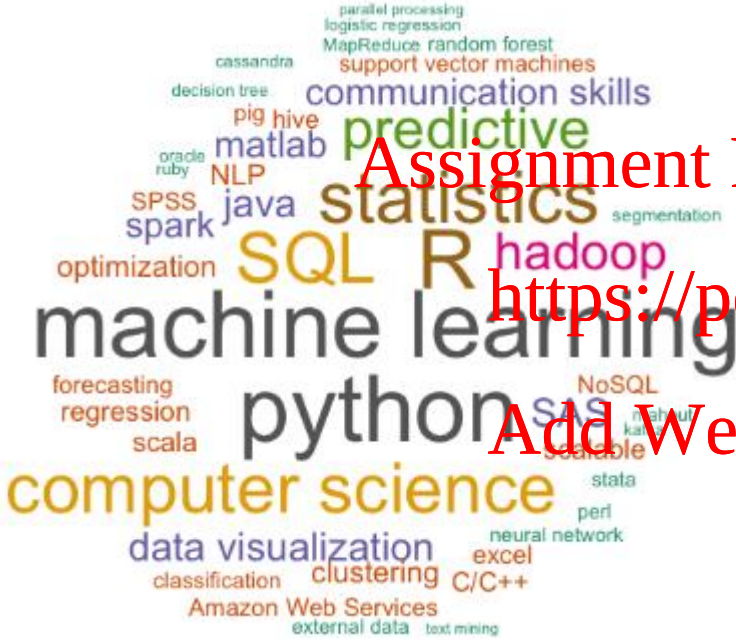
UNSUPERVISED LEARNING



Contents

Introduction

Machine learning focuses primarily on supervised learning but the vast majority of the available data is unlabelled!

- Most of the applications of ML today are based on supervised learning
- The vast majority of the available data is unlabeled!
  - Having the input features X but not the labels y
    - To develop a regular binary classifier to predict whether an item shown in a picture is defective or not, you will need to label every single picture as “defective” or “normal”
    - Labelling generally requires human experts to manually go through all the pictures
    - A long, costly, and tedious task, so usually done on only a small subset of the available pictures
    - The labeled dataset will be quite small and the classifier’s performance will be disappointing
    - Every time any change is made to the system, the labelling process will need to be repeated

Unsupervised Learning

Unsupervised learning refers to the use of ML algorithms to identify patterns in datasets containing data points that are neither classified nor labeled. The algorithms are thus allowed to classify, label and/or group the data points contained within the datasets without having any external guidance in performing that task. The ML algorithms will group data points according to similarities and differences even though there are no categories provided.

Unsupervised learning algorithms can only learn from samples themselves as there is no data labels to learn from

- In unsupervised learning, there is no hidden teacher, the main goals cannot be related to minimizing the prediction error with respect to the ground truth
- Unsupervised learning algorithms have to learn some pieces of information without any formal indication
- The only option is to learn from the samples themselves
- An unsupervised algorithm is usually aimed at discovering the similarities and patterns among samples or reproducing an input distribution given a set of features drawn from it

Unsupervised learning can be more unpredictable than supervised learning, such as creating clutter instead of order

- Unsupervised learning can be more unpredictable than a supervised learning model
  - An unsupervised learning system might, for example, figure out on its own how to sort cats from dogs
  - Such an unsupervised learning might also add unforeseen and undesired categories to deal with unusual breeds, creating clutter instead of order
- ML systems capable of unsupervised learning are often associated with generative learning models
- Chatbots, self-driving cars, facial recognition programs, expert systems and robots are among the systems that may use either supervised or unsupervised learning approaches, or both

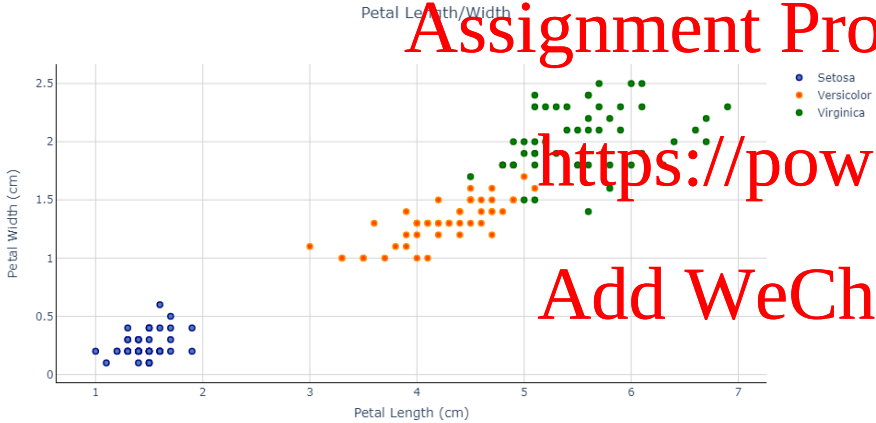
Cluster Analysis / Clustering

Clustering

The task of identifying like with like and assigning them to clusters or group of similar instances. Just like in classification, each instance gets assigned to a group. However, unlike classification, clustering is an unsupervised task. Also, clustering has no notion of correctness.

Classification uses labelled data whereas clustering uses unlabelled data

Samples with labels



Samples without labels

Are you able to identify the clusters?

Clustering algorithms can identify the 3 clusters fairly well making only 5 mistakes out of 150 samples!

Data preparation use cases

- Data analysis
  - When you analyze a new dataset, it can be helpful to run a clustering algorithm, and then analyze each cluster separately
- Dimensionality reduction
  - Once a dataset has been clustered, it is usually possible to measure each instance's affinity with each cluster (affinity is any measure of how well an instance fits into a cluster)
  - Each instance's feature vector x can then be replaced with the vector of its cluster affinities
  - If there are k clusters, then this vector is k-dimensional
  - This vector is typically much lower-dimensional than the original feature vector, but it can preserve enough information for further processing

Data preparation use cases

- Semi-supervised learning
  - If you only have a few labels, you could perform clustering and propagate the labels to all the instances in the same cluster
  - This technique can greatly increase the number of labels available for a subsequent supervised learning algorithm, and thus improve its performance
- Anomaly detection (outlier detection)
  - Any instance that has a low affinity to all the clusters is likely to be an anomaly
  - For example, if you have clustered the users of your website based on their behavior, you can detect users with unusual behavior, such as an unusual number of requests per second
  - Anomaly detection is particularly useful in detecting defects in manufacturing, or for fraud detection

Customer segmentation, recommendation system & image segmentation use cases

- Customer segmentation
  - For marketing campaigns and recommender systems
- Search engines
  - Some search engines let you search for images that are similar to a reference image
  - To build such a system, you would first apply a clustering algorithm to all the images in your database; similar images would end up in the same cluster
  - Then when a user provides a reference image, all you need to do is use the trained clustering model to find this image's cluster, and you can then simply return all the images from this cluster
- Segment an image
  - By clustering pixels according to their color, then replacing each pixel's color with the mean color of its cluster, it is possible to considerably reduce the number of different colors in the image
  - Image segmentation is used in many object detection and tracking systems, as it makes it easier to detect the contour of each object

Clustering algorithms group samples according to their similarities, which capture the distances between samples

$$d_{sim}(\bar{x}_i, \bar{x}_j) = \frac{1}{\delta(\bar{x}_i, \bar{x}_j) + \epsilon}$$

d_{sim} measures the similarity between 2 vectors

$X = \{\bar{x}_1, \bar{x}_2, \cdots, \bar{x}_N\}$ is dataset to be clustered

N is the number of data points in the dataset

ϵ is a constant introduced to avoid division by 0

$$\delta(\bar{x}_i, \bar{x}_j) = \sqrt{\sum_{l=1}^{m} (x_{il} - x_{jl})^2}$$

δ measures the Euclidean distance between 2 vectors

m is the number features in a vector

$\bar{x}_i = (x_{i1}, x_{i2}, \cdots, x_{im})$ is a sample vector

$$C_i = \{\bar{x}_j : d_{sim}(\bar{x}_j, \bar{\mu}_i) > d_{sim}(\bar{x}_j, \bar{\mu}_k)\}$$

C_i, C_k are clusters generated by the clustering algorithm

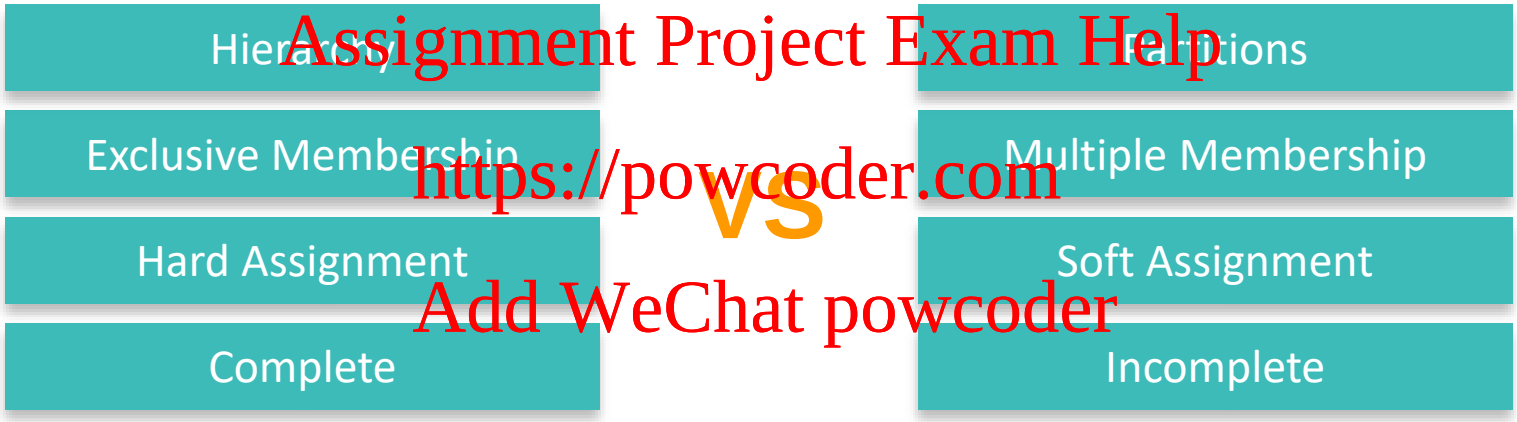
$\bar{\mu}_i$ is a representative vector of C_i

$\bar{\mu}_k$ is a representative vector of C_k

$k \in \{1, 2, \cdots, i-1, i+1, \cdots, K\}$

K is the number of clusters

Cluster algorithms produce different types of clustering results

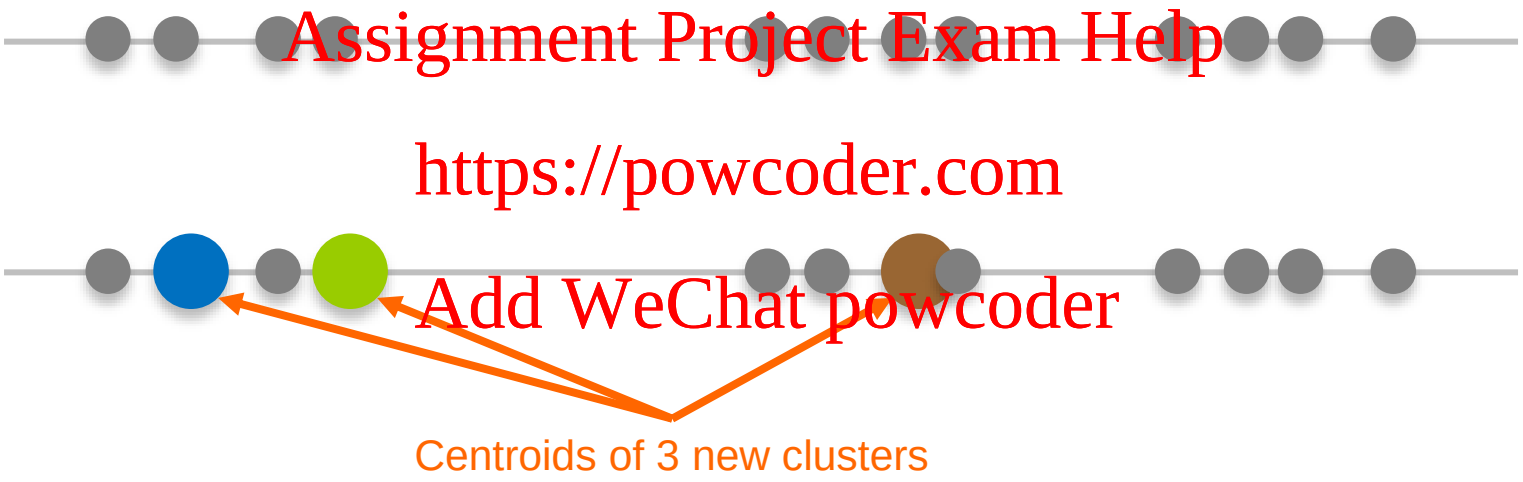


K-Means Clustering

Linear Space

The challenge is to get a computer to identify the same three clusters that are relatively obvious to the naked eyes

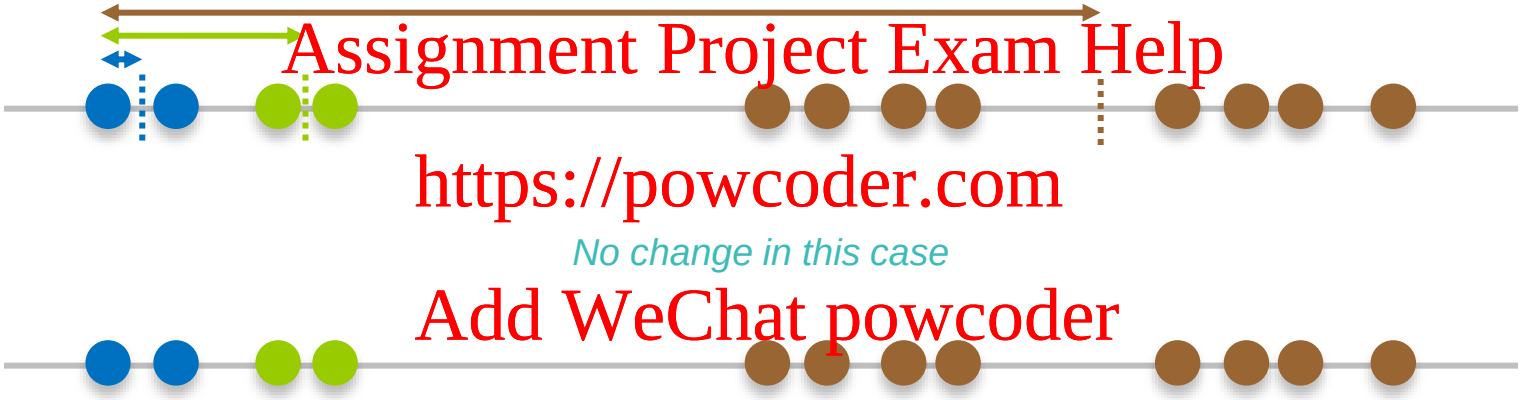
Select the number of clusters (K=3) to identify in the dataset and randomly select 3 data points as cluster centroids



For each data point, find the closest centroid to each data point and assign the corresponding cluster to the data point

For each cluster, calculate the new centroid using the cluster's data points

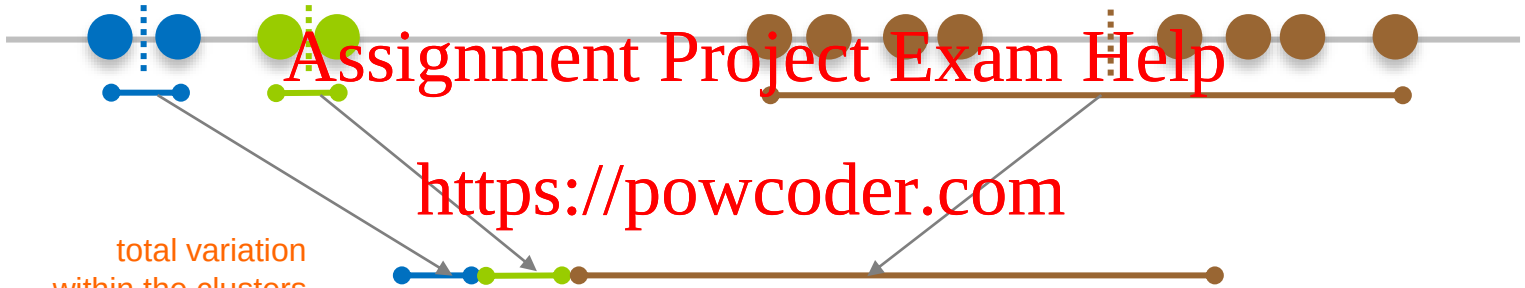
For each data point, re-cluster it to the cluster corresponding to the closest centroid



The clustering algorithm has converged!

Is that the end? No, not when working in a linear space!

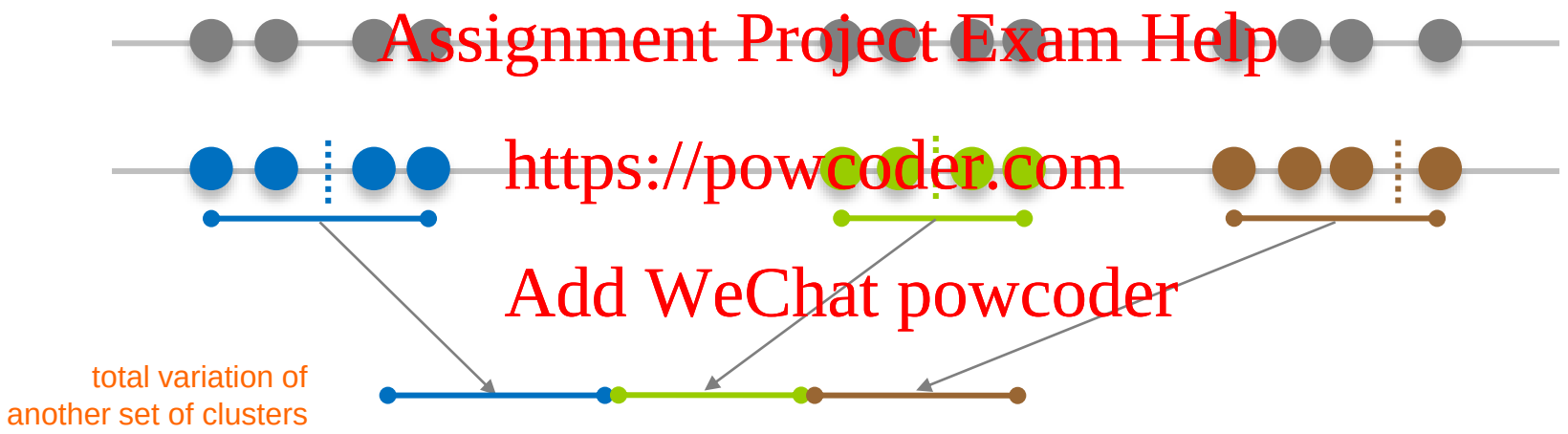
Quality of the clustering can be assessed through adding up the variation within each cluster



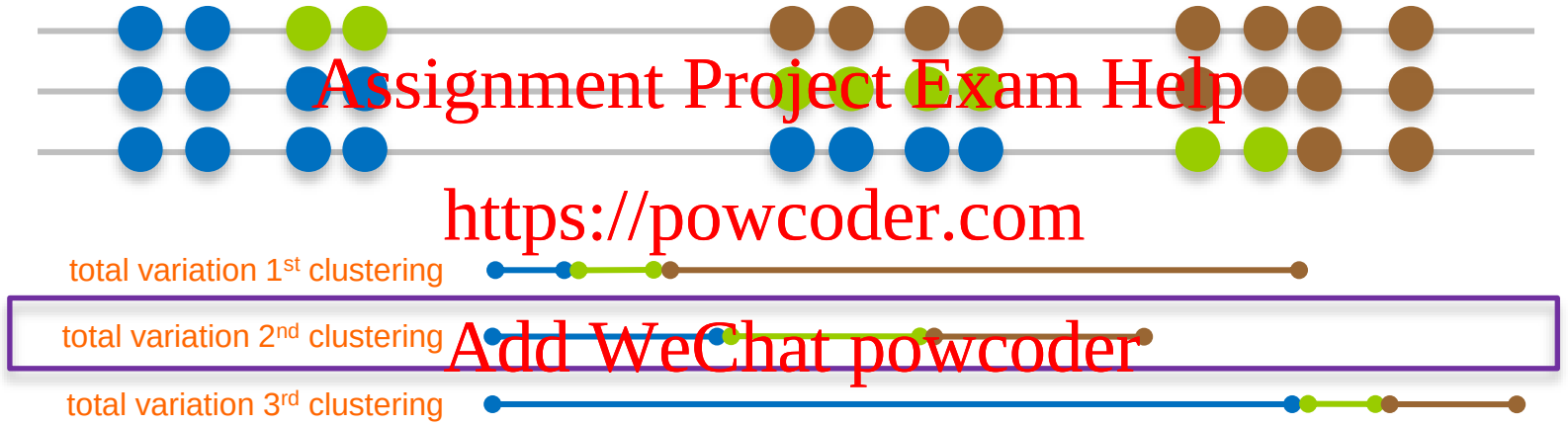
As far as the algorithm goes, it is not clear if this clustering is the best and therefore the predicted clustering.

The algorithm can only repeat the process with different initial centroids and rate its quality using the total variance within the clusters.

Calculate the total variation resulted from using the 3 randomly picked new centroids



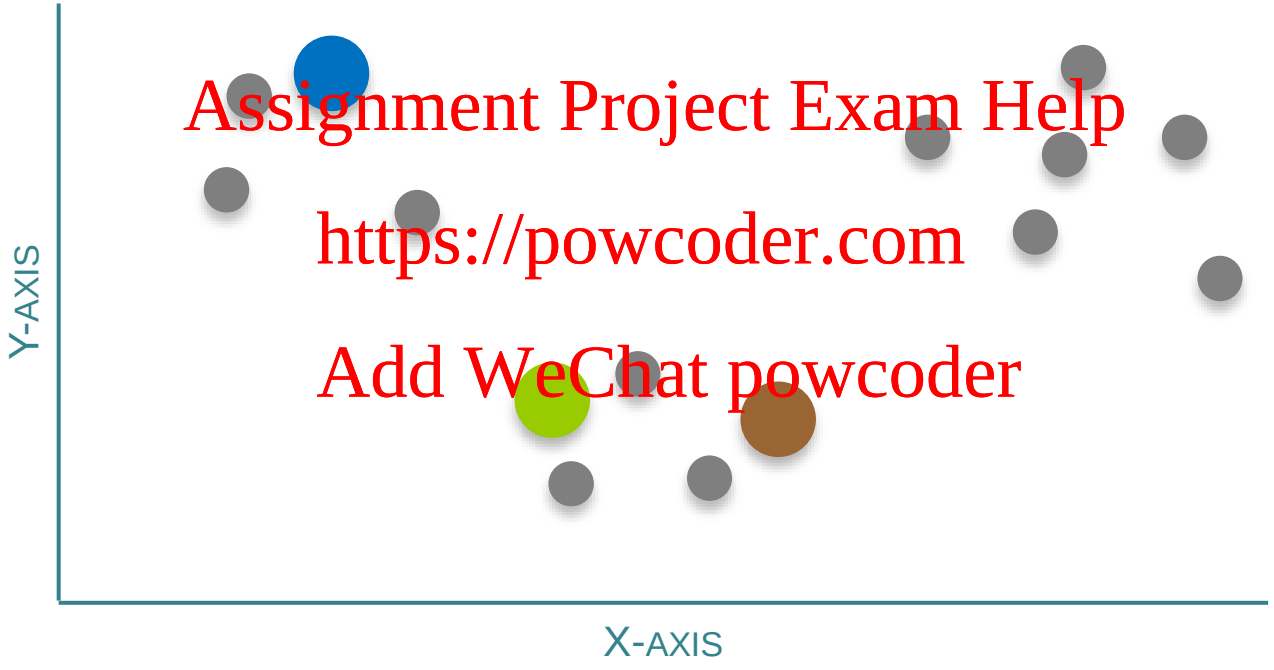
Iterate the clustering with new centroids and record the corresponding total variation



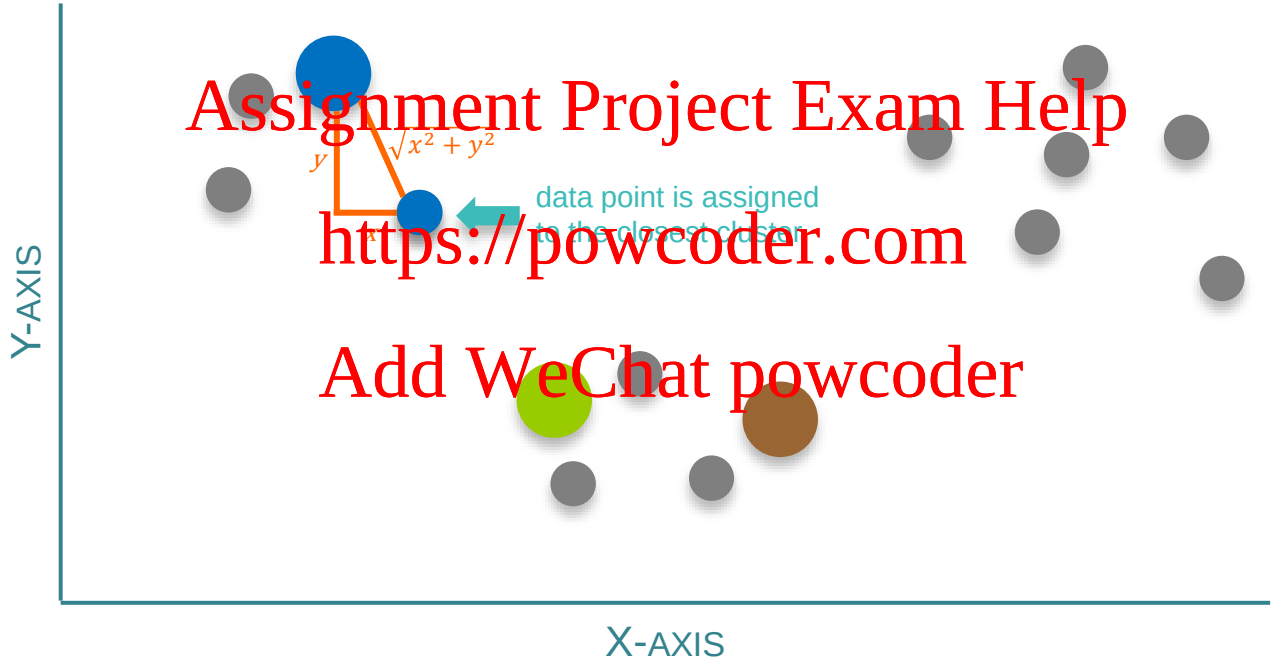
The algorithm will do a few iterations of clustering (it will do as many as you tell it to do) and suggest the one with the least total variation.

Multi-dimensional Space

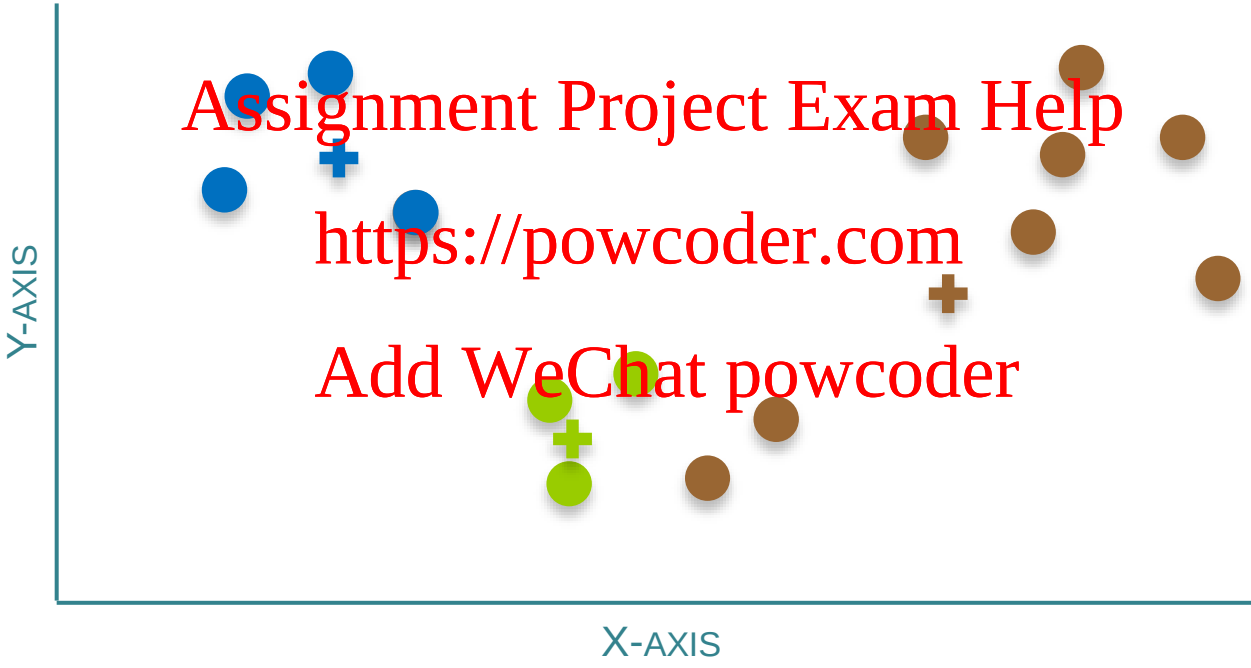
In the same fashion, initial centroids are selected in the multi-dimensional space



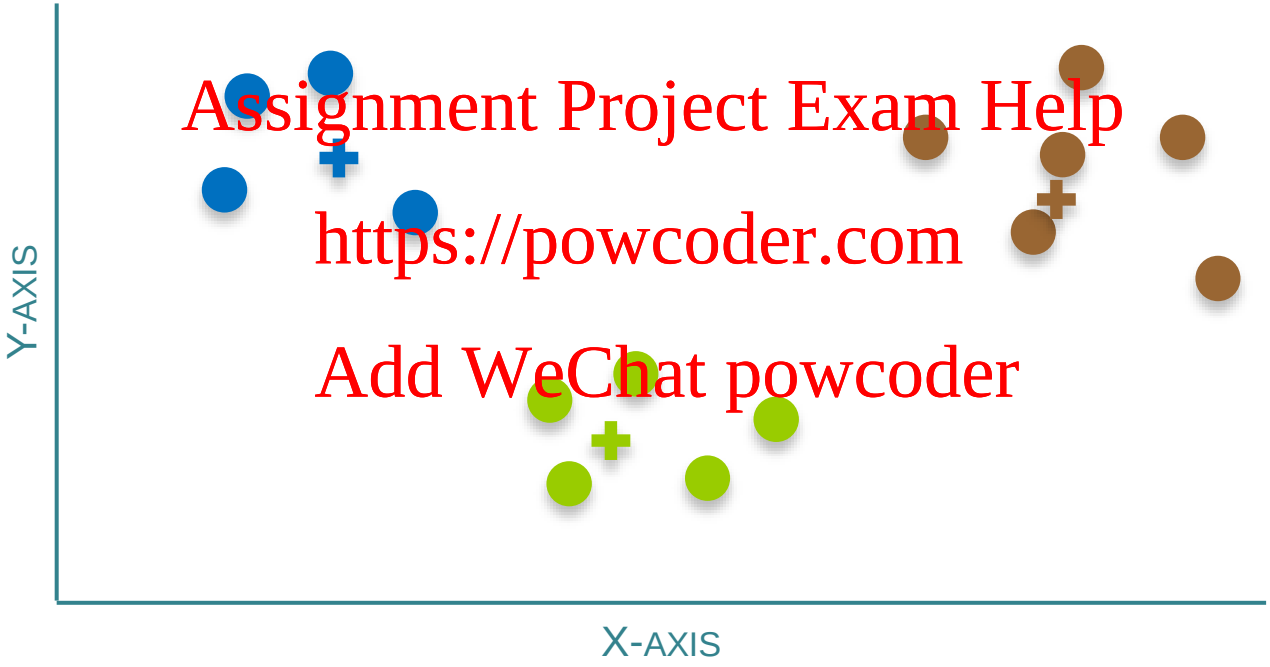
The Euclidean distances of each data point from the three clusters are then measured to decide the clustering



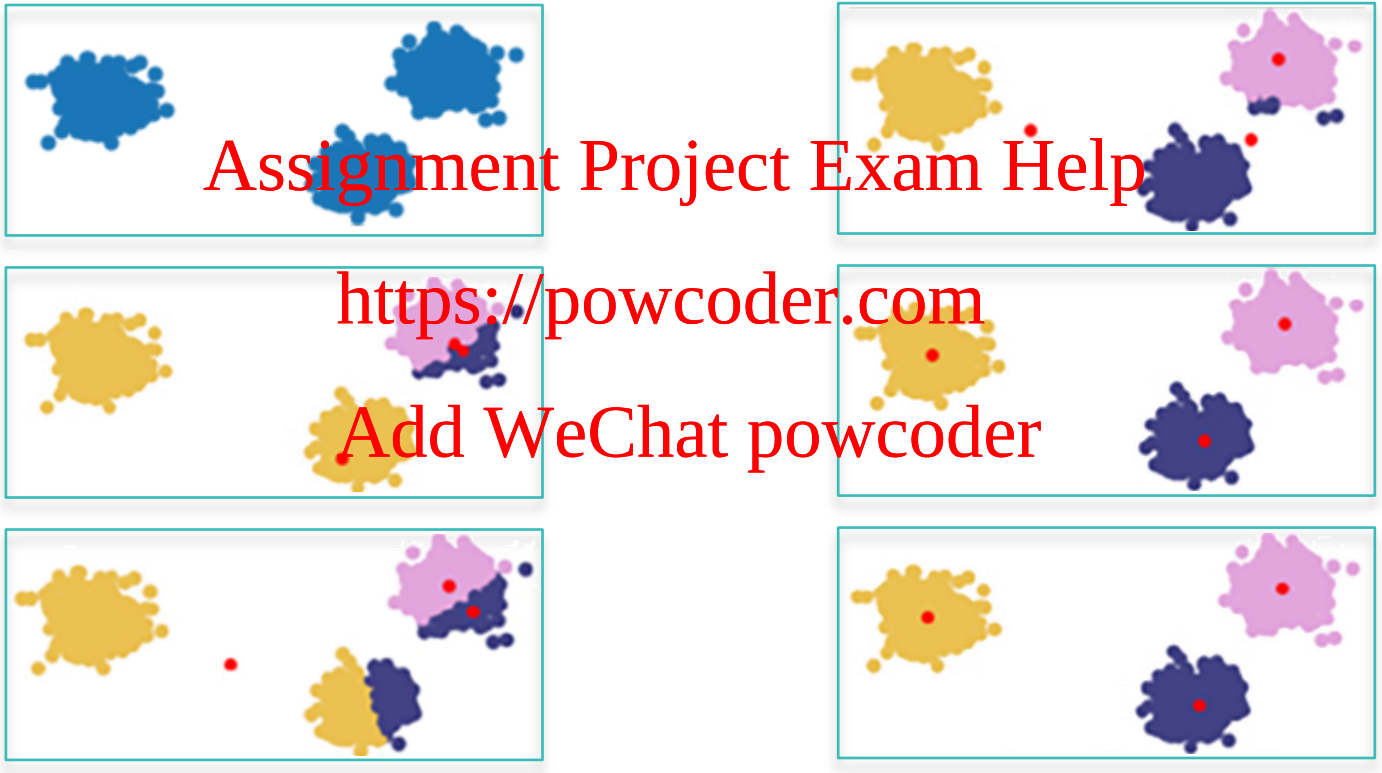
The centre of each cluster is then calculated and all data points will be re-clustered using the new centres

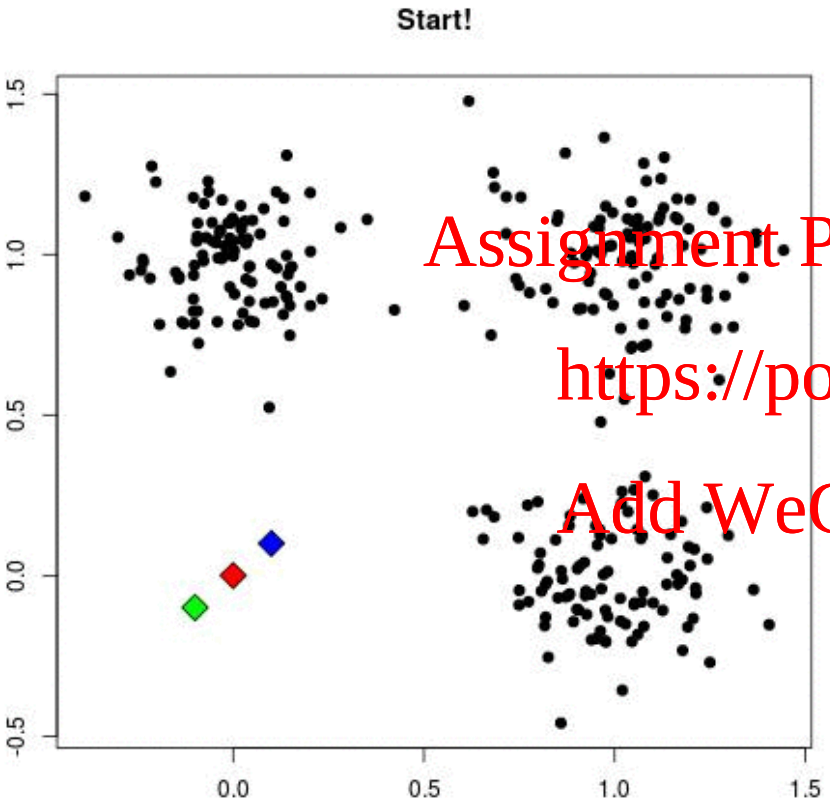


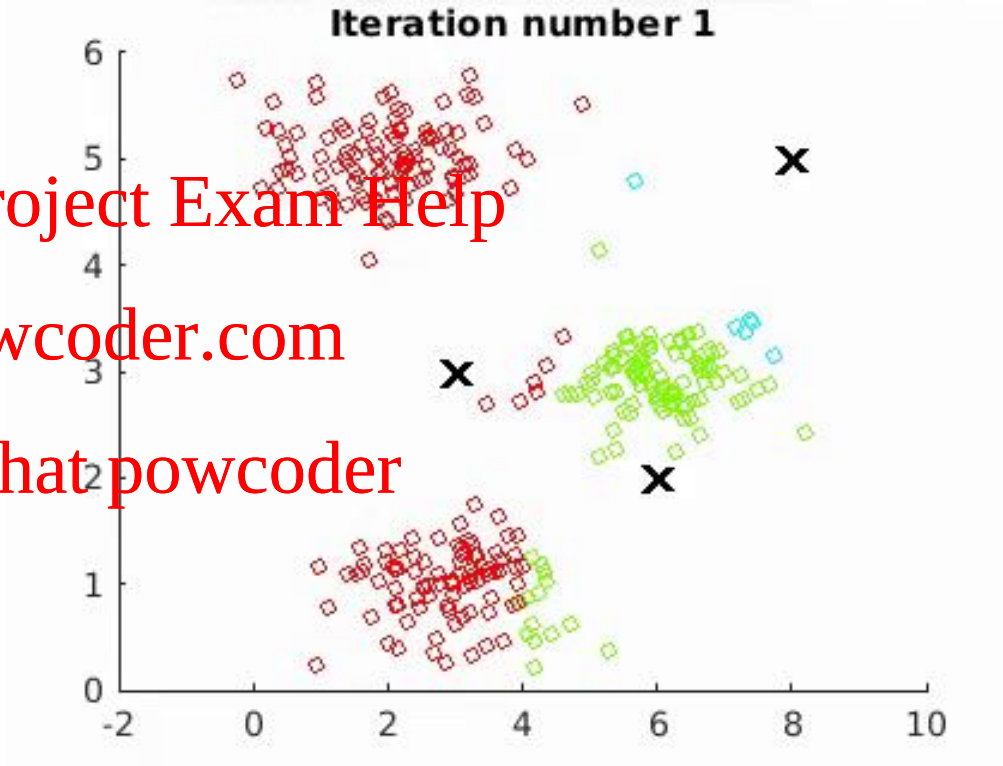
Repeat the process until the centroid values converge or maximum iteration limit has been achieved



Recalculating the centroids effectively formulate an optimal clustering but it may not be globally optimal





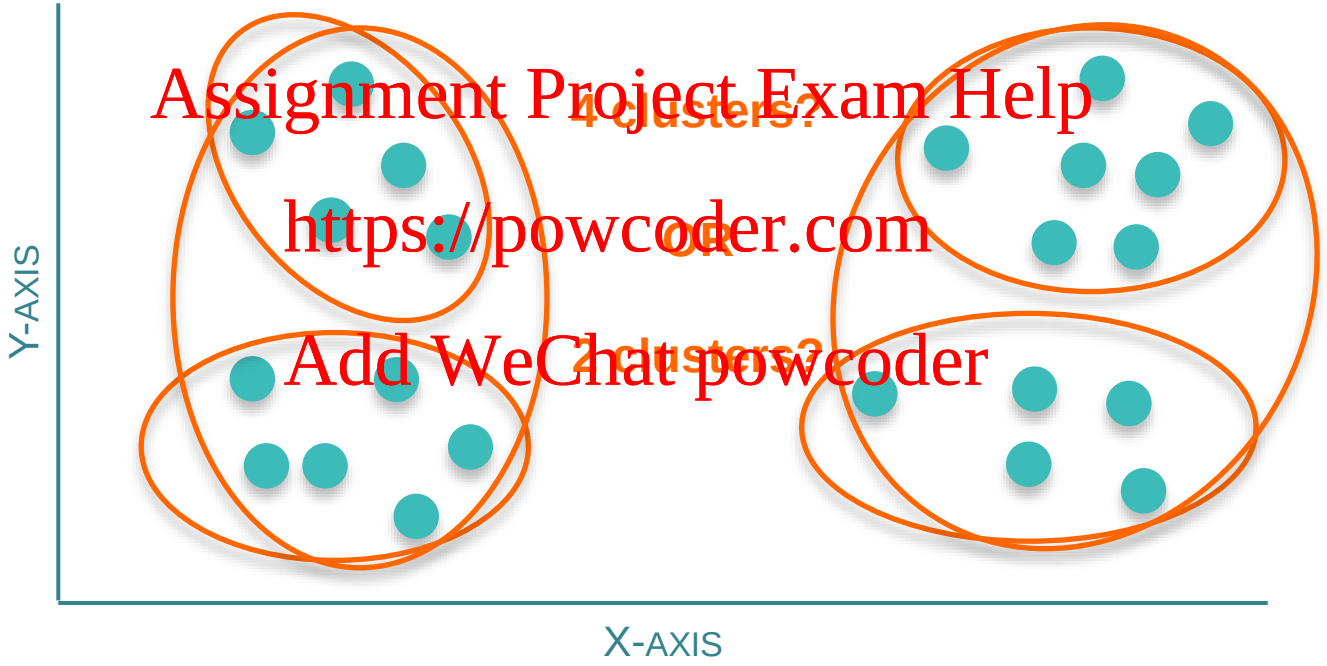


Hyperparameter Tuning

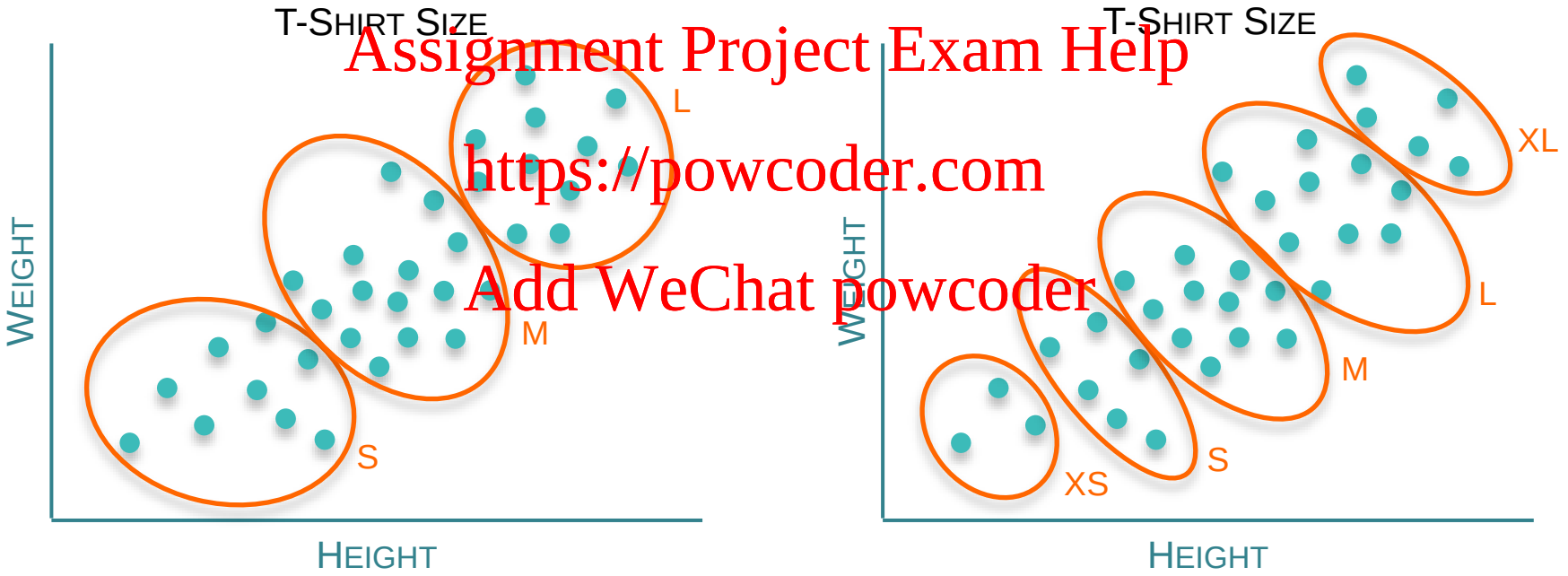
Supervised learning has no ground truth to evaluate model performance

- Understanding the performance of unsupervised learning methods is inherently much more difficult than supervised learning methods because there is no ground truth available
- Moreover, K-means explicitly requests for the number of clusters as a hyperparameter
- K-means performance can be evaluated based on different K clusters
- We can also use the elbow method or the silhouette coefficient to find the optimal K numbers of clusters for the unsupervised learning model

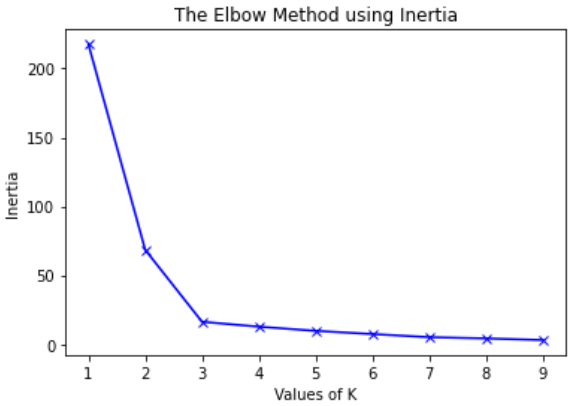
It is genuinely ambiguous how many clusters there are in a dataset and there is no way to decide this automatically



Sometimes the number of clusters to used is imposed by external constraints (e.g. later or downstream processing)



Elbow Method



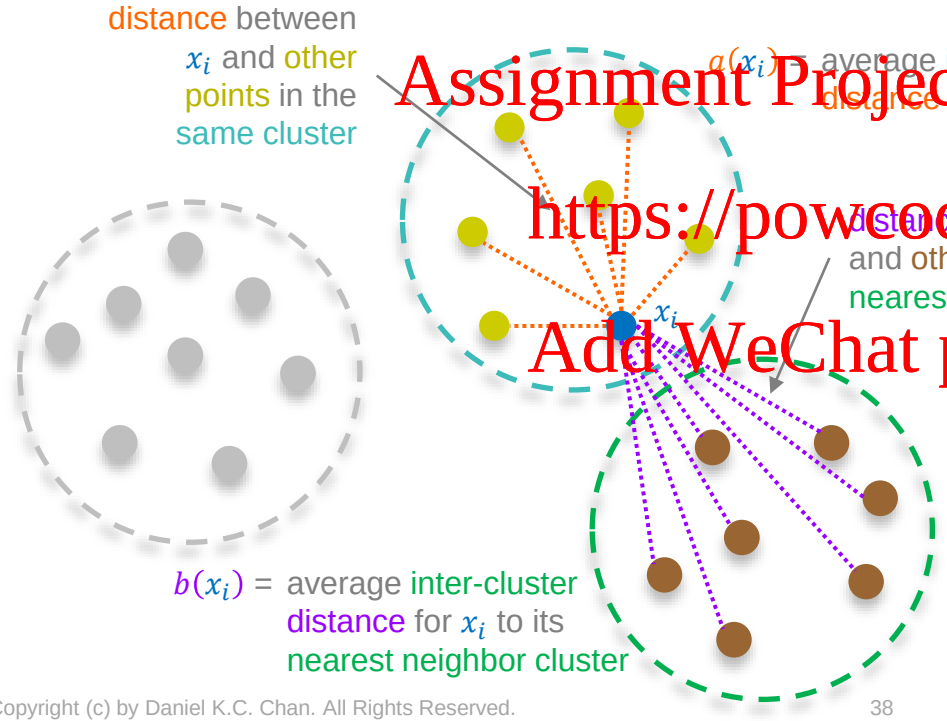
- The elbow method is used to select the optimal number of clusters by examining the visualization of the data
- Inertia is used as the cost functions

$$\sum_{i=1}^{N} \|x_i - \mu_i\|^2$$

μ_i is the centroid closest to the data point x_i
N is the number of data points in the dataset

- The elbow method requires drawing a line plot using the cost function against the number of clusters
- The elbow point is a point of the plot after which the plot starts to flatten out

Ideally, the average intra-cluster distance should be much much less than the inter-cluster distance to the nearest labour cluster



- Objectives
  - Points in the same cluster should be as similar as possible
  - Points in different clusters should be as dissimilar as possible
- When $a(x_i) > b(x_i)$, it is likely that the data point x_i has been **misclassified**

Silhouette Coefficient

$$S(x_i) = \frac{b(x_i) - a(x_i)}{\max(a(x_i),\ b(x_i))}$$

- Evaluates the quality of clustering
- The coefficient ranges from -1 to 1
  - Ideally, $a(x_i) = 0$ and $b(x_i) = \infty$ therefore $S(x_i)=1$ suggesting dense and well separation between clusters
  - In the worst case scenario, $a(x_i) = \infty$ and $b(x_i) = 0$ giving $S(x_i)=-1$ suggesting wrong clustering
  - $S(x_i)$ near 0 suggests overlapping clusters with data points very close to the cluster boundary of the nearest neighbor cluster
- The coefficient is calculated for each data point in the dataset
- Plotting the data points against their silhouette coefficients provides the silhouette plot

Silhouette score is calculated for each data point in the dataset – that is for all data points in all clusters

$$S(x_i) = \frac{b(x_i) - a(x_i)}{\max(a(x_i),\ b(x_i))}$$

1 means the data point is far away from the neighboring clusters meaning minimal confusion and good clustering (positive means the data point is closer to the assigned cluster than it is to neighboring clusters)

0 means the data point lies on the boundary between the assigned cluster and the next closest cluster

-1 means the data point is assigned to an incorrect cluster and the data point in fact likely belongs to a neighboring cluster

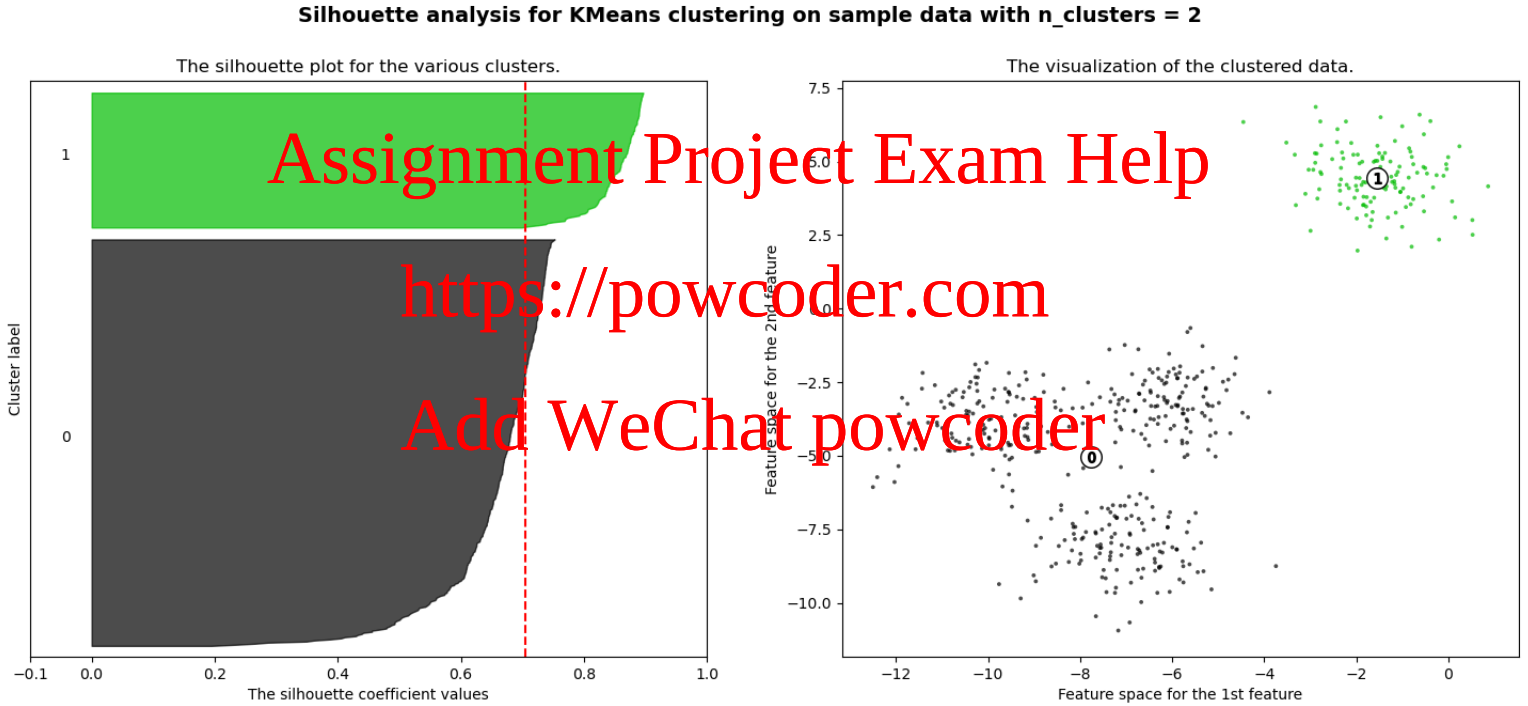
a = Mean Intra−cluster Distance

Mean distance between a data point and all other data points in the same cluster

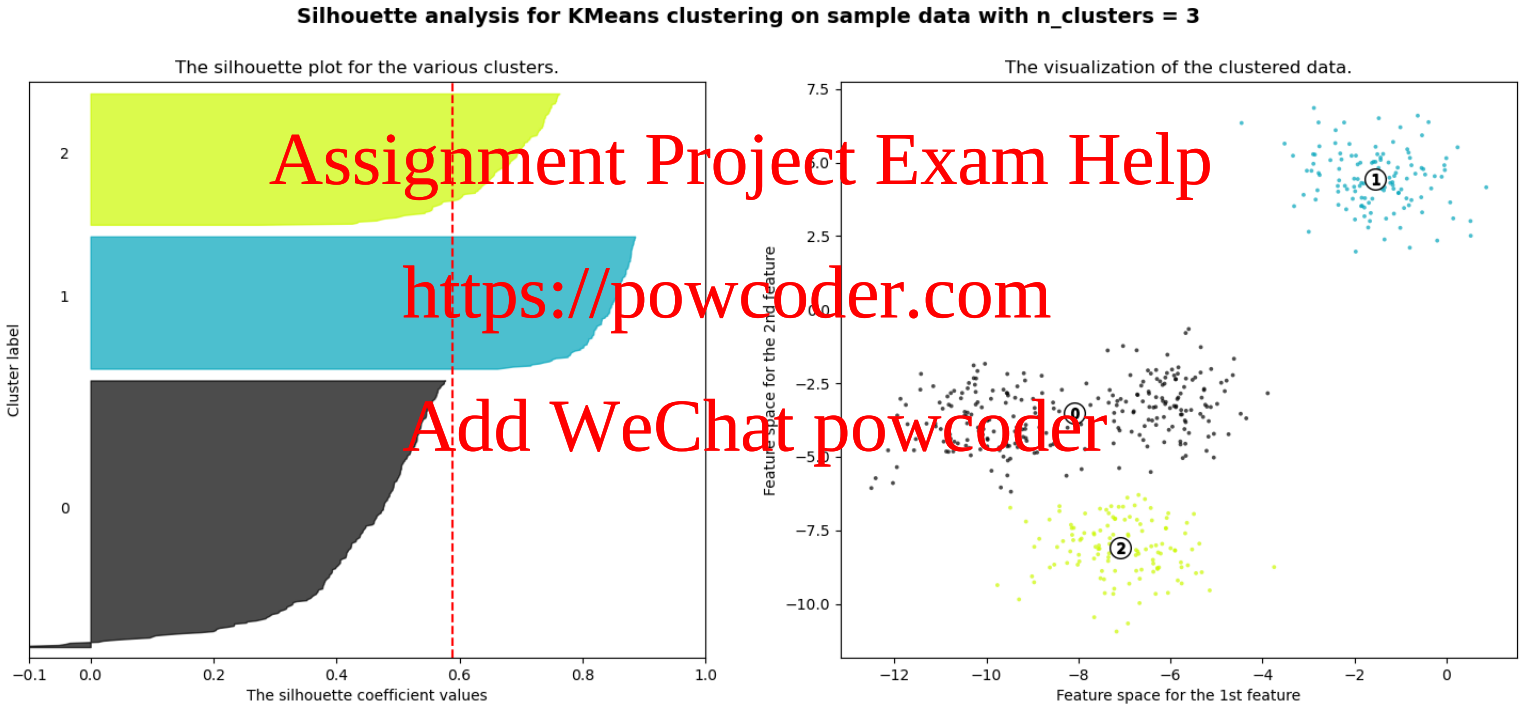
b = Mean Nearest−cluster Distance

Mean distance between a data point and all other data points of the nearest neighbour cluster

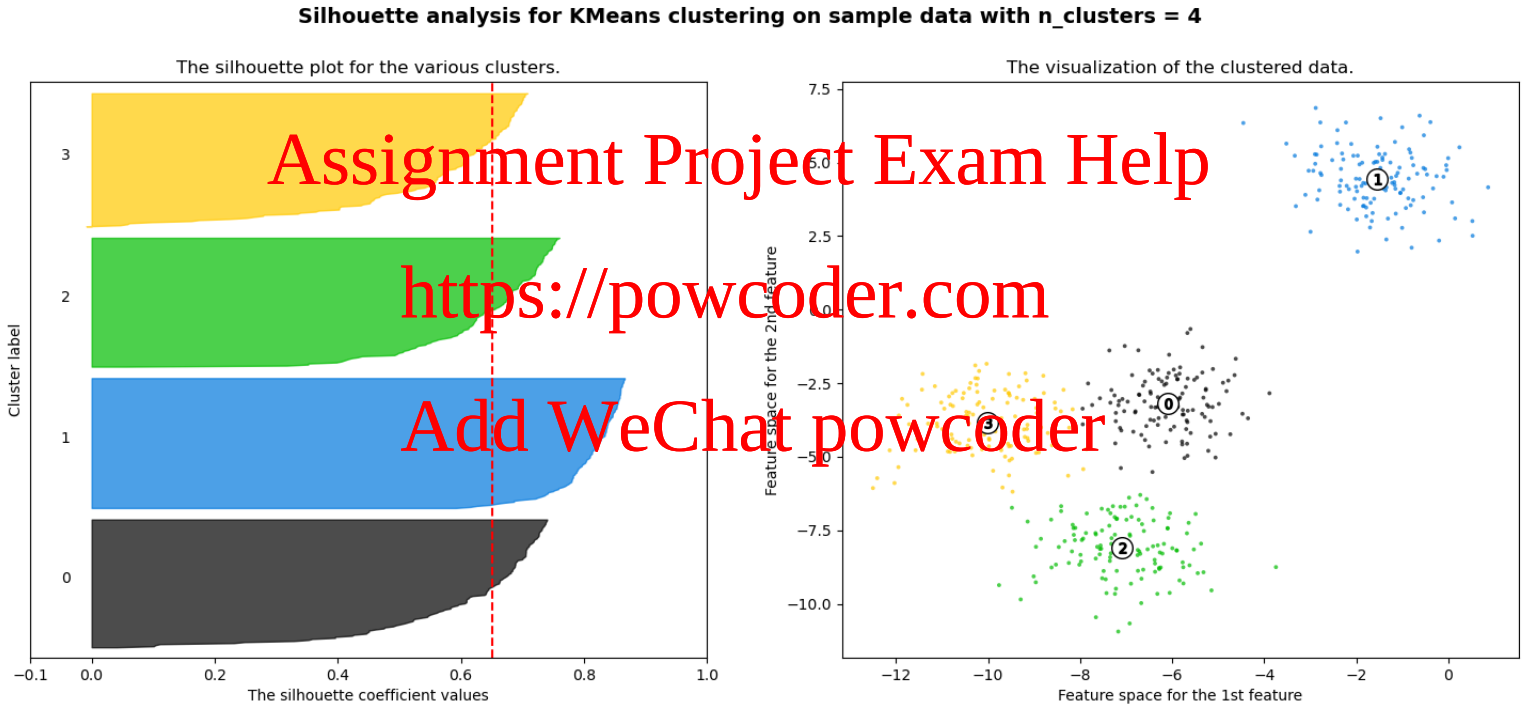
The Silhouette plot shows two clusters that are dense and well-separated



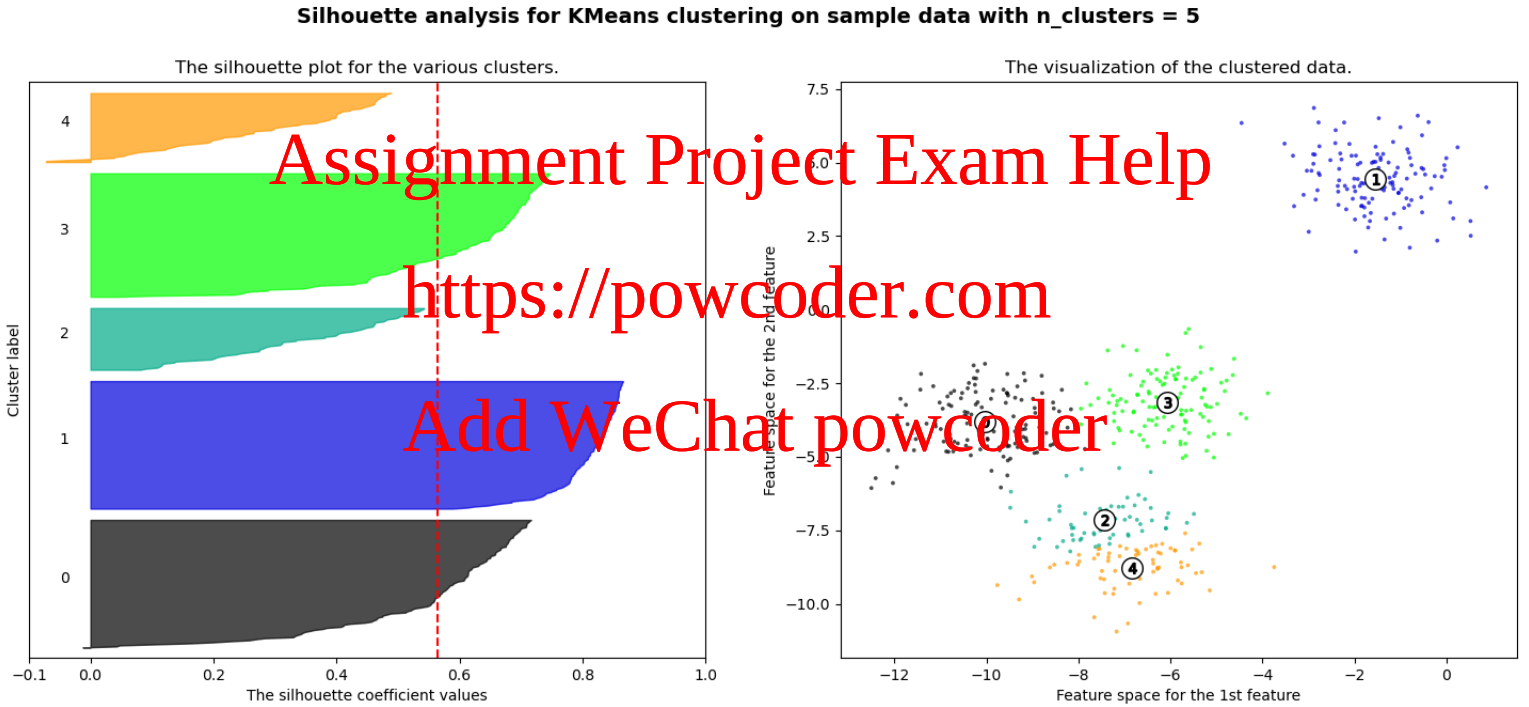
The Silhouette plot shows three clusters that are dense except for one cluster



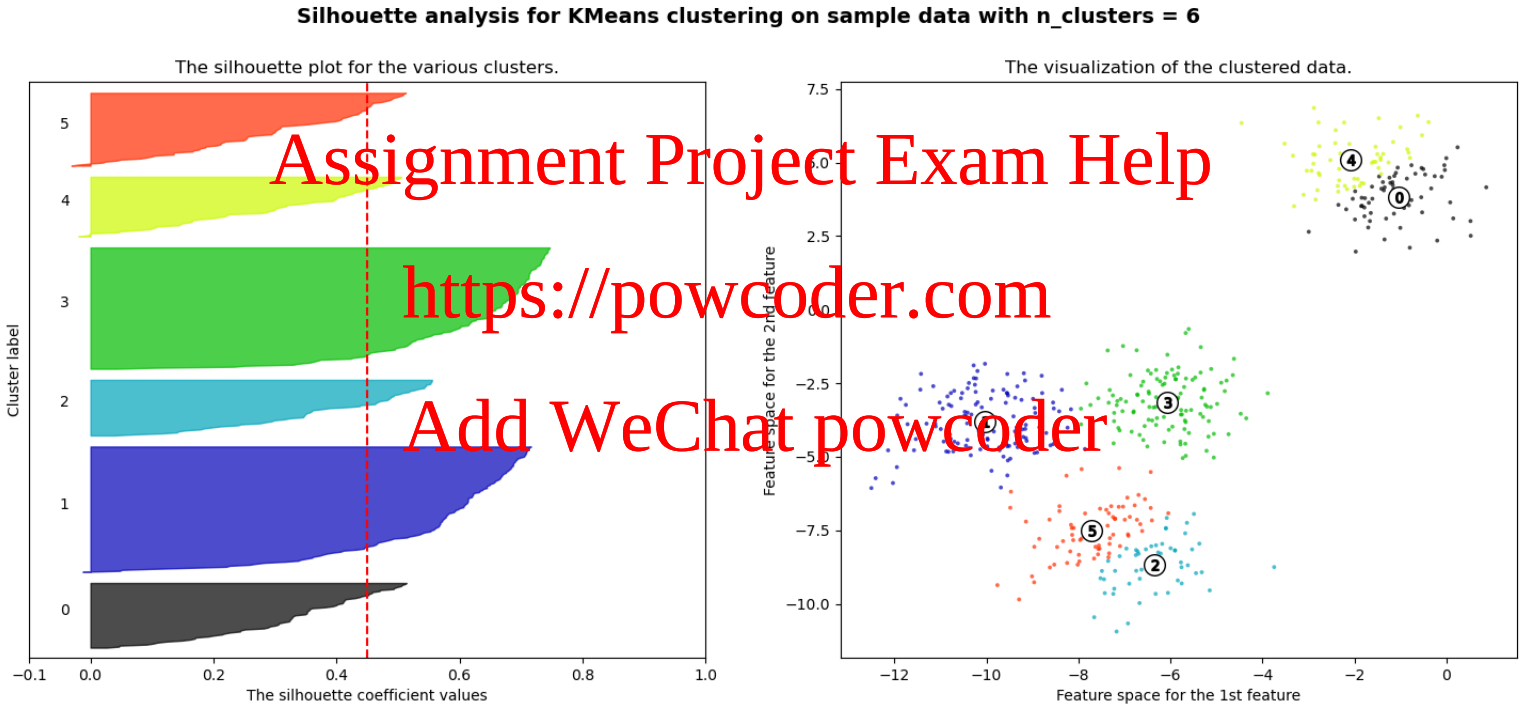
The Silhouette plot shows four clusters that are also dense and well-separated



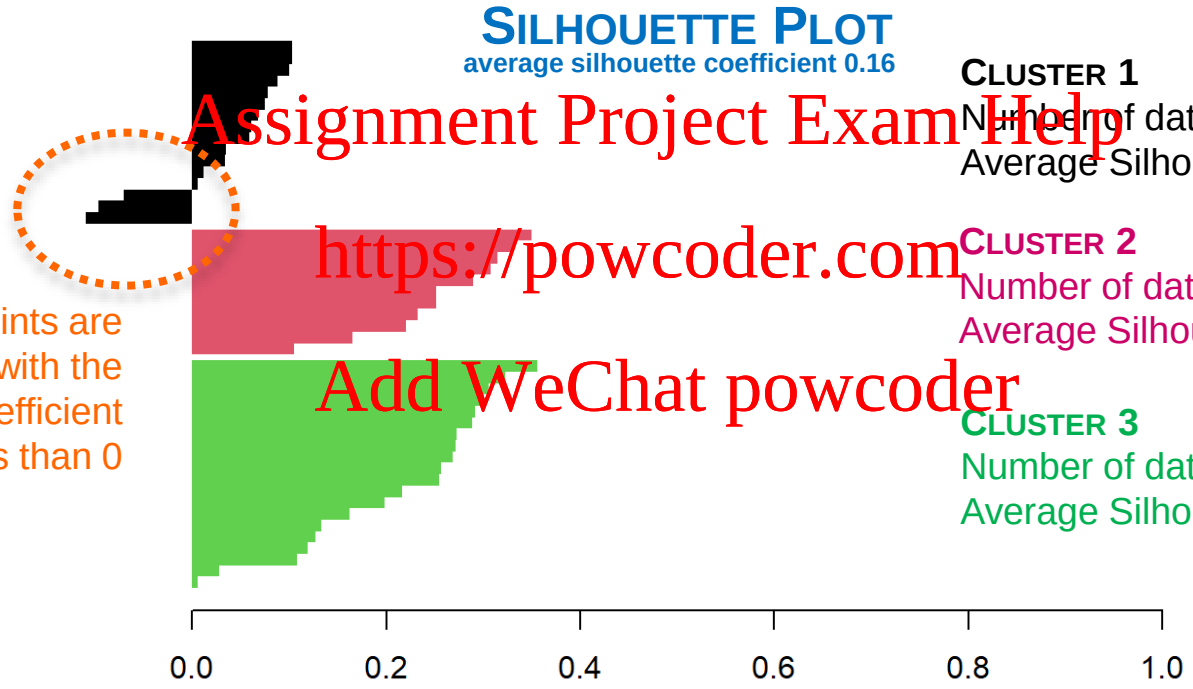
The Silhouette plot shows five clusters that are not so dense



The Silhouette plot shows six clusters that are not dense



Misclassified data points are shown on the left of the Silhouette Plot



Outlier points are those with the Silhouette coefficient value less than 0

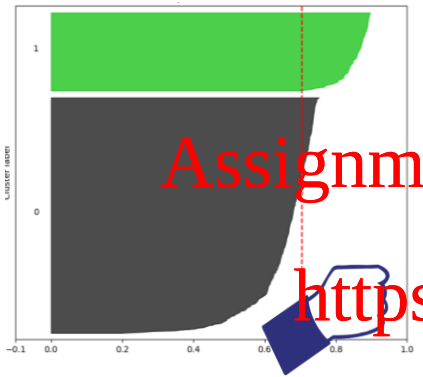
CLUSTER 1
Number of data points: 16
Average Silhouette coefficient: 0.03

CLUSTER 2
Number of data points: 11
Average Silhouette coefficient: 0.26

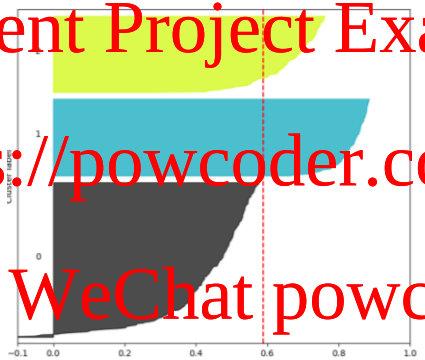
CLUSTER 3
Number of data points: 20
Average Silhouette coefficient: 0.21

The optimal K is chosen based on the number of outliers and the average Silhouette coefficients

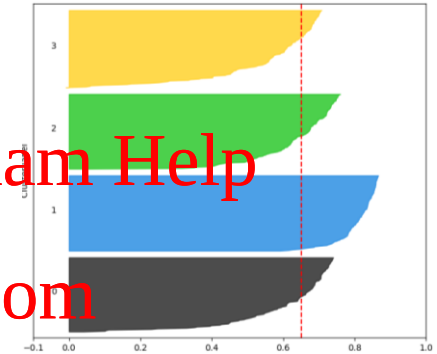
2 CLUSTERS
Average Silhouette coefficient: 0.705



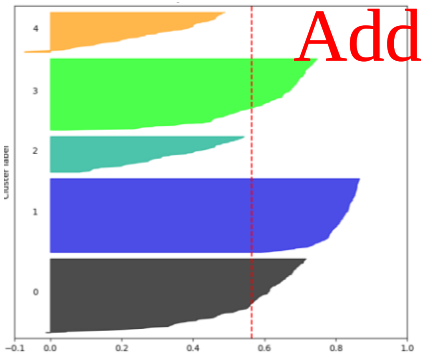
3 CLUSTERS
Average Silhouette coefficient: 0.588



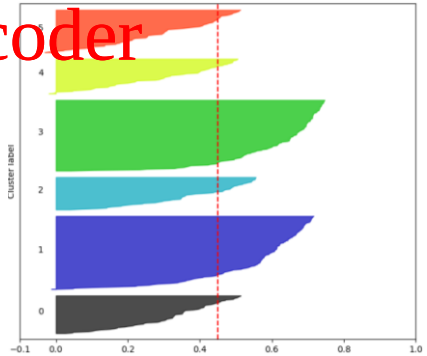
4 CLUSTERS
Average Silhouette coefficient: 0.651



5 CLUSTERS
Average Silhouette coefficient: 0.564



6 CLUSTERS
Average Silhouette coefficient: 0.450



K-Means in a Nutshell

	Property	Description
1	Feature Data Types	Numerical.
2	Target Data Types	Categorical.
3	Key Principles	Likeness is described as a function of Euclidean distance. The goal is to find K centroids (therefore clusters) that minimize the within cluster Euclidean distances. Will group together all data points in the space until no points are left.
4	Hyperparameters	Number of clusters (K).
5	Data Assumptions	Distance metric assumes clusters are spheres. Features are uncorrelated. Normalized.
6	Performance	Fast. Very scalability due to linear time and memory complexity. Even cluster size.
7	Accuracy	Will always converge. Converges to local optimum. May not produce meaningful clusters in a sparse feature space with outliers. Intuition fails in high dimensions and dimensionality reduction is therefore advised as part of the pre-processing.
8	Explainability	

K-Modes Clustering

K-Modes Clustering

K-Modes clustering is an extension of K-Means clustering by replacing cluster means by cluster modes. Modes are updated based on frequency. It is widely used for grouping categorical data. It defines clusters based on the number of matching categories between data points using a simple similarity measure.

The algorithm is essentially the same as K-Means except the cost function is based on equality over categories

$$P(W,Q) = \sum_{l=1}^{K}\sum_{i=1}^{N} w_{il} \cdot d_{sim}(x_i, q_l)$$

P is the cost function for the clustering

W is an NxK matrix of either 0 or 1 representing cluster membership

N is the number of data points in the dataset

K is the number of clusters

Q is the vectors of cluster centroids

X is dataset to be clustered

$$d_{sim}(x_i, q_l) = \sum_{j=1}^{m} \delta(x_{ij}, q_{lj})$$

d_{sim} measures the similarity between 2 vectors

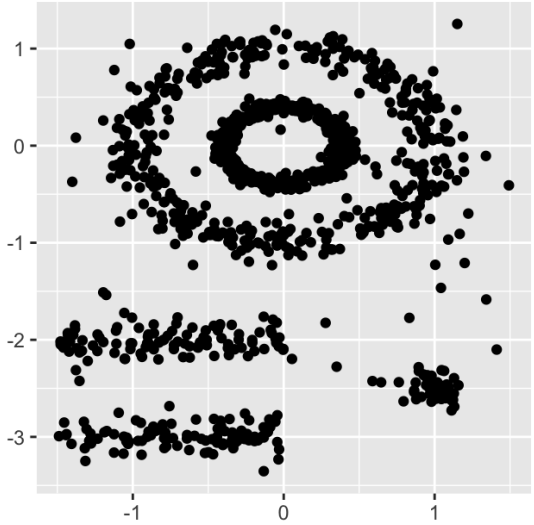
$$\delta(x_{ij}, q_{lj}) = \begin{cases} 1 & \text{if } x_{ij} = q_{lj} \\ 0 & \text{if } x_{ij} \neq q_{lj} \end{cases}$$

δ measures the similarity between 2 features

Density-based Clustering

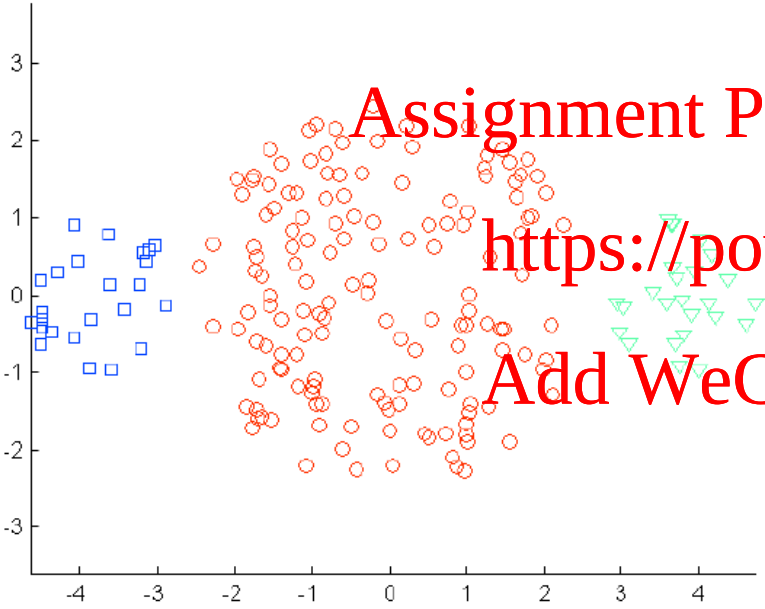
Density-based Spatial Clustering of Applications with Noise (DBSCAN)

Shortcomings of Simple Clustering

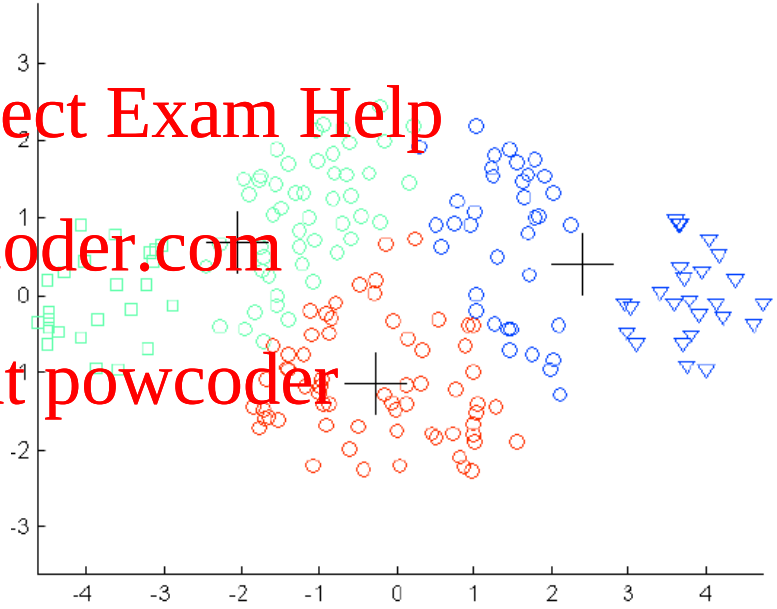


- Clustering algorithms discussed so far are suitable for finding spherical-shaped clusters or convex clusters
- In other words, they work well only for compact and well-separated clusters
- Moreover, they are also severely affected by the presence of noise and outliers in the dataset
- Unfortunately, real life data may exhibit arbitrary shapes and properties (including multiple shapes)

K-Means runs into problem with clusters of different sizes

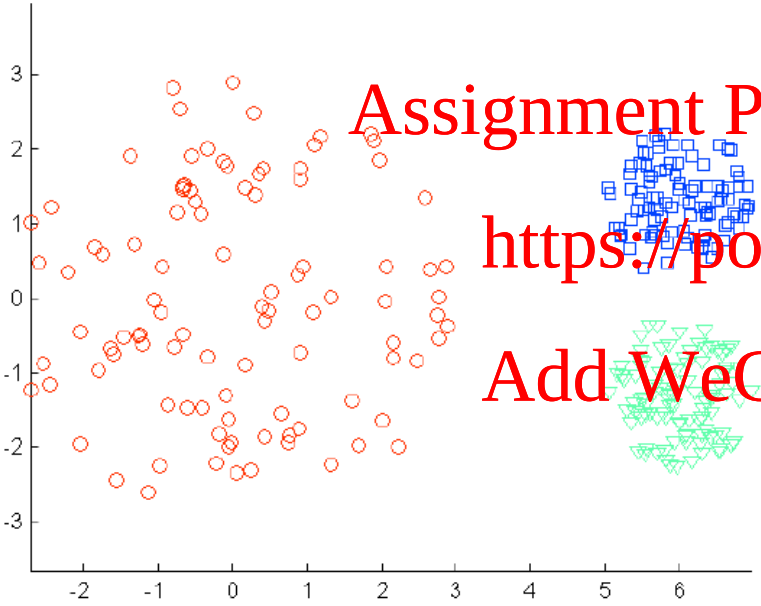


GROUND TRUTH

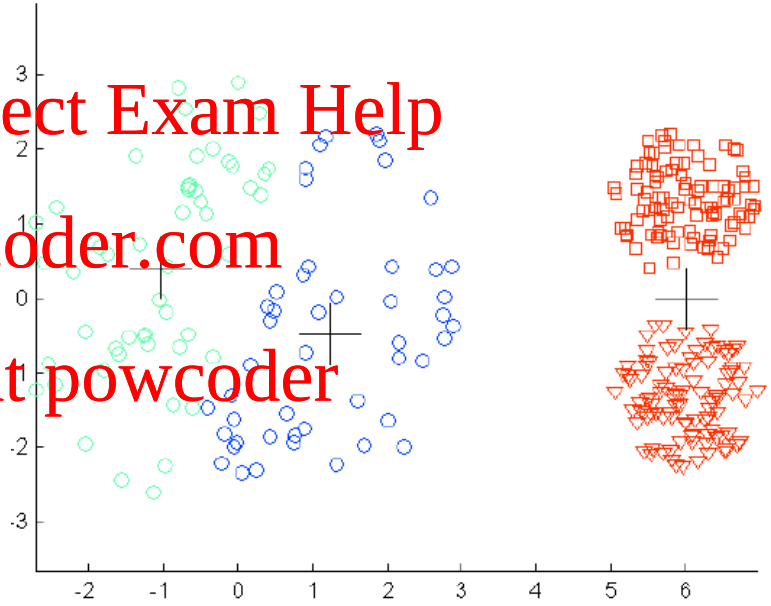


THREE CLUSTERS FROM K-MEANS

K-Means runs into problem with clusters of different densities

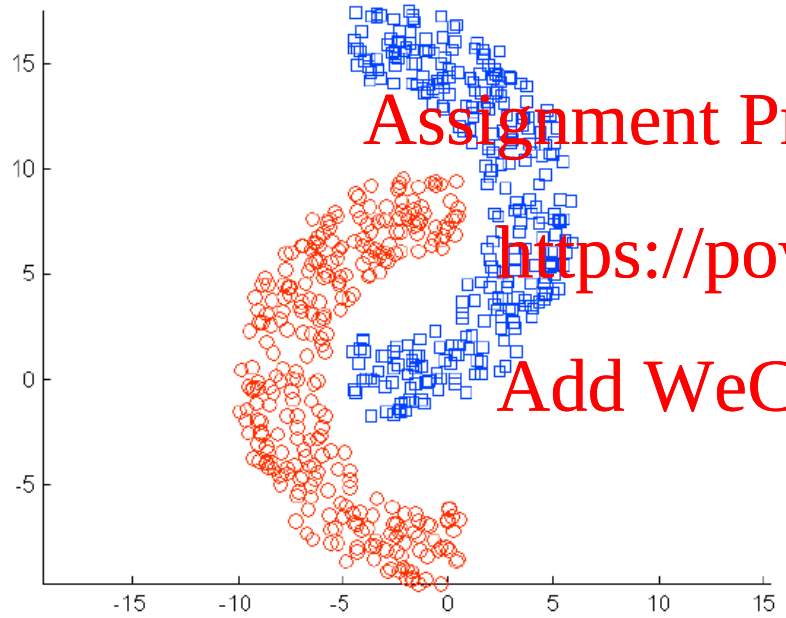


GROUND TRUTH

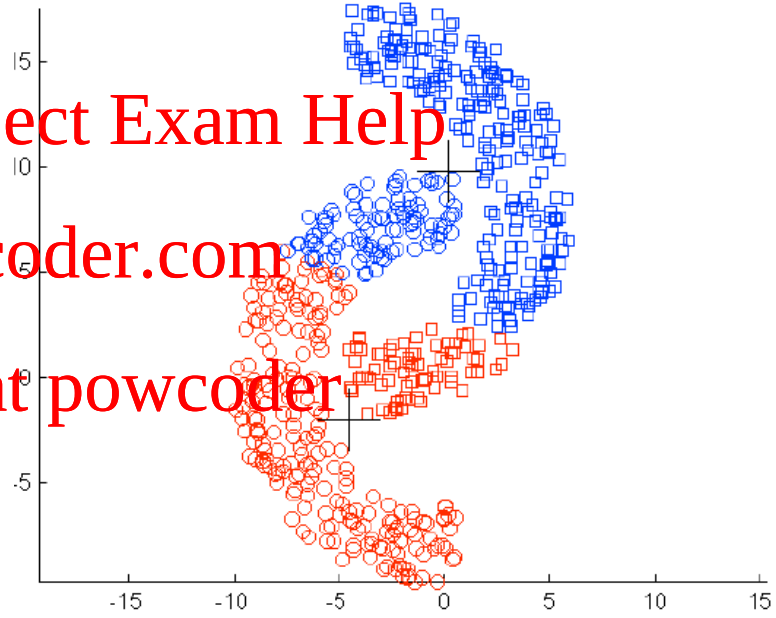


THREE CLUSTERS FROM K-MEANS

K-Means runs into problem with clusters of non-spherical or non-convex shapes

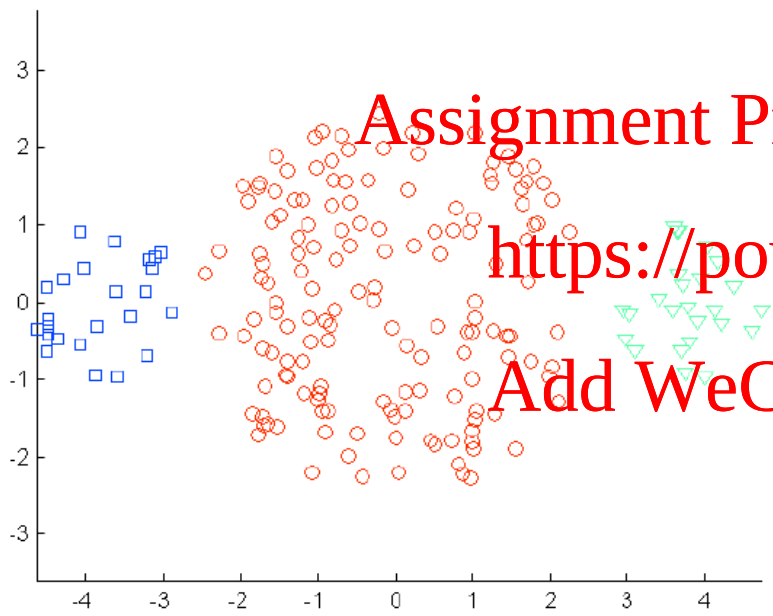


GROUND TRUTH

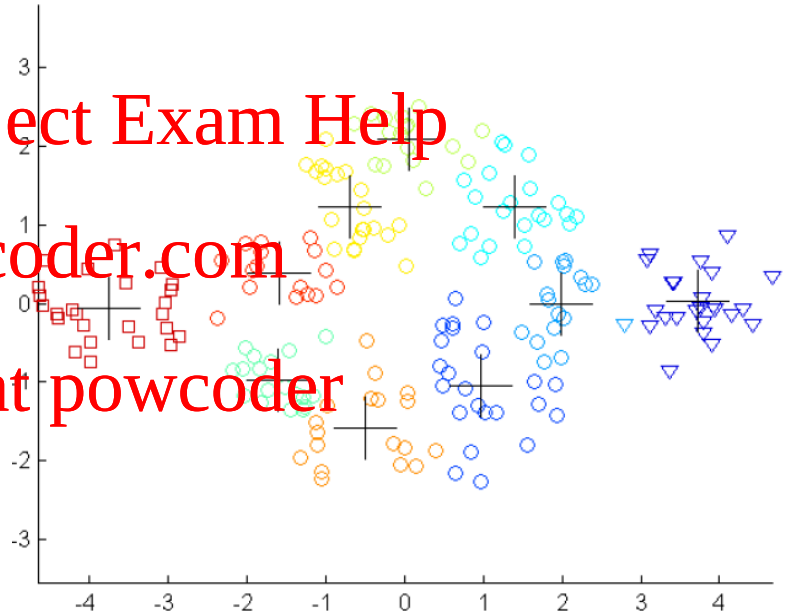


TWO CLUSTERS FROM K-MEANS

Shortcomings of K-Means with cluster size can be dealt with using more clusters first and then put them together

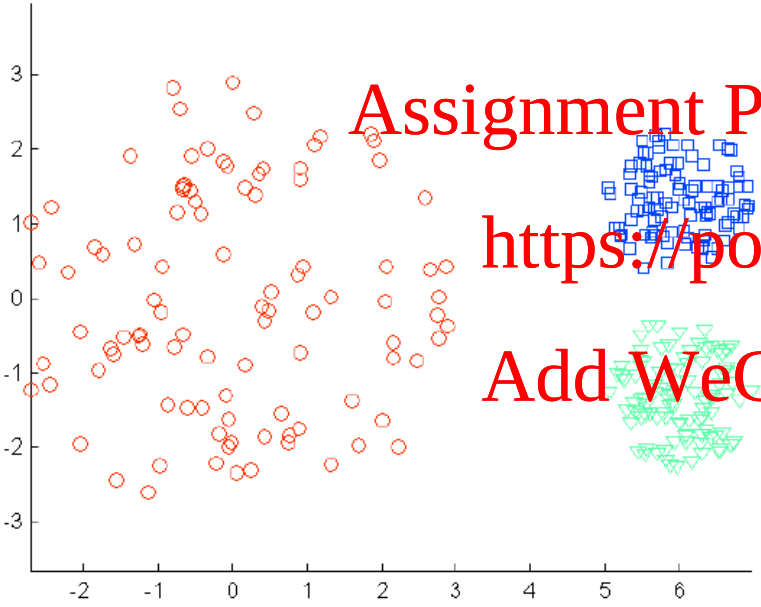


GROUND TRUTH

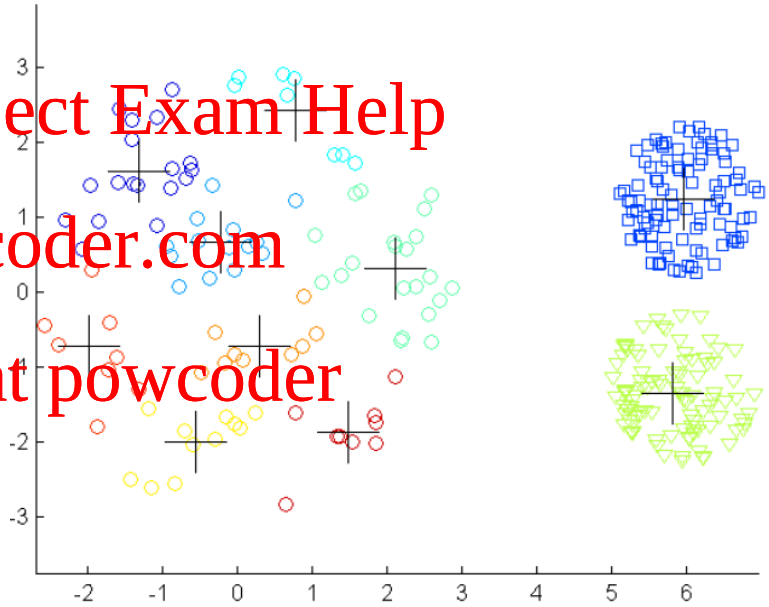


CLUSTERS FROM K-MEANS

Shortcomings of K-Means with cluster densities can be dealt with using more clusters first and then put them together

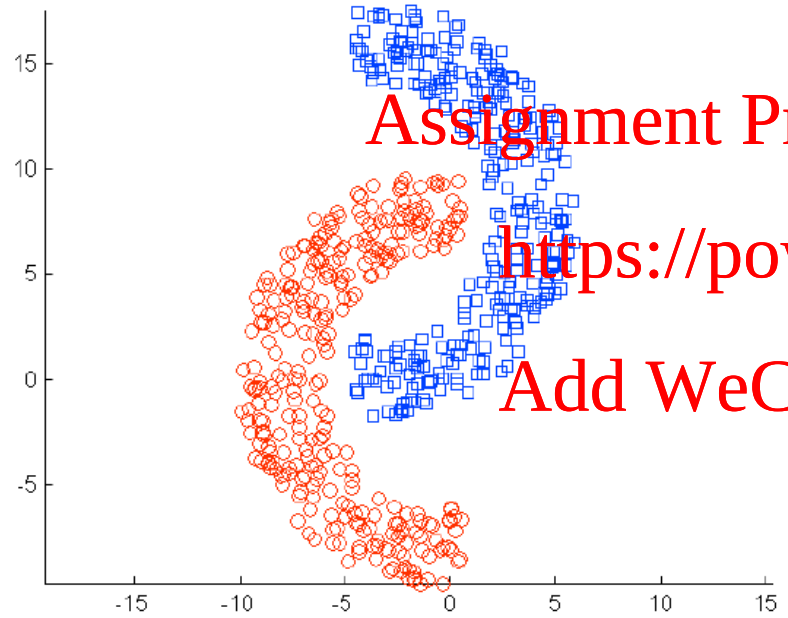


GROUND TRUTH

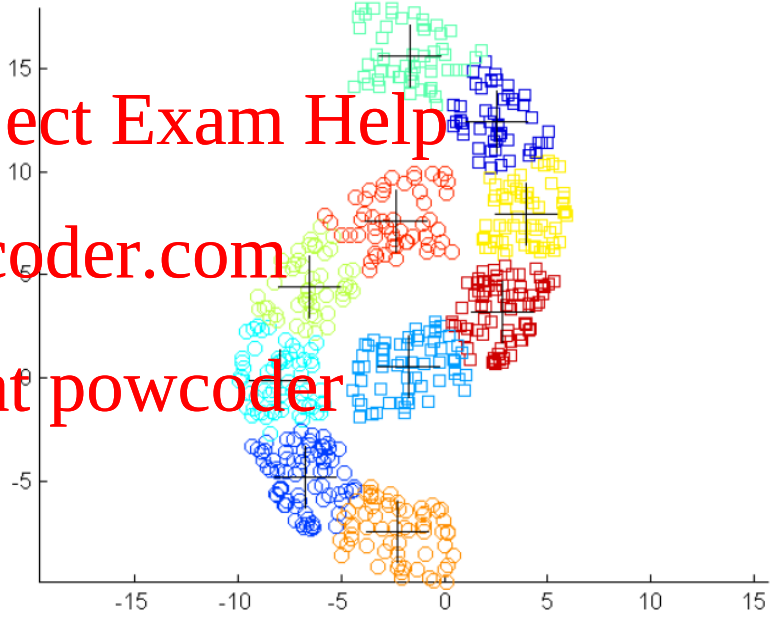


CLUSTERS FROM K-MEANS

Shortcomings of K-Means with cluster shapes can be dealt with using more clusters first and then put them together



GROUND TRUTH

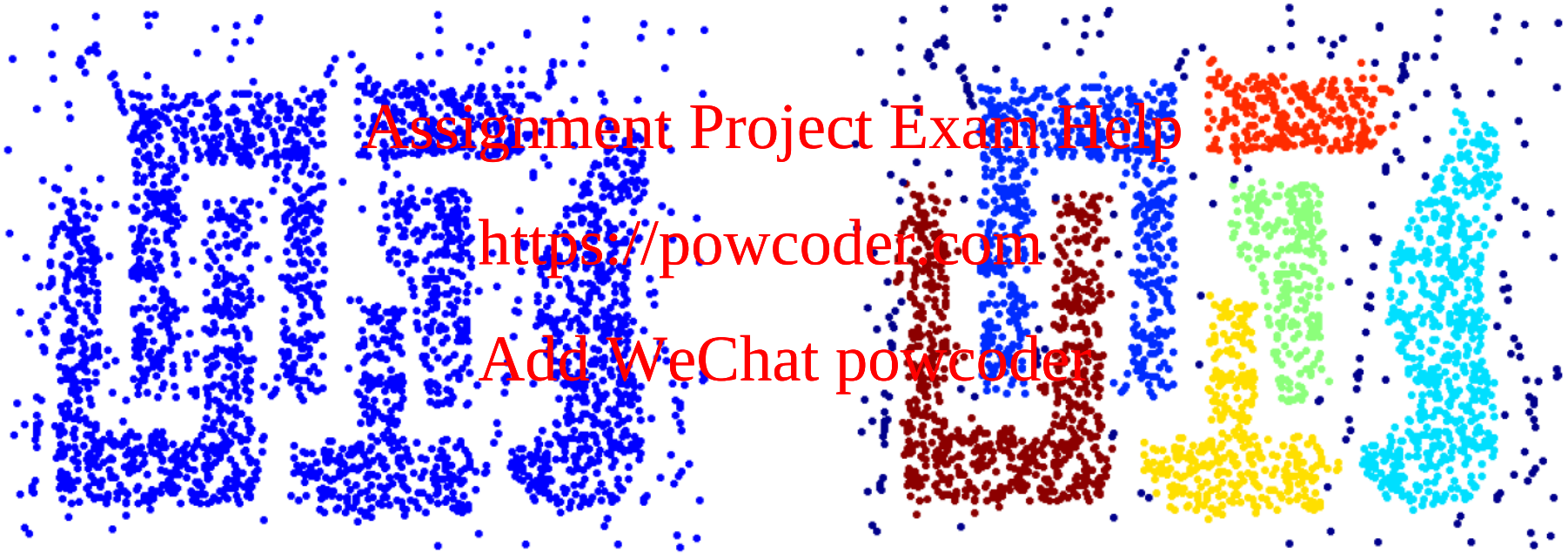


CLUSTERS FROM K-MEANS

DBSCAN

DBSCAN is a density-based clustering algorithm. Given a set of points in some space, it groups together points that are closely packed together (points with many nearby neighbors) and marks as outliers points that lie alone in low-density regions (whose nearest neighbors are too far away).

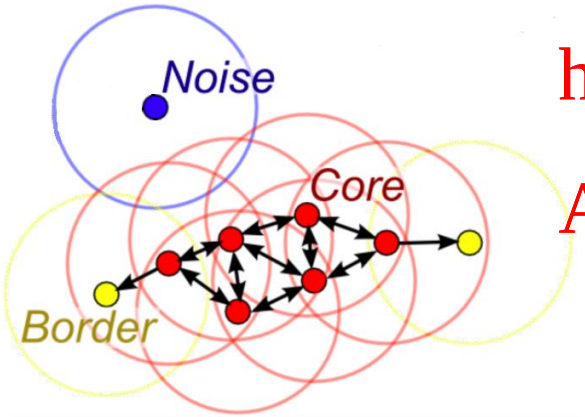
DBSCAN provides a more flexible and direct solution to address the shape and size issues with K-Means



ORIGINAL DATA

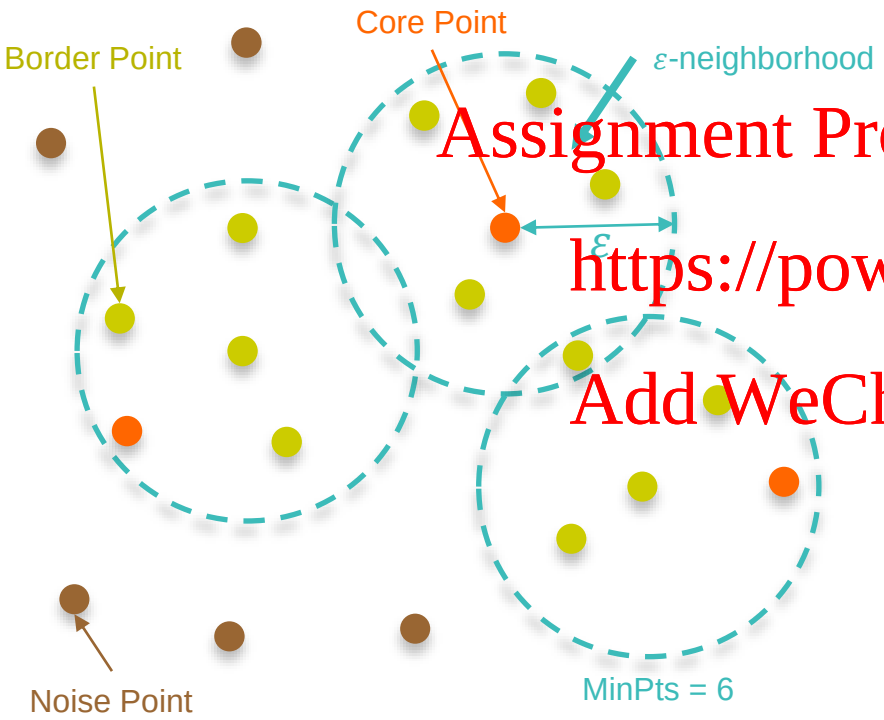
CLUSTERS & NOISE POINTS FROM DBSCAN

DBSCAN

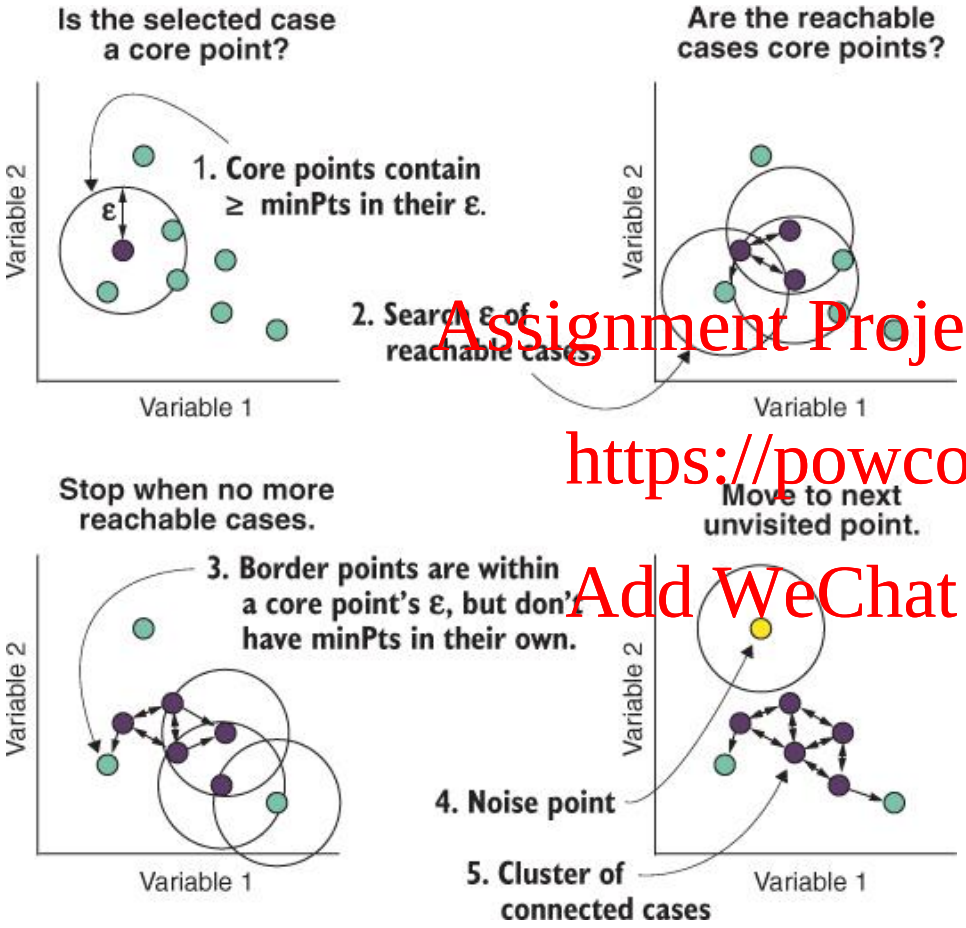


- From each unvisited data point, measure the distance to every other point in the dataset
- All points that fall within the radius of neighborhood will be considered as neighbors
- If the number of neighbors reaches the minimum neighbor point threshold, the points should be grouped together as a new cluster
- Data points not reachable from any cluster will be considered as noise
- Repeat the process until all data points are categorized in clusters or marked as noise

Unlike other clustering algorithms, not all data points are classified - unclassified data points are considered noise



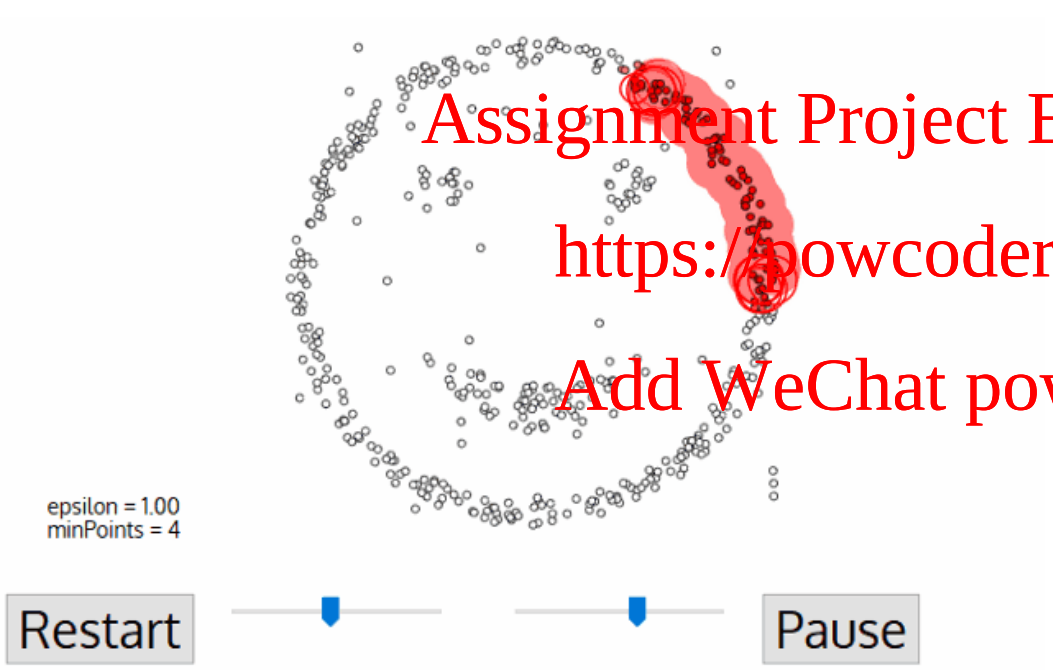
- Core Point
  - At least a minimum number of data points (MinPts) within its radius of neighborhood (ε-neighborhood)
  - All core points within the ε-neighborhood of a core point are grouped as a cluster
- Border Point
  - Lies within the ε-neighborhood of a core point but not a core point itself due to not having enough MinPts in its ε-neighborhood
  - Will be grouped in the cluster of its nearest core point
- Noise Point
  - Not reachable from any cluster
  - A noise point, not enough MinPts in its neighborhood, not associated with a core point
  - Excluded from clustering

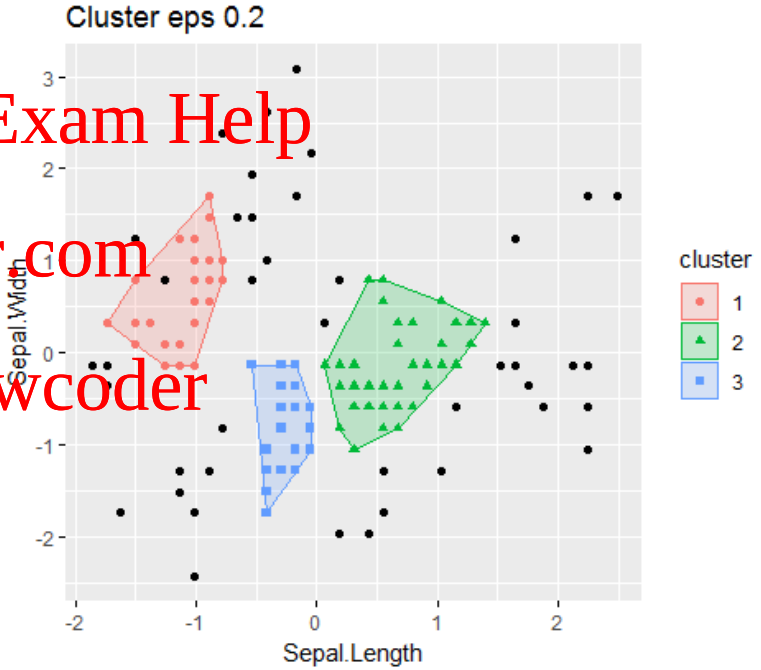


Hyperparameters can be tricky to tune

- Radius of Neighborhood (ε / epsilon)
  - The maximum distance a point to the nearest cluster
  - The greater the value, the fewer clusters are found because clusters eventually merge into other clusters
- Minimum Neighbor Points
  - Required to produce a new cluster
  - A larger value assures a more robust cluster but may exclude some smaller clusters as it attempts to merge them in a larger one
  - Increases with the size of the dataset
  - A smaller value may extract many clusters with possible inclusion of noise

DBSCAN moves through all data points to form clusters based on neighbourhood and density



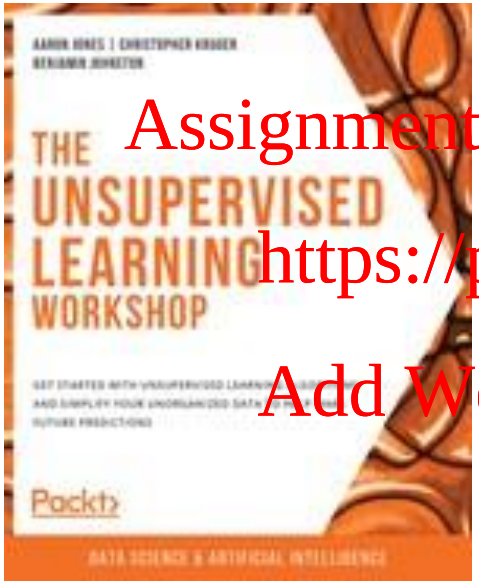


DBSCAN in a Nutshell

	Property	Description
1	Feature Data Types	Numerical. Should be scaled.
2	Target Data Types	Categorical
3	Key Principles	Expands the distance metric with the notion of density and clusters are therefore high density areas. Cluster membership is based on neighbourhood radius and the number of data points in the neighbourhood. Identify core, boundary, and noise points. Noise points are excluded from clustering. Therefore less prone to the distortion caused by outliers.
4	Hyperparameters	K is not required. Neighbourhood radius (epsilon). Minimum data points per neighbourhood.
5	Data Assumptions	Will find clusters of arbitrary shapes and sizes including highly complex data.
6	Performance	It will often immensely outperform K-means (in practice, this often happens with highly intertwined, yet still discrete, data, such as a feature space containing two half-moons). Parameter tuning can be challenging. Finds non-convex and non-linearly separable clusters.
7	Accuracy	Difficulties with clusters of varying density and high-dimensional data.
8	Explainability	

References

References



"The Unsupervised Learning Workshop", Aaron Jones, Christopher Kruger, Benjamin Johnston, Packt Publishing, July 2020

"Hands-On Unsupervised Learning Using Python", Ankur A. Patel, O'Reilly Media, Inc., March 2019

References

- "K-Means Clustering using sklearn and Python", Dhiraj K, October 2019 (https://heartbeat.fritz.ai/k-means-clustering-using-sklearn-and-python-4a054d67b187)
- "K-Means Clustering Explained with Python Example", Ajitesh Kumar, September 2020 (https://vitalflux.com/k-means-clustering-explained-with-python-example/)
- "K-Means Clustering Elbow Method & SSE Plot - Python", Ajitesh Kumar, September 2020 (https://vitalflux.com/k-means-elbow-point-method-sse-inertia-plot-python/)
- "K-Means Silhouette Score Explained with Python Example", Ajitesh Kumar, September 2020 (https://vitalflux.com/kmeans-silhouette-score-explained-with-python-example/)
- "K-Modes Clustering", Shailja Jaiswal, July 2020 (https://medium.com/@shailja.nitp2013/k-modesclustering-ef6d9ef06449)
- "How to Create an Unsupervised Learning Model with DBSCAN", Anasse Bari, Mohamed Chaouchi & Tommy Jung (https://www.dummies.com/programming/big-data/data-science/how-to-create-an-unsupervised-learning-model-with-dbscan/)
- " Scikit-Learn - Clustering: Density-Based Clustering of Applications with Noise [DBSCAN]", June 2020 (https://coderzcolumn.com/tutorials/machine-learning/scikit-learn-sklearn-clustering-dbscan)
- "A Step by Step approach to Solve DBSCAN Algorithms by tuning its hyper parameters", Mohanty Sandip, Mar 2020 (https://medium.com/@mohantysandip/a-step-by-step-approach-to-solve-dbscan-algorithms-by-tuning-its-hyper-parameters-93e693a91289)

References

- "DBSCAN Python Example: The Optimal Value For Epsilon (EPS)", Cory Maklin, Jun 2019 (https://towardsdatascience.com/machine-learning-clustering-dbscan-determine-the-optimal-value-for-epsilon-eps-python-example-3100091cfbc)
- "DBSCAN: Density-Based Clustering Essentials" (https://www.datanovia.com/en/lessons/dbscan-density-based-clustering-essentials/)

THANK YOU