**ΑΡΙΣΤΟΤΕΛΕΙΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΘΕΣΣΑΛΟΝΙΚΗΣ**

**ΠΟΛΥΤΕΧΝΙΚΗ ΣΧΟΛΗ**

**ΤΜΗΜΑ ΜΗΧΑΝΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ**

**ΤΟΜΕΑΣ ΒΙΟΜΗΧΑΝΙΚΗ ΔΙΟΙΚΗΣΗΣ**

**Βαρνάς Γεώργιος**

**ΑΕΜ: 6132**

**4η εργασία στο μάθημα «Ανάλυση Δεδομένων» #MENG228 2023-2024**

Στην παρούσα εργασία ασχολούμαστε με το ίδιο dataset των προηγουμένων εργασιών του μαθήματος, από το οποίο αυτή τη φορά αντλούμε 3 διαφορετικές μεταβλητές, οι οποίες αφορούν στοιχεία εμβολιασμού. Συγκεκριμένα, χρησιμοποιούμε τις μεταβλητές Hepatitis B, Polio, Diptheria, τις οποίες επιχειρούμε να ομαδοποιήσουμε με τις μεθόδους K-means clustering και hierarchical clustering.

Οι αλγόριθμοι K-means clustering και hierarchical clustering που αξιοποιούνται σε προβλήματα ομαδοποίησης(«συσταδοποίησης»)(clustering) αποτελούν μεθόδους μη επιβλεπόμενης μηχανικής μάθησης(unsupervised machine-learning). Αυτό σημαίνει ότι ο αλγόριθμος εκπαιδεύεται σε δεδομένα χωρίς ετικέτες(unlabeled data), όπου δεν υπάρχει προκαθορισμένο output, αλλά μόνο input. Επιδιώκεται ουσιαστικά η εξερεύνηση των δεδομένων με σκοπό την ανάδειξη κρυφών μοτίβων και δομών που πιθανόν να τα χαρακτηρίζουν

Ο αλγόριθμος K-means clustering χωρίζει τα δεδομένα σε K ξεχωριστές συστάδες(clusters), στοχεύοντας στην κατά το μέγιστο δυνατό ελαχιστοποίηση της διακύμανσης των δεδομένων εντός της κάθε συστάδας, δηλαδή στην μέγιστη ομοιότητά τους και την διαφοροποίησή τους από τα δεδομένα των άλλων συστάδων. Ο αλγόριθμος αρχικά θέτει τα κέντρα των K clusters σε τυχαία σημεία στο χώρο και αναθέτει κάθε data point σε εκείνο το cluster, από το κέντρο του οποίου απέχει το λιγότερο(συνήθως με κριτήριο την ευκλείδεια απόσταση). Έτσι έχουν σχηματιστεί τα K clusters, για τα οποία υπολογίζουμε εκ νέου τα κέντρα ως τον μέσο όρο των data points που αποτελούν το κάθε cluster. Η διαδικασία επαναλαμβάνεται έως ότου η ανανέωση των κέντρων των clusters να είναι μηδαμινή από επανάληψη σε επανάληψη ή ενδεχομένως τίθεται όριο επαναλήψεων για τον αλγόριθμο.

Ο αλγόριθμος hierarchical clustering(ιεραρχική ομαδοποίηση) δημιουργεί μια ιεραρχία από συστάδες(clusters), τα οποία παρουσιάζονται με δομή δέντρου, το γνωστό δενδρόγραμμα. Οι δύο βασικότεροι τύποι της ιεραρχικής ομαδοποίησης είναι η ομαδοποίηση συσσωματώματος(agglomerative) και η διαιρετική ομαδοποίηση(divisive). Στην ομαδοποίηση συσσωματώματος κάθε data point αρχικά αποτελεί ένα cluster από μόνο του και σε κάθε επανάληψη ενώνονται τα κοντινότερα ανά μεταξύ τους clusters. Στην διαιρετική ομαδοποίηση αρχικά θέτονται όλα τα data points ως ένα cluster και με διαδοχικές επαναλήψεις διαχωρίζονται σε μικρότερα clusters. Και στις δύο εκδοχές της ιεραρχικής ομαδοποίησης χρησιμοποιείται και πάλι ως μέτρο συνήθως η ευκλείδεια απόσταση για τον υπολογισμό της απόστασης ανάμεσα στα data points που ομαδοποιούνται. Επιπλέον, μια παράμετρος που είναι σημαντική για την ιεραρχική ομαδοποίηση είναι η σύνδεση(linkage), δηλαδή ο τρόπος που μετριούνται οι αποστάσεις μεταξύ των clusters. Οι δύο συνηθέστεροι τρόποι υπολογισμού της απόστασης, -οι οποίοι χρησιμοποιούνται και στην παρούσα εργασία-, είναι με single linkage και με complete linkage. Η single linkage υπολογίζει την απόσταση ως την κοντινότερη απόσταση μεταξύ 2 σημείων, ενός από κάθε cluster. Η complete linkage υπολογίζει την απόσταση ως την μεγαλύτερη απόσταση μεταξύ 2 σημείων, ενός από κάθε cluster.

Στην εργασία αρχικά εφαρμόζουμε ομαδοποίηση για K=3 και Κ=4 clusters. Τα clusters που προκύπτουν έχουν την εξής μορφή:

Εικόνα που περιέχει στιγμιότυπο οθόνης, κείμενο, πολυχρωμία, διάγραμμα

Περιγραφή που δημιουργήθηκε αυτόματα

Βλέπουμε στα 2 παραπάνω σχήματα τα αποτελέσματα της ομαδοποίησης σε k=3 και k=4 clusters αντίστοιχα. Στο διάγραμμα που δημιουργήσαμε φαίνονται οι συστάδες που προκύπτουν βάση των τιμών των 2 από τις 3 μεταβλητές(Polio και Hepatitis B). Πιθανόν η οπτικοποίησή να μην ιδανική για την πλήρη κατανόηση της μορφής των clusters, η οποία θα απαιτούσε ένα 3D διάγραμμα με τις τιμές και των 3 μεταβλητών τις οποίες επεξεργαζόμαστε.

Υπολογίζουμε τα κέντρα των 3 και 4 clusters αντίστοιχα, καθώς και τις «ετικέτες»(labels) δηλαδή το cluster στο οποίο ανήκει κάθε data point που χρησιμοποιούμε.

Εικόνα που περιέχει κείμενο, στιγμιότυπο οθόνης, γραμματοσειρά

Περιγραφή που δημιουργήθηκε αυτόματα

Στο επόμενο ερώτημα επιχειρούμε με την μέθοδο elbow να επιλέξουμε το καταλληλότερο πλήθος clusters για την ομαδοποίηση των δεδομένων μας. Το μέτρο που χρησιμοποιούμε για την επιλογή αυτή είναι το WCSS(within-cluster sum of squares), το οποίο αντικατοπτρίζει πόσο συμπαγή είναι τα clusters. Υπολογίζεται ως το άθροισμα των τετραγώνων των αποστάσεων των data points ενός cluster από το κέντρο του cluster αυτού. Χαμηλό WCSS σημαίνει ότι τα data points εντός του cluster είναι κοντά μεταξύ τους και άρα ορθώς έχουν ομαδοποιηθεί κατά τον τρόπο αυτό. Προφανώς, όσο αυξάνουμε τον αριθμό k των clusters, αναμένουμε την μείωση του WCSS, διότι γίνεται όλο και σαφέστερη η ομαδοποίηση-«διαχωρισμός» των data points στα διαθέσιμα clusters. Ωστόσο, κατά τη διαρκή αύξηση του αριθμού των k clusters, από ένα σημείο και μετά η αύξηση του k μειώνει μηδαμινά το WCSS, υποδεικνύοντας ότι επιπλέον clusters αποτελούν πλεονασμό. Το σημείο αυτό είναι το «elbow point». Παρουσιάζεται το διάγραμμα WCSS-αριθμός K των clusters για τα δεδομένα του προβλήματός μας:

Εικόνα που περιέχει γραμμή, κείμενο, γράφημα, διάγραμμα

Περιγραφή που δημιουργήθηκε αυτόματα

Η ανάλυση έγινε για εύρος τιμών από 1 ως 20 K clusters. Φαίνεται ότι μέχρι Κ=5 clusters το WCSS μειώνεται σημαντικά, ενώ από το σημείο αυτό και έπειτα δεν έχουμε αξιοσημείωτη βελτίωση(μείωση) του WCSS για το ολοένα αυξανόμενο K.

Στο επόμενο ερώτημα εκτελούμε την ιεραρχική ομαδοποίηση αρχικά με complete linkage και έπειτα με single linkage. Λαμβάνουμε τα εξής δενδροδιαγράμματα:

Εικόνα που περιέχει κείμενο, διάγραμμα, στιγμιότυπο οθόνης, γραμμή

Περιγραφή που δημιουργήθηκε αυτόματα

Εικόνα που περιέχει κείμενο, στιγμιότυπο οθόνης, ορθογώνιο παραλληλόγραμμο, γραμμή

Περιγραφή που δημιουργήθηκε αυτόματα

Για να εκλέξουμε τον ιδανικότερο αριθμό clusters βάση ενός δενδροδιαγράμματος, εντοπίζουμε την μέγιστη κάθετη απόσταση μεταξύ οριζόντιων γραμμών, όπου ουσιαστικά διαλέγουμε το ύψος στο οποίο θα «κόψουμε» το δενδροδιάγραμμα. Ο αριθμός των clusters αντιστοιχεί στον αριθμό των κάθετων γραμμών που τέμνονται από την οριζόντια αυτή γραμμή.

Python code:

import numpy as np

import pandas as pd

import matplotlib.pyplot as plt

import seaborn as sns

from sklearn.model\_selection import train\_test\_split

from sklearn.metrics import accuracy\_score

from sklearn.metrics import classification\_report

from sklearn.metrics import confusion\_matrix

from sklearn.cluster import KMeans

import scipy.cluster.hierarchy as sch

from scipy.cluster.\_hierarchy import linkage

from mpl\_toolkits.mplot3d import Axes3D

# 4h ergasia

# Read our data, create the dataframe

df1 = pd.read\_excel(r"c:\Users\Γιωργος\Desktop\MECHANICAL ENGINEERING\4ο ΕΤΟΣ\ΑΝΑΛΥΣΗ ΔΕΔΟΜΕΝΩΝ\1η εργασια\Data Analysis\_2024 1st Case\_Data.xlsx")

print(df1.head())

print(df1.describe())

print(df1.info())

# Check for NaN values

print(df1.isnull().values.any()) #prepei na vgazei False, an den exw NaN values se olo to dataframe

# Remove years 2007,2013 from the dataframe

i = df1[(df1['Year']==2013) | (df1['Year']==2007)].index

print(i)

df1.drop(index=i,inplace=True)

print(df1.info()) #gia epalitheysh, se sygrish me to prohgoumeno info

# Convert Life expectancy into a dummy variable

avg\_life\_expectancy = df1['Life expectancy '].mean()

print(avg\_life\_expectancy)

life\_expectancy\_dummies = pd.get\_dummies(df1['Life expectancy '] > avg\_life\_expectancy ,

                                        prefix='Life expectancy ', drop\_first=True)

life\_expectancy\_dummies = life\_expectancy\_dummies.astype(int)

print(life\_expectancy\_dummies)

print(life\_expectancy\_dummies.value\_counts()) #epalitheysh

df2 = pd.concat([df1, life\_expectancy\_dummies], axis=1)

df2.drop(['Life expectancy ','Year','Country', 'Status'], axis=1, inplace=True)

# Remove everything except Hepatitis B, Polio, Diptheria

df2.drop(['Adult Mortality', 'Alcohol', 'percentage expenditure',

                'Measles ', ' BMI ', 'under-five deaths ', 'Total expenditure',

                 ' HIV/AIDS', 'GDP', 'Population', ' thinness 5-9 years',

                 'Income composition of resources', 'Schooling', 'Life expectancy \_True'], axis=1, inplace=True)

print(df2.info())

k3 = KMeans(n\_clusters=3, random\_state=101)

k3.fit(df2)

print('Cluster centers for k=3:', k3.cluster\_centers\_)

print('Cluster labels for k=3:', k3.labels\_)

k4 = KMeans(n\_clusters=4, random\_state=101)

k4.fit(df2)

print('Cluster centers for k=4:', k4.cluster\_centers\_)

print('Cluster labels for k=4:', k4.labels\_)

fig, axes = plt.subplots(1, 2, figsize=(11, 6))

axes[0].scatter(df2['Hepatitis B'], df2['Polio'], c=k3.labels\_, cmap='viridis')

axes[0].set\_title('K-means Clustering (k=3)')

axes[0].set\_xlabel('Hepatitis B')

axes[0].set\_ylabel('Polio')

axes[1].scatter(df2['Hepatitis B'], df2['Polio'], c=k4.labels\_, cmap='viridis')

axes[1].set\_title('K-means Clustering (k=4)')

axes[1].set\_xlabel('Hepatitis B')

axes[1].set\_ylabel('Polio')

plt.tight\_layout()

plt.show()

# Elbow method to find the best number of K clusters

wcss = []

k\_values = range(1,21)

for k in k\_values:

    kmeans = KMeans(n\_clusters=k, random\_state=101)

    kmeans.fit(df2)

    wcss.append(kmeans.inertia\_)

plt.figure(figsize=(10, 6))

plt.plot(k\_values, wcss, 'bo-', markersize=8)

plt.xlabel('Number of clusters (k)')

plt.ylabel('Within-cluster Sum of Squares (WCSS)')

plt.title('Elbow Method for Determining Optimal k')

plt.grid(True)

plt.show()

# Hierarchical clustering with complete linkage

plt.figure(figsize=(10, 7))

plt.title('Dendrogram - Complete Linkage')

dg\_complete = sch.dendrogram(sch.linkage(df2, method='complete'))

plt.xlabel('Samples')

plt.ylabel('Euclidean Distance')

plt.show()

# Hierarchical clustering with single linkage

plt.figure(figsize=(10, 7))

plt.title('Dendrogram - Single Linkage')

dg\_single = sch.dendrogram(sch.linkage(df2, method='single'))

plt.xlabel('Samples')

plt.ylabel('Euclidean Distance')

plt.show()