Introduction to Data Sciences – TD Clustering

Part I Examples

Example I.

```
import numpy as np

X = np.array([[1], [2], [3], [6], [7], [8], [13],[15], [17]])

from sklearn.cluster import KMeans
kmeans = KMeans(n_clusters=3, init=np.array([[1], [2], [3]]))
kmeans.fit(X)
kmeans.cluster_centers_
kmeans.labels_

from sklearn.cluster import AgglomerativeClustering
cluster = AgglomerativeClustering(n_clusters=3, linkage='single')
cluster.fit(X)
cluster.labels_
```

Example II.

```
import numpy as np
import matplotlib
import matplotlib.pyplot as plt

from scipy.cluster.hierarchy import dendrogram, linkage
Y = np.array([[1], [2], [4], [7], [8], [10], [15],[17], [21]])
linked = linkage(Y, 'single')
labelList = [[1], [2], [4], [7], [8], [10], [15],[17], [21]]

plt.figure(figsize=(10, 7))
dendrogram(linked,
        orientation='top',
        labels=labelList,
        distance_sort='descending',
        show_leaf_counts=True)
plt.show()
```

Example III.

```
import numpy as np
Z = np.array([[11], [21], [22], [23], [26], [27], [28], [33],[35], [37], [47]])

from sklearn.cluster import KMeans
kmeans = KMeans(n_clusters=3, init=np.array([[1], [2], [3]]))
kmeans.fit(Z)
kmeans.cluster_centers_
kmeans.labels_

from sklearn.cluster import AgglomerativeClustering
cluster = AgglomerativeClustering(n_clusters=3, linkage='single')
cluster.fit(Z)
cluster.labels_
```

Example IV.

```
import pandas as pd
import numpy as np
import sklearn.metrics as sm
import matplotlib.pyplot as plt
from sklearn.cluster import KMeans
from sklearn import datasets
from sklearn import metrics

iris = datasets.load_iris()
print(iris)
print(iris.data)
print(iris.feature_names)
print(iris.target)
print(iris.target_names)

x=pd.DataFrame(iris.data)
x.columns=['Sepal_Length','Sepal_width','Petal_Length','Petal_width']
y=pd.DataFrame(iris.target)
y.columns=['Targets']
model=KMeans(n_clusters=3)
model.fit(x)

colormap=np.array(['r','g','b'])
plt.scatter(x.Petal_Length, x.Petal_width,c=colormap[y.Targets],s=40)
plt.show()

plt.scatter(x.Petal_Length, x.Petal_width,c=colormap[model.labels_],s=40)
plt.show()
metrics.adjusted_rand_score(model.labels_, y.Targets)
```

Example V.

```
import pandas as pd
import numpy as np
from sklearn.cluster import KMeans
import matplotlib.pyplot as plt

review = pd.read_csv("google_review_ratings.csv")
review.isna().sum()
review.fillna(review.mean(), inplace=True)
review.isna().sum()
X=review.drop(['User', 'Unnamed: 25'], axis=1)

review_model=KMeans(n_clusters=3)
review_model.fit(X)

review_model.inertia_
```

Part II Exercises

Exercise 1

Generate and explain with help of the ***dendrogram*** how the four linkages methods (signle, average, and complete) work in the Agglomerative Clustering method.

Exercise 2

Comparing the datasets Y and Z in Example 2 and 3 and the result of these examples, what kinds of observation can you get from the two different clustering methods.

Exercise 3

In Example IV, a clustering method is used to classify data. The *metrics.adjusted_rand_score* can be used to evaluate the performance of such classifier. Compare the performances of the classifiers from the *Agglomerative Clustering* method with different linkage method (signle, average, and complete) as well as the performance of the *K-means* classifier.

Exercise 4.

The value of "*inertia_ " in ExampleV can be used to justify the goodness of a clustering method.* Compare the "*inertia_* " value for different number of clusters (*n_clusters*), from 3 to 10, for the data set "*google_review_ratings.csv*".

Final Report

In your report, you have to return

- python programs used for the exercise of Part II
- observations from Part II