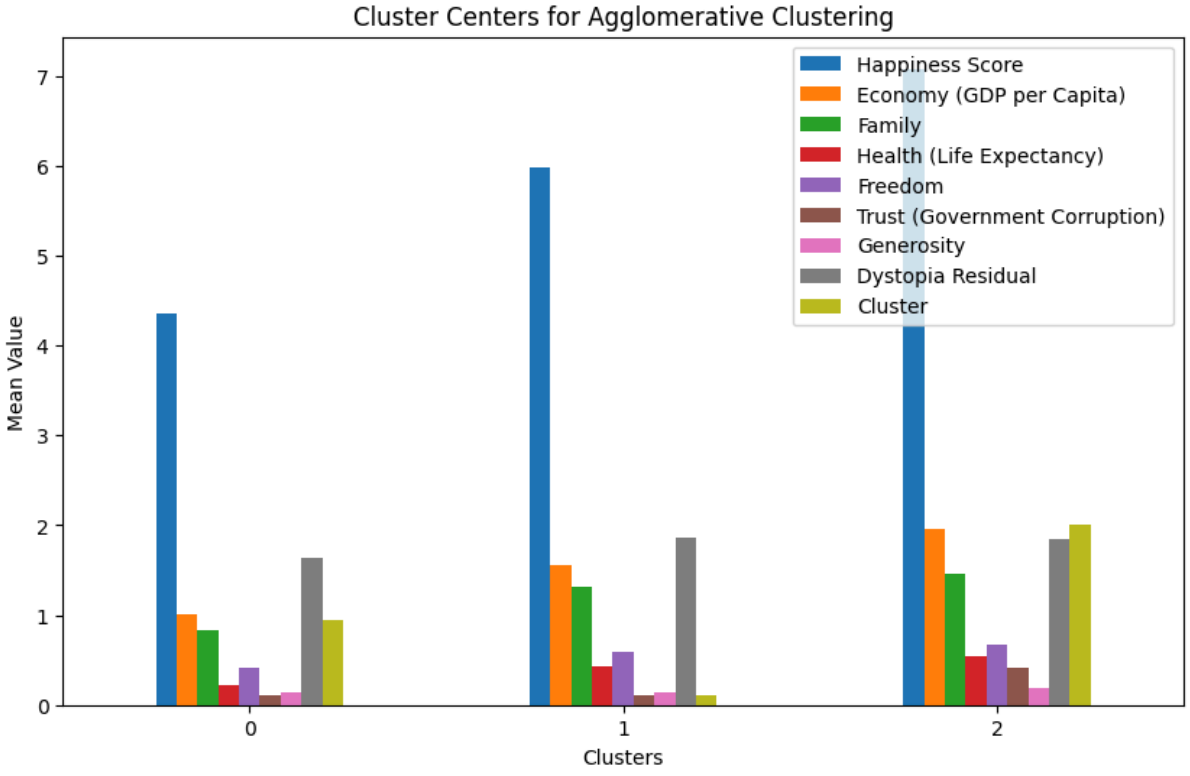
Agglomerative Clustering

3) World Happiness Index 2023

First, we generate a linkage matrix and plot a dendrogram to identify the optimal number of clusters. At the bottom, each data point starts as its own cluster, and as it moves upwards, the clusters begin to merge. Large vertical jumps (like the blue line near the top) indicate significant differences between the clusters being merged.




Agglomerative Clustering (PCA-Reduced Data)
Agg_Cluster
0
1
2
Principal Component 2
Principal Component 1
2
1
0
−1
−2
−3
−4
−6
−4
−2
0
2
4


Cluster Centers for Agglomerative Clustering
Happiness Score
Economy (GDP per Capita)
Family
Health (Life Expectancy)
Freedom
Trust (Government Corruption)
Generosity
Dystopia Residual
Cluster
Mean Value
Clusters
7
6
5
4
3
2
1
0
0
1
2

Analysis: Based on the dendrogram, the optimal number of clusters should be 2. However, we decided to continue to choose 3 as the optimal number as it increases granularity and fits the problem statement. However this suggests that the middle cluster between the highest and lowest happiness score may not be that distinct.

Setting the optimal number of clusters to be 3, in the second plot we used PCA to reduce the dimensionality and plotted the scatterplot.

In plot 3, similarly it shows the happiness score for each cluster allowing us to know that cluster 0 has the lowest average happiness score, cluster 1 with happiness score in the middle and cluster 2, the countries with the highest happiness score.

In plot 2, there exist some overlaps where cluster 0 appears to be in cluster 1 and some countries in cluster 1 appear to be in cluster 2. This supports the observation that there is a distinct difference between the highest and lowest happiness score but not as distinct for those in the middle as supported by the dendrogram.

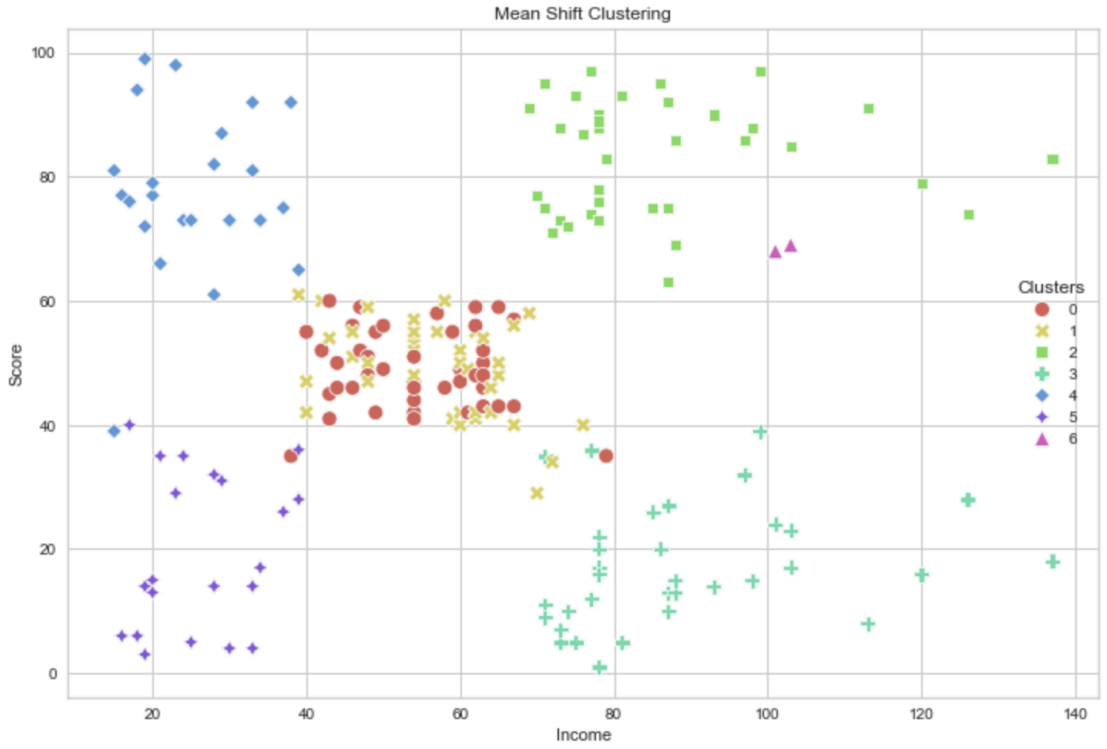
Evaluation of Agglomerative Clustering

- Performance: Agglomerative Clustering can become computationally expensive with large datasets, as it requires calculating the distance between all pairs of points. The time complexity is O(n3)O(n3) for the full linkage method, which makes it less efficient for very large datasets.
- Speed: The performance can be acceptable for medium-sized datasets but tends to slow down significantly as the number of data points increases.
- Parameter Tuning: The choice of linkage method (e.g., single, complete, average, or Ward's linkage) greatly affects the results. Each method considers distances differently and can lead to different cluster shapes and numbers.The selection of the distance metric (e.g., Euclidean, Manhattan) also influences the clustering results.
- Correctness: Agglomerative Clustering is hierarchical and works by merging the closest pairs of clusters iteratively. This may lead to situations where clusters are not formed optimally, especially if the data has a complex structure. However it can struggle with irregularly shaped clusters since it tends to create clusters based on proximity. The algorithm may be sensitive to outliers, as they can influence the distance calculations and impact cluster formation.
- Deterministic: Agglomerative Clustering is generally deterministic, meaning that it will produce the same results across multiple runs, given the same data and parameters. This is advantageous as it provides consistent clustering results.

Mean Shift

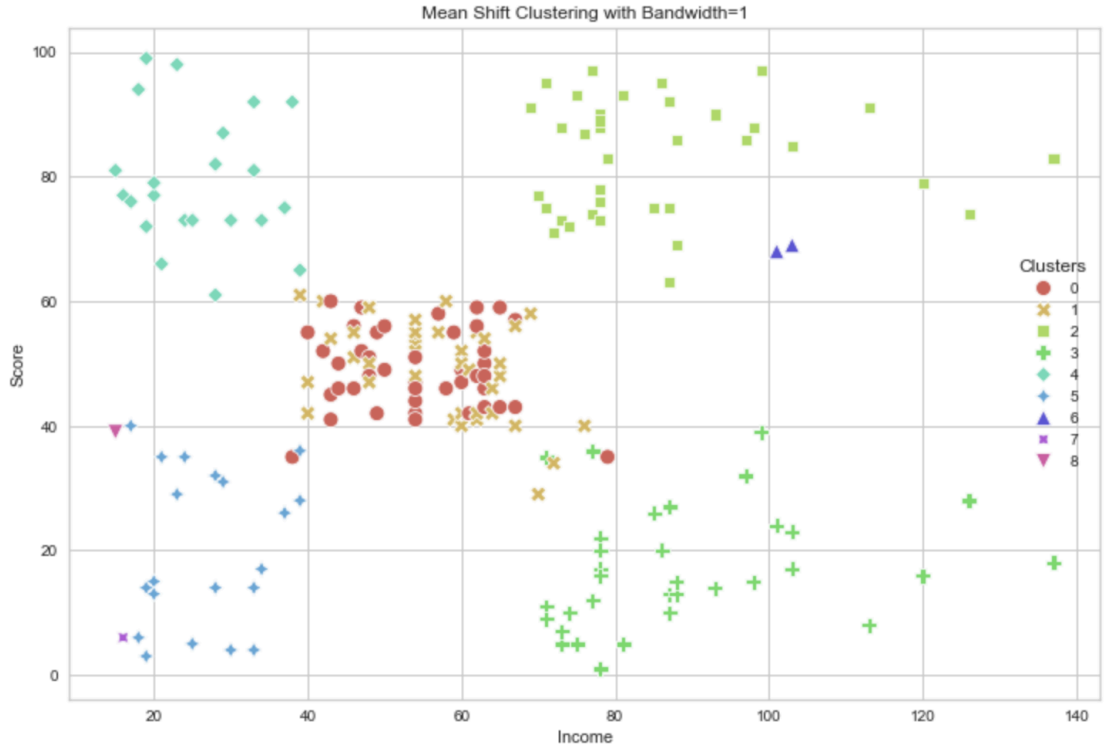
2) Mall Segmentation dataset:

Before Parameter Tuning:



After Parameter Tuning: (1) We found the best silhouette score, bandwidth and number of clusters found by varying bandwidth in the range of (1,100).

```
New best score: 0.5987148193422501, Bandwidth: 1, Clusters: 9
Best Silhouette Score: 0.5987148193422501, Best Bandwidth: 1, Clusters Found: 9
```



(2) Varying Bandwidth: We tested the effect on mean shift using different values of bandwidth. The chosen values are bandwidths = [0.075, 0.1, 0.150, 0.25, 0.3, 0.45]

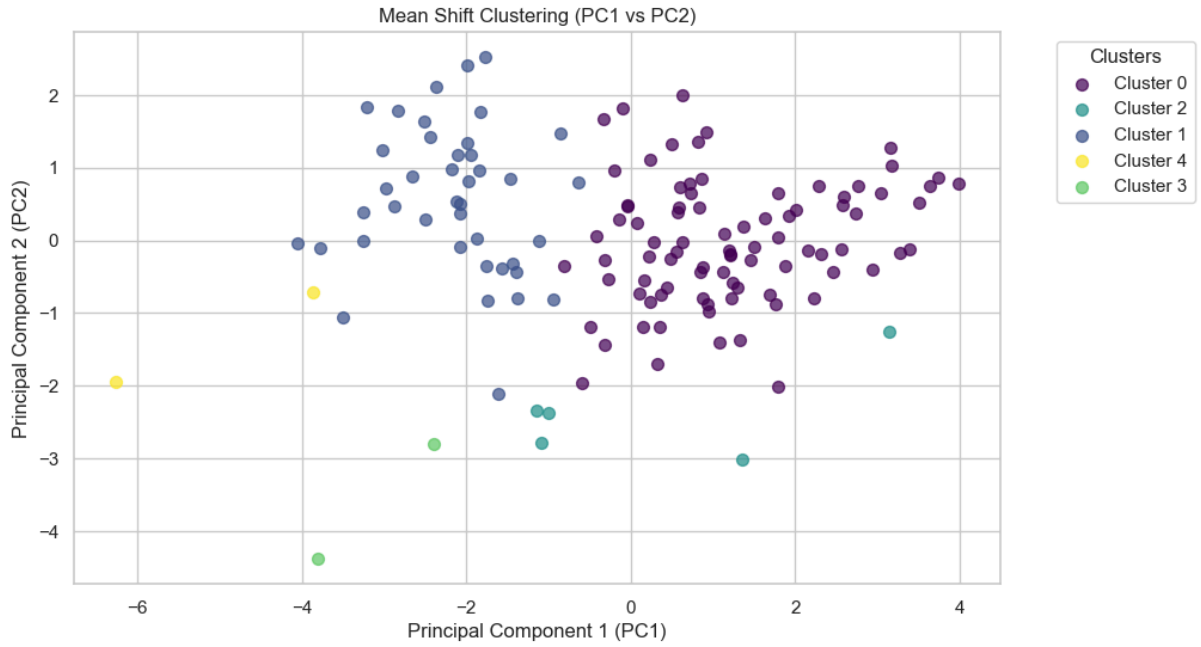
```
Testing MeanShift with Bandwidth = 0.075:
Number of clusters: 198
Clustering took 0.21 s
Silhouette Coefficient: 0.013397210967370033

Testing MeanShift with Bandwidth = 0.1:
Number of clusters: 196
Clustering took 0.21 s
Silhouette Coefficient: 0.02202866932072407

Testing MeanShift with Bandwidth = 0.15:
Number of clusters: 183
Clustering took 0.22 s
Silhouette Coefficient: 0.07809860246024551

Testing MeanShift with Bandwidth = 0.25:
Number of clusters: 131
Clustering took 0.26 s
Silhouette Coefficient: 0.2047471275312107

Testing MeanShift with Bandwidth = 0.3:
Number of clusters: 112
Clustering took 0.27 s
Silhouette Coefficient: 0.24119120949225273

Testing MeanShift with Bandwidth = 0.45:
Number of clusters: 63
Clustering took 0.34 s
Silhouette Coefficient: 0.372952913438738
New best score: 0.013397210967370033, Bandwidth: 0.075, Clusters: 198
New best score: 0.02202866932072407, Bandwidth: 0.1, Clusters: 196
New best score: 0.07809860246024551, Bandwidth: 0.15, Clusters: 183
New best score: 0.2047471275312107, Bandwidth: 0.25, Clusters: 131
New best score: 0.24119120949225273, Bandwidth: 0.3, Clusters: 112
New best score: 0.372952913438738, Bandwidth: 0.45, Clusters: 63

Best Silhouette Score: 0.372952913438738, Best Bandwidth: 0.45, Clusters Found: 63
```

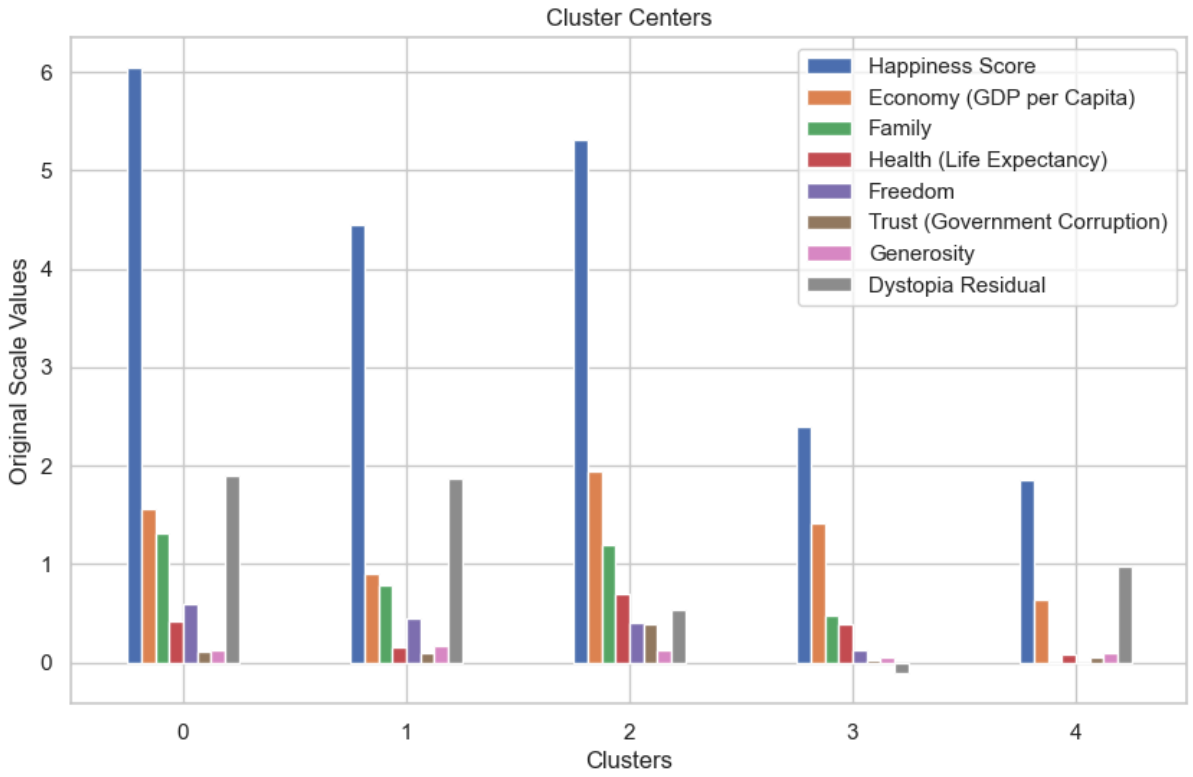
Analysis: We found that as bandwidth increases the number of clusters decreases. As smaller bandwidth results in narrower search area so after points are shifted to local density maximum, still many points will be remained outside that neighbourhood then these points will be clustered into additional clusters, which results in the increase in the number of clusters.

The silhouette coefficient increases as bandwidth increases, mean shift assumes globular clusters and the clusters in the dataset are globular.

3) World Happiness Index 2023

We applied the Mean Shift algorithm to the dataset, using the `estimate_bandwidth()` function, it returned a value of 2.5572. Using the estimated bandwidth, we applied PCA to reduce the dimensionality of the indicators and plotted a scatter plot. We then plot the cluster centres to make sense of each cluster.





Analysis:

In plot 2, we can see that cluster 0 belongs to the countries with the highest happiness score followed by 2, 1, 3 and 4. It gives a more granular view compared to 3 clusters. In plot 1 some clusters have overlapping areas which indicate some similarity. While the clusters that are further apart suggest they have more distinct features.

Feature	Cluster 0	Cluster 1	Cluster 2	Cluster 3	Cluster 4
Happiness Score	6.04	4.45	5.31	2.39	1.86
Economy (GDP per Capita)	1.57	0.91	1.95	1.42	0.64
Family	1.31	0.79	1.20	0.48	0.00
Health (Life Expectancy)	0.43	0.16	0.70	0.40	0.09
Freedom	0.59	0.46	0.41	0.12	0.00
Trust (Gov. Corruption)	0.11	0.10	0.39	0.03	0.06
Generosity	0.13	0.17	0.12	0.06	0.09
Dystopia Residual	1.90	1.86	0.54	-0.11	0.98

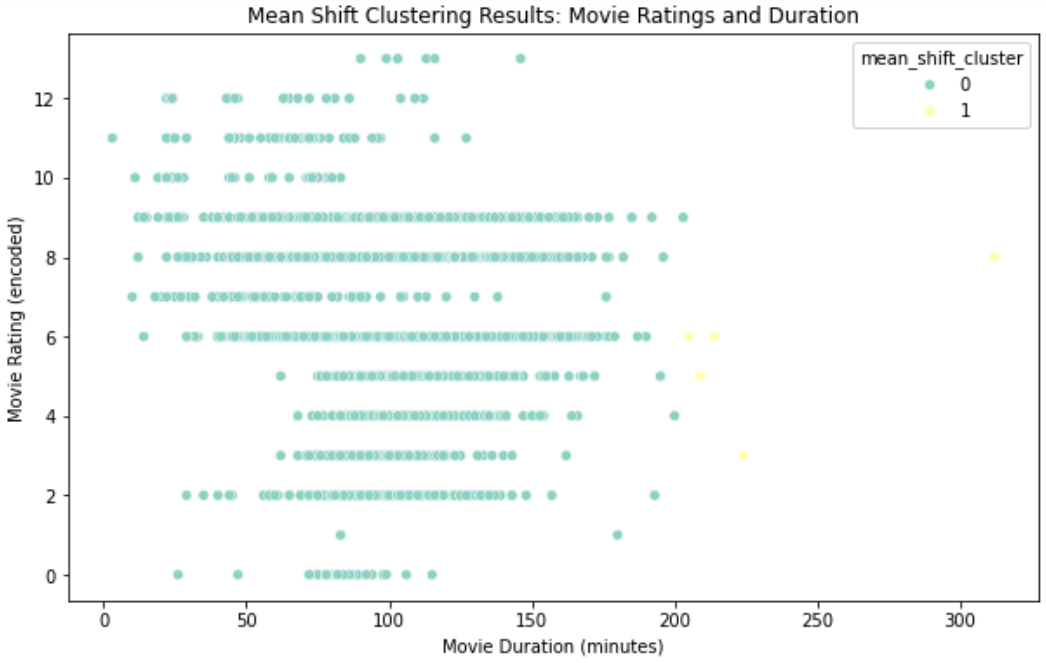
This can be further supported by plot 2 and taking a further look into the actual values.

- Cluster 0: High happiness score, strong GDP, and good life expectancy. Represents countries with overall positive well-being.
- Cluster 1: Moderate happiness and economy, with decent family and freedom scores. Likely a mix of developing and mid-level countries.

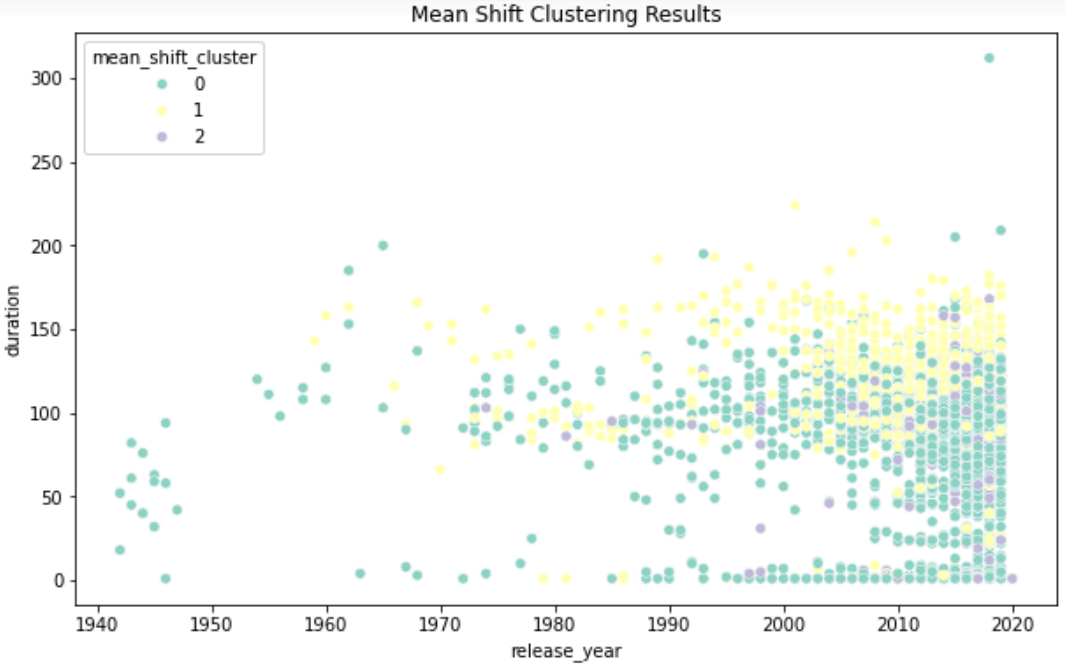
- Cluster 2: High GDP and family support, with good life expectancy and moderate happiness, but relatively lower generosity.
- Cluster 3: Low happiness, freedom, and corruption trust scores. Likely countries with governance and societal challenges.
- Cluster 4: Lowest happiness, GDP, and freedom, but some resilience with a positive dystopia residual.

4) Netflix and Movies Dataset:

The Mean Shift algorithm was applied to the Netflix dataset as another clustering approach. Mean Shift is a non-parametric algorithm that does not require the number of clusters to be specified beforehand, unlike K-means. Instead, it attempts to discover the optimal number of clusters by shifting data points towards areas of higher data point density (modes). In the code, we used the `bandwidth` parameter to control the window size for detecting clusters, with bandwidth estimated using the `estimate_bandwidth()` function.



	duration	rating	mean_shift_cluster
1	67	TV-G	0
2	135	TV-14	0
3	106	TV-14	0
5	107	TV-14	0
6	81	TV-MA	0



	release_year	duration	mean_shift_cluster
0	2019	1.0	0
2	2019	135.0	1
3	2019	106.0	1
4	2019	2.0	2
5	2018	107.0	1

Analysis:
(1) Interpretation of First Image (Movie Ratings and Duration)

The first image shows a scatter plot that maps movie duration (x-axis) against movie ratings (y-axis), with two clusters identified by the Mean Shift algorithm. Cluster 0 (in cyan) contains the majority of movies, with durations ranging from 0 to 200 minutes and various ratings. This cluster represents the bulk of mainstream films.

In contrast, Cluster 1 (in yellow) consists of outliers, characterised by longer durations (100 to 250 minutes) and mid-to-high ratings. These may represent less common, longer films like documentaries or epics. The clustering highlights a clear distinction between typical movies and these outliers.

(2) Interpretation of Second Image (Release Year and Duration)

The second image presents movie durations against their release years, identifying three clusters. Cluster 0 (cyan) includes shorter films (0-100 minutes) spanning from the 1950s to the present, showing that short films have always been prevalent.

Cluster 1 (yellow) captures longer films, primarily released after 1970, suggesting a trend toward extended runtimes in modern cinema. Cluster 2 (purple) represents rare outliers, either very long films or those released during specific periods.

Evaluation of Mean Shift

- Performance: Mean shift is slow and doesn't scale with large datasets.
- Parameter Tuning: Choice of bandwidth value is crucial; it takes trial and error to find the optimal value.
- Correctness: It doesn't cluster every data point into the cluster, but according to the algorithm, it may struggle with irregular-shaped clusters by merging them with nearby clusters. So, the performance can be suboptimal for data with irregular cluster shapes.
- Deterministic: The results may vary due to random initialization for the same bandwidth across different runs.

Conclusion

In this report, we examined several clustering algorithms across different datasets and assessed their effectiveness in various situations. While most algorithms fulfil the fundamental task of identifying clusters, certain methods prove to be more beneficial depending on the specific context. Different datasets exhibit different data characteristics, thus, the model should be chosen in alignment with the specific features and patterns present in the data.