Code Report: K-Means Clustering Implementation

Gauranga Kumar Baishya (MDS202325) , Dipanjoy Saha (MDS202321)

Overview: The code provided includes a robust implementation of the K-means clustering algorithm, along with an array of supporting functions for dataset preprocessing, PCA visualization, and optimized cluster selection. A detailed report on the code's structure, functionality, and recommendations for enhancements is presented below..

Code Structure:

1. **Importing Libraries:** The code's functionality depends on the right libraries being imported. That's why the necessary libraries, such as pandas, numpy, scikit-learn, matplotlib, and others, are imported at the beginning of the code. This ensures that the code runs smoothly and efficiently, providing accurate results.
2. **Functions:**
    - Custom K-means Algorithm: The Custom K-means Algorithm is the one-stop solution for implementing the K-means clustering algorithm. It comes equipped with handy helper functions like assign_clusters and update_centroids, making it an efficient tool for all your clustering needs.
    - K-means ++: K-means++ initialization is the definitive method for selecting centroids. It guarantees optimal performance and results, making it an essential tool for any task that requires cluster analysis.
    - Functions for Jaccard Distance: Includes functions for calculating Jaccard distance and inertia.
    - Functions for Optimal K: Functions for computing inertia and plotting elbow method to find optimal clusters.
3. **Dataset Preprocessing:** Preprocess Enron, Kos, and Nips datasets for clustering.
4. **Clustering:**
    - Clustering Enron Dataset: We have successfully performed a clustering analysis on the Enron dataset, utilizing PCA to visualize the clusters in a lower-dimensional space. Moreover, I have confidently determined the optimal number of clusters through the application of the elbow method, which ensures that the model is both complex enough to capture relevant patterns and simple enough to avoid overfitting.
    - Clustering Kos Dataset: This section aims to cluster the Kos dataset and create visual representations of the resulting clusters, similar to what was done in the Enron dataset.
    - Clustering Nips Dataset: The task is to group the data in the Nips dataset into clusters and create a visual representation of these clusters.

Functionality:

- Custom K-means Algorithm: Implement K-means with custom initialization and convergence options for accurate data analysis.
- Jaccard Distance Functions: This tool offers features to compute the Jaccard distance between different data points.
- Elbow Method for Optimal K: Provides a method for determining the optimal number of clusters using the within-cluster sum of squares (inertia) metric.

- Dataset Preprocessing: Preprocesses datasets for clustering by merging, handling missing values, and converting them to appropriate formats.

- Visualization: By utilizing PCA, we can effectively reduce the dimensions of complex data and visualize their clusters in a two-dimensional space. This technique is incredibly useful for data analysis and can help us gain insights that may not be easily apparent in higher dimensions. So, if you want to analyze complex data sets with ease and precision, consider using PCA for dimensionality reduction and visualization.

Conclusion:

This code provides a detailed implementation of K-means clustering. It includes functions for pre-processing data, visualizing datasets, and determining optimal cluster count. With some improvements to documentation, error handling, and code optimization, it can become even more versatile and reusable.