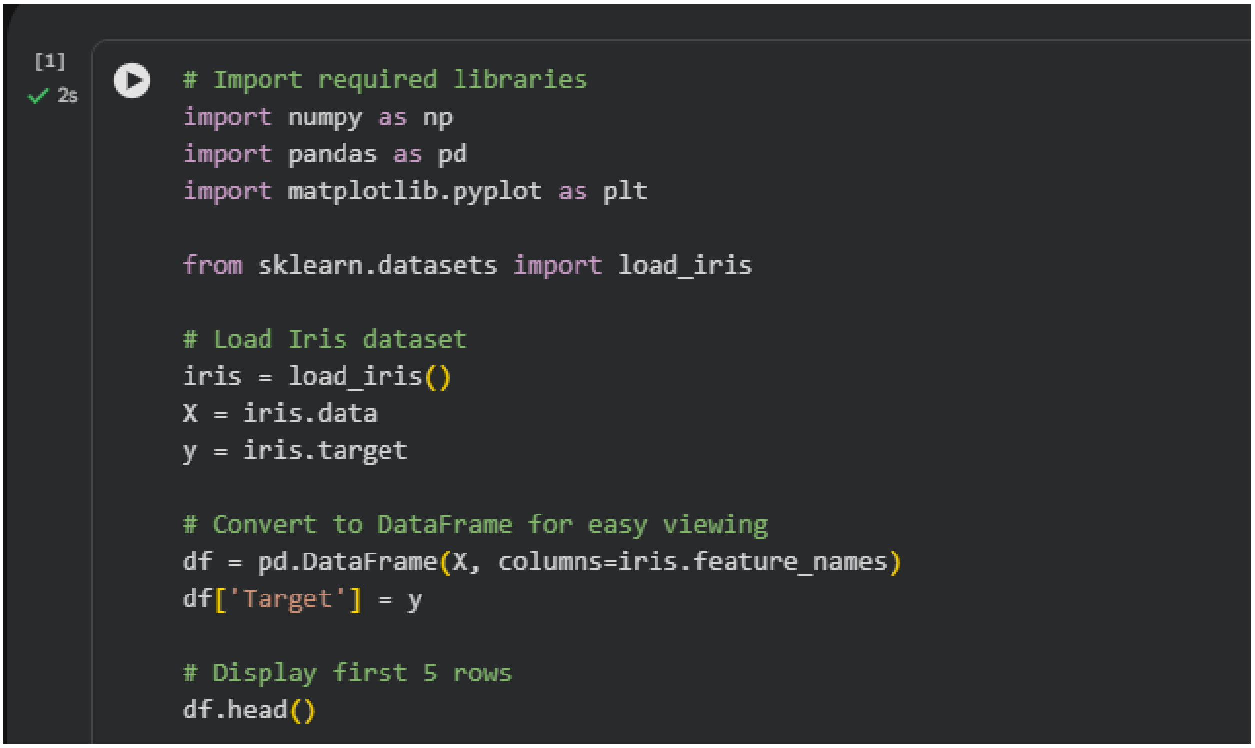
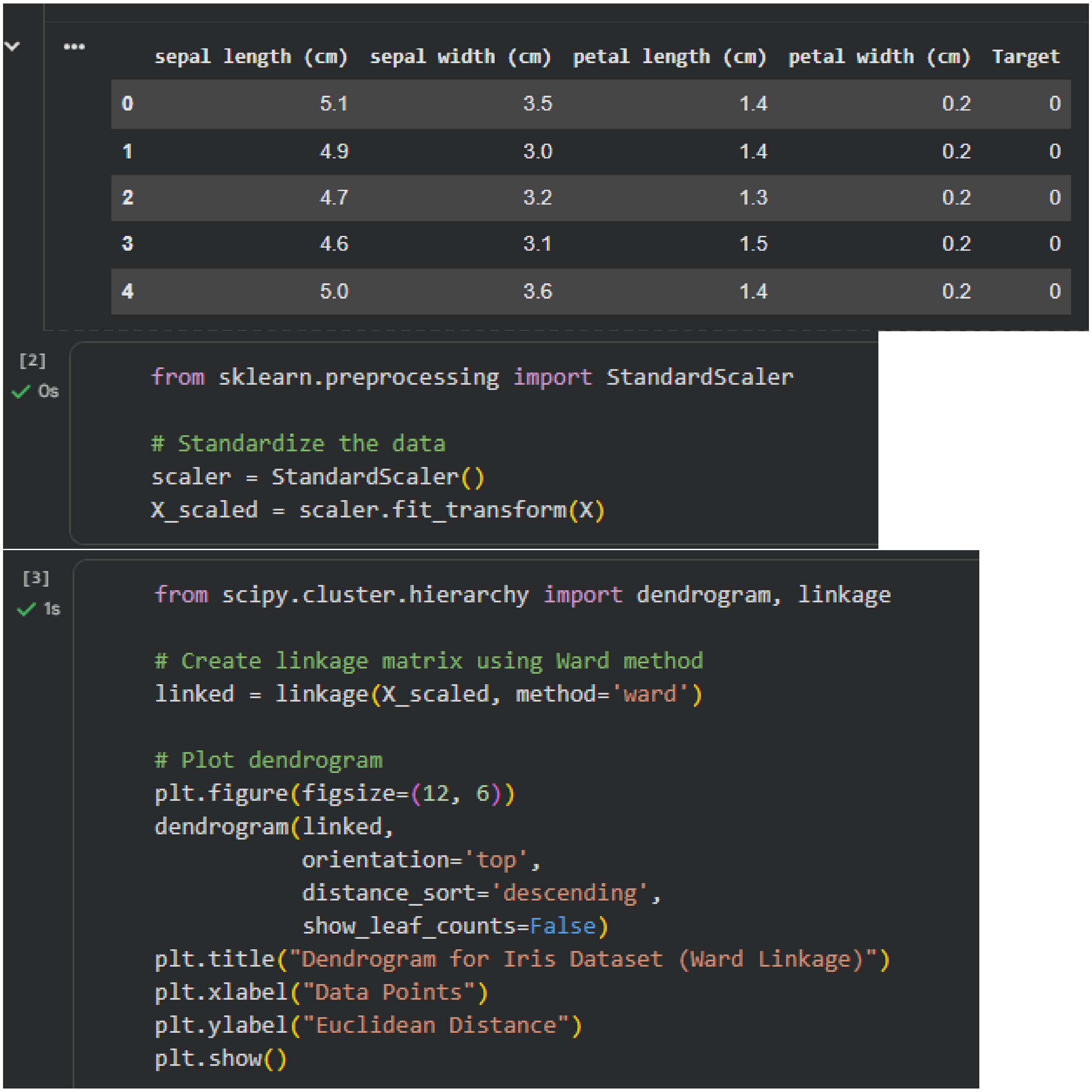
LAB NO:11

HEIRARCHICAL AGGLOMERATIVE CLUSTERING

Task 1: Load and Explore Data
We will use the **Iris dataset** from scikit-learn.
Task 2: Data Preprocessing
Standardize the data before clustering since the algorithm is distance-based.
Task 3: Create Dendrogram
Use scipy to visualize how data points merge at each step.
Task 4: Apply Agglomerative Clustering
Perform clustering using Agglomerative Clustering from scikit-learn.
Task 5: Visualize Clusters
Visualize the clusters using the first two features for simplicity.
Task 6: Evaluation (Optional)
Check clustering performance using the Adjusted Rand Index (ARI).



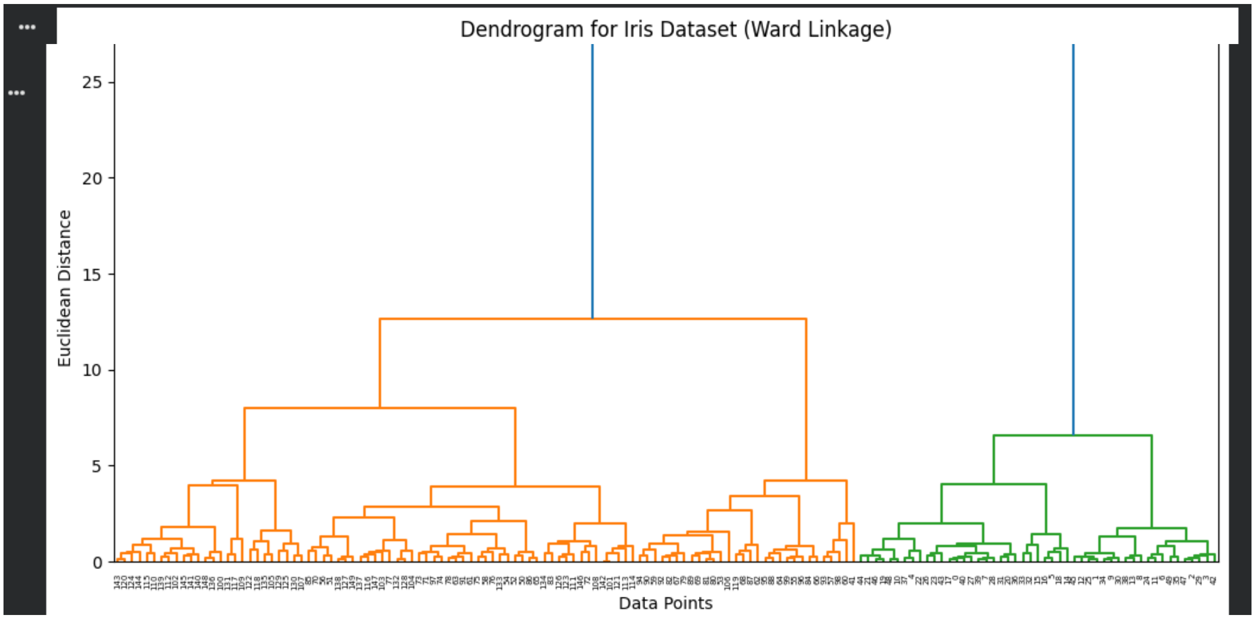
```
# Import required libraries
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt

from sklearn.datasets import load_iris

# Load Iris dataset
iris = load_iris()
X = iris.data
y = iris.target

# Convert to DataFrame for easy viewing
df = pd.DataFrame(X, columns=iris.feature_names)
df['Target'] = y

# Display first 5 rows
df.head()
```

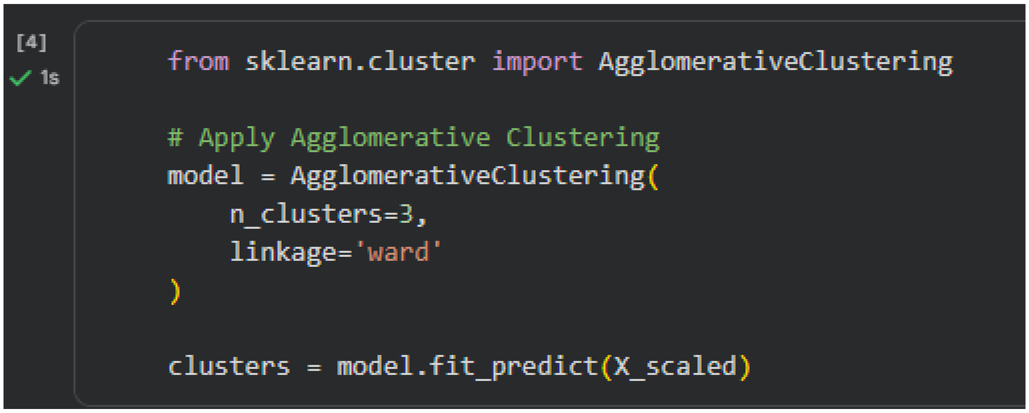


	sepal length (cm)	sepal width (cm)	petal length (cm)	petal width (cm)	Target
0	5.1	3.5	1.4	0.2	0
1	4.9	3.0	1.4	0.2	0
2	4.7	3.2	1.3	0.2	0
3	4.6	3.1	1.5	0.2	0
4	5.0	3.6	1.4	0.2	0

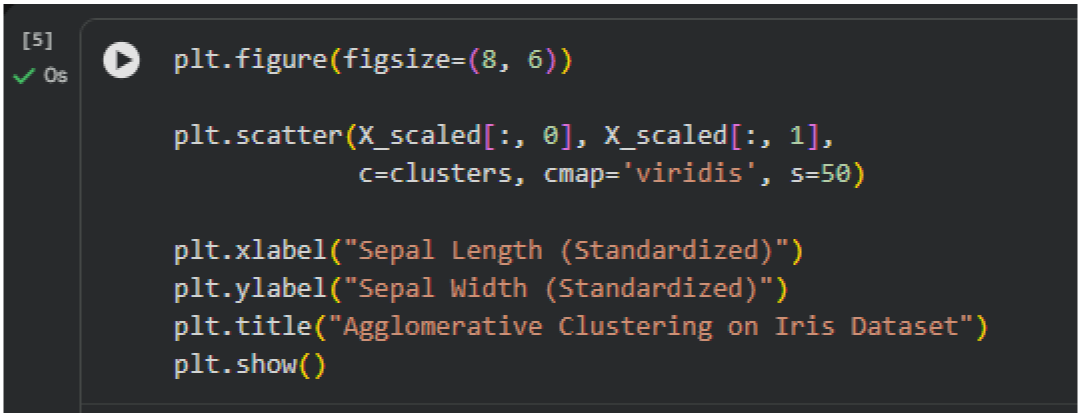
```python
from sklearn.preprocessing import StandardScaler

# Standardize the data
scaler = StandardScaler()
X_scaled = scaler.fit_transform(X)
```

```python
from scipy.cluster.hierarchy import dendrogram, linkage

# Create linkage matrix using Ward method
linked = linkage(X_scaled, method='ward')

# Plot dendrogram
plt.figure(figsize=(12, 6))
dendrogram(linked,
           orientation='top',
           distance_sort='descending',
           show_leaf_counts=False)
plt.title("Dendrogram for Iris Dataset (Ward Linkage)")
plt.xlabel("Data Points")
plt.ylabel("Euclidean Distance")
plt.show()
```

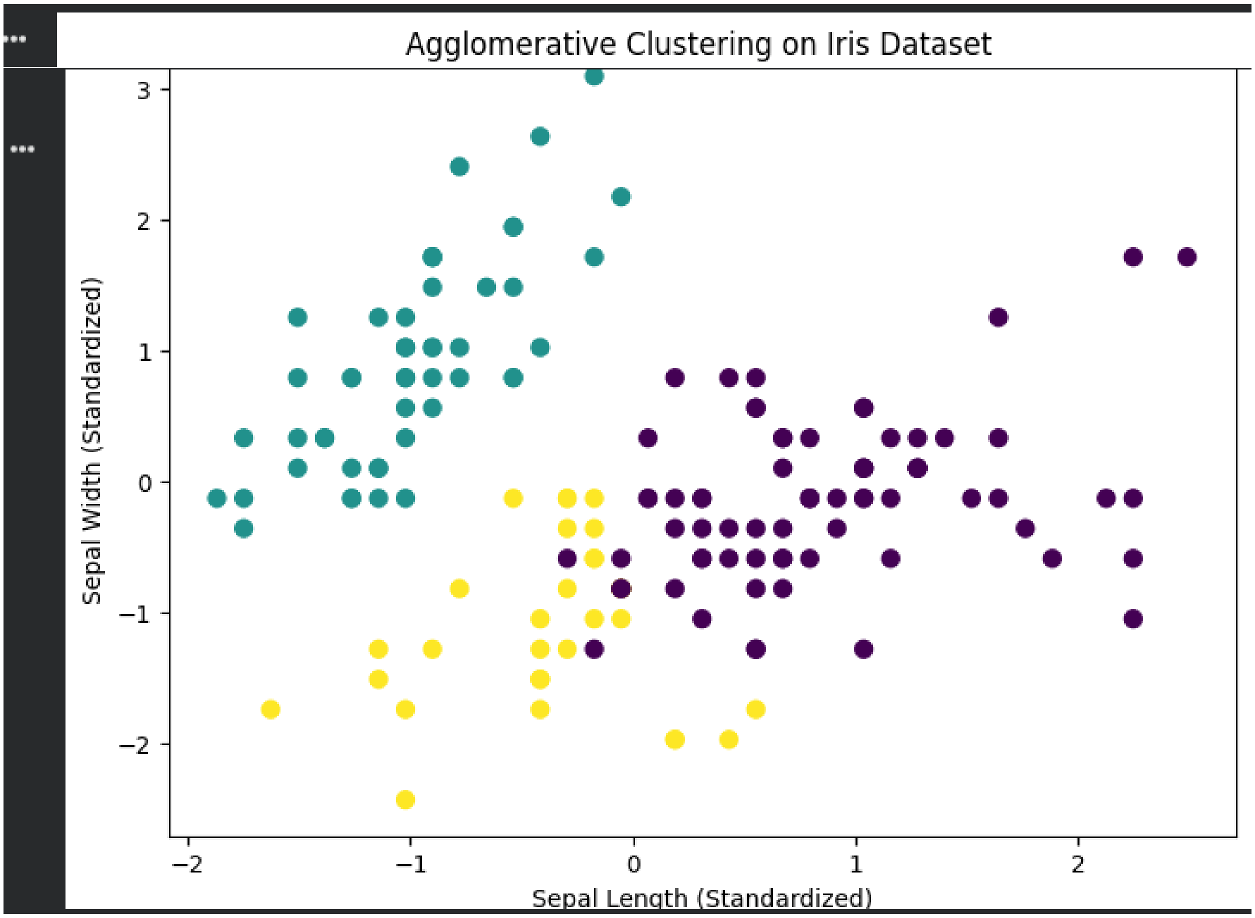


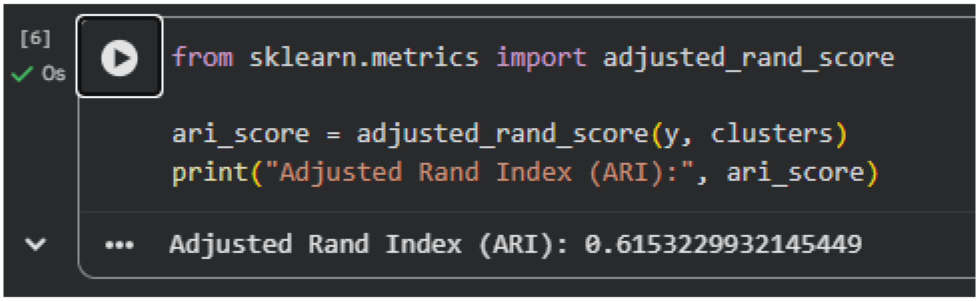


```
from sklearn.cluster import AgglomerativeClustering

# Apply Agglomerative Clustering
model = AgglomerativeClustering(
    n_clusters=3,
    linkage='ward'
)

clusters = model.fit_predict(X_scaled)
```



```
plt.figure(figsize=(8, 6))

plt.scatter(X_scaled[:, 0], X_scaled[:, 1],
            c=clusters, cmap='viridis', s=50)

plt.xlabel("Sepal Length (Standardized)")
plt.ylabel("Sepal Width (Standardized)")
plt.title("Agglomerative Clustering on Iris Dataset")
plt.show()
```





```
from sklearn.metrics import adjusted_rand_score

ari_score = adjusted_rand_score(y, clusters)
print("Adjusted Rand Index (ARI):", ari_score)
```

```
Adjusted Rand Index (ARI): 0.6153229932145449
```