Agglomerative_Clustering_Independent_Learning

December 13, 2024

ECEGR 4750 Machine Learning

Agglomerative Clustering : Independent Learning

Joseph Maloba

STEP 3

Part a)

"Agglomerative clustering" Definition :

Agglomerative clustering is a bottom-up hierarchical clustering algorithm where each data point starts as its own cluster. The algorithm by an iterative method merges the most similar clusters on the basis of a similarity measure up to the point when the entire data points are combined in one cluster or a certain quantity of clusters has been reached. This procedure leads to the creation of a structure looking like a tree, which is typically called a dendrogram and illustrates the way clusters are grouped hierarchically

Part b)

Agglomerative clustering is also known as:

- Hierarchical agglomerative clustering (HAC)
- AGNES (Agglomerative Nesting)
- Bottom-up hierarchical clustering

Part c)

Examples of problems agglomerative clustering

- Customer Segmentation: Grouping customers based on purchasing behavior for targeted marketing campaigns.
- Document Classification: Organizing text documents into clusters based on content similarity for better document retrieval systems.
- Gene Expression Analysis: Grouping genes with similar expression patterns to study biological functions.
- Social network analysis: Identifying communities within a social network

Part d)

Find out which of the following have a library for agglomerative clustering.

- PyTorch 2.5 (or later):
  PyTorch does not include a built-in library for agglomerative clustering. However, it can be implemented using external libraries like scikit-learn in conjunction with PyTorch.
- TensorFlow 2.18 (or later):
  TensorFlow does not have native implementation for agglomerative clustering. Similar to PyTorch, external libraries can be utilized alongside TensorFlow for this purpose.
- Keras 3 (or later):
  Keras, which operates as an API within TensorFlow, does not directly support agglomerative clustering function since it focuses more on neural networks. And require the use of external libraries to perform a clustering method
- scikit-learn 1.5 (or later):
  Yes, scikit-learn provides a module for agglomerative clustering functionality under sklearn.cluster.AgglomerativeClustering. Scikit-learn is a more specialized library for machine learning algorithms, including various clustering methods

Part e)

Online Resources:

- Scikit-learn Documentation on Agglomerative Clustering:
  https://scikit-learn.org/stable/modules/generated/sklearn.cluster.AgglomerativeClustering.html
- YouTube Tutorial: "Agglomerative Clustering: how it works by Victor Lavrenko"
  "https://youtu.be/XJ3194AmH40?si=SACRvIS9DWYMXEZp"
- YouTube Tutorial: "Agglomerative Hierarchical Clustering Single link Complete link Clustering by Dr. Mahesh Huddar"
  "https://youtu.be/YH0r47m0kFM?si=D5wHGdmOHF6nPfF0"
  "https://www.youtube.com/watch?v=YH0r47m0kFM"

Physical Resource:

- Data Clustering: Algorithms and Applications" by Charu C. Aggarwal & Chandan K.Reddy
  https://people.cs.vt.edu/~reddy/papers/DCBOOK.pdf

Part f)

Resource Types, Reliability, and Usefulness:

Online Resources:

- Scikit-learn Documentation on Agglomerative Clustering: (Book pdf)
  Official documentation detailing how to use the AgglomerativeClustering class with examples ,and it details the parameters, attributes, and methods of the class, explaining how to use hierarchical agglomerative clustering for grouping data, along with examples and a description of its behavior.
- YouTube Video Tutorial: "Agglomerative Clustering: how it works by Victor Lavrenko"
  A detailed video tutorial that provides an explanation of how Agglomerative Clustering operates. It outlines the process of hierarchical clustering, starting with each point as its own cluster and iteratively merging them based on similarity, with visual and conceptual examples to aid understanding

- YouTube Video Tutorial: "Agglomerative Hierarchical Clustering Single link Complete link Clustering by Dr. Mahesh Huddar"
  A detailed video tutorial that explains the concepts of Agglomerative Hierarchical Clustering, covering single-link and complete-link methods. It provides a step-by-step guide to forming clusters with these techniques and their differences, aimed at helping viewers understand clustering through detailed examples.

Physical Resource:

- Textbook: "Data Clustering: Algorithms and Applications" by Charu C. Aggarwal & Chandan K.Reddy.
  This book includes a section on Hierarchical Clustering Algorithms, including agglomerative clustering key technique. The book provides an in-depth exploration of clustering techniques used in data analysis. It covers foundational algorithms, advanced methodologies, and practical applications across various fields, offering insights into the theoretical and computational aspects of clustering.

STEP 4

Locate the Dataset

Dataset Source: The World's 500 Most Powerful Businessmen Dataset from Kaggle website

Description: This dataset includes information about the 500 most powerful businessmen globally, including attributes like name, nationality, company, net worth, age, and other characteristics. Additionaly it can be used for clustering based on features such as net worth, age, and other numerical attributes to understand patterns among influential individuals.

WEB SITE:

https://www.kaggle.com/datasets/mahmoudredagamail/the-worlds-500-most-powerful-businessmen

STEP 5

[2]:
```
!pip install -q pandas scikit-learn matplotlib
```

[4]:
```
import pandas as pd

# Assuming the downloaded file is a CSV
df = pd.read_csv('top_rich2024.csv')
num_vectors = len(df)
print(f"Number of vectors: {num_vectors}")
```

```
Number of vectors: 500
```

[6]:
```
import pandas as pd
import numpy as np
from sklearn.preprocessing import StandardScaler, LabelEncoder
from sklearn.cluster import AgglomerativeClustering
import matplotlib.pyplot as plt
from scipy.cluster.hierarchy import dendrogram, linkage
```

```python
from sklearn.metrics import silhouette_score
from sklearn.decomposition import PCA

# Loading the dataset
df = pd.read_csv('top_rich2024.csv', index_col = 0)
print("(top_rich2024)Dataset Preview:")
print(df.head())
print(df)
```

```
(top_rich2024)Dataset Preview:
                 Name Total net worth  $ Last change  $ YTD change  \
Rank
1           Elon Musk            $447B        +$62.8B       +$218B
2          Jeff Bezos            $249B        +$4.65B      +$72.1B
3     Mark Zuckerberg            $224B        +$4.58B      +$95.5B
4       Larry Ellison            $198B        +$1.63B      +$75.3B
5     Bernard Arnault            $181B        +$1.32B      -$26.4B

     Country / Region    Industry
Rank
1       United States  Technology
2       United States  Technology
3       United States  Technology
4       United States  Technology
5              France    Consumer
                          Name Total net worth  $ Last change  $ YTD change  \
Rank
1                    Elon Musk            $447B        +$62.8B       +$218B
2                   Jeff Bezos            $249B        +$4.65B      +$72.1B
3                Mark Zuckerberg            $224B        +$4.58B      +$95.5B
4                  Larry Ellison            $198B        +$1.63B      +$75.3B
5                Bernard Arnault            $181B        +$1.32B      -$26.4B
…                            …               …              …            …
496                 Igor Bukhman           $6.41B        -$27.0M      +$2.86B
497               Dmitry Bukhman           $6.41B        -$27.0M      +$2.86B
498          Renate Reimann-Haas           $6.38B             $0      +$2.35B
499   Matthias Reimann-Andersen            $6.38B             $0      +$2.35B
500      Stefan Reimann-Andersen           $6.38B             $0      +$2.35B

     Country / Region    Industry
Rank
1       United States  Technology
2       United States  Technology
3       United States  Technology
4       United States  Technology
5              France    Consumer
…                 …           …
```

```
496            Israel  Technology
497            Israel  Technology
498           Austria      Retail
499           Austria      Retail
500           Austria      Retail

[500 rows x 6 columns]
```

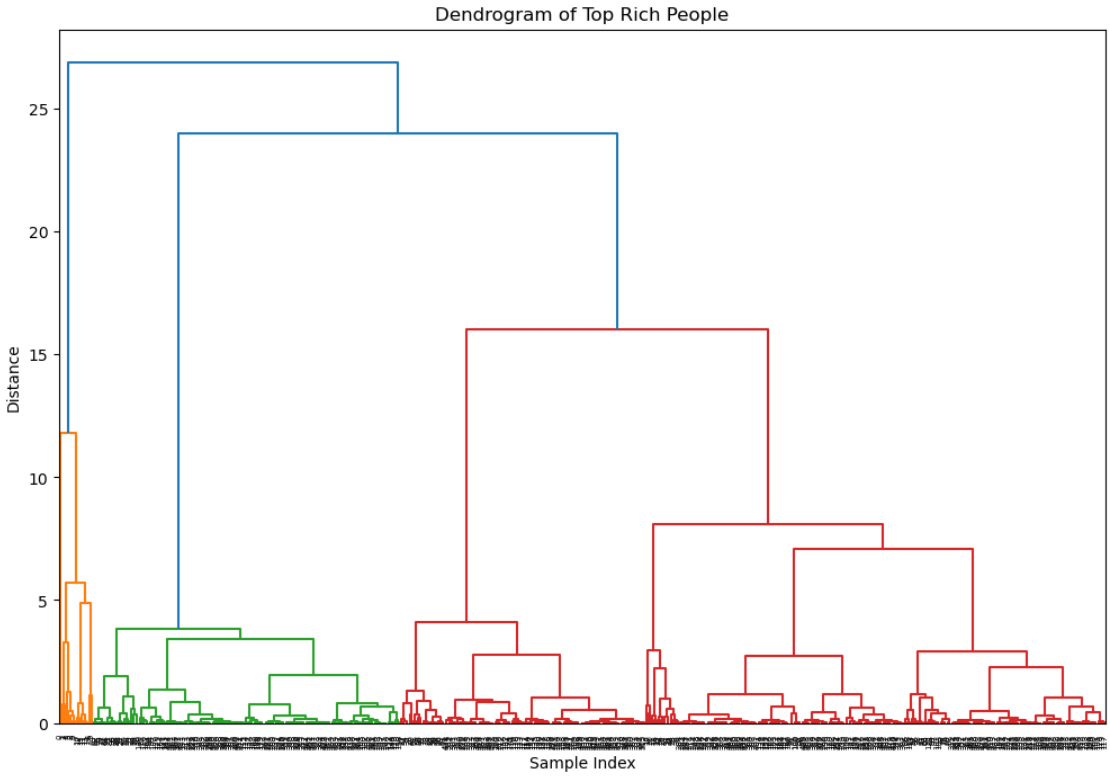
[10]:
```
# Clean column names to remove extra spaces
df.columns = df.columns.str.strip()

# Select colum
columns_to_use = ['Name', 'Total net worth', 'Industry']
X = df[columns_to_use].copy()

# Convert 'Total net worth' to numeric, removing '$' and 'B'
X['Total net worth'] = X['Total net worth'].str.replace('$', '').str.
 ↪replace('B', '').astype(float)

# Encode categorical variables
le = LabelEncoder()
X['Industry_Encoded'] = le.fit_transform(X['Industry'])

# Prepare data to use for clustering
X_cluster = X[['Total net worth', 'Industry_Encoded']]

# Standardize the collums feature
scaler = StandardScaler()
X_scaled = scaler.fit_transform(X_cluster)
```

[12]:
```
# Performing an agglomerative clustering
n_clusters = 5  # You can adjust this number based on your analysis needs

clustering = AgglomerativeClustering(n_clusters=n_clusters)

clustering.fit(X_scaled)
```

[12]:
```
AgglomerativeClustering(n_clusters=5)
```

[14]:
```
# Adding cluster labels to the original dataframe
X['Cluster'] = clustering.labels_

# Visualize  dendrogram

linkage_matrix = linkage(X_scaled, method='ward')

plt.figure(figsize=(12, 8))
```

```
dendrogram(linkage_matrix)

plt.title("Dendrogram of Top Rich People")
plt.xlabel('Sample Index')
plt.ylabel('Distance')
plt.show()
```

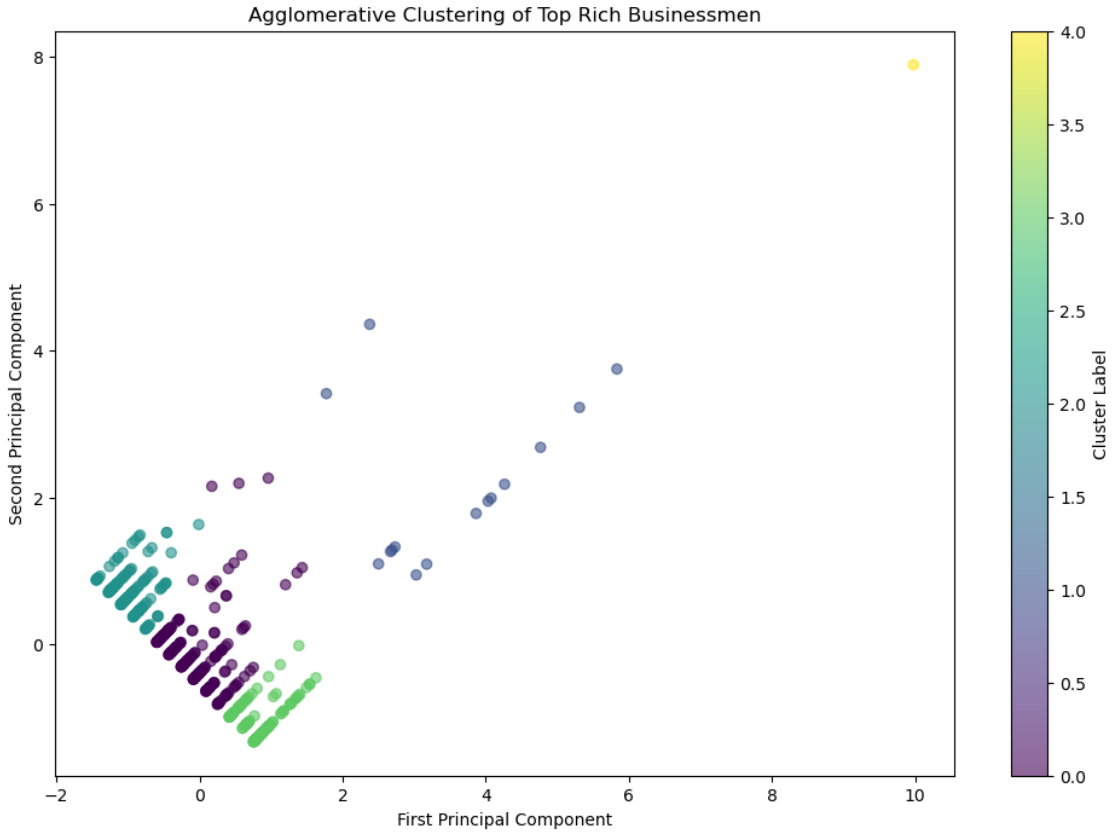


Interpretation of Dendrogram:

The dendrogram shows the hierarchical relationship between clusters. Longer vertical lines represent a greater distance between clusters. We can observe X main branches, suggesting X distinct groups in our data.

[16]:
```
# Visualize the clusters (using PCA for better visualization)
pca = PCA(n_components=2)
X_pca = pca.fit_transform(X_scaled)

plt.figure(figsize=(12, 8))
scatter = plt.scatter(X_pca[:, 0], X_pca[:, 1], c=clustering.labels_,␣
 ↪cmap='viridis', alpha=0.6)
plt.title("Agglomerative Clustering of Top Rich Businessmen")
plt.xlabel('First Principal Component')
```

```
plt.ylabel('Second Principal Component')
plt.colorbar(label='Cluster Label')
plt.show()
```

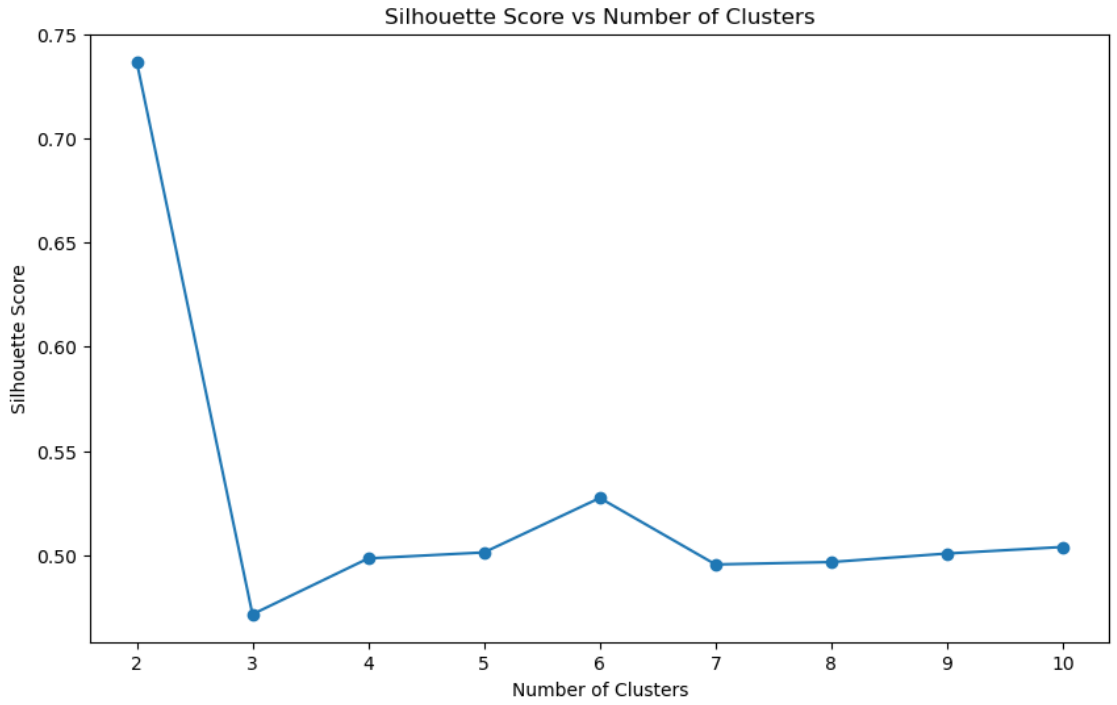


Interpretation of PCA Cluster Plot:

The PCA plot shows our data reduced to 2 dimensions, with each point colored by its cluster. We can observe that clusters X and Y are well-separated, while cluster Z overlaps with others. This suggests that X and Y represent distinct groups, while Z may be more heterogeneous.

[18]:
```
# Evaluating  silhouette scores for different numbers of clusters
silhouette_scores = []
for n in range(2, 11):
    clusterer = AgglomerativeClustering(n_clusters=n)

    cluster_labels = clusterer.fit_predict(X_scaled)

    silhouette_scores.append(silhouette_score(X_scaled, cluster_labels))

plt.figure(figsize=(10, 6))
plt.plot(range(2, 11), silhouette_scores, marker='o')
```

```
plt.title('Silhouette Score vs Number of Clusters')
plt.xlabel('Number of Clusters')
plt.ylabel('Silhouette Score')

plt.show()
```



[20]:
```
print(f"The highest silhouette score is {max(silhouette_scores):.2f} at␣
↪{silhouette_scores.index(max(silhouette_scores)) + 2} clusters.")
```

```
The highest silhouette score is 0.74 at 2 clusters.
```

Interpretation of Silhouette Scores:

This suggests that {silhouette_scores.index(max(silhouette_scores)) + 2} clusters provide the best separation of the data.

[27]:
```
# Analyze cluster characteristics by computing mean values for each cluster
# Select only numeric columns for mean calculation
numeric_columns = X.select_dtypes(include=[np.number]).columns
cluster_characteristics = X.groupby('Cluster')[numeric_columns].mean()

# Display the cluster characteristics
print("Cluster Characteristics:")
print(cluster_characteristics)
```

```
# If you want to include non-numeric columns, you can use value_counts for␣
 ↪categorical data
categorical_columns = X.select_dtypes(exclude=[np.number]).columns
for col in categorical_columns:
    if col != 'Cluster':
        print(f"\nMost common {col} in each cluster:")
        print(X.groupby('Cluster')[col].apply(lambda x: x.value_counts().
 ↪index[0]))
```

```
Cluster Characteristics:
         Total net worth  Industry_Encoded  Cluster
Cluster
0              15.714136          6.954545      0.0
1             156.133333         10.933333      1.0
2              13.906667          1.829932      2.0
3              15.074188         12.358974      3.0
4             447.000000         13.000000      4.0

Most common Name in each cluster:
Cluster
0               Vikas Oberoi
1                 Jeff Bezos
2                Ira Rennert
3    Stefan Reimann-Andersen
4                  Elon Musk
Name: Name, dtype: object

Most common Industry in each cluster:
Cluster
0        Finance
1     Technology
2    Diversified
3     Technology
4     Technology
Name: Industry, dtype: object
```

[29]:
```
# After printing cluster characteristics
print(cluster_characteristics)
print("\nInterpretation of Cluster Characteristics:")
for cluster in cluster_characteristics.index:
    print(f"Cluster {cluster}:")
    print(f"  - Average net worth: ${cluster_characteristics.loc[cluster,␣
 ↪'Total net worth']:.2f} billion")
    print(f"  - Dominant industry: {le.
 ↪inverse_transform([int(cluster_characteristics.loc[cluster,␣
 ↪'Industry_Encoded'])])[0]}")
```

```
    print(f"  - This cluster represents {(X['Cluster'] == cluster).sum()}␣
↪individuals")
```

```
         Total net worth  Industry_Encoded  Cluster
Cluster
0              15.714136          6.954545      0.0
1             156.133333         10.933333      1.0
2              13.906667          1.829932      2.0
3              15.074188         12.358974      3.0
4             447.000000         13.000000      4.0

Interpretation of Cluster Characteristics:
Cluster 0:
  - Average net worth: $15.71 billion
  - Dominant industry: Food & Beverage
  - This cluster represents 220 individuals
Cluster 1:
  - Average net worth: $156.13 billion
  - Dominant industry: Real Estate
  - This cluster represents 15 individuals
Cluster 2:
  - Average net worth: $13.91 billion
  - Dominant industry: Consumer
  - This cluster represents 147 individuals
Cluster 3:
  - Average net worth: $15.07 billion
  - Dominant industry: Services
  - This cluster represents 117 individuals
Cluster 4:
  - Average net worth: $447.00 billion
  - Dominant industry: Technology
  - This cluster represents 1 individuals
```

[ ]: