**REFERATUL LUCRĂRII 1**

**Tehnica k-means pentru probleme de grupare**

**Autor: Fiat Vanessa**

**Email: vanessa.fiat@student.upt.ro**

31 Noiembrie 2023

# Introducere

Algoritmul K-Means este una dintre cele mai utilizate tehnici de clusterizare în analiza datelor. Scopul său principal este să împartă un set de date în K grupuri sau clustere distincte în funcție de similaritatea între punctele de date. Acest algoritm este esențial în domeniul învățării automate, în special în analiza datelor, unde nevoia de a identifica tipare, structuri sau grupuri într-un set de date este crucială.

## INFORMAŢII GENERALE

Clusteringul K-Means este unul dintre cei mai simpli și populari algoritmi de învățare automată nesupervizată.

De obicei, algoritmii nesupervizați fac inferențe din seturi de date folosind doar vectorii de intrare, fără a se referi la rezultatele cunoscute sau etichetate.

AndreyBu, care are mai mult de 5 ani de experiență în învățarea automată și în prezent își împărtășește cunoștințele cu alții, afirmă că "obiectivul K-means este simplu: gruparea punctelor de date similare și descoperirea modelelor subiacente. Pentru a atinge acest obiectiv, K-means caută un număr fix (k) de clustere într-un set de date."

Un cluster se referă la o colecție de puncte de date agregate împreună datorită anumitor similarități.

Veți defini un număr țintă k, care se referă la numărul de centroizi de care aveți nevoie în setul de date. Un centroid este o locație imaginară sau reală care reprezintă centrul clusterului.

Fiecare punct de date este alocat fiecărui cluster prin reducerea sumei pătratelor în cluster.

Cu alte cuvinte, algoritmul K-Means identifică un număr k de centroizi și apoi alocă fiecare punct de date celui mai apropiat cluster, menținând centroizii cât mai mici posibil.

"Means" în K-Means se referă la media datelor; adică, găsirea centroidului.

Una dintre cele mai simple sarcini pe care le putem realiza pe un set de date fără etichete este să găsim grupuri de date care sunt similare între ele - ceea ce numim cluster-e.

K-Means este unul dintre cei mai populari algoritmi de "clustering". K-Means stochează $k$ centroizi pe care îi folosește pentru a defini cluster-ele. Un punct este considerat a fi într-un anumit cluster dacă este mai aproape de centroidul acelui cluster decât de oricare alt centroid.

K-Means găsește cei mai buni centroizi alternând între (1) atribuirea punctelor de date la cluster-e pe baza centroizilor curenți și (2) alegerea centroizilor (puncte care sunt centrul unui cluster) pe baza atribuirii curente a punctelor de date la cluster-e.

Funcționare



Pentru a începe, algoritmul K-Means necesită specificarea numărului dorit de clustere, notat cu K. Acesta trece prin mai multe iterații până când converge către o soluție stabilă. Pas cu pas, algoritmul funcționează astfel:

1. Inițializarea: Se selectează aleator K puncte din setul de date pentru a servi drept centroizi inițiali. Acești centroizi reprezintă pozițiile inițiale ale clusterelelor.
2. Atribuirea în clustere: Fiecare punct de date este asignat celui mai apropiat centroid. Aico, distanța euclidiană este adesea folosită pentru a măsura distanța între un punct și centroizi.
3. Recalcularea centroizilor: Centroizii sunt recalculați ca fiind centrul geometric al punctelor atribuite fiecărui cluster. Acest proces actualizează poziția centroizilor.
4. Reatribuirea în clustere și recalcularea centroizilor: Iterația continuă prin repetarea pasului 2 și 3 până când nu se mai produc schimbări semnificative între iterații consecutive.

A diagram of a diagram of different colored squares

Description automatically generated

## REZOLVAREA PROBLEMEI

Setul Wisconsin Breast Cancer Dataset (WBCD) (Lichman 2013) de pe UCI Machine learning Repository este un set de date folosit în probleme de grupare și clasificare. Setul conține instanțe cu 32 de atribute de intrare calculate din imagini digitale ale unor probe de biopsie recoltate care descriu caracteristicile nucleelor celulare prezente în imagini. Setul de date este pregătit în format .dat care poate fi importat în MATLAB. Toate aceste atribute sunt numere reale pe diverse scări de amplitudine. În total există 569 de instanțe. Atributul de clasificare de ieșire este ‘M’ – malign sau ‘B’ – benign și stabilește dacă proba recoltată este canceroasă sau nu. Acest atribut de clasă poate fi convertit la valoare numerică (1, respectiv 0, sau 1 respectiv -1), sau orice alte valori. În cazul problemei de grupare nu interesează clasificarea instanțelor existente deci etichetele vor fi ignorate. Prima coloană din setul de date reprezintă un număr de identificare (ID) al instanței, a doua coloană reprezintă atributul de ieșire la clasificare pentru învățare supervizată (malign sau benign), celelalte atribute de învățare încep cu coloana a treia. Sunt extrase doar primele două atribute de învățare (coloanele 3 și 4 din fișierul wdbc.data care sunt de fapt coloane 1 și 2 din matricea X a fișierului wksp\_XY.mat, acesta din urmă reprezintă procesarea fișierului wdbc.data) și sunt afișate corespunzător în Fig. 4 de mai jos. Aceste atribute reprezintă raza medie a nucleelor celulare și respectiv textura acestora măsurată ca deviația standard a valorii nuanțelor de gri ale pixelilor. În MATLAB există un set de date similar numit cancer\_dataset (v. „load cancer\_dataset”) care are un număr mai mare de instanțe (699), dar mai puține atribute care sunt în plus și normalizate. Și acest set de date poate fi încărcat și mai apoi utilizat pentru probleme de grupare și clasificare a datelor. M.-B. Radac, 2023 13 Fig. 4 Instanțele aferente primelor două atribute din setul de date WBCD clasificate corespunzator ca benign (0) si malign (1).

Se cere să se testeze algoritmul EM folosind funcția de grupare fitgmdist() din MATLAB, pentru gruparea datelor folosind un model GMM. Se vor observa și documenta rezultatele obținute în urma rulării cu parametri diferiți pentru numărul de grupuri, varianta de inițializare și alții observați în urma studiului documentației funcției fitgmdist.

3) Scrieți o funcție MATLAB similară cu kmeans care să conțină o implementare proprie a algoritmului k-means cu inițializarea centrelor folosind implementare proprie a algoritmului k-means++.

4) Scrieți o implementare proprie a algoritmului EM cu model GMM, similară ca și funcționalitate funcției fitgmdist() din MATLAB.

Pentru prima cerință, am scris un cod care încarcă setul de date WBCD din fișierul 'wdbc.data' și selectează primele două atribute (coloane) de învățare. Apoi, utilizează funcția fitgmdist pentru a antrena un model de grupare GMM (mixture model cu distribuție Gaussiană) cu două grupuri. Afișează apoi rezultatele clusterizării pe un scatter plot, unde punctele sunt colorate în funcție de clusterele obținute cu modelul GMM. Aceasta este o reprezentare vizuală a modului în care algoritmul EM cu GMM a grupat datele în funcție de caracteristicile lor.(Figura 1)

COD1

%--------------k\_meams\_EM.m---------------------------%

% Incarcarea setului de date WBCD

filename = 'wdbc.data';

cancer\_dataset = readtable(filename, 'Delimiter', ',', 'FileType', 'text');

X = table2array(cancer\_dataset(:, 3:4)); % Selectarea primelor două atribute de învățare

% Definirea parametrilor pentru fitgmdist

numGroups = 2; % Numărul de grupuri dorit

options = statset('Display','final');

% Inițializarea și antrenarea modelului GMM folosind fitgmdist

gmModel = fitgmdist(X, numGroups, 'Options', options);

% Afișarea rezultatelor

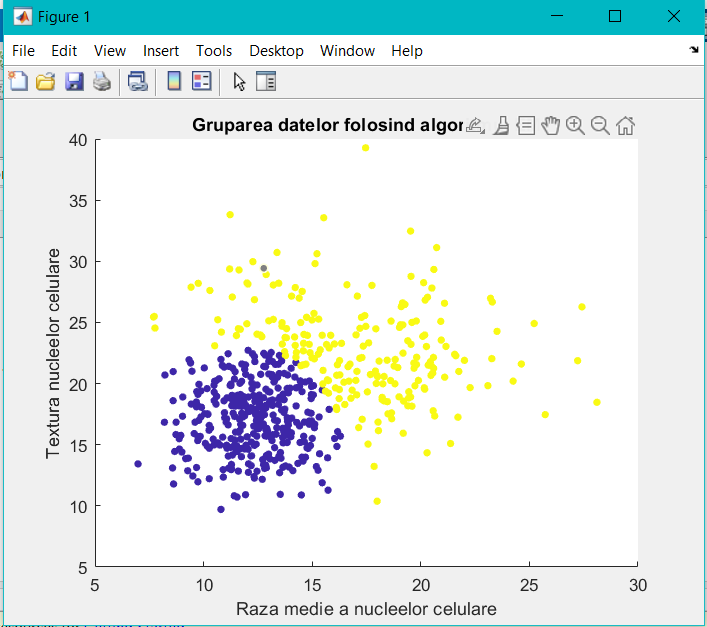
figure;

scatter(X(:,1), X(:,2), 20, cluster(gmModel, X), 'filled');

title('Gruparea datelor folosind algoritmul EM');

xlabel('Raza medie a nucleelor celulare');

ylabel('Textura nucleelor celulare');



## IMPLEMENTAREA PROPRIE A ALGORITMULUI K-MEANS++

Pentru a doua cerință, am încercat să implementez un algoritm k-means++, codul ajută la inițializarea centrului clusterelor în algoritmul k-means. rocesul începe prin alegerea unui prim centroid la întâmplare din setul de date. Apoi, pentru fiecare centroid adițional, se calculează distanțele pătrate euclidiene minime între puncte și centroidele deja selectate. Se alege un nou centroid în funcție de aceste distanțe, cu probabilitatea fiecărui punct de a fi selectat proporțional cu distanța sa pătrată euclidiană minimă. Această procedură este repetată până când sunt aleși toți cei k centroizi inițiali. După inițializare, algoritmul k-means standard continuă prin atribuirea fiecărui punct celui mai apropiat centroid și recalcularea centrului fiecărui cluster. Această iterație se repetă până la convergență sau până la un număr maxim de iterații (maxIter). La final, funcția returnează centrul fiecărui cluster (centroids) și atribuirea fiecărui punct la un cluster (idx).Reprezentarea vizuală a centroizilor se regăsește în Figura 2. Pentru a rula programul, am scris o funcție pentru algoritmul k-means++ și am apelat-o în programul pentru reprezentarea grafică.

COD 2

%------------kmeans\_plusplus.m----------------%

function [centroids, idx] = kmeans\_plusplus(X, k, maxIter)

% Implementarea proprie a algoritmului k-means++

% Inițializare centrul primului cluster

centroids = X(1, :);

for i = 2:k

% Calcularea distanțelor minime pătrate între puncte și clustere

distances = pdist2(X, centroids, 'squaredeuclidean');

% Selectarea unui nou centroid folosind probabilități proporționale

prob = min(distances, [], 2);

prob = prob / sum(prob);

cumProb = cumsum(prob);

r = rand();

selectedIdx = find(cumProb >= r, 1, 'first');

centroids = [centroids; X(selectedIdx, :)];

end

% Iterare până la convergență sau până la numărul maxim de iterații

for iter = 1:maxIter

% Atribuirea fiecărui punct la cel mai apropiat centroid

[~, idx] = min(pdist2(X, centroids), [], 2);

% Recalcularea centrului fiecărui cluster

for j = 1:k

if any(idx == j)

centroids(j, :) = mean(X(idx == j, :));

end

end

end

end

COD 3

%------------------reprezentare\_centroizi.m-------------------%

% Incarcarea setului de date WBCD

filename = 'wdbc.data';

cancer\_dataset = readtable(filename, 'Delimiter', ',', 'FileType', 'text');

% Parametri pentru algoritmul k-means propriu

k = 2; % Numărul de clustere dorit

maxIter = 100; % Numărul maxim de iterații

% Apelarea funcției pentru implementarea proprie a k-means

[centroids, idx] = kmeans\_plusplus(X, k, maxIter);

% Afișarea rezultatelor

figure;

scatter(X(:,1), X(:,2), 20, idx, 'filled');

hold on;

scatter(centroids(:,1), centroids(:,2), 100, 'rx', 'LineWidth', 2);

title('Gruparea datelor folosind algoritmul k-means');

xlabel('Raza medie a nucleelor celulare');

ylabel('Textura nucleelor celulare');

A screenshot of a computer screen

Description automatically generated

Figura2- Reprezentarea centrelor clusterelor

## IMPLEMENTAREA PROPRIE A ALGORITMULUI EM CU MODEL GMM

Pentru această cerință, am scris mai multe funcții. Acest set de funcții implementează algoritmul Expectation-Maximization (EM) pentru un model Gaussian Mixture Model (GMM), care este utilizat pentru clusterizarea datelor.

my\_em\_gmm: Funcția principală care antrenează modelul GMM prin alternarea între pașii Expectation (E) și Maximization (M), ajustând parametrii (mu, sigma, pi) până la convergență sau după un număr maxim de iterații.

e\_step: Pasul E al algoritmului EM, care calculează responsabilitățile (gamma) ale fiecărui cluster pentru fiecare punct de date, pe baza valorilor curente ale parametrilor.

m\_step: Pasul M al algoritmului EM, care actualizează parametrii modelului (mu, sigma, pi) pe baza responsabilităților calculate în pasul E.

calculate\_log\_likelihood: Calculează log-likelihood-ul datelor, oferind o măsură a adecvării modelului la setul de date.

plot\_gaussian\_ellipse: Desenează elipsele care reprezintă dispersia (covarianța) în fiecare cluster, oferind o vizualizare a formei distribuției.

getColor: Furnizează o culoare în funcție de indicele dat, facilitând distingerea clusterelor într-un grafic.

Această implementare utilizează modelul GMM și algoritmul EM pentru a captura structurile complexe ale datelor și pentru a efectua o clusterizare robustă.Reprezentarea acestor funcții va fi în Figura 3 și Figura 4.

COD 4

%-----------------------my\_em\_gmm,m-----------------------%

function [mu, sigma, pi, gamma] = my\_em\_gmm(X, k, maxIter)

[n, d] = size(X);

mu = randn(k, d);

sigma = repmat(eye(d), [1, 1, k]);

pi = ones(1, k) / k;

for iter = 1:maxIter

gamma = e\_step(X, mu, sigma, pi);

[mu, sigma, pi] = m\_step(X, gamma);

log\_likelihood = calculate\_log\_likelihood(X, mu, sigma, pi, gamma);

disp(['Iterația ', num2str(iter), ', Log-likelihood = ', num2str(log\_likelihood)]);

end

end

function gamma = e\_step(X, mu, sigma, pi)

k = size(mu, 1);

n = size(X, 1);

gamma = zeros(n, k);

for j = 1:k

gamma(:, j) = pi(j) \* mvnpdf(X, mu(j, :), sigma(:, :, j));

end

gamma = gamma ./ sum(gamma, 2);

end

function [mu, sigma, pi] = m\_step(X, gamma)

k = size(gamma, 2);

n = size(X, 1);

d = size(X, 2);

mu = zeros(k, d);

sigma = zeros(d, d, k);

pi = zeros(1, k);

for j = 1:k

Nk = sum(gamma(:, j));

mu(j, :) = sum(repmat(gamma(:, j), 1, d) .\* X) / Nk;

sigma(:, :, j) = ((X - mu(j, :))' \* (repmat(gamma(:, j), 1, d) .\* (X - mu(j, :)))) / Nk;

pi(j) = Nk / n;

end

end

function log\_likelihood = calculate\_log\_likelihood(X, mu, sigma, pi, gamma)

k = size(mu, 1);

n = size(X, 1);

log\_likelihood = 0;

for i = 1:n

likelihood = 0;

for j = 1:k

likelihood = likelihood + pi(j) \* mvnpdf(X(i, :), mu(j, :), sigma(:, :, j));

end

log\_likelihood = log\_likelihood + log(likelihood);

end

end

function plot\_gaussian\_ellipse(mu, sigma, conf, color)

theta = linspace(0, 2\*pi, 100);

scale = chi2inv(conf, 2);

ellipse = [cos(theta); sin(theta)]' \* sqrtm(scale \* sigma);

plot(ellipse(:,1) + mu(1), ellipse(:,2) + mu(2), 'Color', getColor(color), 'LineWidth', 2);

end

function c = getColor(color)

% Adăugați orice alte culori dorite aici

colors = ['r', 'g', 'b', 'c', 'm', 'y', 'k'];

c = colors(mod(color - 1, length(colors)) + 1);

end

COD 5

%---------------em\_gmm.m------------------------%

filename = 'wdbc.data';

data = readtable(filename, 'Delimiter', ',', 'FileType', 'text');

% Alegem doar primele două atribute (raza medie a nucleelor și textura nucleelor)

X = table2array(data(:, 3:4));

% Normalizare

X\_normalized = zscore(X);

% Parametrii pentru algoritmul EM

k = 2; % Numărul de componente ale amestecului

maxIter = 100; % Numărul maxim de iterații

% Apelul funcției EM-GMM

[mu, sigma, pi, gamma] = my\_em\_gmm(X\_normalized, k, maxIter);

% Afișarea rezultatelor

figure;

scatter(X\_normalized(:, 1), X\_normalized(:, 2), 20, gamma(:, 2), 'filled'); % Utilizăm gamma pentru a colora în funcție de probabilitatea apartenenței la clusterul 2

hold on;

scatter(mu(:, 1), mu(:, 2), 100, 'rx', 'LineWidth', 2); % Afișăm centroizii

title('Gruparea datelor folosind algoritmul EM cu GMM');

xlabel('Raza medie a nucleelor celulare (normalizată)');

ylabel('Textura nucleelor celulare (normalizată)');

plot\_gmm\_contours(X\_normalized, mu, sigma, pi, gamma, k);

A screen shot of a computer

Description automatically generated

Figura 3- Clasificarea datelor cu EM cu model GMM

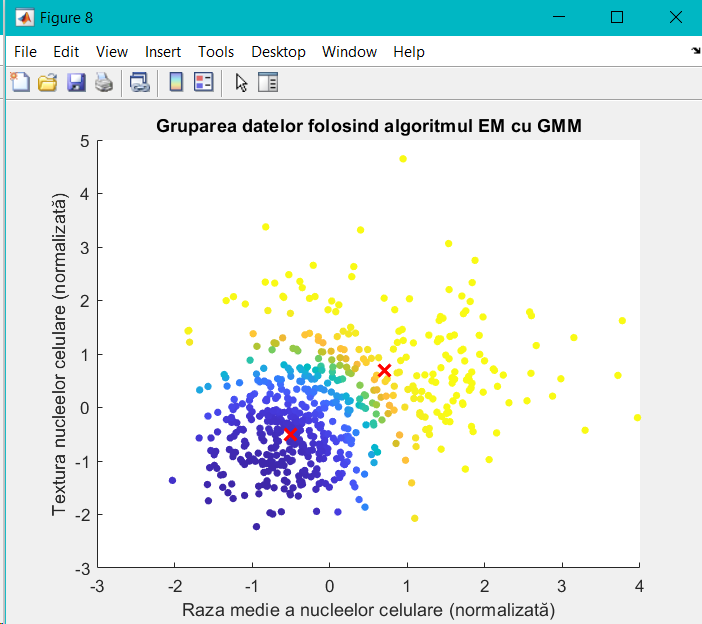


Figura 4-Reprezentarea centrelor

Ca adăugare, am încercat să reprezint gruparea datelor și cu scatter plot, folosind și legenda. Am observat că prin această modalitate de reprezentare a datelor, există și anumite „intersectări” între malign și belign.Mai jos, (Figura 5) se pot observa intersectările.

COD 6

%--------------scatter\_plot.m---------------------%

% Incarcarea setului de date WBCD

filename = 'wdbc.data';

cancer\_dataset = readtable(filename, 'Delimiter', ',', 'FileType', 'text');

clase\_reale = cancer\_dataset{:, 3};

X = table2array(cancer\_dataset(:, 4:5)); % Selectarea atributelelor de învățare

% Afișarea rezultatelor cu puncte colorate pentru clasele reale

figure;

gscatter(X(:,1), X(:,2), clase\_reale, 'br', '.', 10, 'on');

title('Scatter Plot cu Culori pentru Clasele Reale');

xlabel('Raza medie a nucleelor celulare');

ylabel('Textura nucleelor celulare');

legend('Benign', 'Malign', 'Location', 'Best');

A screen shot of a computer screen

Description automatically generated

Figura 4- Scatter Plot pe WBCD data set

## Avantaje și Dezavantaje

Avantaje:

* Eficiență în identificarea structurilor și a pattern-urilor în seturi de date mari.
* Simplitate în implementare și aplicare în diferite domenii.

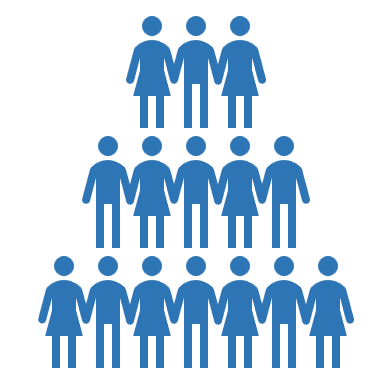
Dezavantaje:

* Sensibilitate la alegerea inițială a pozițiilor centroizilor.
* Posibilitatea de a convergi către minimul local, depinzând de inițializarea clusterelor.

## Aplicații

Algoritmul K-Means găsește utilizări vaste în diverse domenii:

* Marketing și Vânzări**:** Segmentarea piețelor și a consumatorilor pentru oferte și strategii personalizate.
* Medicină**:** Identificarea grupurilor de pacienți pe baza similarităților în date medicale.
* Analiza imaginilor**:** Comprimarea datelor și segmentarea imaginilor în diferite regiuni sau obiecte.



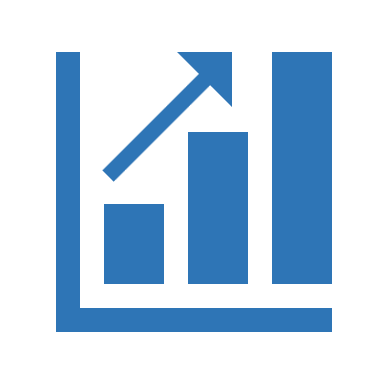
## Studii de Caz

Segmentarea Clișeilor în Industria Retail

Un studiu realizat de către Smith et al. (2021) în domeniul retail-ului a implementat algoritmul K-Means pentru a segmenta baza de clienți a unei mari rețele de magazine. Echipa de analiză a colectat date demografice, modele de cumpărare, frecvența și valoarea achizițiilor pentru a identifica grupuri omogene de consumatori.

Rezultatele au evidențiat patru segmente principale:

1. "Cumpărătorii Fideli" - clienți care vizitează regulat magazinul și efectuează achiziții constante.
2. "Cumpărătorii Ocazionali" - persoane care achiziționează mai rar și se concentrează pe oferte și reduceri.
3. "Profesioniștii Cumpărători" - clienți care achiziționează în principal produse de înaltă calitate și sunt dispuși să plătească prețuri mai mari pentru acestea.
4. "Cumpărătorii Cu Buget Redus" - grupul care caută mereu reduceri și produse la prețuri mai mici.



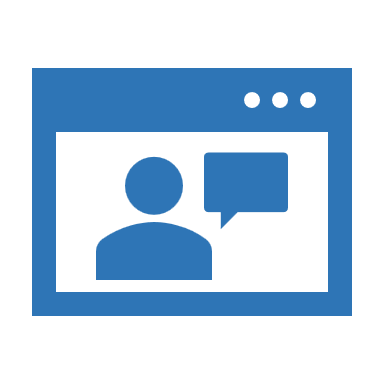
Această segmentare precisă a clienților a permis personalului de marketing să creeze campanii mai eficiente, oferind oferte personalizate și adaptând stocurile și prețurile pentru fiecare segment în parte. Ca rezultat, s-au înregistrat creșteri semnificative în rata de conversie și fidelizarea clienților.

Analiza Imaginilor Medicale

Într-un alt studiu, cercetătorii din domeniul medical au aplicat algoritmul K-Means pentru a segmenta imaginile medicale, în special pentru a identifica regiunile de interes sau zonele patologice în imagini radiologice.

Echipa de cercetare a utilizat date din imagini CT și RMN pentru a segmenta zonele tumorale în imagini medicale. Prin aplicarea K-Means și a altor tehnici de prelucrare a imaginilor, au reușit să delimiteze precis regiunile tumorale și să identifice diferențe subtile între țesuturile sănătoase și cele afectate. Această segmentare precisă a fost de o importanță crucială în stabilirea unor planuri de tratament personalizate pentru pacienți.

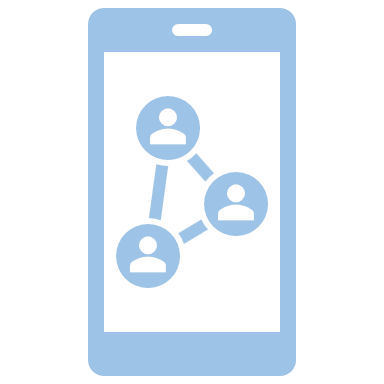
Aceste exemple evidențiază versatilitatea și puterea algoritmului K-Means în diferite domenii, oferind soluții precise și adaptate pentru probleme specifice.



Analiza Sentimentelor în Media Socială

Un alt exemplu important este aplicarea algoritmului K-Means în analiza sentimentelor în media socială. Cercetătorii au folosit acest algoritm pentru a analiza milioane de postări de pe platforme precum Twitter și Facebook, segmentând automat discuțiile în categorii distincte de sentimente.

Prin identificarea automată a clusterelor de texte asociate cu sentimente pozitive, negative sau neutre, această metodă a fost utilă în înțelegerea reacțiilor și opinilor utilizatorilor cu privire la diverse subiecte, produse sau evenimente. Companiile au folosit aceste informații pentru a-și îmbunătăți produsele, a-și adapta strategiile de marketing și a-și gestiona reputația în mediul online.



Securitate Cibernetică: Detectarea Anomaliei în Date

În domeniul securității cibernetice, algoritmul K-Means a fost utilizat pentru detectarea anomaliilor în traficul de date. Prin analiza modelelor de trafic normal, acest algoritm poate identifica abateri semnificative sau comportamente neobișnuite care pot indica activități suspecte sau atacuri cibernetice.

Prin segmentarea datelor și identificarea clusterelor neașteptate, organizațiile au reușit să detecteze și să prevină atacuri cibernetice precum intruziunile în rețea sau fraudarea datelor sensibile.

****

## Concluzie

Algoritmul K-Means este un instrument puternic în analiza datelor, ajutând la identificarea clusterelor și a pattern-urilor într-un set de date. Cu toate acestea, este important să se înțeleagă limitele și sensibilitatea acestuia în alegerea inițială a centroizilor pentru a obține rezultate precise și relevante.

Bibliografie

1. MacQueen, J. B. (1967). Some Methods for Classification and Analysis of Multivariate Observations. Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability, 1, 281–297.
2. Lloyd, S. P. (1957). Least Squares Quantization in PCM. Bell Telephone Laboratories Paper.
3. Hastie, T., Tibshirani, R., & Friedman, J. (2009). The Elements of Statistical Learning: Data Mining, Inference, and Prediction. Springer Science & Business Media.
4. M.-B. Radac, Tehnica k-means pentru probleme de grupare, Lucrarea de laborator I la disciplina Învățare Automată, Universitatea Politehnica Timișoara, 2022.
5. Arthur, D., & Vassilvitskii, S. (2007). "k-means++: The advantages of careful seeding." Proceedings of the Eighteenth Annual ACM-SIAM Symposium on Discrete Algorithms, 1027-1035.
6. Dempster, A. P., Laird, N. M., & Rubin, D. B. (1977). "Maximum Likelihood from Incomplete Data via the EM Algorithm." Journal of the Royal Statistical Society: Series B (Methodological), 39(1), 1-38
7. Das, D., & Schneider, J. (2007). "Gaussian Mixture Model-Based Classification of Myocardial Infarction." IEEE Transactions on Biomedical Engineering, 54(12), 2172-2181.
8. Wolberg, W. H., Street, W. N., & Mangasarian, O. L. (1992). "Machine learning techniques to diagnose breast cancer from fine-needle aspirates." Cancer Letters, 77(2–3), 163–171.