深 圳 大 学 实 验 报 告

课程编号：**2801000049**

课程名称：机器学习

实验项目名称：**Machine Learning Task 3**

学院：电子与信息工程学院

专业：电子信息工程

指导教师：麦晓春

报告人：廖祖颐 学号：**2022110131** 班级：文华班

实验时间：**2024.5.27**

实验报告提交时间：**2024.6.6**

教务部制

1. Experimental Purposes and Requirements

(1). Load 5 datasets

noisy_circles.txt(n_clusters=2)

noisy_moons.txt(n_clusters=2)

blobs.txt(n_clusters=3)

aniso.txt(n_clusters=3)

no_structure.txt(n_clusters=3)

(2). Drawing scatterplots to visualize data sets

(3). Use K-means Clustering on all the data sets separately, and visualize the clustering results

(4). Use DBScan on all the data sets separately, adjust the parameter eps for different datasets, and visualize the clustering results

2. Experiment Contents and Process

My program is mainly divided into two large sections, one is the parameter selection part, the other is the clustering result comparison part

Datasets:

The experiment involves five different synthetic datasets:

- noisy_circles: Points arranged in a noisy circular pattern.
- noisy_moons: Points distributed in two touching crescent shapes with noise.
- blobs: Randomly distributed blobs of points.
- aniso: Anisotropically distributed data.
- no_structure: Randomly scattered points with no clear structure.

Part one (parameter selection of K-means)

K-means Clustering:

- K-means clustering is performed for each dataset for a range of cluster numbers (k values from 1 to 7).
- For each k, the clustering process involves randomly initializing cluster centers (centroids), assigning each data point to the nearest centroid, and iteratively adjusting the

centroids until the positions stabilize or a maximum number of iterations is reached.

Calculation of Inertia:

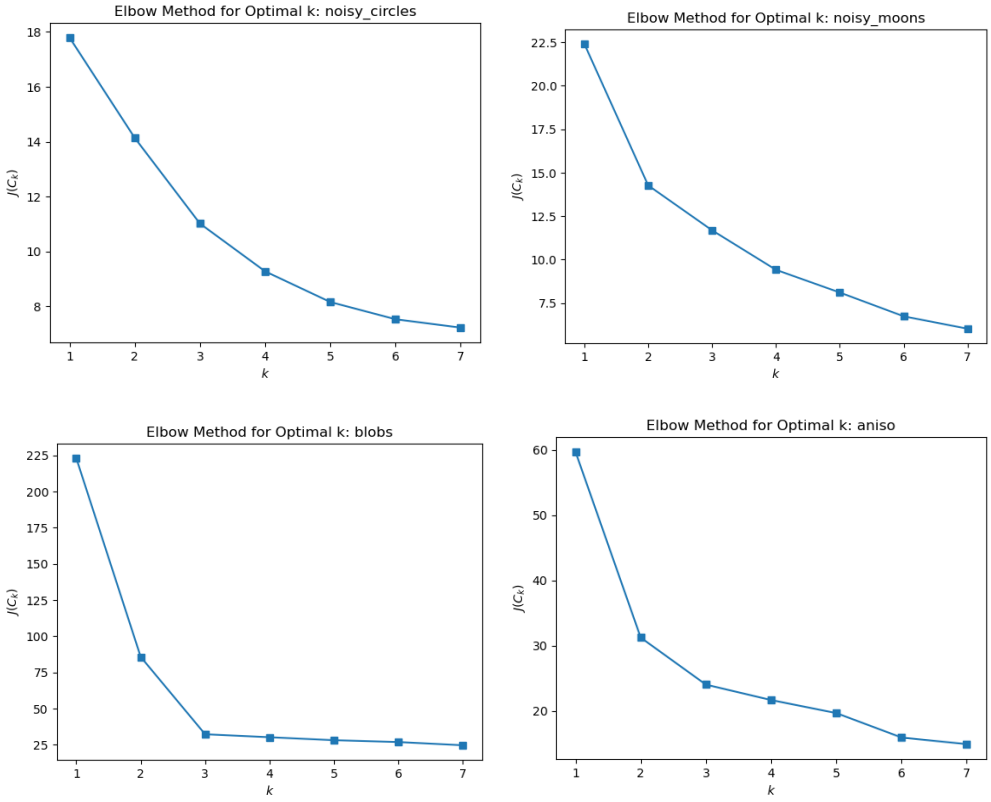
- Inertia, which measures the sum of squared distances of samples to their closest cluster center, is calculated for each k. This measure helps in assessing how well a dataset is clustered by K-means.
- The square root of inertia is calculated to potentially improve the visualization and interpretation of results.

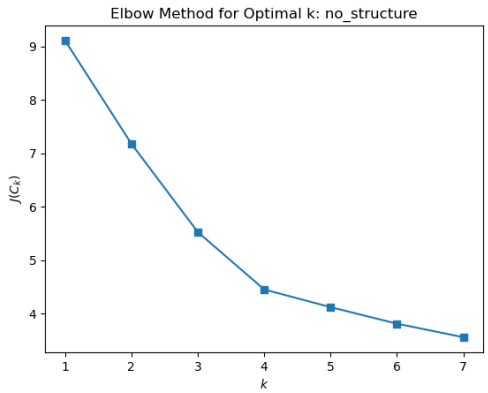
Visualization:

- Inertia values for different k's are plotted to visualize the "elbow" method. The elbow point on the plot typically indicates the optimal number of clusters, where further increases in k result in diminishing returns in terms of decreased inertia.

Results:

- The experiment produces a series of plots, one for each dataset, illustrating how inertia changes with different numbers of clusters.
- The optimal number of clusters for each dataset is identified based on where the plot shows an "elbow" — the point after which increases in the number of clusters result in smaller reductions in inertia.
- The plot for each dataset is reviewed to determine the elbow point, which suggests the most appropriate number of clusters for that particular dataset.





So we can get the number of clusters

```
# Datasets file names and number of clusters
datasets = {
    "noisy_circles.txt": 2,
    "noisy_moons.txt": 2,
    "blobs.txt": 3,
    "aniso.txt": 3,
    "no_structure.txt": 3
}
```

Part one (parameter selection of DBScan)

Distance Sorting and Derivative Calculation:

- For each dataset, distances to the k-th nearest neighbor are sorted in ascending order. These distances are crucial as they represent the density around each point.
- The first derivative of the sorted distances is calculated to identify where the rate of increase in distance shows significant change.
- Subsequently, the second derivative is computed to pinpoint more precisely the "elbow point" where the acceleration in distance increase is the highest. This point typically indicates a good choice for the eps parameter.

Visualization:

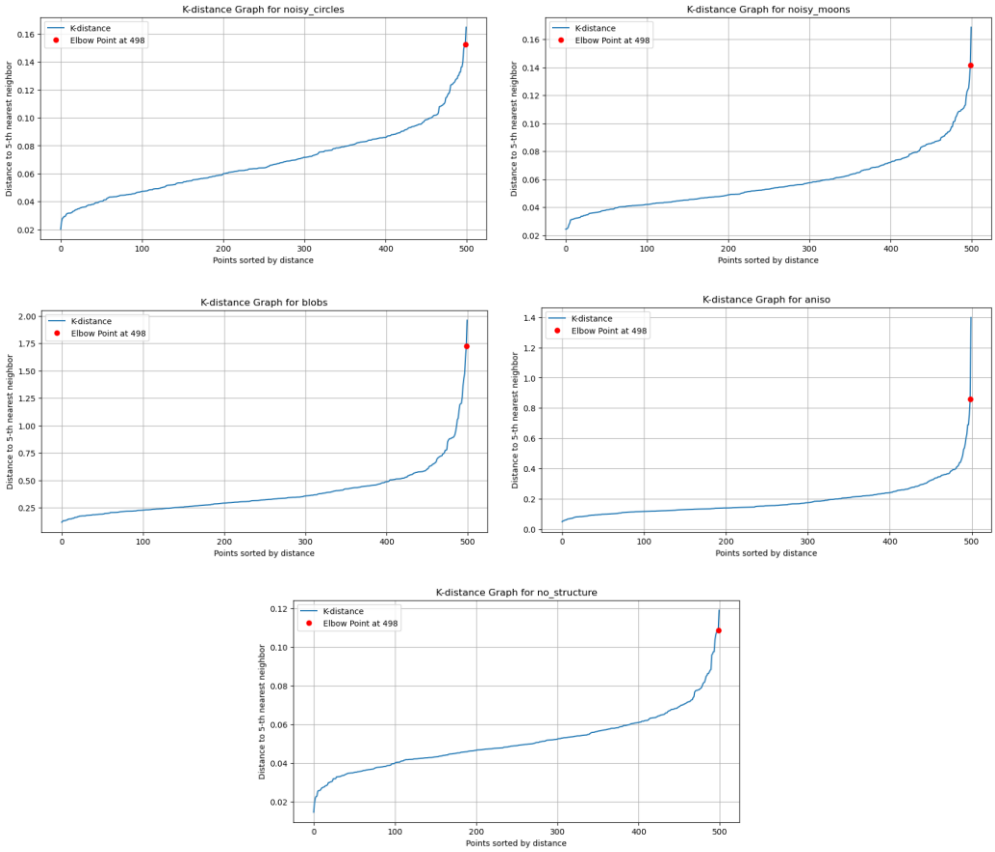
- For each dataset, a k-distance graph is plotted showing the sorted distances. The identified elbow point is marked clearly to visualize the suggested eps value.
- The graph includes annotations or markers at the elbow point to highlight the recommended eps value for DBSCAN clustering.

Elbow Point Detection:

- The elbow point is determined algorithmically by finding the maximum in the second derivative of the sorted distances, indicating the greatest change in density.

Results:

- The experiment produces a series of k-distance graphs, one for each dataset. Each graph visualizes the sorted distances to the k-th nearest neighbor and marks the elbow point.
- These results help to determine the optimal eps values for DBSCAN clustering by identifying points where the increase in distance suggests a transition to a less dense region.



So we can get the eps parameters for DBSCAN, as the photo show, aniso.txt's eps parameter is not selected according to the clearly marked points in the image, because that parameter is less effective

```
# Set the eps parameters for DBSCAN
eps_values = {
    "noisy_circles.txt": 0.14,
    "noisy_moons.txt": 0.12,
    "blobs.txt": 1.75,
    "aniso.txt": 0.4,
    "no_structure.txt": 0.11
}
```

Part two——clustering result comparison

Clustering Execution:

- K-means Clustering: Implemented for each dataset with specified n_clusters, using n_init='auto' to allow the algorithm to set the number of initialization runs automatically.
- DBSCAN Clustering: Configured with specific eps and min_samples for each dataset, taking into account the density and spread of the data.

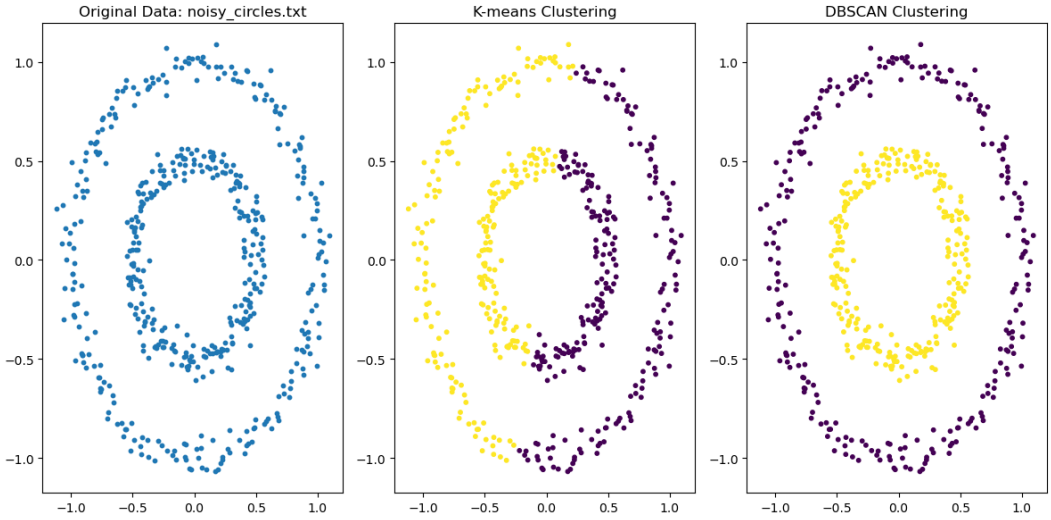
Visualization:

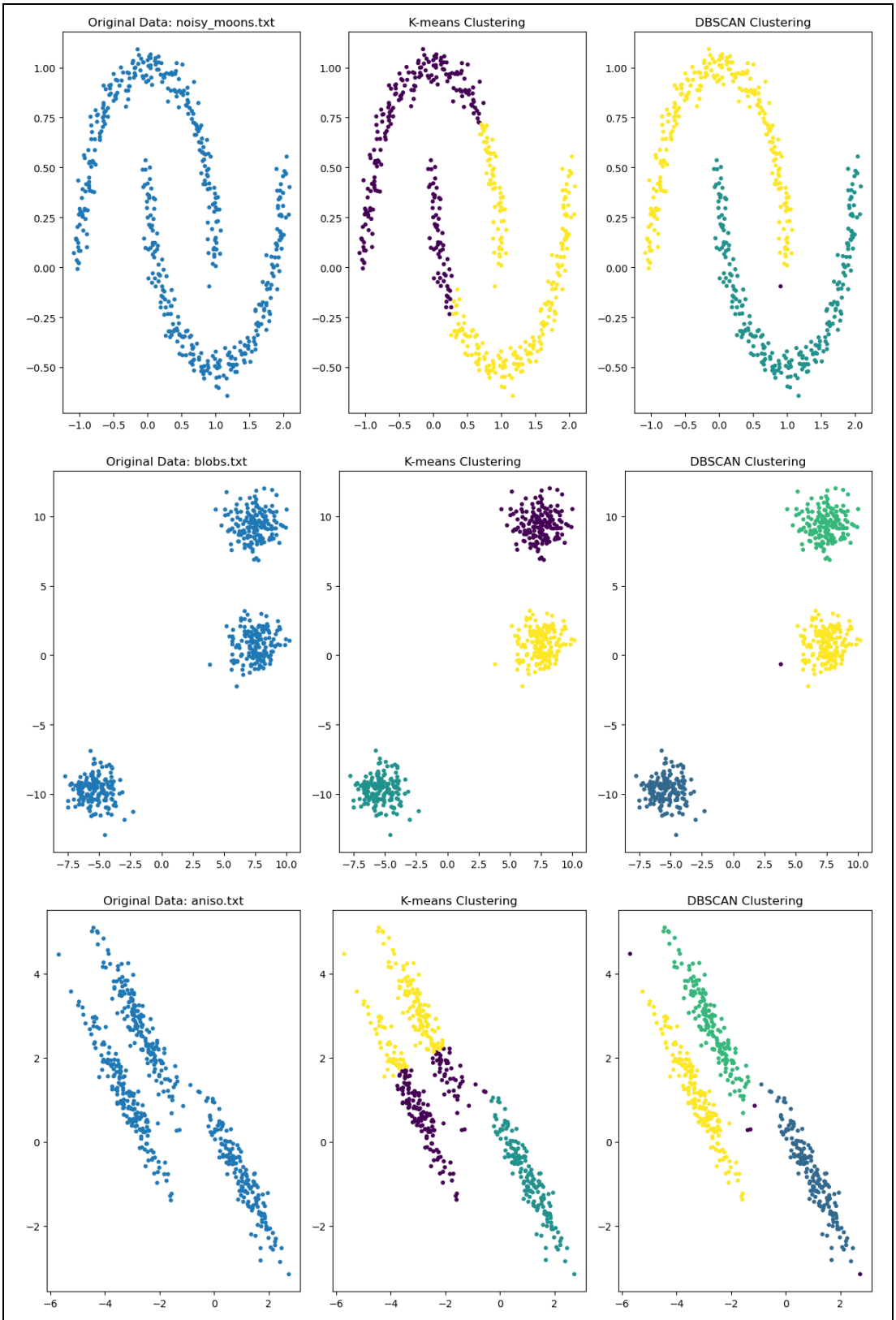
Three plots are generated for each dataset:

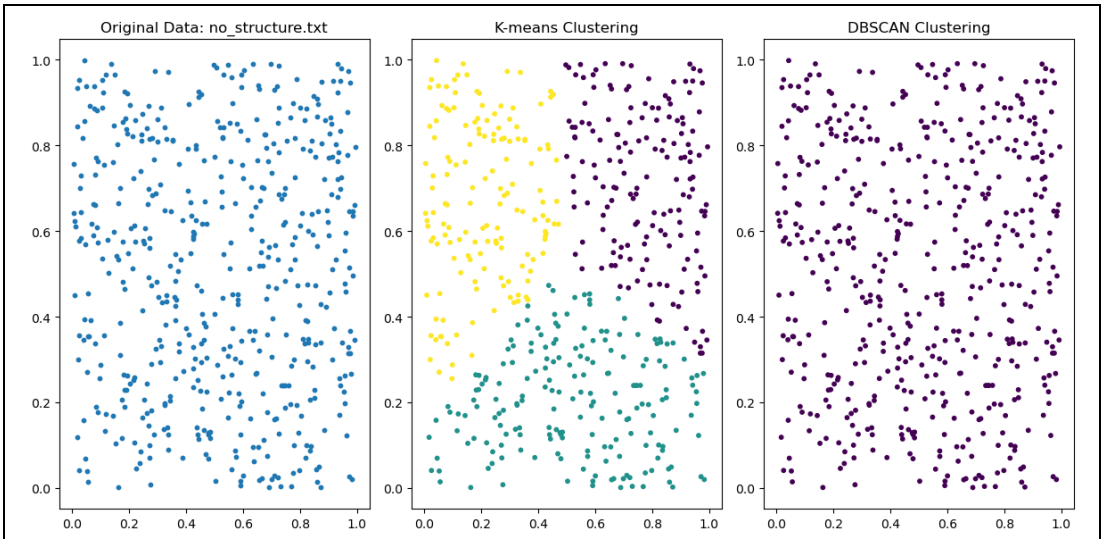
- Original Data: Displays unclustered data points.
- K-means Clustering Results: Shows clustering outcomes with points colored based on their assigned clusters.
- DBSCAN Clustering Results: Illustrates how DBSCAN identifies core, border, and noise points with colors indicating different clusters.

Results:

- Visualization of clustering results for each dataset provides insights into the effectiveness of K-means and DBSCAN for handling different types of data distributions.
- Comparison between the two clustering methods highlights their respective strengths and weaknesses: K-means in handling spherical clusters and DBSCAN in capturing clusters of arbitrary shapes and coping with noise.




Original Data: noisy_moons.txt
K-means Clustering
DBSCAN Clustering
Original Data: blobs.txt
K-means Clustering
DBSCAN Clustering
Original Data: aniso.txt
K-means Clustering
DBSCAN Clustering


Original Data: no_structure.txt
K-means Clustering
DBSCAN Clustering
0.0
0.2
0.4
0.6
0.8
1.0

3. Discussion and Conclusions

PART one (parameter selection of K-means)

Detailed Analysis of Each Dataset:

1. **Noisy Circles:**
   - Inertia Plot: The plot shows a sharp decline in inertia from k=1 to k=2, and a gradual decrease thereafter.
   - Optimal k: The Elbow Method suggests k=2 as optimal, aligning with the visual composition of two concentric noisy circles in the dataset.
2. **Noisy Moons:**
   - Inertia Plot: There is a rapid decrease in inertia from k=1 to k=2, indicating that two clusters are significantly better than one for capturing the dataset's structure.
   - Optimal k: The elbow at k=2 matches the two interlocking crescent shapes of the noisy moons.
3. **Blobs:**
   - Inertia Plot: The plot shows a sharp drop from k=1 to k=3, then levels off, suggesting little gain in further dividing the data.
   - Optimal k: The optimal number of clusters appears to be 3, consistent with the three visually distinct blobs.
4. **Aniso:**
   - Inertia Plot: The decline in inertia from k=1 to k=3 is steep, with more subtle reductions up to k=5.
   - Optimal k: Given the stretched nature of the data distributions, k=3 is reasonable, capturing the major directions of data spread.
5. **No Structure:**
   - Inertia Plot: The plot declines steadily without a clear elbow, reflecting the lack of inherent clustering structure.
   - Optimal k: Choosing an optimal k is less meaningful for this dataset; however, k=2 or k=3 might be selected to minimize inertia without overfitting to noise.

Discussion:

- K-means Suitability: K-means performs well with isotropic clusters (e.g., blobs) but struggles with more complex shapes or overlapping clusters (e.g., noisy moons or aniso). This is due to its reliance on minimizing Euclidean distance to centroids, which biases towards spherical clusters.
- Choosing k: The Elbow Method is effective for identifying a reasonable number of clusters but can be subjective, especially in datasets without a clear structure. The method relies heavily on visual interpretation of the inertia plot.

Experiment Conclusions:

- Effectiveness of Elbow Method: The method proved effective in most cases but highlighted its limitations with datasets lacking clear or uniform

clustering patterns.

- Algorithm Selection: This analysis reinforces the importance of selecting the right clustering algorithm based on data characteristics. For non-spherical or unevenly distributed data, algorithms like DBSCAN or hierarchical clustering might be more appropriate.

PART one(parameter selection of DBScan)

Detailed Analysis of Each Dataset:

1. **Noisy Circles**
   Identified at the 498th point, where there is a significant jump in distance to the 5th nearest neighbor, indicating a transition between the densities of the inner and outer circles.
2. **Noisy Moons**
   Also at the 498th point, showing a sharp increase in distance, likely reflecting the sparse region between the two moon shapes.
3. **Blobs**
   Occurs at the 498th point with a large distance increase, which could be due to the spacing between the distinct blobs.
4. **Aniso**
   Detected at the 498th point with a marked increase in distance.
5. **No Structure**
   Found at the 498th point, indicating a significant jump in distances due to the overall low density of the dataset.

Discussion:

- **K-Distance Graph:** The k-distance graph proved to be a valuable tool in all datasets for identifying the eps parameter. The graph shows where distances between points begin to increase significantly, which is critical for setting the radius of neighborhood for the DBSCAN algorithm.
- **Influence of Data Distribution:** The characteristics of each dataset heavily influenced where the elbow point appeared. Uniformly distributed datasets like blobs showed clear and distinct elbows, whereas more complex or less structured datasets presented challenges in determining a single clear elbow point.

Experiment Conclusion:

This experiment highlighted the effectiveness of the k-distance method in estimating the eps parameter for DBSCAN across various synthetic datasets. It demonstrated that while the method is generally effective, the complexity of the data structure can significantly impact the ease and clarity of identifying the optimal eps. For practical applications, it might be necessary to combine this approach with other methods or heuristic adjustments based on the specific characteristics and requirements of the data.

PART two——clustering result comparison

Dataset Characteristics and Clustering Results:

1. **Noisy Circles**:
   - **Original Data**: Consists of two concentric circles with noise.
   - **K-means Clustering**: Struggles to meaningfully separate the two circles due to its reliance on centroid-based clustering which does not handle circular structures well.
   - **DBSCAN Clustering**: Effectively identifies the two circular structures as separate clusters due to its density-based approach.
2. **Noisy Moons**:
   - **Original Data**: Two interlocking crescent shapes with added noise.
   - **K-means Clustering**: Unable to capture the true shape of the moons, dividing them somewhat arbitrarily.
   - **DBSCAN Clustering**: Successfully delineates the two moons, recognizing the non-linear separation between them.
3. **Blobs**:
   - **Original Data**: Three well-separated clusters of points.
   - **K-means Clustering**: Excellently partitions the three blobs, demonstrating its effectiveness with well-separated, globular clusters.
   - **DBSCAN Clustering**: Also performs well, correctly identifying the three clusters, highlighting its versatility.
4. **Aniso**:
   - **Original Data**: Data points are stretched or elongated along a diagonal axis.
   - **K-means Clustering**: Partially effective but does not perfectly align with the natural distribution of the data.
   - **DBSCAN Clustering**: More effectively adapts to the anisotropic nature of the data, clustering based on density rather than distance to a centroid.
5. **No Structure**:
   - **Original Data**: Random scatter of points without any clear clusters.
   - **K-means Clustering**: Imposes structure where there is none, creating arbitrary clusters.
   - **DBSCAN Clustering**: Most points are classified as noise, which is a more accurate representation of the dataset's lack of structure.

Discussion:

1. **Algorithm Suitability**:
   - **K-means**: Performs well with isotropic clusters (e.g., blobs) but fails with nonlinear shapes or overlapping clusters due to its reliance on Euclidean distances to centroids.
   - **DBSCAN**: Excels in datasets with complex structures or varying densities and is particularly adept at handling outliers and noise.
2. **Challenges and Considerations**:
   - **Parameter Sensitivity**: Both algorithms have critical parameters that

significantly affect their performance (e.g., number of clusters for K-means, and eps and min_samples for DBSCAN). Choosing these parameters requires careful consideration of the dataset characteristics or additional diagnostic tools like the elbow method for K-means and k-distance plots for DBSCAN.

- **Data Density and Distribution**: DBSCAN's performance advantage in datasets with non-uniform density distributions highlights the importance of choosing a clustering algorithm based on the specific data characteristics.

Conclusion:

The experiment underscores the importance of selecting the right clustering tool for specific types of data. K-means is suitable for simple, well-separated clusters but may not be ideal for complex structures. In contrast, DBSCAN offers robust performance across a broader range of datasets, particularly where the cluster model does not conform to spherical shapes.

Some problem:

When I use the k-means, the program always report an strange error, I tried to solve it in jupyter notebook and computer command line without success, In fact, this error has no effect on the results of the experiment

```
F:\anaconda\Lib\site-packages\sklearn\cluster\_kmeans.py:1382: UserWarning: KMeans is known to have a memory leak on Windows with MKL, when there are less chunks than available threads. You can avoid it by setting the environment variable OMP_NUM_THREADS=2.
  warnings.warn(
```

指导教师批阅意见： 成绩评定： 指导教师签字： 年　　月　　日
备注：

注：1、报告内的项目或内容设置，可根据实际情况加以调整和补充。

2、教师批改学生实验报告时间应在学生提交实验报告时间后 10 日内。