Task 2.3

Q4 need to rewritten.

Centroids don't change: The algorithm stops when the centroids of the clusters no longer move. This means that the data points have been assigned to the closest clusters and there is no further improvement to be made.

Maximum number of iterations reached: The algorithm can also be stopped after a certain number of iterations, even if the centroids are still moving. This is done to prevent the algorithm from running forever.

Minimum change in the objective function: The objective function for K-means clustering is the sum of the squared distances between each data point and its assigned centroid. The algorithm can be stopped when the change in the objective function is less than some threshold.

Q5

The best number of clusters can be determined using the mean silhouette scores calculated for different values of K. The highest mean silhouette score generally indicates the most appropriate number of clusters.

Q6

K-means clustering is a widely used algorithm due to its simplicity and efficiency, but it does come with some limitations:

1. **Sensitive to Initial Centroids:** K-means clustering's performance heavily depends on the initial placement of centroids. If the initial centroids are chosen poorly or in a way that doesn't represent the data well, it might converge to a suboptimal solution or get stuck in a local minimum.
2. **Assumes Spherical Clusters:** K-means assumes that clusters are spherical and isotropic, which might not hold true for all datasets. It struggles when dealing with clusters of non-uniform size, density, or irregular shapes.
3. **Requires Predefined K:** The user needs to specify the number of clusters (K) beforehand, which might not be known in real-world applications. Determining the optimal number of clusters (e.g., using techniques like the elbow method or silhouette analysis) can be subjective and might not always yield the "correct" number of clusters.
4. **Sensitive to Outliers:** Outliers or noise in the data can significantly affect K-means clustering results. Outliers might create their own small clusters or influence the positions of centroids, leading to less accurate cluster assignments.
5. **May Converge to Local Optima:** The algorithm might converge to a local minimum instead of the global minimum, especially if it's run multiple times with different initializations and selects the clustering solution with the lowest inertia or within-cluster variation, which might not be the most optimal for the data.
6. **Scalability:** While K-means is relatively fast and efficient, it might struggle with high-dimensional data or datasets with a large number of samples, as the computational complexity increases with the size of the dataset.
7. **Not Suitable for Non-Numeric Data:** K-means is designed for numerical data and might not perform well with categorical or mixed-type data without proper preprocessing.

Understanding these limitations helps in choosing appropriate clustering algorithms or preprocessing techniques based on the characteristics of the data and the specific goals of the analysis. Other clustering methods like hierarchical clustering, DBSCAN, or Gaussian Mixture Models can be considered to address some of these drawbacks in different scenarios.

Task 2.4

Q5

1. **Computational Complexity**: KNN can be computationally expensive, especially with large datasets. The algorithm needs to compute distances for each new instance against all existing instances in the dataset during prediction, which can become inefficient as the dataset grows.
2. **Memory Usage**: Since KNN stores all training data, memory usage can be significant for large datasets. This can be a drawback in scenarios where memory resources are limited.
3. **Sensitive to Outliers**: Outliers or noisy data can significantly impact the performance of KNN. As it relies on distance measures, outliers can distort the decision boundaries and affect predictions.
4. **Choosing Optimal K**: Selecting the right value of K is crucial. A small value of K can make the model sensitive to noise, while a large value of K can make it less sensitive to local patterns. Determining the optimal K value is often more of an empirical task.
5. **Impact of Irrelevant Features**: KNN considers all features equally when computing distances. Irrelevant or less important features can negatively impact the algorithm's performance.
6. **Imbalanced Data**: In datasets where classes are imbalanced (i.e., one class has significantly more instances than others), KNN tends to favor the majority class, leading to biased predictions.
7. **No Training Phase**: KNN doesn't have a training phase; it stores the entire dataset and performs computations during prediction. While this can be advantageous in some scenarios, it also means it doesn't learn from data and might not generalize well to unseen data.