QM-7063 Data Mining
Professor: Dr. Abdulrashid
Learning Practice 4 – Noah L. Schrick

**Imports and Initial Work**

```
# Learning Practice 4 for the University of Tulsa's QM-7063 Data Mining Course
# Cluster Analysis
# Professor: Dr. Abdulrashid, Spring 2023
# Noah L. Schrick - 1492657

%matplotlib inline

from pathlib import Path

import pandas as pd
from sklearn import preprocessing
from sklearn.metrics import pairwise
from scipy.cluster.hierarchy import dendrogram, linkage, fcluster
from sklearn.cluster import KMeans
import matplotlib.pylab as plt
import seaborn as sns
from pandas.plotting import parallel_coordinates


pd.options.mode.chained_assignment = None  # default='warn'
```

**15.1 University Rankings.**

The dataset on American College and University Rankings (available from www.dataminingbook.com) contains information on 1302 American
colleges and universities offering an undergraduate program. For each university, there are 17 measurements, including continuous measurements (such as tuition and graduation rate) and categorical measurements (such as location by state and whether it is a private or public school).

Note that many records are missing some measurements. Our first goal is to estimate these missing values from “similar” records. This will be done by clustering the complete records and then finding the closest cluster for each of the partial records. The missing values will be imputed from the information in that cluster.

**a.**

**Remove all records with missing measurements from the dataset.**

```
raw_university_df = pd.read_csv('Universities.csv')
university_df = raw_university_df.dropna() # 1302 rows -> 471 rows
```

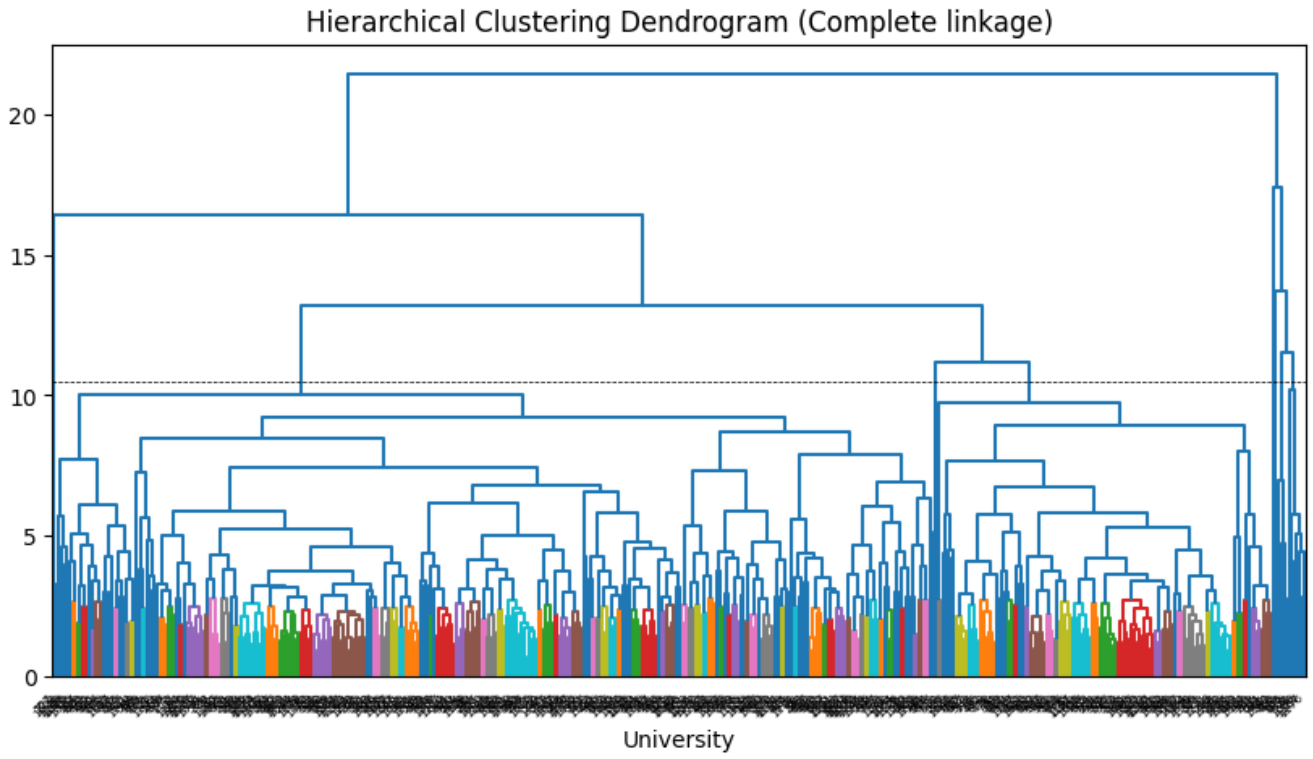
| | index | College Name | State | Public (1)/ Private (2) | # appli... | # appl.... | # new ... | % new ... | % new ... | # FT u... | # PT u... | in-stat... | out-of- |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 27 | 121 | Adams State College | CO | 1 | -0.40246... | -0.32110... | -0.23121... | -0.65006... | -0.57329... | -0.35143... | -0.36580... | -1.45385... | -2.064 |
| 164 | 524 | Adrian College | MI | 2 | -0.42210... | -0.38580... | -0.48568... | -0.32537... | -0.27808... | -0.54118... | -0.45184... | 0.334137... | 0.156 |
| 0 | 0 | Alaska Pacific University | AK | 2 | -0.72531... | -0.76563... | -0.79257... | -0.65006... | -0.57329... | -0.70974... | 0.046284... | -0.33472... | -0.69 |
| 83 | 271 | Albertson College | ID | 2 | -0.62858... | -0.63263... | -0.68008... | 0.540459... | 0.312344... | -0.61786... | -0.48936... | 0.741982... | 0.678 |
| 165 | 525 | Albion College | MI | 2 | -0.30647... | -0.13697... | -0.31858... | 0.486344... | 0.607557... | -0.42168... | -0.49518... | 0.808687... | 0.763 |
| 339 | 957 | Albright College | PA | 2 | -0.51785... | -0.48884... | -0.60472... | 0.107540... | 0.361546... | -0.55468... | -0.31792... | 1.121732... | 1.164 |
| 468 | 1284 | Alderson-Broaddus College | WV | 2 | -0.62981... | -0.62504... | -0.66479... | -0.37949... | -0.57329... | -0.59194... | -0.46542... | 0.192388... | -0.024 |
| 266 | 768 | Alfred University | NY | 2 | -0.34747... | -0.25479... | -0.33714... | 0.486344... | 0.951971... | -0.37114... | -0.44472... | 1.294477... | 1.385 |
| 340 | 958 | Allegheny College | PA | 2 | -0.12160... | -0.06508... | -0.32404... | 0.865149... | 1.050376... | -0.39748... | -0.48742... | 1.390910... | 1.508 |
| 378 | 1035 | Allentown Coll. of St. Francis de Sales | PA | 2 | -0.48324... | -0.51241... | -0.53591... | 0.540459... | 0.410748... | -0.52105... | -0.10315... | 0.051364... | -0.205 |
| 166 | 526 | Alma College | MI | 2 | -0.46163... | -0.39259... | -0.43216... | 0.865149... | 0.853567... | -0.48336... | -0.49777... | 0.573768... | 0.463 |
| 133 | 432 | Amherst College | MA | 2 | 0.283490... | -0.42774... | -0.39612... | 2.975630... | 1.985216... | -0.42189... | -0.51265... | 1.876699... | 2.130 |
| 100 | 325 | Anderson University | IN | 2 | -0.47415... | -0.46128... | -0.39066... | -0.48772... | -0.77010... | -0.37349... | -0.33410... | 0.125682... | -0.110 |
| 167 | 527 | Andrews University | MI | 2 | -0.49527... | -0.54276... | -0.50097... | -0.75829... | -1.60653... | -0.42339... | -0.30499... | 0.106831... | -0.134 |
| 413 | 1120 | Angelo State University | TX | 1 | 0.096411... | -0.02474... | 0.256975... | -0.21714... | -0.08127... | 0.134296... | 0.462250... | -1.55282... | -1.262 |
| 327 | 916 | Antioch University | OH | 2 | -0.59764... | -0.55994... | -0.57741... | -0.16303... | -0.57329... | -0.61058... | -0.50100... | 1.100161... | 1.136 |
| 210 | 645 | Appalachian State University | NC | 1 | 1.022724... | 1.038858... | 1.233348... | -0.43360... | 0.361546... | 1.365764... | 0.153672... | -1.57095... | -0.874 |
| 6 | 31 | Arkansas College (Lyon College) | AR | 2 | -0.59887... | -0.69054... | -0.67134... | 0.973378... | 0.902769... | -0.64955... | -0.39814... | -0.13823... | -0.447 |

(After normalization)

**b.**

**hierarchical clustering using complete linkage and Euclidean distance**

```
# Normalize
university_df_num = university_df.select_dtypes(include='number') # get numeric cols only
university_df_num = university_df_num.drop('Public (1)/ Private (2)', axis=1) # drop the discrete
column
university_df_num_norm = (university_df_num -
university_df_num.mean(numeric_only=True))/university_df_num.std(numeric_only=True) #
normalize
university_df.update(university_df_num_norm) # merge

university_dist = pairwise.pairwise_distances(university_df_num_norm,
                                              metric='euclidean')
pd.DataFrame(university_dist, columns=university_df.index, index=university_df.index).head(5)

uni_hclust = fcluster(linkage(university_df_num_norm, 'complete'), 6, criterion='maxclust')

Z = linkage(university_df_num_norm, method='complete')

fig = plt.figure(figsize=(10, 6))
fig.subplots_adjust(bottom=0.23)
plt.title('Hierarchical Clustering Dendrogram (Complete linkage)')
plt.xlabel('University')
dendrogram(Z, labels=university_df_num_norm.index, color_threshold=2.75)
plt.axhline(y=10.5, color='black', linewidth=0.5, linestyle='dashed')
plt.xticks(rotation=45, ha='right')
plt.show()

# reasonable number of clusters for describing the data:
# At distance of 10.5 (horizontal line in the dendrogram image) data can be reduced to 9 clusters
```




**c.**

**Compare the summary statistics for each cluster**

```
cutree = cluster.hierarchy.cut_tree(Z, n_clusters=[5, 10])
clust_stats = university_df_num_norm.agg(['mean', 'median'])
print(clust_stats)
```

| | index | # appli... | # appl.... | # new ... | % new ... | % new ... | # FT u... | # PT u... | in-stat... | out-of-... | room | board | add. fe... | estim. ... | estim. ... | % fac. ... | stud./f |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | mean | -1.88572... | 1.131437... | 3.630028... | 9.051499... | -3.77145... | -2.07430... | 3.771458... | 7.354343... | 2.130873... | -8.29720... | -2.75316... | 3.771458... | 1.131437... | 9.805791... | 2.413733... | 1.5085 |
| 1 | median | -0.36858... | -0.33388... | -0.36882... | -0.27126... | -0.08127... | -0.39576... | -0.32245... | 0.081816... | -0.11020... | -0.18383... | -0.07045... | -0.27826... | -0.29894... | -0.16417... | 0.167526... | -0.144 |

**d.**

**Use the categorical measurements to categorize**

```
state_table = tabulate(university_df[['State']], cutree)
pub_priv_table = tabulate(university_df[['Public (1)/ Private (2)']], cutree)
print(state_table)
```

**e.**

**Other external information**

```
# There are multiple external factors that can explain these clusters. Notably, that these clusters are
# built with only partial information. Since the pre-processing step removed all entries with NaNs, the
```

# total number of entries was reduced from 1302 to 471, which is a very large amount of missing data.
# Second, school funding priorities can affect some of the school data. Depending on how funding is allocated
# to sports, liberal arts, research, campus maintenance, events, etc, the underlying data may change.
# The socioeconomic factors involved with private vs public universities may also change the data.

**f.**

**Compute the Euclidean distance of this record from each of the clusters that you found above (using only the measurements that you have)**

```
tufts_df = raw_university_df.loc[raw_university_df['College Name'] == 'Tufts University']
tufts_df = tufts_df.drop(['# PT undergrad'], axis=1)
tufts_df_num = tufts_df.select_dtypes(include='number') # get numeric cols only

tufts_dist = pairwise.pairwise_distances(tufts_df_num, Y=university_df_num_norm,
metric='euclidean')

# Closest cluster:
print(raw_university_df.iloc[np.where(tufts_dist == tufts_dist.min())[1][0]]['College Name'])

# impute missing (from raw data – non-normalized)
tufts_df['# PT undergrad'] = clust_stats['# PT undergrad']['mean']
print(tufts_df['# PT undergrad'])
```

```
Colorado Christian University
475    3.771458e-17
```

**# Problem 15.4**

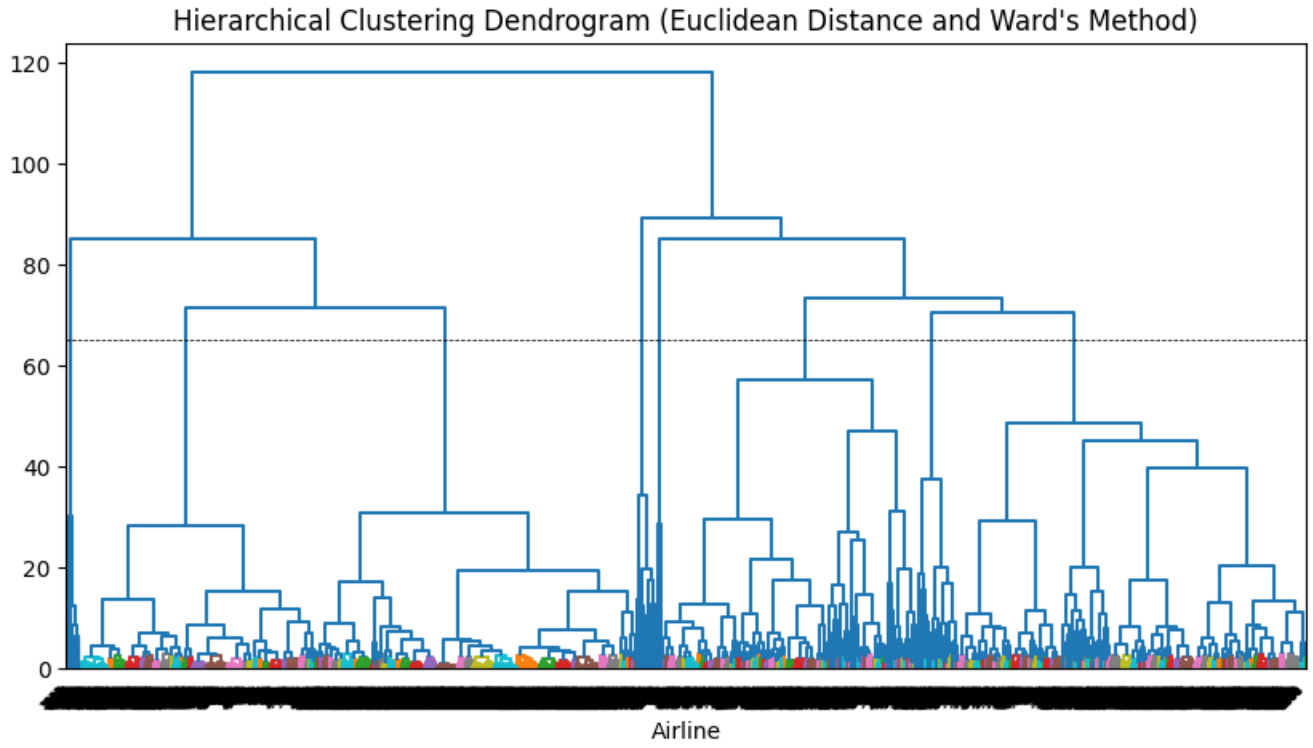
The file EastWestAirlinesCluster.csv contains information on 3999 passengers who belong to an airline's frequent flier program. For each passenger, the data include information on their mileage history and on different ways they accrued or spent miles in the last year. The goal is to try to identify clusters of passengers that have similar characteristics for the purpose of targeting different segments for different types of mileage offers.

**a.**

**Apply hierarchical clustering with Euclidean distance and Ward's method. Make sure to normalize the data first. How many clusters appear?**

```
raw_airlines_df = pd.read_csv('EastWestAirlinesCluster.csv')

# Normalize
airlines_df_norm = raw_airlines_df.apply(preprocessing.scale, axis=0)

# a: hclust with euclidean and ward's
Z = linkage(airlines_df_norm, method='ward', metric='euclidean')
fig = plt.figure(figsize=(10, 6))
fig.subplots_adjust(bottom=0.23)
```

```
plt.title("Hierarchical Clustering Dendrogram (Euclidean Distance and Ward's Method)")
plt.xlabel('Airline')
dendrogram(Z, labels=airlines_df_norm.index, color_threshold=2.75)
plt.axhline(y=65, color='black', linewidth=0.5, linestyle='dashed')
plt.xticks(rotation=45, ha='right')
plt.show()
```




**b.**

**What would happen if the data were not normalized?**

Data must be normalized to ensure that likeness is preserved through different data. It allows for scale to be enforced, and ensures that unwanted data is filtered out. It is an important step for optimizing the analysis approaches, specifically for distance measurements in this example.

**c.**

**Compare the cluster centroid to characterize the different clusters, and try to give each cluster a label.**

```
kmeans = KMeans(n_clusters=8, random_state=0).fit(airlines_df_norm)
# Number of clusters is based on the dendrogram from part a

centroids = pd.DataFrame(kmeans.cluster_centers_, columns=airlines_df_norm.columns)
#pd.set_option('precision', 3)
print(centroids)

# Cluster membership
```

```
memb = pd.Series(kmeans.labels_, index=airlines_df_norm.index)
for key, item in memb.groupby(memb):
    print(key, ': ', ', '.join(str(item.index[0])))
```

```
   Days_since_enroll    Award?
0          -0.955105 -0.461496
1           0.820704 -0.192639
2           0.208676  0.907008
3           0.239873  0.337527
...
4 :  1, 1, 4, 1
5 :  7, 0
6 :  8
7 :  4
```

```
0 :  1, 1, 2, 9
1 :  0
2 :  6, 5
3 :  1, 0, 6
4 :  1, 1, 4, 1
5 :  7, 0
6 :  8
7 :  4
```