K- MEANS CLUSTERING ALGORITHM

Submitted by-
Sweta Das
40016932

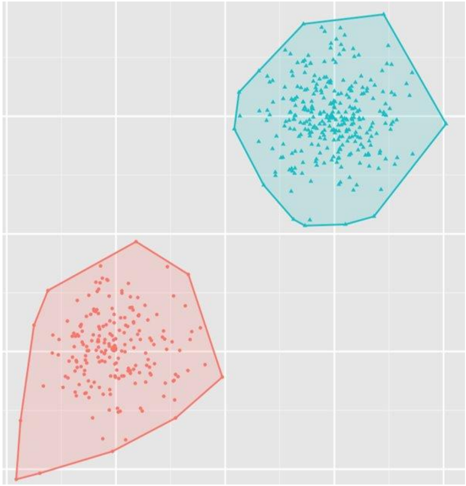
CONTENTS

Introduction to clustering:

Clustering is the classification of objects into different groups, or more precisely, the partitioning of a data set into subsets (clusters), so that the data in each subset (ideally) share some common trait - often according to some defined distance measure.The main advantage of clustering over classification is that it helps find useful features that distinguish different groups.

The problem statement revolves around finding a solution by grouping data points into clusters so that:

- Points within each cluster are similar to each other(intra class distance minimized)
- Points from different clusters are dissimilar(Inter class distance maximized)



Clustering is useful whenever diverse and varied data can be exemplified by a much smaller number of groups. It results in meaningful and actionable (information) data structures that
reduce complexity and provide insight into patterns of relationships.

Types of clustering:

1. Hierarchical algorithms: these find successive clusters using previously established clusters.

- Agglomerative ("bottom-up"): Agglomerative algorithms begin with each element as a separate cluster and merge them into successively larger clusters.
- Divisive ("top-down"): Divisive algorithms begin with the whole set and proceed to divide it into successively smaller clusters.

2. Partitional clustering: Partitional algorithms determine all clusters at once. They include:

- K-means and derivatives
- Fuzzy c-means clustering
- QT clustering algorithm

Introduction to k-means clustering:

K-means clustering is one of the simplest and most popular unsupervised machine learning algorithms. Typically, unsupervised algorithms make inferences from datasets using only input vectors without referring to known, or labelled, outcomes.The **k-means algorithm** is perhaps the most commonly used clustering method. Having been studied for several decades, it serves as the foundation for many more sophisticated clustering techniques.

Why K- MEANS?

- Relatively simple to implement.
- Scales to large data sets.
- Guarantees convergence.
- Can warm-start the positions of centroids.
- Easily adapts to new examples.
- Generalizes to clusters of different shapes and sizes, such as elliptical clusters. Best suited for spherical clusters.

Deep dive into K-Means Algorithm:

Suppose We are given a database of ‘n’ objects and the partitioning method constructs ‘k’ partitions of data. Each partition will represent a cluster and $k \leq n$. It means that it will classify the data into k groups, which satisfy the following requirements −

· Each group contains at least one object.
· Each object must belong to exactly one group.

Points to remember −

· For a given number of partitions (say k), the partitioning method will create an initial partitioning.

· Then it uses the iterative relocation technique to improve the partitioning by moving objects from one group to another.

Key points:

· The k-means algorithm assigns each of the n examples to one of the k clusters.

· Where suitable k is a number that has been determined ahead of time (but must be given in advance at the beginning).

· The goal is to minimize the differences within each cluster and maximize the differences between the clusters.

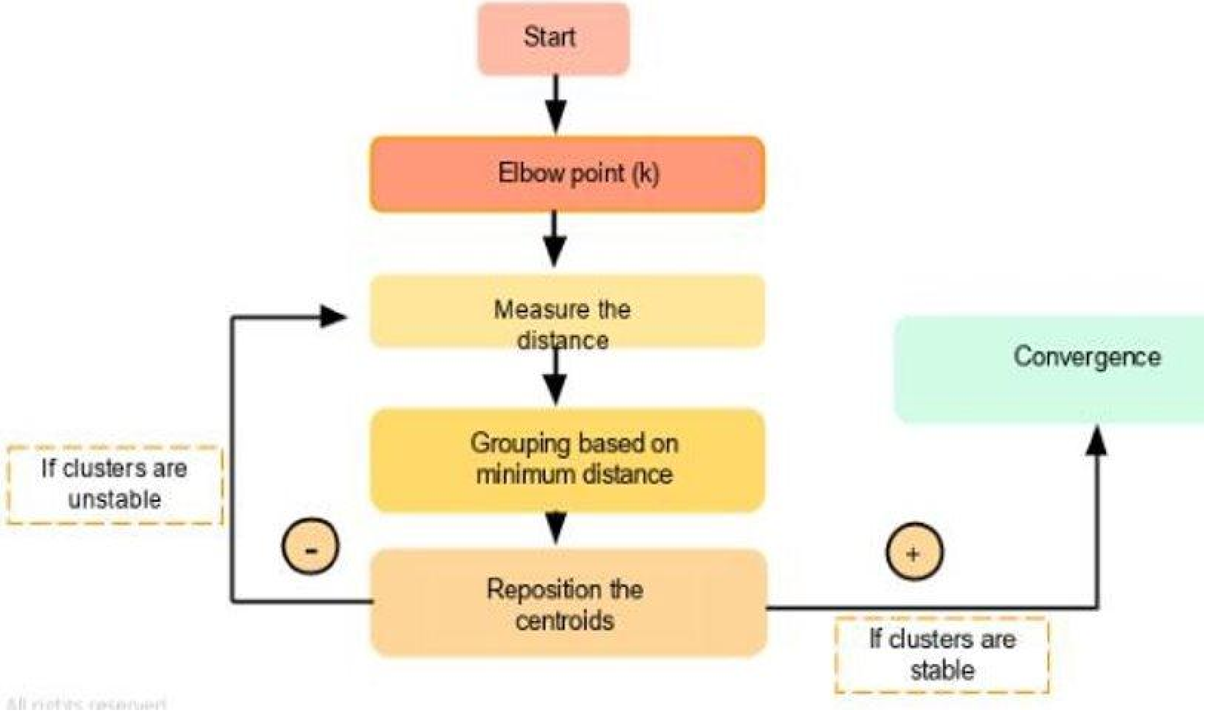
The objective of K-means is simple: group similar data points together and discover underlying patterns. To achieve this objective, K-means looks for a fixed number (k) of clusters in a dataset. We have to define a target number k, which refers to the number of centroids we need in the dataset. A centroid is the imaginary or real location representing the center of the cluster.Every data point is allocated to each of the clusters through reducing the in-cluster sum of squares. In other words, the K-means algorithm identifies k number of centroids, and then allocates every data point to the nearest cluster, while keeping the centroids as small as possible. The 'means' in the K-means refers to averaging of the data; that is, finding the centroid.

Algorithm:

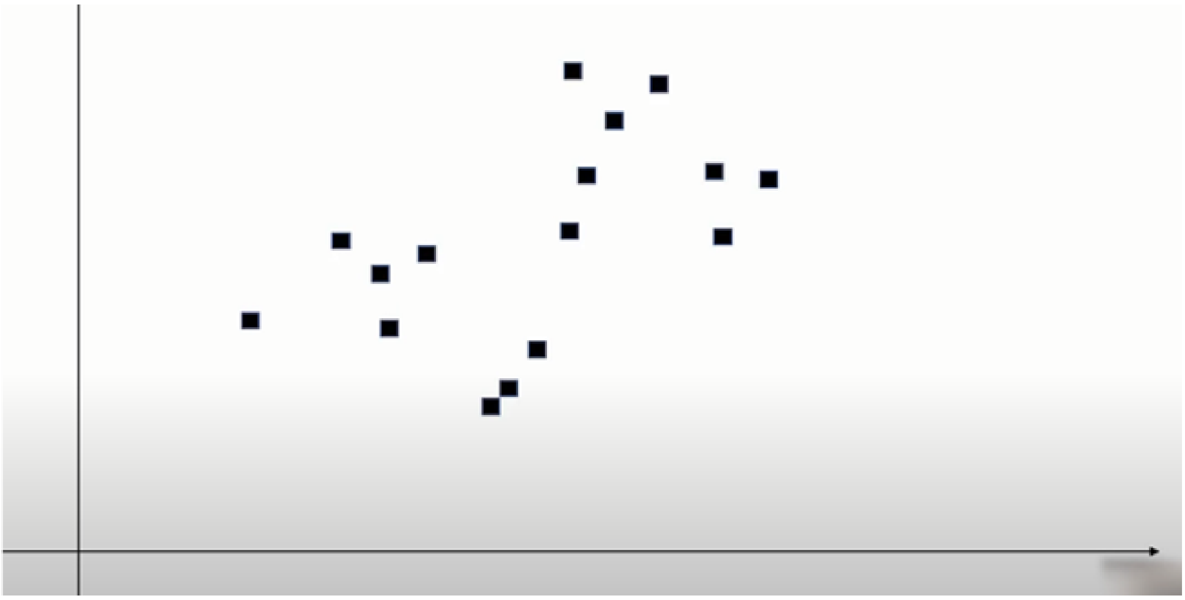
Algorithm essentially involves two phases (updation and assignment).

- First, it assigns examples to an initial set of k clusters.
- Then, it updates the assignments by adjusting the cluster boundaries.
- The process of updating and assigning occurs several times until changes no longer improve the cluster fit.
- At this point, the process stops and the clusters are finalized.

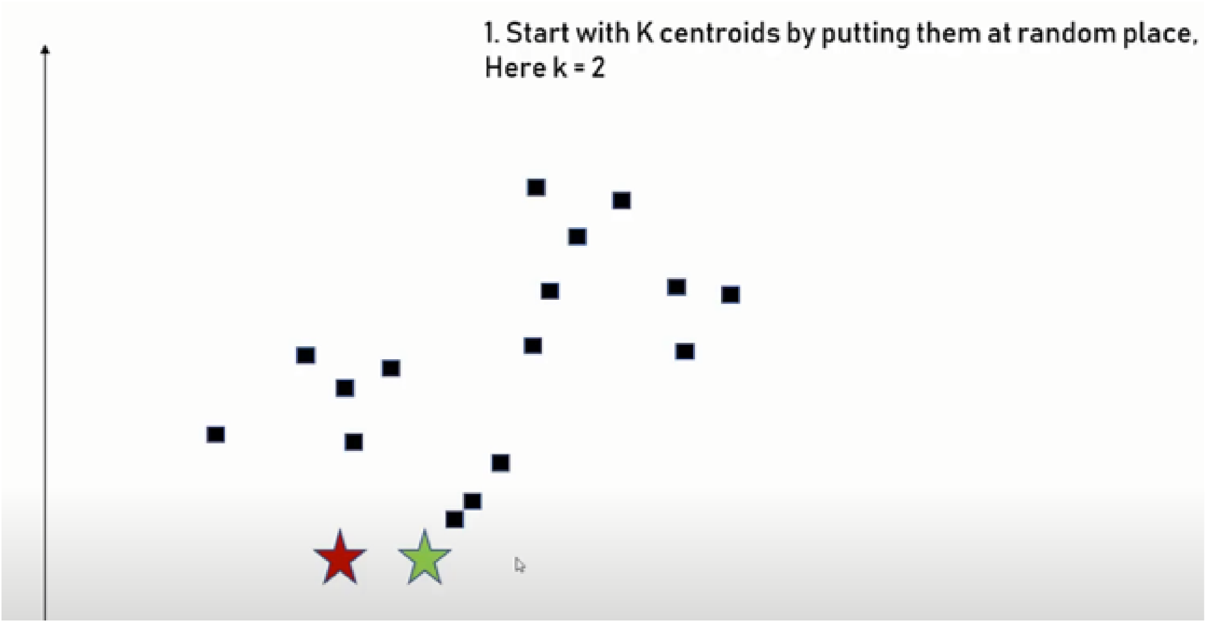
Flowchart:



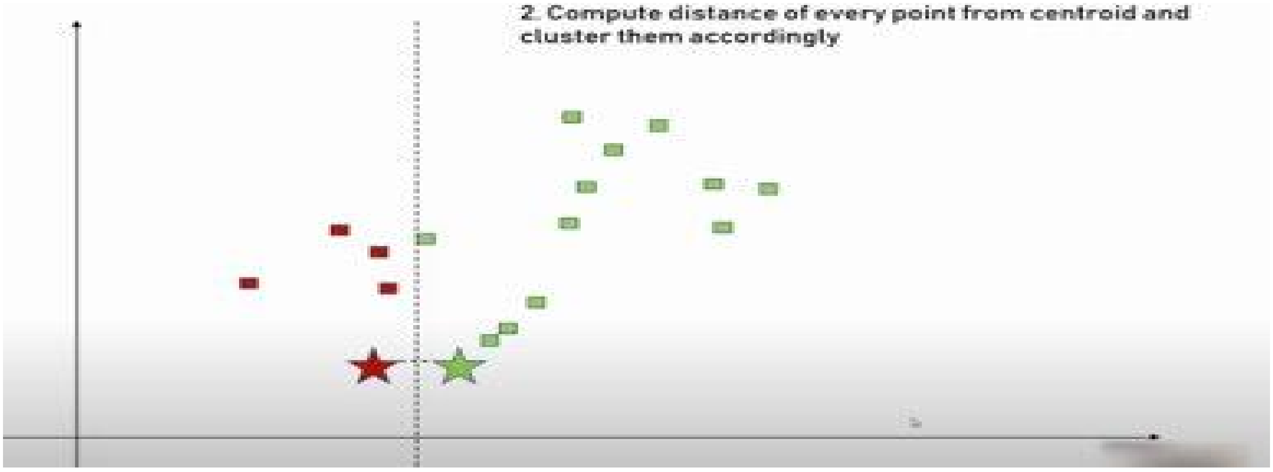
Visualization Of The Algorithm:



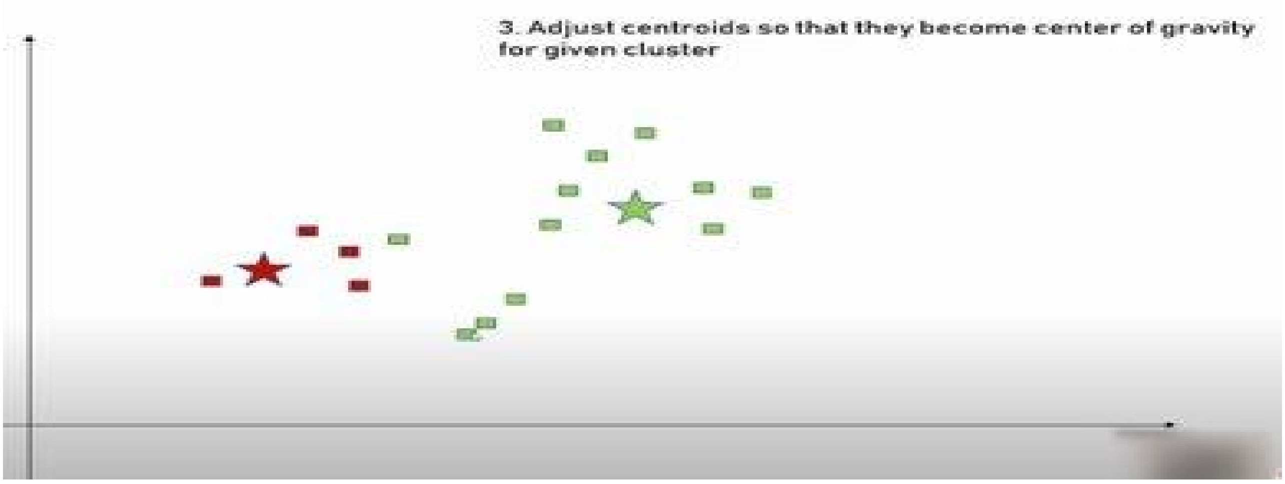
This figure represents the dataset



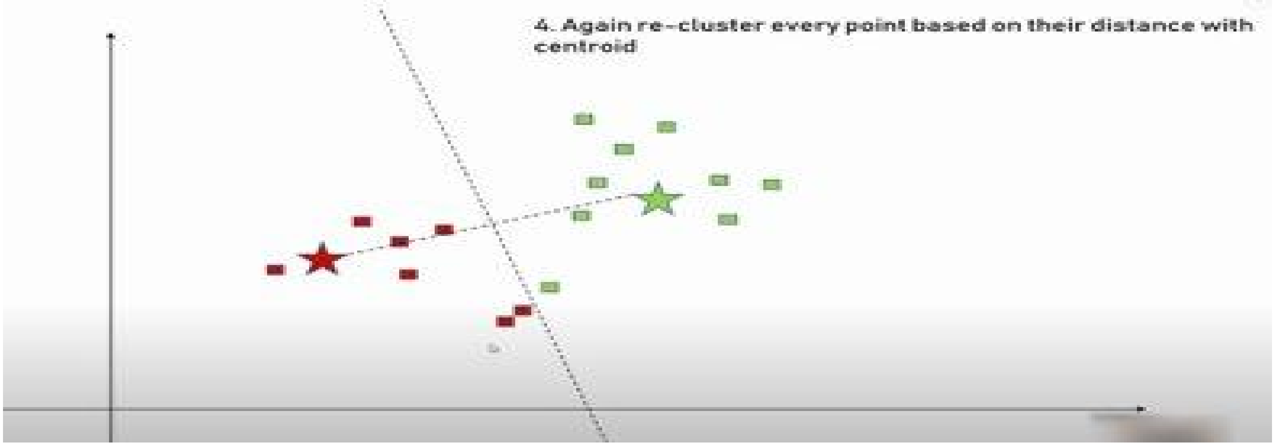
1. Start with K centroids by putting them at random place,
Here k = 2



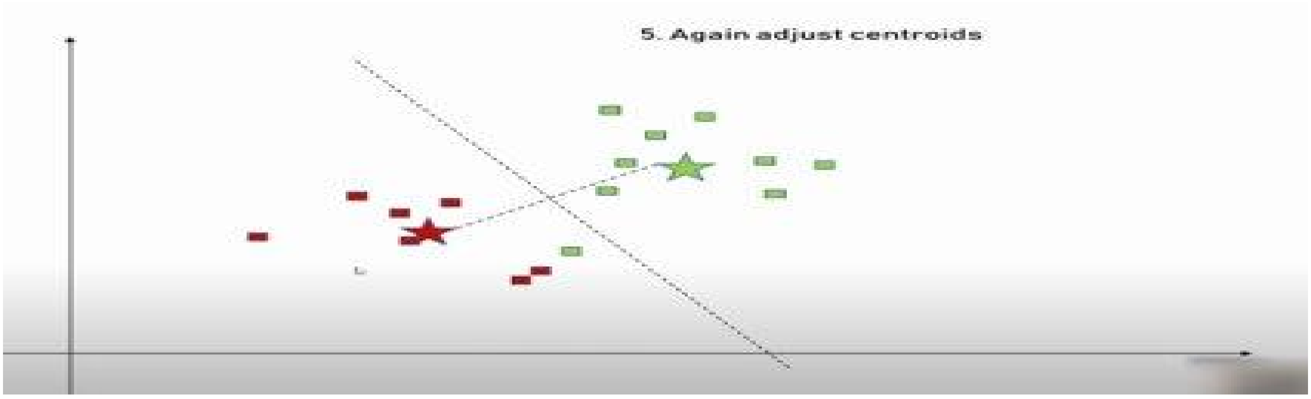
2. Compute distance of every point from centroid and cluster them accordingly

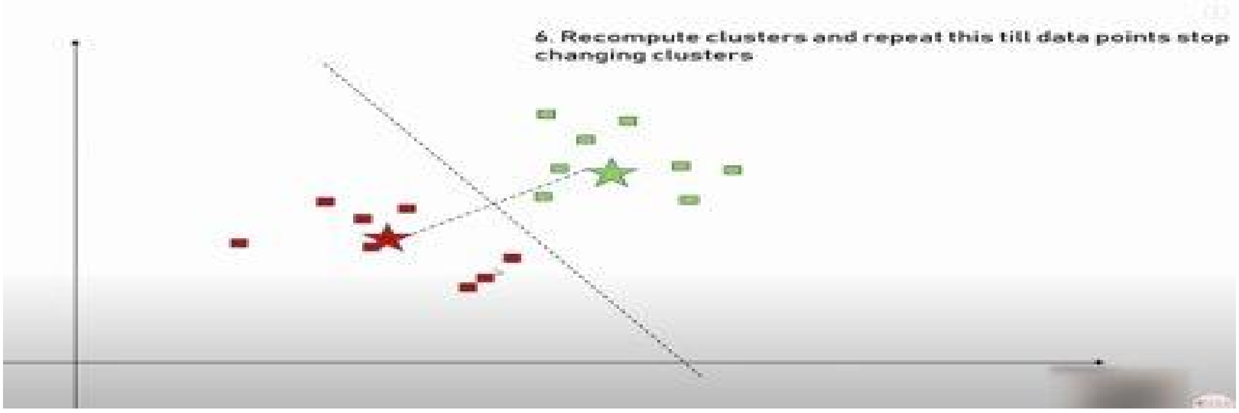


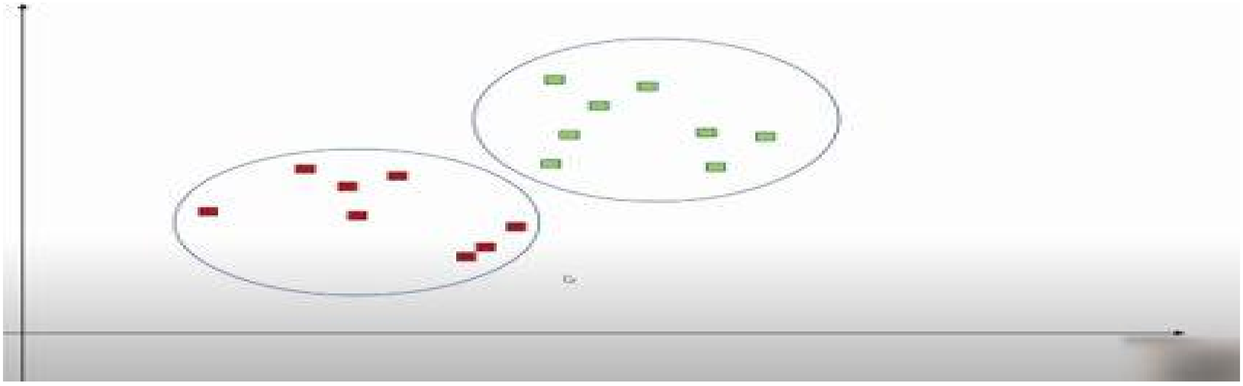
3. Adjust centroids so that they become center of gravity for given cluster



4. Again re-cluster every point based on their distance with centroid




6. Recompute clusters and repeat this till data points stop changing clusters



Stopping Criteria:

Basically there is three stopping criteria:
(i) Changes (data points movement between the cluster) does not improve cluster fit criteria i.e., homogeneity in cluster).
(ii) Data points stop shifting (data point's movement stop).
(iii) Set in advance, number of iterations

Unless k and n are extremely small, it is not feasible to compute the optimal clusters across all the possible combinations of examples. Instead, the algorithm uses a heuristic process that finds **locally optimal** solutions. Simply, this means that it starts with an initial guess for the cluster assignments, and then modifies the assignments slightly to see whether the changes improve the homogeneity within the clusters.

Note:

Due to the heuristic nature of k-means, we may end up with somewhat different final results by making only slight changes to the starting conditions. If the results vary dramatically, this could indicate a problem. For instance, the data may not have natural groupings or the value of k has been poorly chosen. With this in mind, it's a good idea to try a cluster analysis more than once to test the robustness of your findings.

Mathematical Explanation:

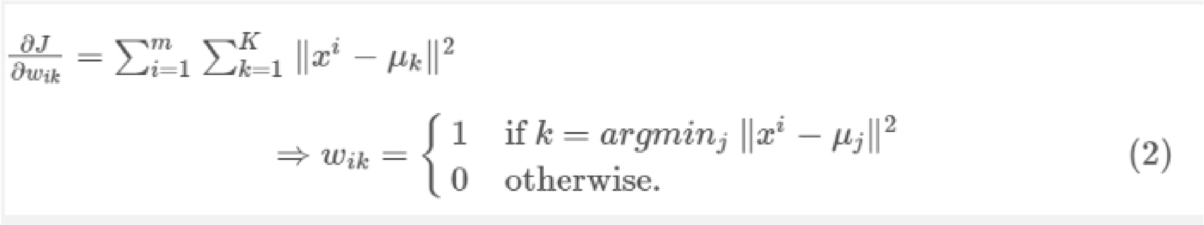
The approach used follows Expectation- Maximization. The E-step is assigning the data points to the closest cluster. The M-step is computing the centroid of each cluster. Below is a breakdown of how we can solve it mathematically.

The objective function is:

$$J = \sum_{i=1}^{m} \sum_{k=1}^{K} w_{ik} \|x^{i} - \mu_{k}\|^{2} \qquad (1)$$

where w_{ik}=1 for data point xi if it belongs to cluster k; otherwise, w_{ik}=0. Also, μk is the centroid of xi's cluster.

It's a minimization problem of two parts. We first minimize J w.r.t. w_{ik} and treat μk fixed. Then we minimize J w.r.t. μk and treat w_{ik} fixed. Technically speaking, we differentiate J w.r.t. w_{ik} first and update cluster assignments (E-step). Then we differentiate J w.r.t. μk and recompute the centroids after the cluster assignments from the previous step (M-step). Therefore, E-step is:

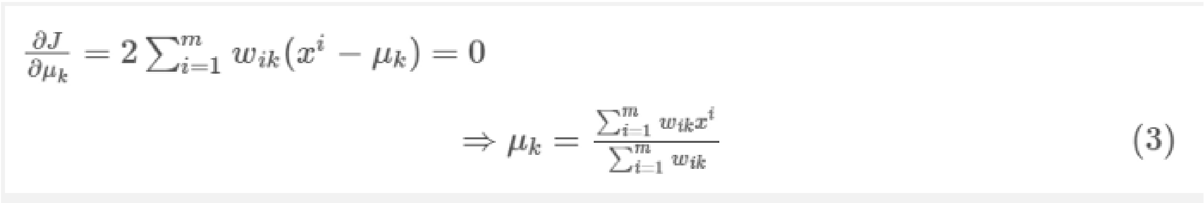


$$\frac{\partial J}{\partial w_{ik}} = \sum_{i=1}^{m} \sum_{k=1}^{K} \|x^i - \mu_k\|^2$$

$$\Rightarrow w_{ik} = \begin{cases} 1 & \text{if } k = argmin_j \; \|x^i - \mu_j\|^2 \\ 0 & \text{otherwise.} \end{cases} \qquad (2)$$

In other words, assign the data point xi to the closest cluster judged by its sum of squared distance from cluster's centroid.

And M-step is:



$$\frac{\partial J}{\partial \mu_k} = 2\sum_{i=1}^{m} w_{ik}(x^i - \mu_k) = 0$$

$$\Rightarrow \mu_k = \frac{\sum_{i=1}^{m} w_{ik}x^i}{\sum_{i=1}^{m} w_{ik}} \qquad (3)$$

Which translates to recomputing the centroid of each cluster to reflect the new assignments.

Few things to note here:

- Since clustering algorithms including k-means use distance-based measurements to determine the similarity between data points, it’s recommended to standardize the data to have a mean of zero and a standard deviation of one since almost always the features in any dataset would have different units of measurements such as age vs income.
- Given k-means iterative nature and the random initialization of centroids at the start of the algorithm, different initializations may lead to different clusters since k-means algorithms may be stuck in a local optimum and may not converge to a global optimum. Therefore, it’s recommended to run the algorithm using different initializations of centroids and pick the results of the run that yielded the lower sum of squared distance.
- Assignment of examples isn’t changing is the same thing as no change in within-cluster variation:

$$\frac{1}{m_k}\sum_{i=1}^{m_k}\left\|x^i - \mu_{c^k}\right\|^2 \tag{4}$$

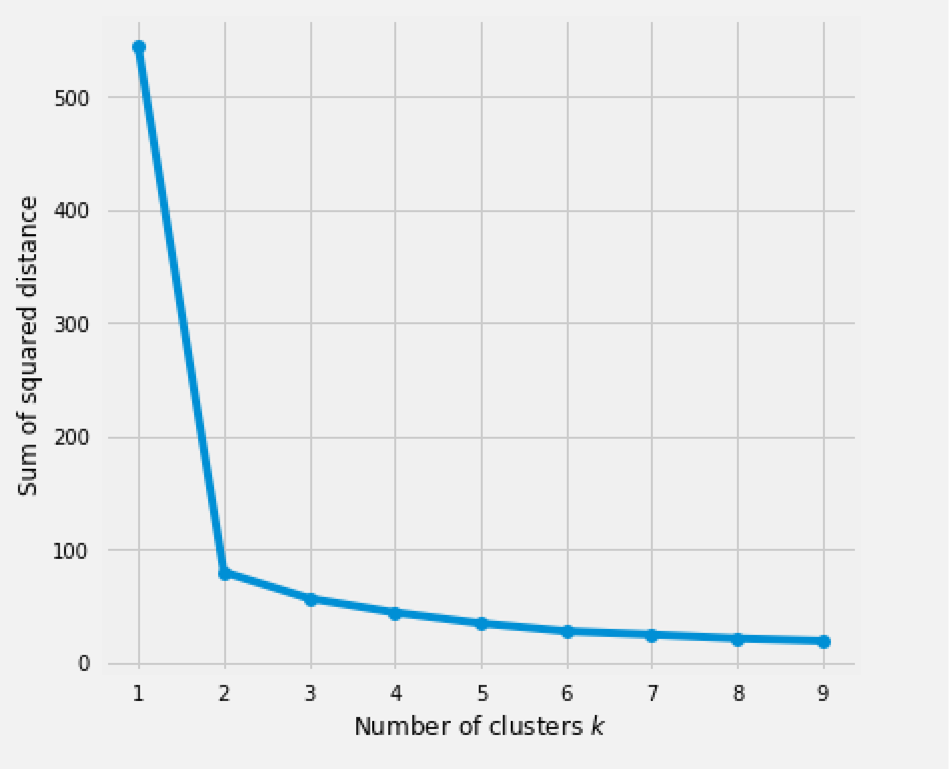
How to determine clusters?

Two metrics that may give us some intuition about k are:

- Elbow method
- Silhouette analysis

Elbow Method

The **Elbow** method gives us an idea on what a good k number of clusters would be based on the sum of squared distance (SSE) between data points and their assigned clusters' centroids. We pick k at the spot where SSE starts to flatten out and form an elbow.The following figure represents an evaluation of SSE for different values of k and see where the curve might form an elbow and flatten out.



The graph above shows that k=2 is not a bad choice. Sometimes it’s still hard to figure out a good number of clusters to use because the curve is monotonically decreasing and may not show any elbow or has an obvious point where the curve starts flattening out.

Silhouette analysis

-can be used to determine the degree of separation between clusters. For each sample:

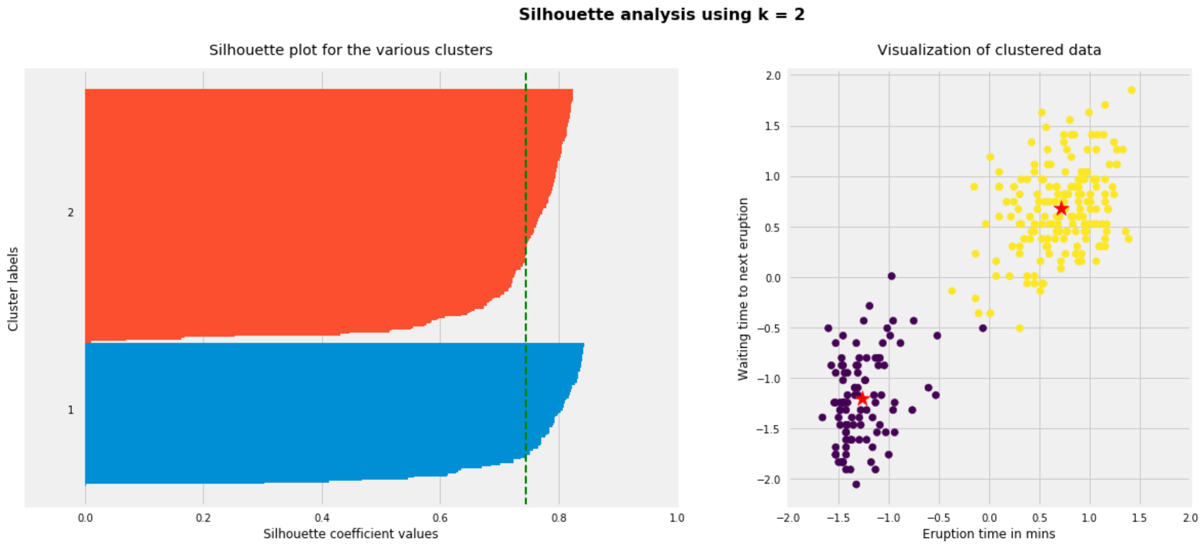
- Compute the average distance from all data points in the same cluster (ai).
- Compute the average distance from all data points in the closest cluster (bi).
- Compute the coefficient:

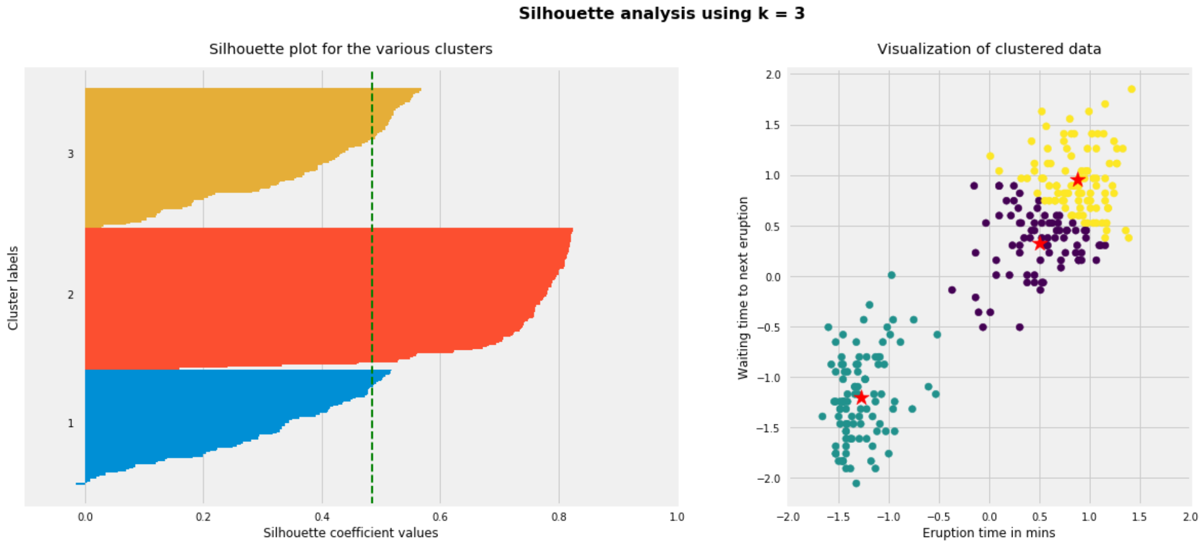
$$\frac{b^i - a^i}{max(a^i, b^i)}$$

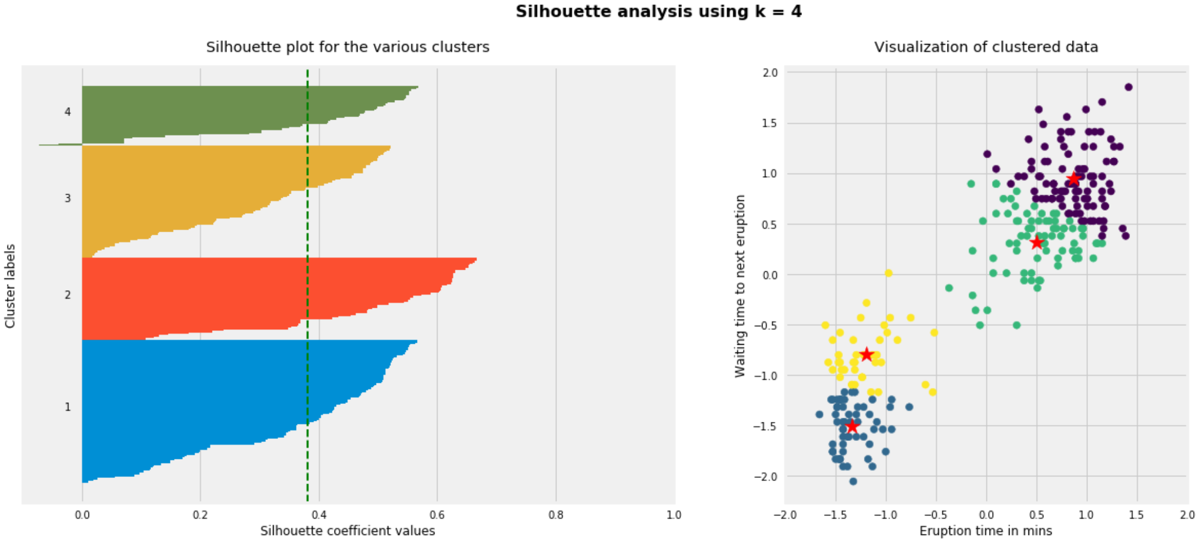
The coefficient can take values in the interval [-1, 1].

- If it is 0 –> the sample is very close to the neighboring clusters.
- If it is 1 –> the sample is far away from the neighboring clusters.
- If it is -1 –> the sample is assigned to the wrong clusters.

Therefore, we want the coefficients to be as big as possible and close to 1 to have a good cluster. The figure here again is used to run the silhouette analysis and it is actually obvious that there are most likely only two groups of data points.







As the above plots show, `n_clusters=2` has the best average silhouette score of around 0.75 and all clusters being above the average shows that it is actually a good choice. Also, the thickness of the silhouette plot gives an indication of how big each cluster is. The plot shows that cluster 1 has almost double the samples than cluster 2. However, as we increased `n_clusters` to 3 and 4, the average silhouette score decreased dramatically to around 0.48 and 0.39 respectively. Moreover, the thickness of the silhouette plot started showing wide fluctuations. The bottom line is: Good `n_clusters` will have a well above 0.5 silhouette average score as well as all of the clusters having higher than the average score.

Elbow methods can result in ambiguity so silhouette analysis comes into picture, both of them are used together for best results.

K-means Applications In Real Life:

1. Document classification:

clusters documents in multiple categories based on tags, topics, and the content of the document. This is a very standard classification problem and k-means is a highly suitable algorithm for this purpose. The initial processing of the documents is needed to represent each document as a vector and uses term frequency to identify commonly used terms that help classify the document. The document vectors are then clustered to help identify similarity in document groups.

2. Delivery store optimization

optimize the process of good delivery using truck drones by using a combination of k-means to find the optimal number of launch locations and a genetic algorithm to solve the truck route as a traveling salesman problem.

3. Identifying crime localities

With data related to crimes available in specific localities in a city, the category of crime, the area of the crime, and the association between the two can give quality insight into crime-prone areas within a city or a locality.

4. Customer segmentation

Clustering helps marketers improve their customer base, work on target areas, and segment customers based on purchase history, interests, or activity monitoring. Telecom providers can cluster prepaid customers to identify patterns in terms of money spent in recharging, sending sms, and browsing the internet. The classification would help the company target specific clusters of customers for specific campaigns.

5. Fantasy league stat analysis

Analyzing player stats has always been a critical element of the sporting world, and with increasing competition, machine learning has a critical role to play here. As an interesting exercise, if we would like to create a fantasy draft team and like to identify similar players based on player stats, k-means can be a useful option.

6. Insurance fraud detection

Machine learning has a critical role to play in fraud detection and has numerous applications in automobile, healthcare, and insurance fraud detection. utilizing past historical data on fraudulent claims, it is possible to isolate new claims based on its proximity to clusters that indicate fraudulent patterns. Since insurance fraud can potentially have a multi-million dollar impact on a company, the ability to detect frauds is crucial.

7. Rideshare data analysis

The publicly available uber ride information dataset provides a large amount of valuable data around traffic, transit time, peak pickup localities, and more. Analyzing this data is useful not just in the context of Uber but also in providing insight into urban traffic patterns and helping us plan for the cities of the future

8. Cyber-profiling criminals

Cyber-profiling is the process of collecting data from individuals and groups to identify significant correlations. The idea of cyber profiling is derived from criminal profiles, which provide information on the investigation division to classify the types of criminals who were at the crime scene.

9. Call record detail analysis

A call detail record (cdr) is the information captured by telecom companies during the call, sms, and internet activity of a customer. This information provides greater insights about the customer’s needs when used with customer demographics. We can cluster customer activities for 24 hours by using the unsupervised k-means clustering algorithm. It is used to understand segments of customers with respect to their usage by hours.

10. Automatic clustering of it alerts

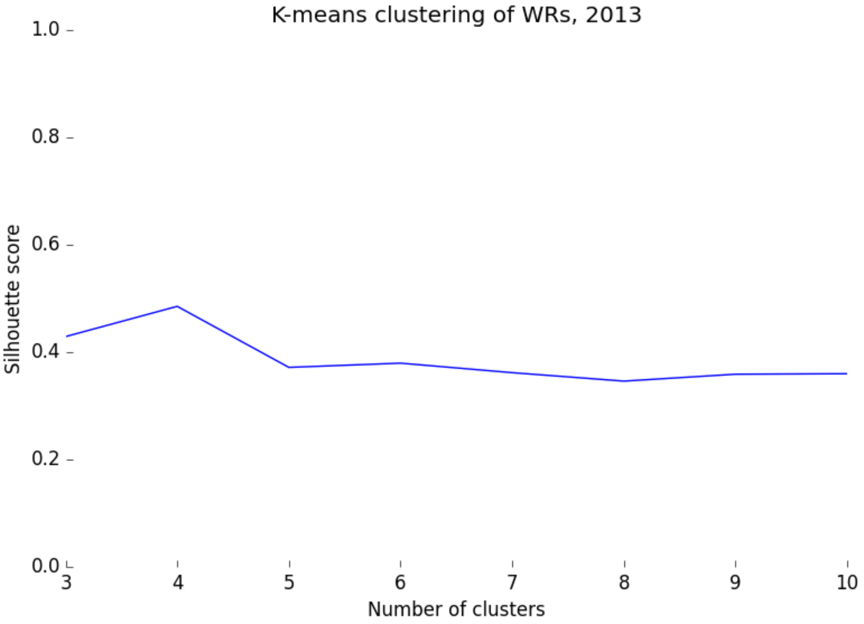
Large enterprise IT infrastructure technology components such as network, storage, or database generate large volumes of alert messages. Because alert messages potentially point to operational issues, they must be manually screened for prioritization for downstream processes. Clustering can provide insight into categories of alerts and mean time to repair, and help in failure predictions.

Analysis of a real life example- fantasy league predictions:

The analysis is done through an example of implementation of k- means clustering for wide receivers using their 2013 statistics. The basic idea is about taking the entire data set and dividing the observations into *k* sections and have each of the observations be as similar to each other as possible (and potentially as dissimilar to every other cluster as possible).

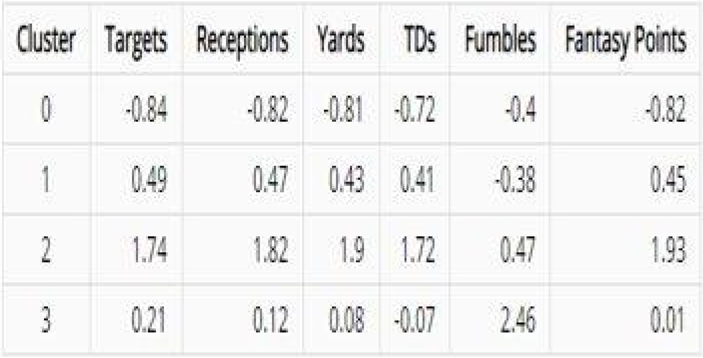
The wide receivers who played in 2013 were clustered using the following variables: targets, receptions, receiving yards, receiving touchdowns, fumbles, and fantasy points.

SILHOUETTE ANALYSIS:



We see that the silhouette score is maximized at *k* = 4, meaning 4 clusters of wide receivers.

The results of the clusters by applying k means are obtained as follows:



Cluster	Targets	Receptions	Yards	TDs	Fumbles	Fantasy Points
0	-0.84	-0.82	-0.81	-0.72	-0.4	-0.82
1	0.49	0.47	0.43	0.41	-0.38	0.45
2	1.74	1.82	1.9	1.72	0.47	1.93
3	0.21	0.12	0.08	-0.07	2.46	0.01

On analysis see cluster 2 is the best cluster with high receivers based on the input parameters that was fed to our algorithm to give a clustered output.

Drawbacks

K-means algorithm is good in capturing structure of the data if clusters have a spherical-like shape. It always tries to construct a nice spherical shape around the centroid. That means, the minute the clusters have complicated geometric shapes, kmeans does a poor job in clustering the data. In other words, data points in smaller clusters may be left away from the centroid in order to focus more on the larger cluster.It requires specifying the number of clusters (k) in advance. It cannot handle noisy data and outliers. It is not suitable to identify clusters with non-convex shapes.It is sensitive to initial conditions. Different initial conditions may produce different results of the cluster. The algorithm may be trapped in the local optimum.

Conclusion

K-means clustering is one of the most popular clustering algorithms and usually the first thing practitioners apply when solving clustering tasks to get an idea of the structure of the dataset. The goal of k-means is to group data points into distinct non-overlapping subgroups. It does a very good job when the clusters have a kind of spherical shape. However, it suffers as the geometric shapes of clusters deviate from spherical shapes. Moreover, it also doesn’t learn the number of clusters from the data and requires it to be pre-defined. It is the backbone of many sophisticated clustering algorithms.

Some points to conclude the k-means algorithm are written below:

- Scale/standardize the data when applying k means algorithm.
- The Elbow method in selecting the number of clusters doesn’t usually work because the error function is monotonically decreasing for all ks.
- Kmeans give more weight to the bigger clusters.
- Kmeans assumes spherical shapes of clusters (with radius equal to the distance between the centroid and the furthest data point) and doesn’t work well when clusters are in different shapes such as elliptical clusters.
- If there is overlapping between clusters, kmeans doesn’t have an intrinsic measure for uncertainty for the examples belonging to the overlapping region in

order to determine for which cluster to assign each data point.

- K Means may still cluster the data even if it can't be clustered such as data that comes from uniform distributions.

References:

1.https://towardsdatascience.com/k-means-clustering-algorithm-applications-evaluation-methods-and-drawbacks-aa03e644b48a
2.https://dzone.com/articles/10-interesting-use-cases-for-the-k-means-algorithm
3.http://thespread.us/clustering.html
4.https://en.wikipedia.org/wiki/K-means_clustering
5.https://towardsdatascience.com/understanding-k-means-clustering-in-machine-learning-6a6e67336aa1
6.https://www.youtube.com/watch?v=EItlUEPCIzM
7.https://www.analyticsvidhya.com/blog/2019/08/comprehensive-guide-k-means-clustering/
8.https://stanford.edu/~cpiech/cs221/handouts/kmeans.html
9.https://blogs.oracle.com/ai-and-datascience/post/introduction-to-k-means-clustering
10.https://www.ijert.org/research/analysis-and-study-of-k-means-clustering-algorithm-IJERTV2IS70648.pdf
11.https://www.mdpi.com/2571-8800/2/2/16/pdf
12.https://datafloq.com/read/7-innovative-uses-of-clustering-algorithms/6224