Projecte d'Algorísmia: Avaluació experimental de l'algorisme k-means i variants

Sergi Padrés Masdemont, Jan Antón Villanueva, Alexis David Monroy Arroyo i Gemma Bachs Prim

Curs 2023-2024, Quadrimestre de Primavera

March 27, 2024

Contents

1 Introducció

L'algoritme K-means és àmpliament reconegut com una eina crucial en l'anàlisi de dades i l'aprenentatge automàtic. El seu objectiu fonamental és agrupar un conjunt de dades en K grups o clústers, basant-se en les similituds entre els diferents components de les dades. Aquest algoritme opera de manera iterativa, inicialitzant centroides aleatoris per a cada clúster i ajustant-los progressivament per tal de minimitzar la suma dels quadrats de les distàncies entre els punts de les dades i els centroides dels clústers. A través d'aquest procés iteratiu d'assignació i ajustament, l'algoritme K-means cerca trobar una agrupació òptima del conjunt de dades, proporcionant una representació estructurada i significativa del mateix.

El projecte es dividirà en tres parts distintes. En primer lloc, procedirem a implementar dues variants de l'algorisme k-means. Després d'analitzar diverses opcions, hem optat per les següents dues versions: d'una banda, l'algorisme de Lloyd, i de l'altra, el K-means++.

En segon lloc, realitzarem una anàlisi de la qualitat dels algorismes mitjançant diversos mètodes d'avaluació de clusterings, així com un mètode per optimitzar el nombre de clústers, denominat k. Hem decidit utilitzar l'índex de Calinski–Harabasz per avaluar la qualitat interna, el Rand Index per a la qualitat externa i l'Elbow Method per determinar el nombre ideal de clústers K per a ambdós algorismes.

Finalment, conclourà el projecte amb una anàlisi experimental dels algorismes treballats, utilitzant un conjunt de datasets proporcionats pels docents de l'assignatura.

2 Algorisme de k-means

En aquest treball, implementarem dues versions de l'algorisme k-means i posteriorment en farem el seu estudi.

2.1 Algorisme de Lloyd

L'algorisme de Lloyd és la versió més bàsica de k-means i és un algorisme d'aprenentatge no supervisat. Classifica punts en K clústers en funció de la seva similitud i es basa en calcular distàncies euclidianes entre els punts.

Representarem així els K clústers de punts: $C_1...C_k$, que hauran de complir les següents dues condicions:

- $\bigcup_i C_i = \{1, ..., n\}$: tot punt del dataset és assignat a un clúster
- $C_i \cap C_i' = \emptyset$: els clústers no se solapen.

És a dir, si x_i és assignat al clúster j llavors $i \in C_j$.

Tenint en compte que $\parallel xi - xj \parallel_2$ és la distància euclidiana entre els vectors xi i xj, l'objectiu de l'algorisme de Lloyd és triar una agrupació que minimitzi les distàncies euclidianes per parelles de punts dins de cada clúster (normalitzada per les mides de clúster):

$$Z(C_1, ..., C_k) = \sum_{l=1}^{k} \frac{1}{2|C_l|} \sum_{i,j \epsilon C_l} \parallel x_i - x_j \parallel_2^2$$

Com més petita la Z, el clustering tindrà més qualitat. La Z és la nostra mesura de qualitat del clustering.

Matemàticament, representem així el centroid de cada clúster l:

$$\mu_l = \frac{1}{|C_l|} \sum_{i \epsilon C_l} x_i$$

De manera que, bàsicament, el centroide del clúster és la mitjana de tots els punts del clúster.

En resum, l'algorisme de Lloyd busca minimitzar la Z, és a dir:

$$minZ(C_1, ..., C_k)$$

2.1.1 Algorisme

L'algorisme de Lloyd és el següent:

1. Seleccionar a l'atzar K punts del dataset com centroides inicials de cada clúster $C1, ..., Ck$. Aquests centroides defineixen les fronteres inicials de les cel·les Voronoi.

2. Calcular, per tots els punts del dataset, la distància del punt a tots els centroides.

3. Assignar cada punt al clúster (cel·la Voronoi) amb el centroide més proper (ex. calculant la distància euclidiana).

4. Calcular la mitjana de punts en cada clúster per obtenir K nous centroides.

5. Repetir passos 2 i 4 fins que l'assignació de clústers no canviï (convergència), o el màxim nombre d'iteracions s'hagi assolit.

2.1.2 Cost

El cost de l'algorisme de Lloyd depèn de diversos factors, però en general es considera que té una complexitat $O(n * k * d * i)$, on:

- n és el nombre de punts del dataset.
- k és el nombre de clústers.
- d és la dimensionalitat de les dades (nombre de característiques o atributs per punt del dataset).
- i és el nombre d'iteracions necessàries per a la convergència.

Això s'explica perquè:

A cada iteració, l'algorisme ha d'assignar cada punt del dataset (n) al centroide (k) més proper. Això implica calcular la distància entre cada punt del dataset i tots els centroides. Per tant, $n * k$.

El càlcul de la distància depèn de la dimensionalitat de les dades (d). Com més gran sigui la dimensionalitat, més complex serà el càlcul de la distància.

L'algorisme s'executa de manera iterativa fins que els centroides convergeixen. El nombre d'iteracions (i) necessari per a la convergència pot variar en funció dels punts del dataset i la inicialització dels centroides.

En conclusió, l'eficiència de l'algorisme de Lloyd depèn molt del nombre d'iteracions que es fan abans que convergeixi.

2.1.3 Diagrames de Voronoi

Les fronteres dels clústers formen una tesselació Voronoi de l'espai, basada en els centroides dels clústers. Els diagrames de Voronoi són útils en àmbits com la meteorologia, les ciències naturals i les ciències socials.

2.2 Algorisme de kmeans++

K-Means++ és una variant de l'algorisme K-Means que es diferencia únicament en el càlcul inicial dels centroides. A l'algoritme K-Means aquesta assignació es feia de forma aleatòria i, per tant, estava lluny de ser l'òptima.

2.2.1 Algorisme

1. Escollim de forma aleatòria el primer centroide C1.
2. Calculem la distància entre tots els data points i el seu centroide més proper.

$$D_i = max_{j:1->k} \parallel x_i - c_j \parallel^2$$

3. Escollim la distància màxima, que és la distància més gran d'un data point Xi al seu centroide més proper.
4. Utilitzem el datapoint Xi com a nou centroide.
5. Repetim els passos 2,3 i 4 fins que tots els nous centroides siguin seleccionats.

2.2.2 Cost

Amb la inicialització k-means++, es garanteix que l'algorisme trobi una solució que sigui $O(logk)$ competitiva per a la solució òptima de k-means.

3 Mètodes d'avaluació de clusterings i d'optimització de k

En aquesta part implementarem tres tipus d'algorismes per a l'avaluació de clusterings, tan interna com externament, i per obtenir-ne la k-òptima en cada cas.

3.1 Avaluació interna: índex de Calinski–Harabasz

L'índex de Calinski-Harabasz és una mesura interna de qualitat que consisteix en mesurar la relació entre la dispersió interna dels clústers (WCSS) i la separació o distància entre clusters (BCSS).

Donats BCSS, WCSS, n (el nombre de punts) i k (el nombre de clústers, als quals estan assignats els punts), definim l'índex de Calinski-Harabasz així:

$$CH = \frac{BCSS/(k-1)}{WCSS/(n-k)}$$

Siguin c_i cada centroide, i c el centroide global, la separació entre clústers (BCSS) es mesura com la suma ponderada de les distàncies Euclidianes quadrades entre els centroides de cada clúster i el centroide global del dataset, així:

$$BCSS = \sum_{i=1}^{k} n_i \parallel c_i - c \parallel^2$$

Com més gran sigui el BCSS millor, ja que significarà que els clústers estan ben separats els uns dels altres.

La dispersió interna dels clústers (WCSS) la calculem com la suma de les distàncies Euclidianes quadrades entre els punts que pertanyen al clúster i el seu respectiu centroide, per a cada clúster.

$$WCSS = \sum_{i=1}^{k} \sum_{x \epsilon C_i} \parallel x - c_i \parallel^2$$

Com més petit sigui el valor de WCSS millor, ja que significa que els clústers són compactes i estan ben cohesionats.

3.2 Avaluació externa: Rand Index

El Rand Index és una mesura de similitud utilitzada per comparar dues particions diferents d'un mateix conjunt de dades. Aquesta mesura avalua la similitud entre dues particions basant-se en quants dels parells d'elements es classifiquen de la mateixa manera o de manera diferent en ambdues particions.

La fórmula matemàtica per calcular el Rand Index RI(X, Y) entre dues particions X i Y es defineix com:

$$RI(X,Y) = \frac{a+b}{a+b+c+d}$$

On:

- a és el nombre de parells d'elements que estan en el mateix clúster en les dues particions.
- b és el nombre de parells d'elements que estan en clústers diferents en les dues particions.
- c és el nombre de parells d'elements que estan en el mateix clúster en la partició X però en clústers diferents en la partició Y.
- d és el nombre de parells d'elements que estan en el mateix clúster en la partició Y però en clústers diferents en la partició X.

El denominador $a + b + c + d$ representa el total de parells d'elements en el conjunt de dades, mentre que el numerador $a + b$ representa els parells que estan classificats de la mateixa manera en ambdues particions, és a dir, el valor del Rand Index varia entre 0 i 1, que varia de menys a més coincidència.

3.3 Obtenció K-optima

Per determinar el nombre òptim de clústers en un conjunt de dades, hem decidit utilitzar la tècnica Elbow Method. Havent executat els algorismes de clustering amb diferents valors de k, es calcula la suma de les distàncies quadrades de cada punt al seu centroide corresponent, per a tots els punts en el conjunt de dades.

Per obtenir-ne la k òptima fem un gràfic amb aquestes sumes en funció del nombre de clústers i es busca el punt on hi ha una reducció significativa, que d'aquí és d'on en surt el nom del mètode, ja que la forma és semblant a un "colze". Aquest punt indica el nombre òptim de clústers i ajuda a millorar la interpretació i eficàcia del clustering.

En el nostre treball, representarem el gràfic automatitzant-ne el càlcul, per poder extreure'n la k òptima gràficament.

4 Experimentació

Aquesta última part del projecte la dedicarem a l'experimentació dels algorismes especificats fins ara amb un conjunt de datasets proporcionats pel professorat.

4.1 Experiment 1: Variants de k-means Dataset6

El primer experiment que farem serà comparar el clustering generat per les dues variants de l'algorisme de k-means estudiades: l'algorisme de Lloyd i l'algorisme de kmeans++ en un mateix dataset. Per aquest primer experiment hem decidit començar amb el Dataset6 que és de dues dimensions i se'n poden extreure gràfics representables.

4.1.1 Algorisme de Lloyd

Després d'executar l'algorisme de Lloyd en el Dataset6, hem obtingut el clustering de la següent figura per a k=3:

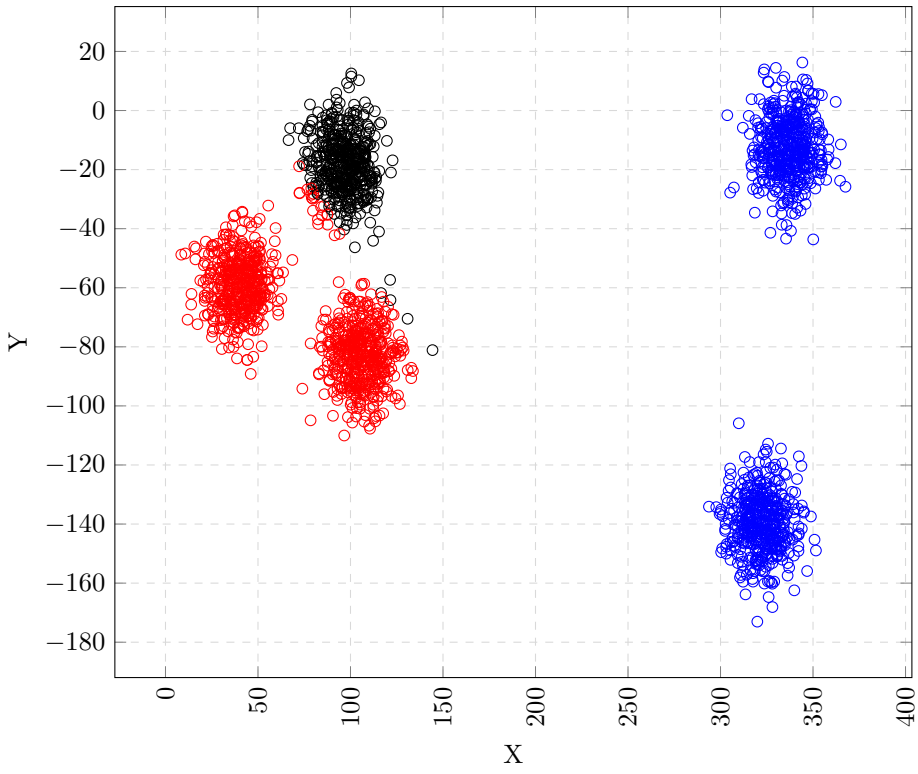

Figure 1: Execució de l'algorisme de Lloyd amb el Dataset6 amb k=3

Per a k=5 ens ha donat aquest resultat:

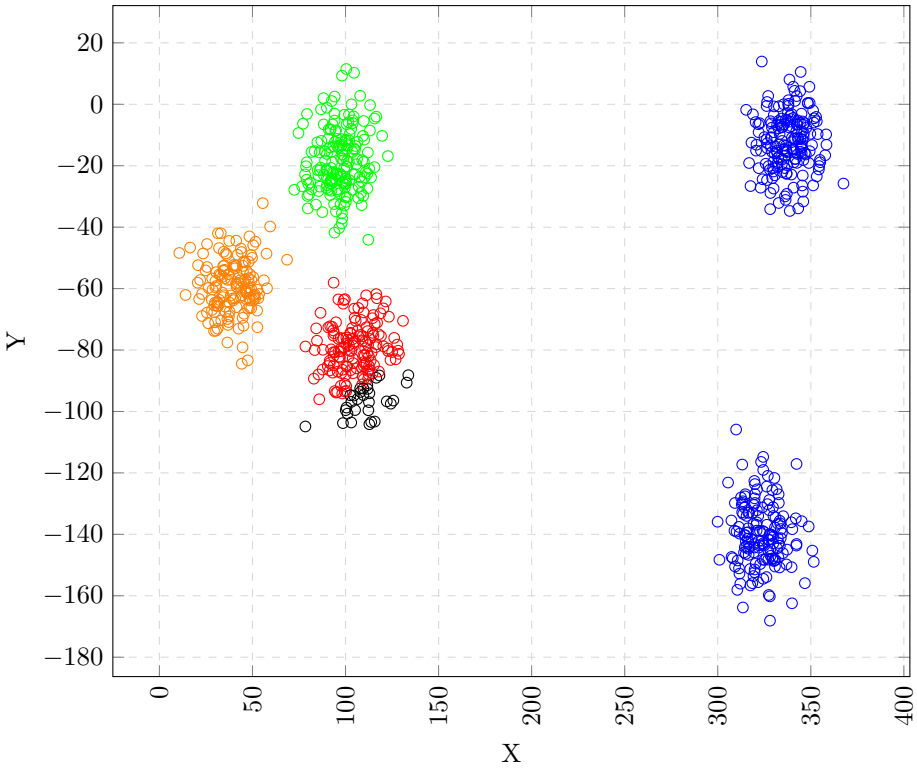

Figure 2: Execució de l'algorisme de Lloyd amb el Dataset6 amb k=5

4.1.2 Algorisme de kmeans++

Després d'executar l'algorisme de kmeans++ en el Dataset6, hem obtingut el clustering de la figura 3 per a k=3 i el de la figura 4 per a k = 5.

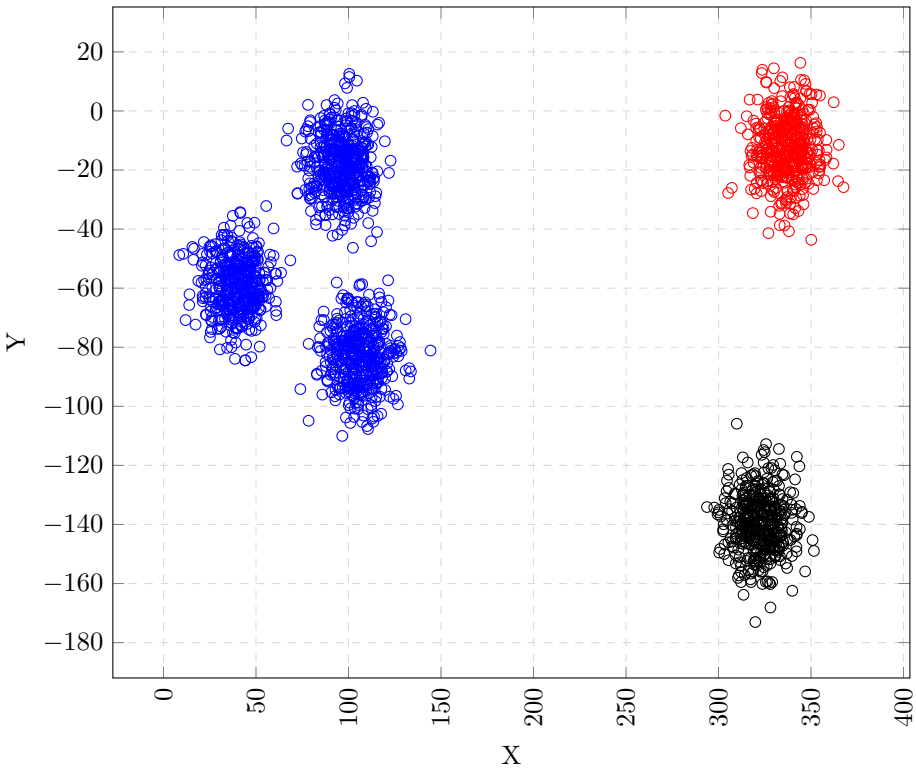

Figure 3: Execució de l'algorisme de kmeans++ amb el Dataset6 amb k=3

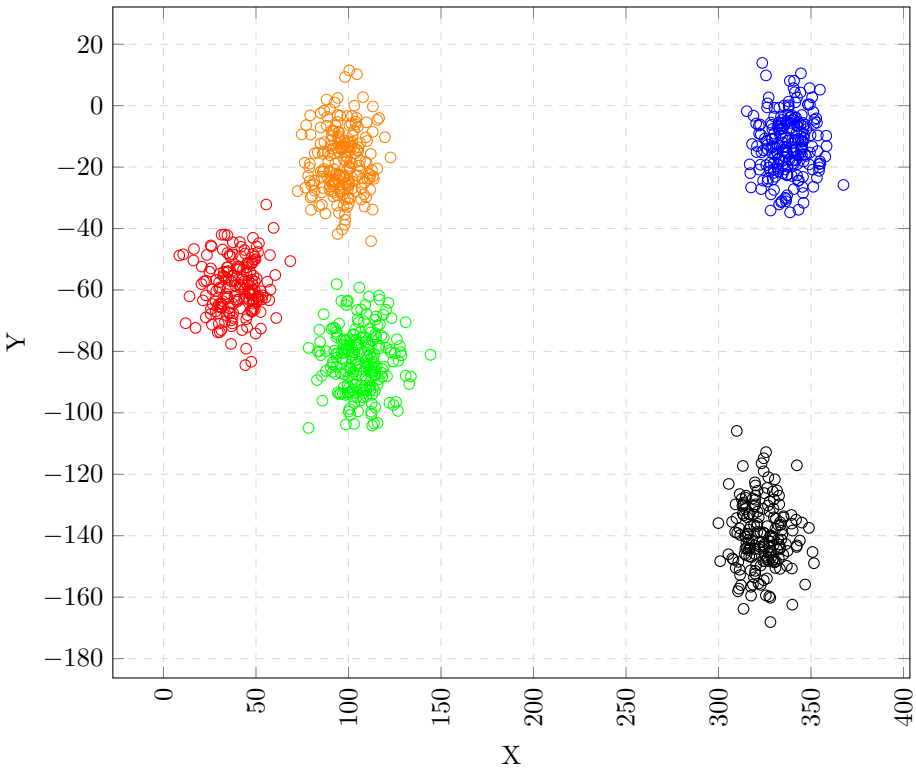

Figure 4: Execució de l'algorisme de kmeans++ amb el Dataset6 amb k=5

4.1.3 Comparació d'algorismes. Rand Index

Com hem esmentat anteriorment, la mesura externa dels algorismes la farem a partir del Rand Index, i algunes de les observacions més interessants que hem extret són amb els següents datasets:

Tant el Dataset1 com el Dataset2, Dataset3, Dataset4 i Dataset6, entre els dos algorismes obtenen un Rand Index d'1, això indica que les dues agrupacions generades pels algorismes són idèntiques. Aquest resultat podria suggir que el dataset1 pot tenir una estructura clara que facilita la seva clusterització i que ambdós algorismes convergeixen cap a la mateixa solució.

Amb un altre conjunt de dades, com el Dataset5 s'obté un Rand Index de 0.760918. Això indica que les dues agrupacions generades pels algorismes són relativament similars, però no idèntiques. Podria significar que el dataset5 té una estructura més complexa o que els algorismes estan trobant diferents solucions a causa de la seva inicialització o algoritme d'optimització de k.

4.2 Experiment 2: K-òptima

Per determinar la K òptima, com es va esmentar anteriorment, fem servir el mètode de l'elbow. En aquesta secció, examinarem el nombre òptim de clústers en un dataset específic, ja sigui que no se'ns demani cap valor K en particular i hàgim de determinar-lo, o bé per verificar si algun dels datasets proporcionats no correspon amb la K òptima utilitzant els nostres algorismes.

Cal recalcar que a causa de les diferècies entre ambdós algorismes pot ocórrer que el nombre de clústers ideals per un dataset sigui diferent per cada algorisme, a més a més, al ser un mètode gràfic no és gaire precís i cada observador pot determinar una k diferent.

Primerament, provem amb un dataset on no se'ns requereix cap valor K específic, com ara el Dataset5, i observem els gràfics resultants per a cada algorisme:

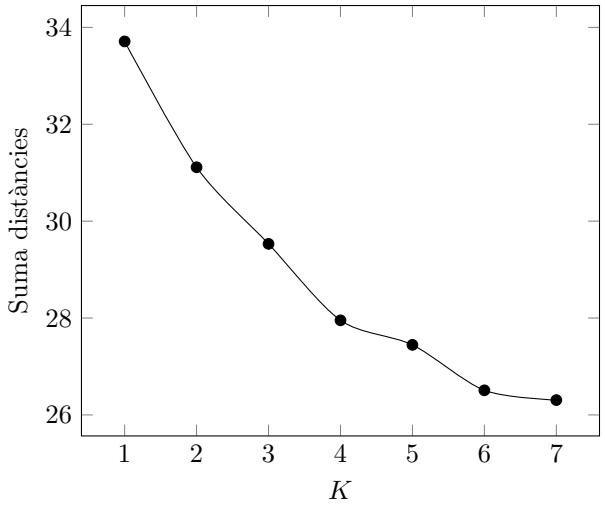

Figure 5: Mètode de l'elbow pel Dataset5 per l'algorisme de Kmeans++

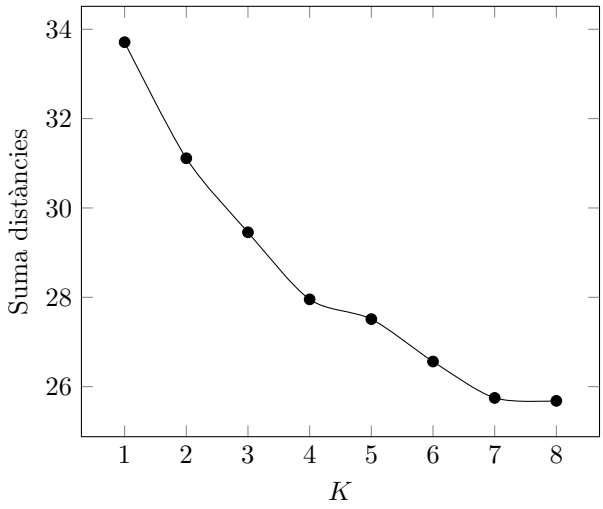

Figure 6: Mètode de l'elbow pel Dataset5 per l'algorisme de Lloyd

En aquest cas veiem que la k òptima a l'algorisme k-means++ i LLoyd és 4.

Ara podem provar un Dataset6 en el que ens indiquen el valor de la k a utilitzar (k = 3) i comprovar si aquest valor coincideix amb el valor obtingut al gràfic.

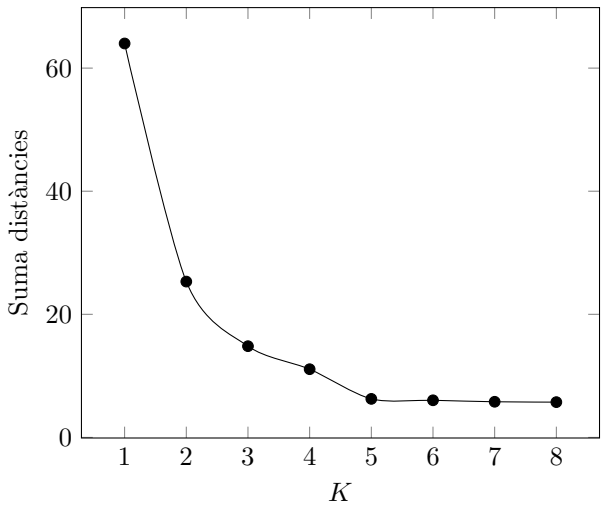

Figure 7: Mètode de l'elbow pel Dataset6 per l'algorisme de Kmeans++

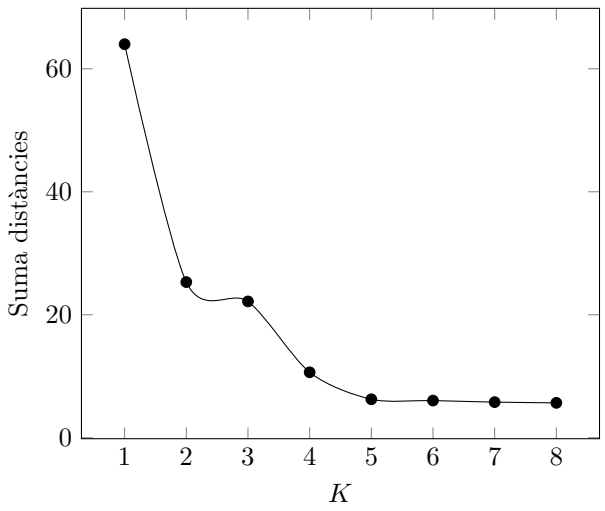

Figure 8: Mètode de l'elbow pel Dataset6 per l'algorisme de Lloyd

Veiem que pel k-means++ la k-òptima és 3 mentre que per l'algorisme de Lloyd és 4. Per k-means++ la k-òptima és idèntica mentre que per lloyd quasi.

Finalment mirem el Dataset3 en el que ens indiquen el valor de la k a utilitzar (k = 2) i comprovem si aquest valor coincideix amb el valor obtingut al gràfic.

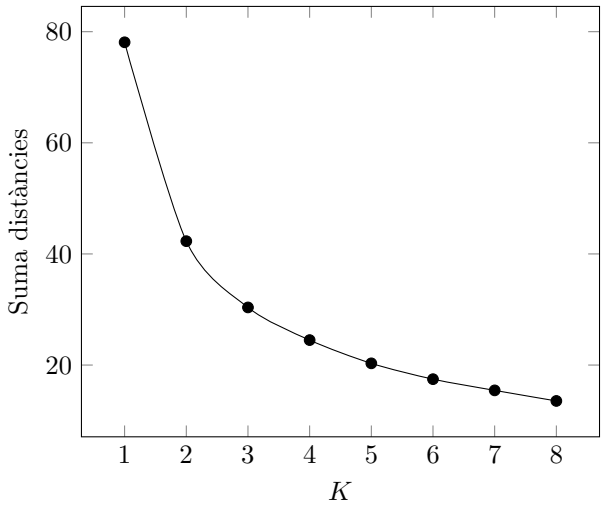

Figure 9: Mètode de l'elbow pel Dataset3 per l'algorisme de K-means++

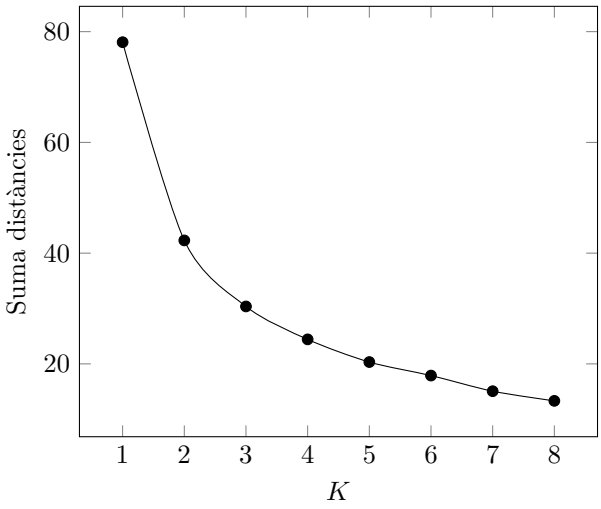

Figure 10: Mètode de l'elbow pel Dataset3 per l'algorisme de Lloyd

Com es pot observar, la k-òptima pels dos algorismes coincideix. Aquesta k és 3 i no 2 com s'indicava al Dataset3.

4.3 Expertiment 3: Temps d'execucio dels codis

A continuació exposem i comentarem els temps d'execució dels diferents Datasets en els diferents algorismes emprats:

Table 1: Temps d'execució de cada algorisme segons el dataset (en segons)

	DS1 (k=7)	DS2 (k=2)	DS3 (k=2)	DS4 (k=5)	DS5 (k=8)	DS6 (k=3)
LLoyd	0,3926432	0,006257214	0,06691898	0,02858108	0,2717968	0,5651252
K-means++	0,5889782	0,007012488	0,07443594	0,02192066	0,3011806	0,6129618
Rand Index	0,773896	0,000675714	0,0615627	0,00476351	0,126376	13,1135
Elbow Lloyd	1,18767	0,0150686	0,553448	0,0946945	0,682379	0,50471
Elbow K++	1,00535	0,0135278	0,598495	0,0404778	0,694256	3,40305
CH Index	0,044983	0,000920839	0,00757894	0,0037436	0,0206456	0,054809

Com es pot observar a la taula anterior, el temps d'execució de l'algorisme de Lloyd generalment és menor que el temps d'execució del K-means++. Aquesta discrepància es deu probablement al fet que, mentre el primer selecciona punts aleatoris del dataset com a centroides inicials, el K-means++ calcula centroides òptims.
A més, es pot notar que el temps d'execució dels dos algorismes augmenta significativament a mesura que el nombre de punts en el dataset i les seves dimensions creixen.

4.4 Expertiment 4: Mesura Interna de Qualitat

Per mesurar la qualitat interna dels algorismes de clustering, com es va esmentar anteriorment, fem servir l'índex de Calinski-Harabasz que ens mostra el ratio entre la dispersió interna dels clústers i la separació entre clústers.

Al calcular l'índex per cada algorisme i dataset obtenim la seguent taula:

Table 2: Puntuació obtinguda per cada algorisme i Dataset

	DS1 (k=7)	DS2 (k=2)	DS3 (k=2)	DS4 (k=5)	DS5 (k=8)	DS6 (k=3)
LLoyd	13387,1	447,917	7222,91	241,776	650,821	76799,8
K-means++	27975,2	469,983	7224,29	274,825	743,21	161109

Per si sola aquest índex no significa molt, però si comparem els valors obtinguts entre iteracions o entre algorismes podem arribar a certes conclusions.

En aquest cas com mostra la taula, l'algorisme de K-means++ obté millors puntuacions que l'algorisme de lloyd grácies a com inicialitza els clústers. Aquesta implementació li permet aconseguir puntuacions iguals o més grans que l'algorisme de lloyd fins i tot duplicar-li la puntuació als datasets 1 i 6 a costa de un augment en temps d'execució i complexitat.

5 Metodologia, organització i procès d'autoaprenentatge

5.1 Metodologia del Treball

Hem començat el treball analitzant els requisits del projecte i planificant la nostra estratègia per a cada part del projecte. Hem definit els algorismes a implementar i les mesures a utilitzar per a la seva posterior avaluació.

Després d'informar-nos de diferents variants del Kmeans, hem triat i implementat dues variants de l'algorisme de K-means que són les que ens han semblat més interessants a tractar que, posteriorment, hem implementat diferents mesures per determinar-ne la qualitat.

Quan ja hem tingut tota la part d'implementació hem començat amb l'experimentació. Hem dut a terme experiments amb diversos conjunts de dades per avaluar l'eficiència i la qualitat dels algorismes. Hem mesurat el temps d'execució, les mesures internes i externes de qualitat del clustering, i hem aplicat els mètodes per a determinar el nombre òptim de clústers.

Finalment, hem analitzat els resultats obtinguts a partir dels experiments i hem interpretat les conclusions i hem comparat les prestacions dels diferents algorismes.

5.2 Organització del Treball

Per començar, pel que fa a la divisió de tasques hem assignat tasques específiques a cada membre de l'equip, això ha inclòs la implementació d'algorismes, la recopilació de dades, l'execució d'experiments i l'anàlisi de resultats.

També hem mantingut una comunicació regular a través de reunions periòdiques per a discutir el progrés del projecte, resoldre problemes i prendre decisions importants per estar sempre al corrent de problemes que ens havien anat sorgint durant el projecte. Aquestes reunions tratavem de fer-les de forma presencial per facilitar l'enteniment i quan no ha sigut possible de forma virtual.

A més a més, hem utilitzat sistemes de control de versions com Git per poder compartir i anar traballant sobre les versions actualitzades de codi entre nosaltres.

5.3 Procés d'aprenentatge

Durant el desenvolupament d'aquest projecte, hem experimentat un notable creixement i aprenentatge en diversos aspectes. En primer lloc, hem adquirit un coneixement profund sobre el funcionament dels algorismes de clustering, una comprensió que ha anat més enllà de les expectatives inicials. Això ens ha permès implementar-los de manera efectiva i entendre les seves aplicacions en

l'anàlisi de dades.

Un dels reptes més destacats ha estat l'adquisició de nous coneixements tècnics, com l'ús de LaTeX per generar gràfics i altres funcionalitats, ampliant així el nostre ventall de competències en l'àmbit de la ciència de dades. Aquesta nova habilitat no només ens ha beneficiat en aquest projecte, sinó que també serà útil en futurs treballs i investigacions.

Hem de reconèixer que el procés també ha implicat moments de desafiament i fins i tot estrès, especialment en qüestions relacionades amb el processament de les dades per a la seva posterior utilització. Aquestes dificultats, però, ens han permès créixer i desenvolupar habilitats de resolució de problemes de manera eficient.

A més, hem dedicat temps a investigar i explorar diverses variants de l'algorisme k-means, consultant múltiples fonts d'informació, inclosos llibres de referència, per comprendre més a fons el seu funcionament i la seva implementació. Aquesta tasca de recerca ha estat crucial per aprofundir en els aspectes teòrics i pràctics de l'algorisme.

Finalment, hem après sobre les tècniques experimentals per avaluar els algorismes, des del càlcul fins a l'ús de mètriques d'avaluació. Aquesta comprensió ens ha proporcionat una visió més completa i crítica dels resultats obtinguts, permetent-nos fer interpretacions significatives i prendre decisions informades en el procés d'anàlisi de dades.

6 Conclusions

Després de l'experimentació amb els dos algorismes desenvolupats i l'ús dels mètodes d'avaluació esmentats, es poden extreure diverses conclusions rellevants:

Pel que fa a l'Elbow Method, s'observa que la k-òptima és molt semblant pels dos algorismes, tot i que pot no ser exactament la mateixa. Aquesta k-òptima també és consistent amb els valors de k suggerits pels datasets analitzats, indicant una certa coherència en la selecció del nombre de clústers.

Utilitzant el Rand Index, es pot veure que és possible que els dos algorismes donin com a resultat la mateixa assignació de clústers, com es pot observar en el Dataset5, on el valor del Rand Index és 1. Tot i això, en general, els valors del Rand Index no coincideixen exactament, però indiquen una agrupació similar. Comparant els gràfics generats per ambdós algorismes, es pot apreciar que les diferències no són significatives.

A través de l'experimentació amb diversos conjunts de dades, s'ha mostrat com els algorismes de Lloyd i K-means++ funcionen en diferents contextos. Tot i que en alguns casos les agrupacions generades són molt similars, en altres casos poden presentar diferències moderades. Un aspecte rellevant és que el temps d'execució de l'algorisme de Lloyd és lleugerament menor, ja que la inicialització de K-means++ implica un cost computacional addicional que no sempre es compensa amb una convergència més ràpida.

Observant les gràfiques dels resultats del punt 4.1.1 i 4.1.2 de l'experimentació, queda evident de forma visual com l'algorisme K-means++ millora significativament l'agrupació per clústers dels punts. A més, aquesta millora es pot veure de forma numèrica observant que obté unes puntuacions en tots els casos més grans que l'algorisme de lloyd amb l'índex de Calinski–Harabasz.

Tot i que pot ser una mica més costós en temps computacional, els seus resultats justifiquen aquesta inversió, demostrant la seva eficàcia en la generació de clústers més precisos i significatius.

References

[1] Natasha Sharma. K-Means Clustering Explained. Font: https://www.deeplearning.ai/the-batch/k-means-clustering-group-think/

[2] Wikipedia. K-means. Font: https://en.wikipedia.org/wiki/K-means

[3] GeeksForGeeks. ML — K-means++ Algorithm. Font: https://www.geeksforgeeks.org/ml-k-means-algorithm/

[4] Wikipedia. Font: https://https://en.wikipedia.org/wiki/Calinski

[5] Cornell. K-means. Font: https://www.cs.cornell.edu/courses/cs4780/2022sp/notes/LectureNotes04.html

[6] Edo Liberty. Lecture 10: k-means clustering. Font: https://www.cs.yale.edu/homes/el327/datamining2013aFiles/10_k_means_clustering.pdf

[7] Font: https://neptune.ai/blog/k-means-clustering

[8] K-Means clustering in c++. Robert Andrew Martin 2023. Font: https://reasonabledeviations.com/2019/10/02/k-means-in-cpp/

[9] Advances in Information Retrieval Theory. Font: https://link-springer-com.recursos.biblioteca.upc.edu/book/10.1007/978-3-642-04417-5

[10] Advances in Information and Communication. Font: https://link-springer-com.recursos.biblioteca.upc.edu/book/10.1007/978-3-030-73103-8

[11] Emerging Trends in Knowledge Discovery and Data Mining. Font; https://link-springer-com.recursos.biblioteca.upc.edu/book/10.1007/978-3-642-36778-6