ADP_Red\ml\5.clustering.py

```python
# %% 5. Machine Learning - Clustering Analysis
import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt

import scipy.stats as stats


# %% 1. 데이터 수집
df = pd.read_csv('../../ADP_Python/data/USArrests.csv')

print(df.shape)
print(df.info())

# Check binary variable
for i, var in enumerate(df.columns):
    print(i, var, len(df[var].unique()))

# Check data summary
print(df.describe())


# %% Hierarchical Clustering Analysis
from scipy.cluster.hierarchy import dendrogram, linkage, fcluster

# single linkage
model = linkage(df.iloc[:,1:], metric='euclidean', method='single')

# # ward linkage
model = linkage(df.iloc[:,1:], metric='euclidean', method='ward')

dendrogram(model,
           labels=list(df.iloc[:,0]),
           distance_sort='descending',
           color_threshold=250)
plt.show()

assignment = fcluster(model, 250, 'distance')
print(assignment)


# %% Non-Hierarchical Clustering Analysis (k-Means)
df = pd.read_csv('../../ADP_Python/data/iris.csv')
X = df.copy().drop('target', axis=1)
y = df.copy()['target']

# k-Means
from sklearn.cluster import KMeans

# Scree Plot
sse = []

for k in range(1, 11):
    kmeans = KMeans(n_clusters=k, n_init=10)
    kmeans.fit(X)
    sse.append(kmeans.inertia_)
```

```

plt.plot(sse, marker='o')
plt.show()

# Visualization
df_result = X
df_result['prediction'] = KMeans(n_clusters=3, n_init=10).fit(X).predict(X)

sns.pairplot(df_result,
             diag_kind='kde', hue='prediction')
plt.show()


# %% Non-Hierarchical Clustering Analysis (DBScan)
from sklearn.cluster import dbscan


# Gaussian Mixture
```