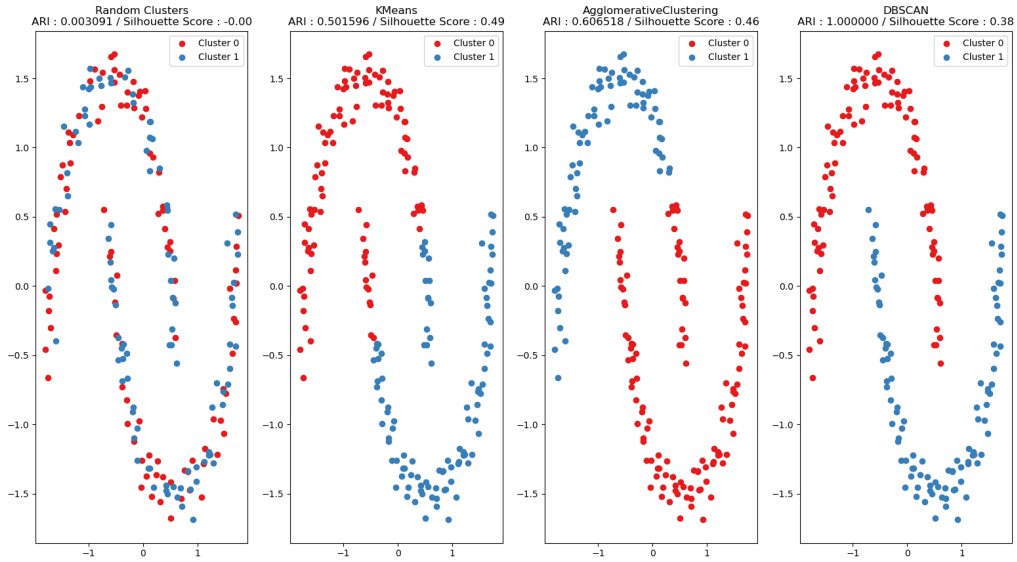
Q6. Clustering Algorithm Comparison

```
import numpy as np
import os

import matplotlib as mpl
import matplotlib.pyplot as plt
from matplotlib.cm import get_cmap

from sklearn.datasets import make_moons
from sklearn.preprocessing import StandardScaler
from sklearn.metrics.cluster import adjusted_rand_score     # Criterion used to assess the accuracy of clustering (prediction vs cluster label)
from sklearn.metrics.cluster import silhouette_score        # Criterion used to assess the density of clustering (data vs cluster label)

from sklearn.cluster import KMeans
from sklearn.cluster import AgglomerativeClustering
from sklearn.cluster import DBSCAN

################################################
### Comparison of Various Clustering Algorithms ###
################################################

### Prepare dataset ##########################################################################################################################
X, y = make_moons(n_samples=200, noise=0.05, random_state=0)    # Moon dataset with 200 samples

X_standardized = StandardScaler().fit_transform(X)      # Standardize the given datset

# Assign random cluster label for each data in the datset
random_state = np.random.RandomState(seed=0)
random_clusters = random_state.randint(low=0, high=2, size=len(X))

### Prepare various clustering algorithms
clusterer = [KMeans(n_clusters=2), AgglomerativeClustering(n_clusters=2), DBSCAN()]

### Plot clustering results of various clustering algorithms ##################################################################################
# Plot current random dataset ################################################################################################################
plt.subplot(1, 4, 1)
cmap = get_cmap('Set1')         # Prepare color map / Each cluster uses an distinctive color
legend = []
for label in np.unique(random_clusters):

    # Plot only the points that correspond to certain cluster label using X_standardized[random_clusters==label]
    # Assign the color to the points in the dataset according to their labels
    plt.scatter(X_standardized[random_clusters==label][:, 0], X_standardized[random_clusters==label][:, 1],
                c=cmap.colors[label], label='Cluster '+ str(label))

    legend.append('Cluster '+ str(label))

plt.legend(legend, loc='best')
plt.title('Random Clusters\nARI : {:.6f} / Silhouette Score : {:.2f}'.format(adjusted_rand_score(y, random_clusters),
                                                                             silhouette_score(X_standardized, random_clusters)))

# Plot the clustering result of each clustering algorithm ####################################################################################
for i in range(len(clusterer)):     # For each clustering algorithm...

    plt.subplot(1, 4, i+2)

    clusters = clusterer[i].fit_predict(X_standardized)     # Cluster the standardized dataset with current clusterer

    cmap = get_cmap('Set1')     # Prepare color map / Each cluster uses an distinctive color
    legend = []
    for label in np.unique(clusters):

        # Plot only the points that correspond to certain cluster label using X_standardized[clusters==label]
        # Assign the color to the points in the dataset according to their labels
        plt.scatter(X_standardized[clusters==label][:, 0], X_standardized[clusters==label][:, 1], c=cmap.colors[label], label='Cluster '+ str(label))

        legend.append('Cluster '+ str(label))

    plt.legend(legend, loc='best')

    # Display criterion used to assess the accuracy of clustering for current clusterer (adjusted_rand_score)
    # and criterion used to assess the density of clustering (silhouette_score)
    plt.title('{}\nARI : {:.6f} / Silhouette Score : {:.2f}'.format(clusterer[i].__class__.__name__,
                                                                    adjusted_rand_score(y, clusters),
                                                                    silhouette_score(X_standardized, clusters)))

    # This plot shows that density of clustering is not an effective criterion for determining the accuracy of clustering algorithm
    # It is recommended to use the score of clustering algorithm as the criterion for the effectiveness of clustering algorithm

plt.show()
```



- **Clustering 알고리즘을 판단하는 기준으로 Density 의 정도를 사용하는 것은 부적합함 .**
- **위 그래프에서 DBSCAN 과 K-Means, Agglomerative Clustering 을 비교하면 Density 척도인 Sillhouette Score 는 K-Means 가 제일 높지만 근거리로 가까이 있는 서로 다른 Cluster 를 분리 못 하는 것을 볼 수 있음 .**
- **Cluster 생성 기준에 중심점 - 데이터 거리를 주로 반영하는 K-Means 와 Agglomerative Clustering 은 근거리로 가까이 있는 서로 다른 Cluster 를 분리 못 하는 단점이 있음 .**
- **DBSCAN 은 거리 + Cluster 생성 기준이라는 좀 더 복잡한 기준으로 Clustering 을 하기에 근거리로 가까이 있는 서로 다른 Cluster 를 잘 분리할 수 있음 . 그러나 거리 기준 (eps) 와 Cluster 생성 기준 (min_samples) 에 크게 영향을 받기에 Parameter Tuning 이 쉽지 않음 .**