**Variations of K-means Algorithm Overview (Python)**

This overview explains the implementation of three variations of the K-means clustering algorithm: K-means, K-means++, and Bisecting K-means in Python.

**Algorithm Overview**

**K-Means**: A standard K-means clustering algorithm that randomly initializes centroids and iteratively updates them to minimize the sum of squared distances within each cluster.

**K-Means++:** An improved version of K-means that initializes centroids in a smarter way to speed up convergence. The initial centroids are selected from the dataset points such that the probability of choosing a point is proportional to the squared distance to the nearest existing centroid.

**Bisecting K-Means**: A hierarchical clustering approach that starts with the entire dataset as a single cluster and iteratively splits the cluster with the highest sum of squared distances into two using the K-means algorithm. The process continues until the desired number of clusters is reached.

**In the provided code, there are functions for each of these algorithms:**

k\_means\_clustering(k, D): Implements the K-means clustering algorithm

k\_means\_plus\_plus\_clustering(k, D): Implements the K-means++ clustering algorithm.

Bisecting\_kmeans(dataset, k): Implements the Bisecting K-means clustering algorithm.

**Additionally, the code includes supporting functions for each algorithm:**

K-means

**Initialize\_centroids(k, D):** Randomly selects k centroids from the dataset D.

**Assign\_clusters(D, centroids):** Assigns each data point to the nearest centroid.

**Update\_centroids(D, assignments, k):** Updates centroids by calculating the mean of all data points assigned to each centroid.

**Has\_converged(previous\_centroid, centroids, tolerance=1e-4):** Checks whether the centroids have converged using a predefined tolerance.

K-means++

**K\_means\_plus\_plus\_init(k, D):** Implements the K-means++ centroid initialization algorithm.

Bisecting K-means

This algorithm is directly implemented in the bisecting\_kmeans(dataset, k) function.

Additional functions

**Read\_data(file\_name):** Reads the dataset from a the file.

**Silhouette\_score(D, assignments, centroids):** Computes the Silhouette coefficient for the given clustering assignments and centroids.

**K\_means\_silhouette(data):** Runs K-means clustering and computes the Silhouette coefficient.

**Plot\_silhouette(k\_values, silhouette\_scores):** Plots the Silhouette coefficient versus the number of clusters k.

In the main function, the script reads a dataset, performs clustering using the three algorithms for various k values, computes the Silhouette coefficient for each clustering, and plots the results.