Università degli Studi di Catania
Dipartimento di Matematica ed Informatica
Corso di Laurea Magistrale in Informatica

ESPLORAZIONE DI DATI CON TECNICHE DI CLUSTERING

Prof. Alfredo Ferro
Dott. Salvatore Alaimo
Dott. Giovanni Micale

———

Francesco Anastasio

Sommario

Introduzione

1.1 Scopo del progetto

Il progetto si pone l'obiettivo di esplorare i due dataset forniti dal docente, ***weibull.params*** e ***lognorm.params***.

Per fare ciò sono state utilizzate diverse tecniche di clustering, nello specifico:

- Clustering Gerarchico
- K-means
- DBSCAN

È stato utilizzato lo strumento RStudio e il linguaggio di programmazione R per l'esplorazione dei dati.

1.2 Linguaggio R e RStudio

R è un linguaggio per l'elaborazione statistica e grafica. Esso fornisce un'ampia varietà di tecniche statistiche.

RStudio è un IDE gratuito ed opensource per R.

Entrambi, R come linguaggio di programmazione ed RStudio come IDE, sono stati utilizzati per sviluppare il progetto.

1.3 Clustering Gerarchico

Il clustering gerarchico è un approccio al clustering in cui ciascun punto viene inizialmente posto in un cluster diverso e successivamente vengono combinati tra loro secondo una nozione di distanza opportunamente definita.

Una classica nozione di distanza, nel caso di spazio euclideo è la ***Distanza euclidea***, definita dalla formula in figura 1:

$$D(x, y) = \sqrt{\sum_{i=1}^{n} (x_i - y_i)^2}$$

Figura 1: Definizione di distanza euclidea tra due punti.

Il vantaggio del clustering gerarchico è che non necessita di nessun parametro iniziale, ad esso è infatti spesso associato un albero, chiamato ***dendogramma***, che descrive in che modo i cluster sono stati via via combinati. Effettuando un taglio del ***dendogramma*** ad un determinato livello, i sottoalberi ottenuti costituiscono i cluster prodotti dall'algoritmo.

1.4 K-means

K-means è un popolare algoritmo di Unsupervised learning. Esso può essere utilizzato solamente su spazi eucludei ed è basato su "assegnamento di punti". Anche se solitamente gli algoritmi di apprendimento non supervisionato non hanno necessità di alcun parametro, il K-means ha bisogno di avere, a priori, la conoscenza del numero di cluster da ricercare (parametro "K").
Vi sono diverse tecniche per stimare questo parametro, una delle quali (che è anche la stessa utilizzata per lo studio dei dataset oggetto di questa relazione) è la seguente:
All'aumentare del parametro "K" la distanza media dei punti dai centroidi diminuisce.

La diminuzione è dapprima drastica, poi sempre più contenuta. Il valore ideale di "K" è quello a partire dal quale la distanza media varia poco, solitamente al di sotto di una soglia definita.
Un esempio può essere osservato nella Figura 2. Nel caso specifico il numero "K" ideale potrebbe essere 8.

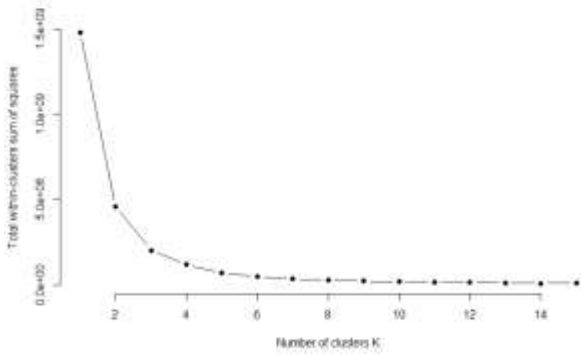

Figura 2: Esempio di un grafico per il calcolo del miglior valore del parametro "K" nell'algoritmo K-means.

1.5 DBSCAN

DBSCAN è, così come il K-means, un popolare algoritmo di unsupervised learning che, al contrario di quest'ultimo, sfrutta una nozione di "densità" per operare.
È caratterizzato da due parametri, ε e ***MinPts***. ε è il raggio legato alla grandezza dei cluster, ***MinPts*** invece è il numero minimo di punti che un cluster deve avere per essere considerato tale.
La scelta dei due parametri risulta quindi cruciale.
Per quanto riguarda ***MinPts***, è consigliato scegliere un ***MinPts >= D + 1*** dove ***D*** è la dimensionalità dello spazio. ***MinPts*** deve essere tanto più alto quanto:

1. Più ampio è il dataset di punti.
2. Maggiore è il rumore presente.

La stima del valore ε va fatta dopo aver stimato il valore ***MinPts***. Per stimare ε, si possono ordinare i punti del dataset sulla base della distanza dal k-esimo punto più vicino, dalla distanza più alta a quella più bassa e plottare tali distanze. Il valore ottimale di ε è l'ordinata del punto del grafico in cui la curva piega maggiormente. Un esempio si può vedere in Figura 3:

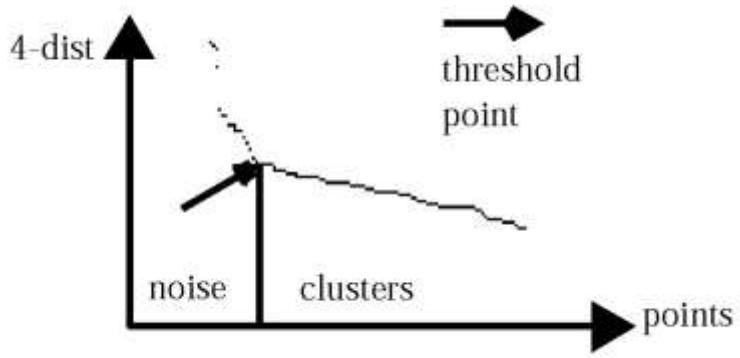

Figura 3: Scelta del miglior ε nel caso di DBSCAN.

Dataset 1 – Weibull.params

2.1 Esplorazione e pulizia dei dati

La matrice fornitomi è costituita da due colonne di numeri decimali, ***shape*** e ***scale***, e da 2299 righe totali.

Da un semplice Plot è apparso subito chiaro che i dati contenevano qualche ***outlier*** (Figura 4), si è quindi proceduto alla pulizia dei dati, andando ad eliminare le righe contenenti valori di Scale > 5000.

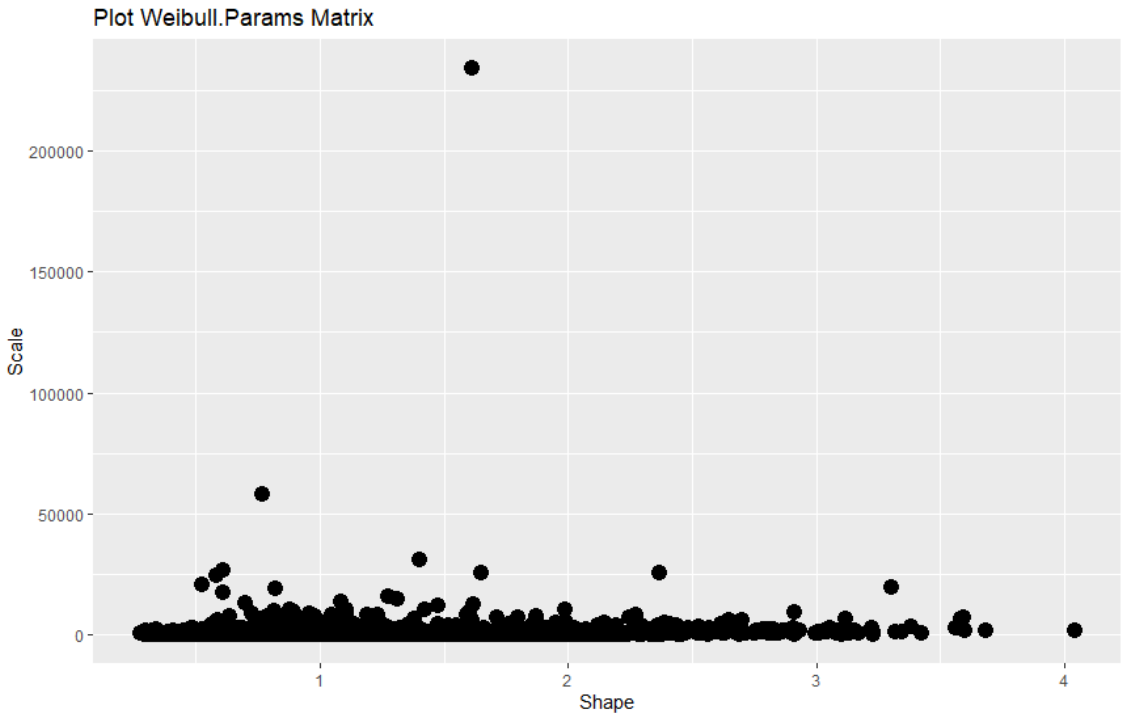

Figura 4: Plot della matrice Weibull.Params.

Le righe sono così passate dalle iniziali 2299 a 2233, il grafico è risultato essere il seguente (Figura 5):

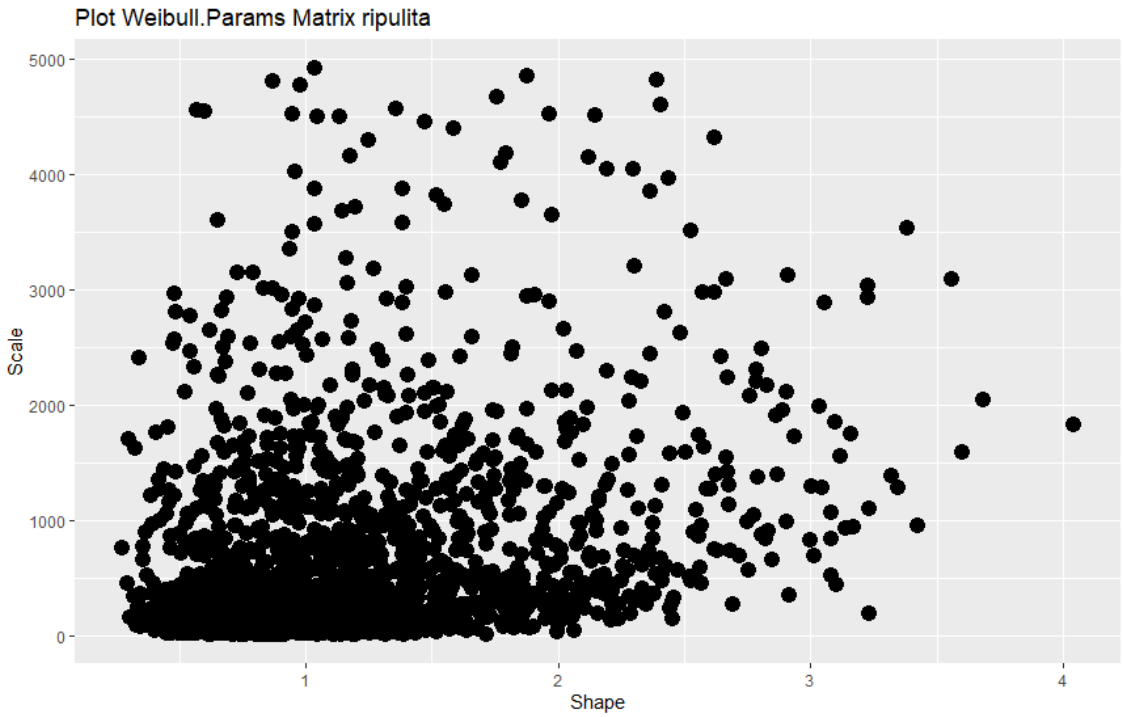

Figura 5: Plot Matrice Weibull.Params dopo eliminazione righe con Scale > 5000.

A partire da questa matrice sono stati applicati i tre algoritmi di Clustering presentati precedentemente.

2.2 Risultati Clustering Gerarchico

Il dendogramma relativo all'applicazione dell'algoritmo di clustering gerarchico al dataset ripulito è il seguente (Figura 6):

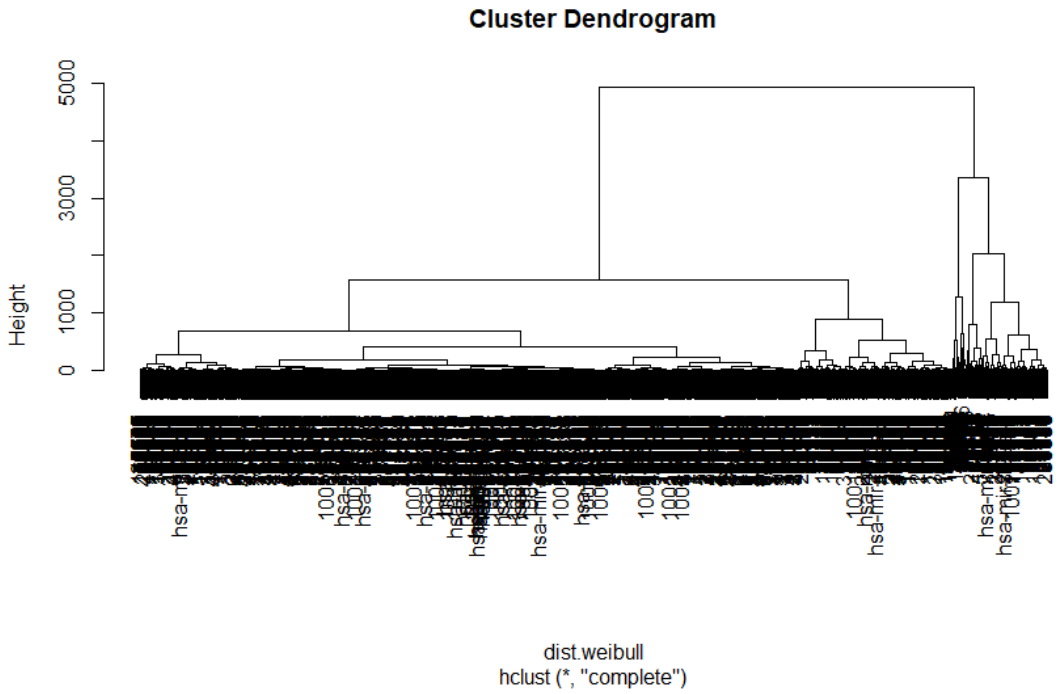

Figura 6: Dendogramma clustering gerarchico.

Risultando illeggibile si è proceduto a vari tagli del dendogramma a diverse altezze (in figura 7 l'esempio del taglio ad altezza 12, con 13 cluster totali) senza raggiungere apparentemente grandi risultati.

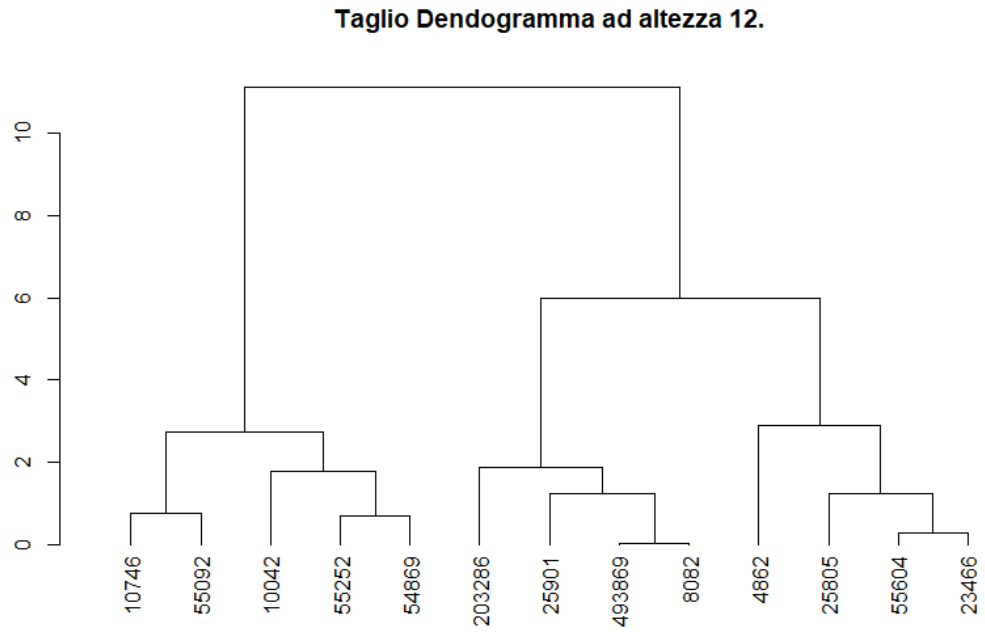

Figura 7: Taglio dendogramma ad altezza 12.

Il numero di elementi per ogni cluster risulta essere il seguente:

1	2	3	4	5	6	7	8	9	10	11	12	13
1376	268	32	74	17	245	29	111	14	36	5	18	8

Figura 8: Numero di elementi per ognuno dei 13 cluster.

2.3 Risultati DBSCAN

Prima di applicare l'algoritmo DBSCAN si è proceduto alla stima dei due parametri necessari all'algoritmo, ***MinPts*** ed ε:

- ***MinPts*** è stato posto uguale a 5.
- ε è stato calcolato applicando, via codice, quanto descritto nel paragrafo ***1.5***. È risultato essere un buon ε il valore 21, è possibile evincere ciò dalla figura 9.

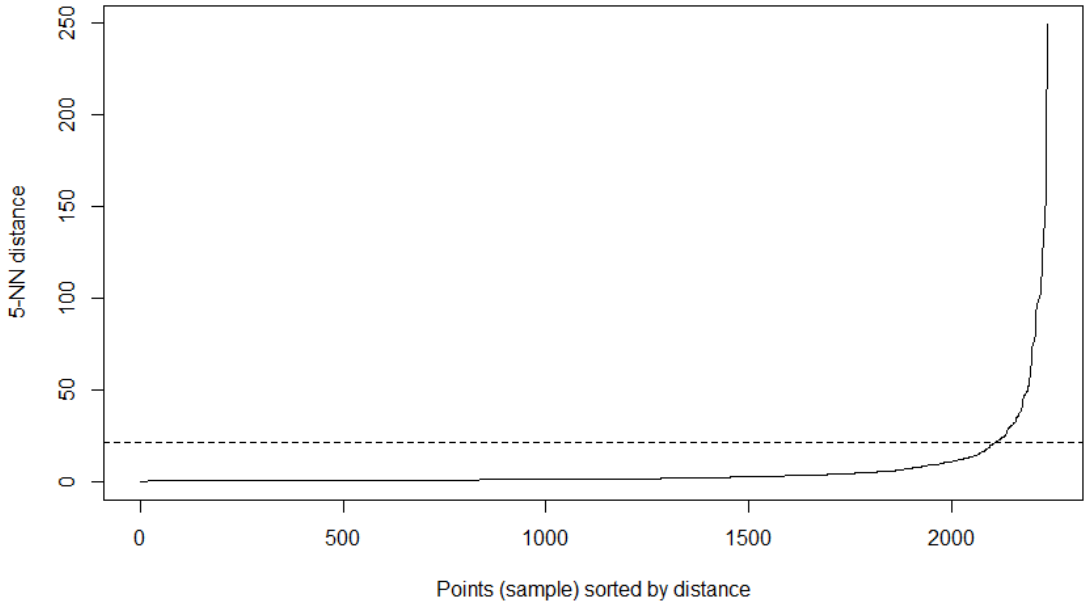

Figura 9: punti ordinati sulla base della distanza di un 5-nn. La linea tratteggiata corrisponde al valore di y 21.

Applicando, quindi, l'algoritmo DBSCAN alla matrice ripulita con i parametri ε = 21 e ***MinPts*** = 5 il risultato è il seguente (Figura 10):

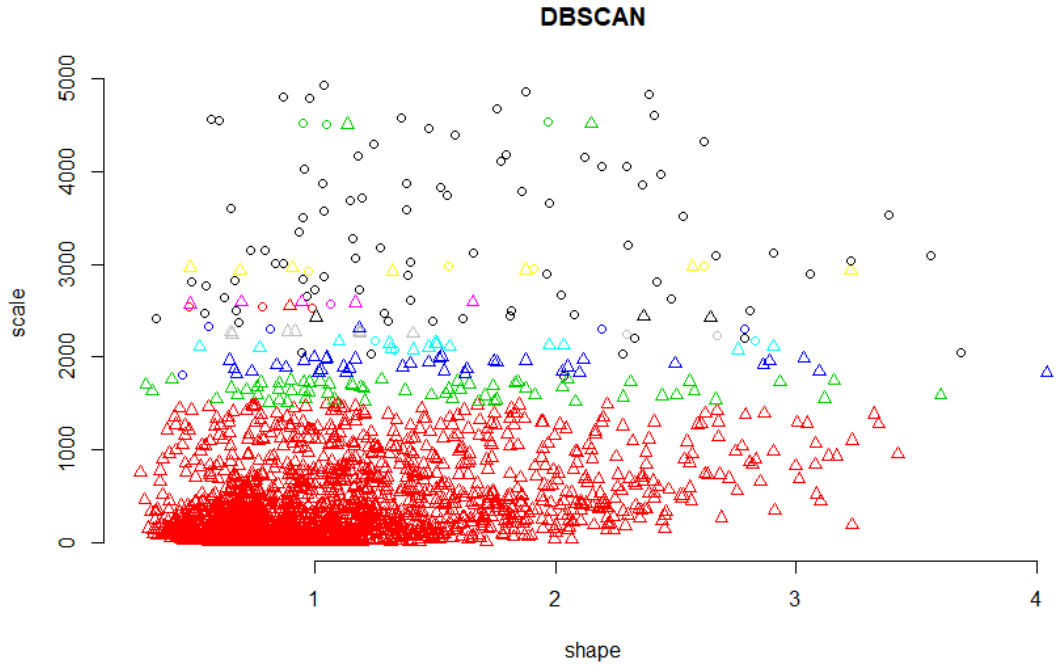

Figura 10: Plot cluster creati con algoritmo DBSCAN e parametri ε = 21 e MinPts = 5

2.4 Risultati K-means

Per l'applicazione dell'algoritmo K-means è stato calcolato il parametro K, necessario all'algoritmo. Per fare ciò è stata utilizzata la strategia descritta nel paragrafo 1.4. Il risultato si può osservare in figura 11:

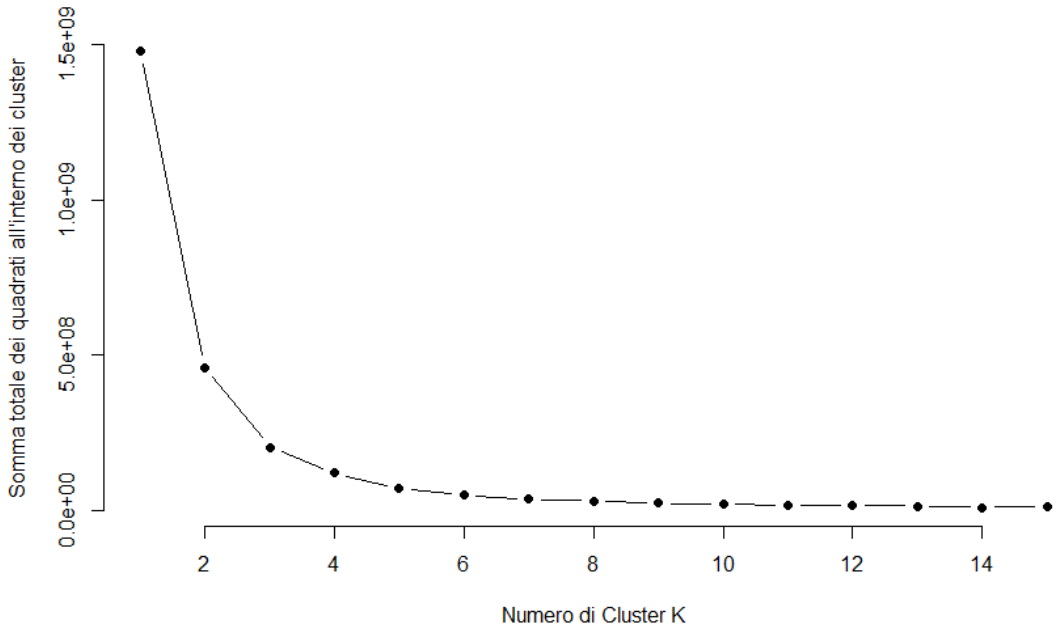

Figura 11: Calcolo del miglior K per il dataset Weibull.params.

Il miglior K è risultato essere il 5, quindi si è proceduto all'applicazione dell'algoritmo K-means. A seguire (figura 12) il plot dei risultati.

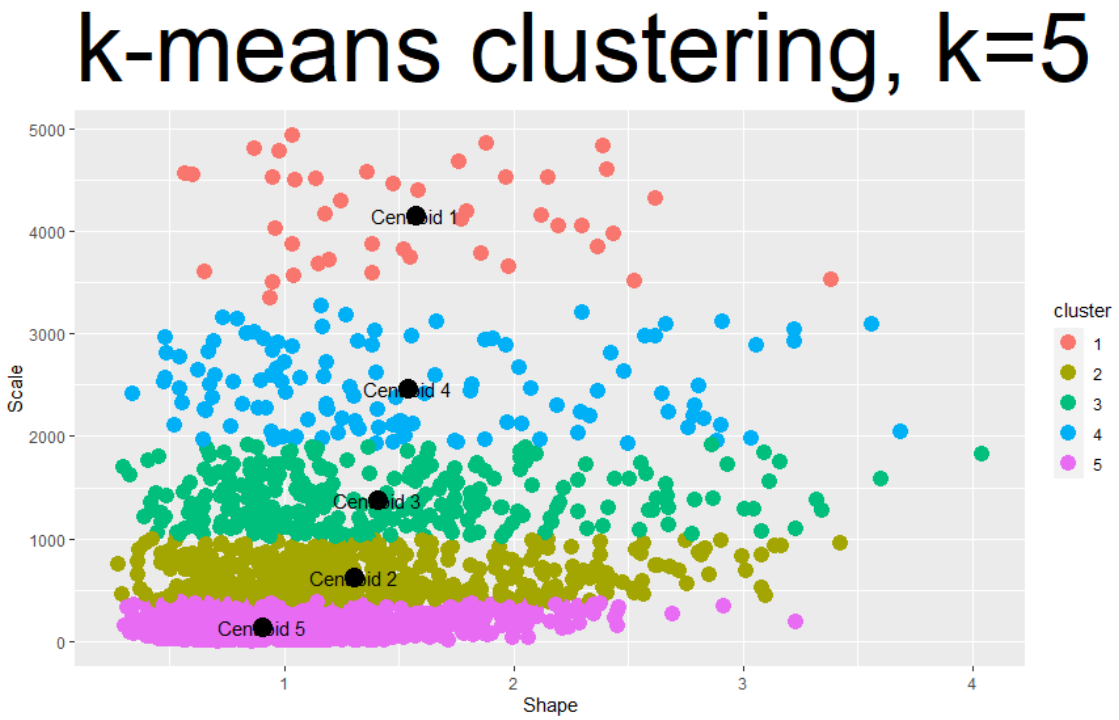

Figura 12: Plot del risultato dell'applicazione dell'algoritmo K-means con K = 5.

Dataset 2 – Lognorm.params

2.1 Esplorazione e pulizia dei dati

Il dataset Lognorm.params è formato da due colonne, ***meanlog*** e ***sdlog***, e conta 8611 osservazioni.

Anche in questo caso l'esplorazione è iniziata con un semplice plot delle due colonne (figura 13).

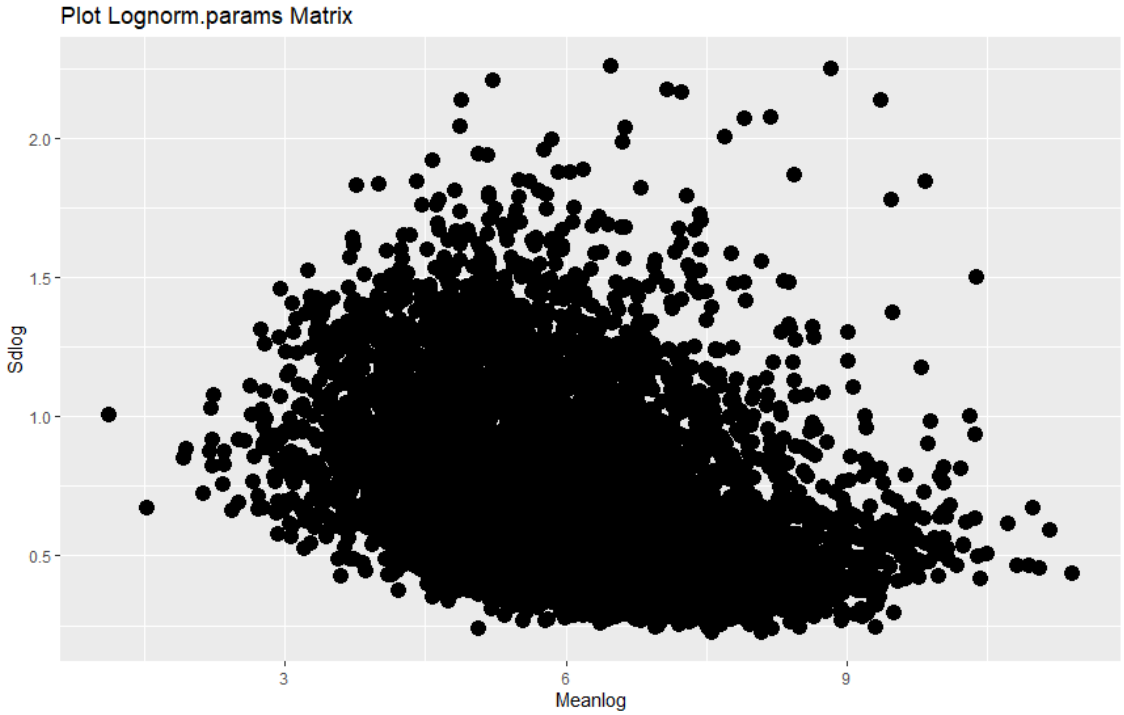

Figura 13: Plot dataset Lognorm.params.

In questo caso, rispetto al precedente, non risultano esserci outlier, di conseguenza non si è proceduto a nessun filtering dei dati.

2.2 Risultati Clustering Gerarchico

Così come per lo studio del dataset al paragrafo 2.2, anche in questo caso il plot del dendogramma risulta essere incomprensibile. Si è quindi proceduto ad un taglio dello stesso ad altezza 1 (arbitraria) e di seguito è possibile vedere il plot del taglio (Figura 14) e il numero di elementi per ognuno dei 13 cluster (figura 15):

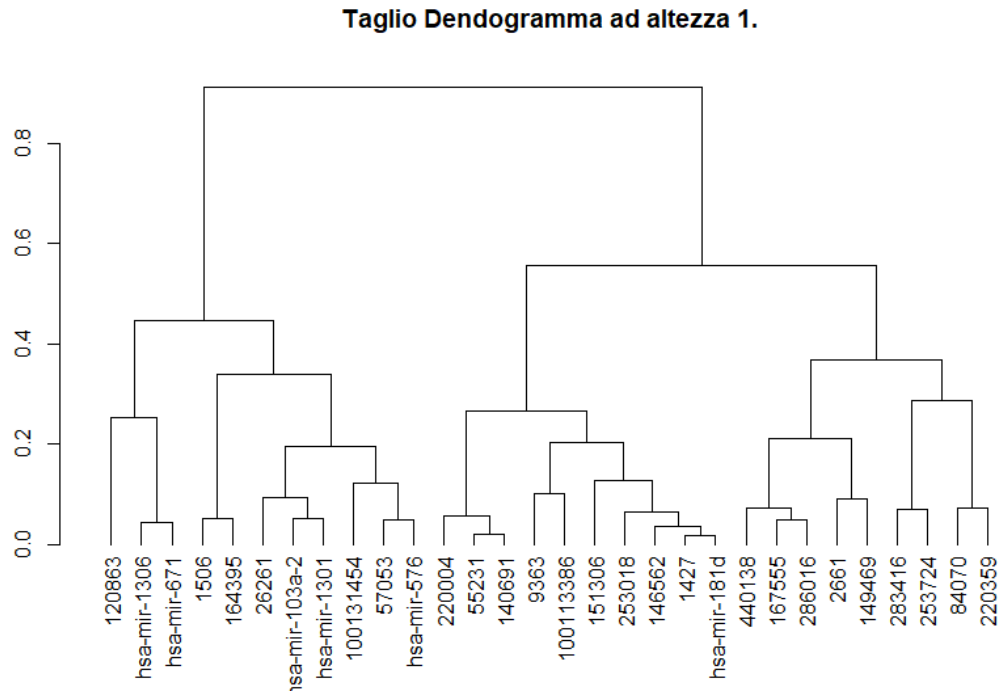

Figura 14: Plot taglio ad altezza 1 del dendogramma output del clustering gerarchico effettuato sulla matrice Lognorm.params.

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
1589	525	1005	918	307	345	105	1084	382	729	431	147	92	199	164	170	12	7	112	3	120	38	7	26
25	26	27	28	29	30																		
30	22	28	6	6	2																		

Figura 15: Numero di elementi per ognuno dei 30 cluster (taglio ad altezza 1).

2.3 Risultati DBSCAN

Anche in questo caso, prima di applicare l'algoritmo si è andati alla ricerca dei due parametri necessari al funzionamento dello stesso, ***MinPts*** ed ε:

- ***MinPts*** è stato posto uguale a 10 (rispetto alla precedente matrice abbiamo più osservazioni).
- ε è stato calcolato applicando, via codice, quanto descritto nel paragrafo ***1.5***. È risultato essere un buon ε il valore 0.12, è possibile evincere ciò dalla figura 16.

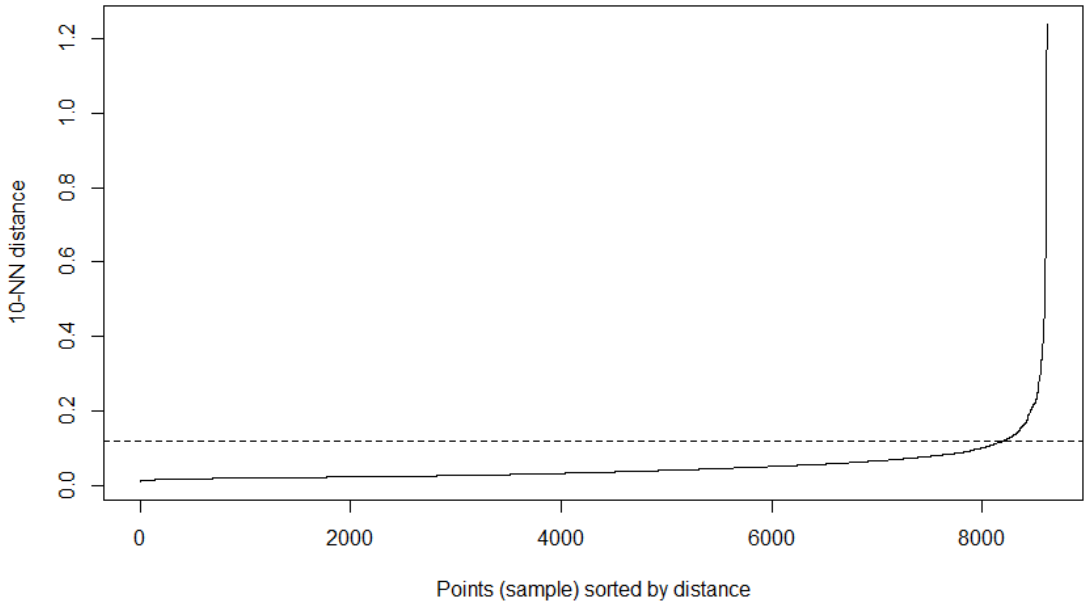

Figura 16: punti ordinati sulla base della distanza di un 10-nn. La linea tratteggiata corrisponde al valore di y 0.12.

L'output dell'algoritmo DBSCAN applicato con i parametri trovati, ossia MinPts = 10 e epsilon = 0.12 è il seguente (Figura 17):

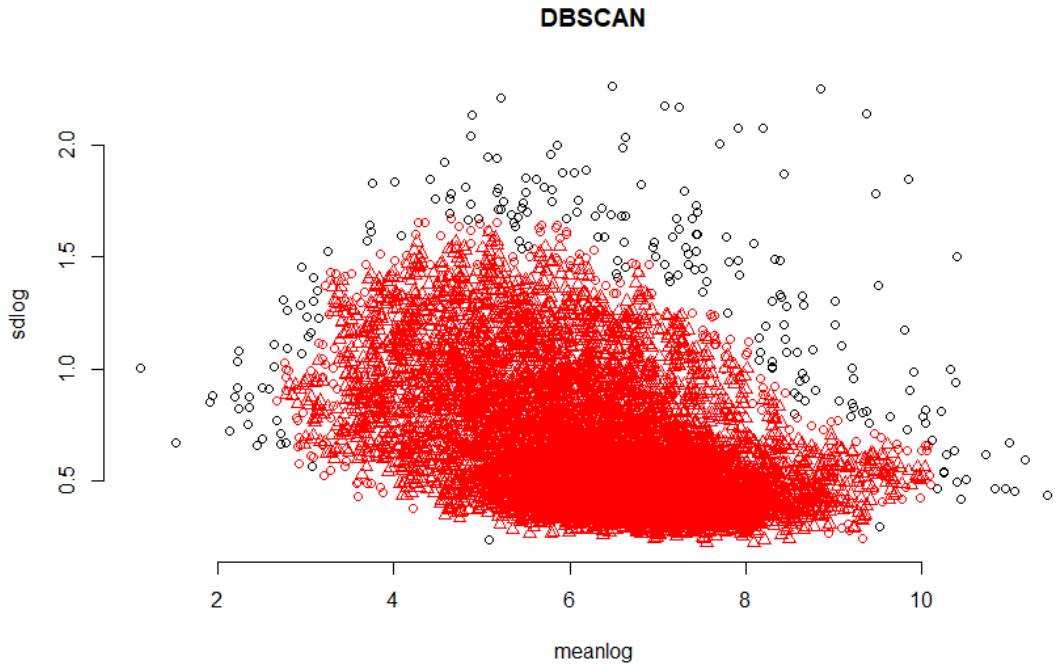

Figura 17: Plot cluster creati con algoritmo DBSCAN e parametri $\varepsilon = 0.12$ e MinPts = 10

Come si può facilmente evincere dal plot precedente, l'algoritmo non riesce correttamente a clusterizzare i dati.

2.4 Risultati K-means

Per l'applicazione dell'algoritmo k-means si è proceduto dapprima al calcolo del parametro k, che come si evince dalla figura 18 risulta essere 10.

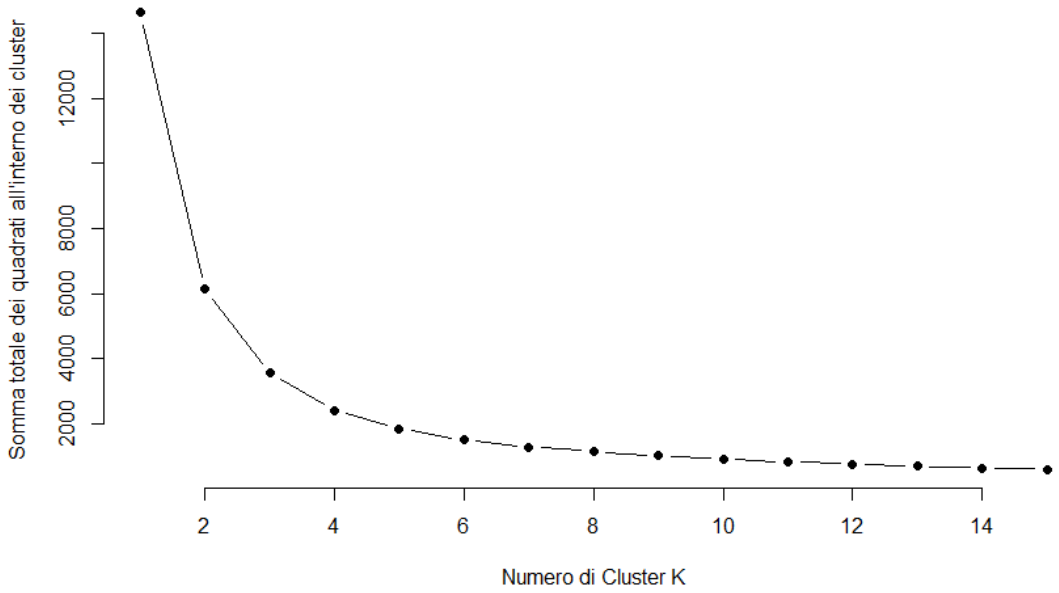

Figura 18: Calcolo del miglior K per il dataset lognorm.params.

Applicando l'algoritmo di cluster con un k = 10, il risultato è il seguente (Figura 19):

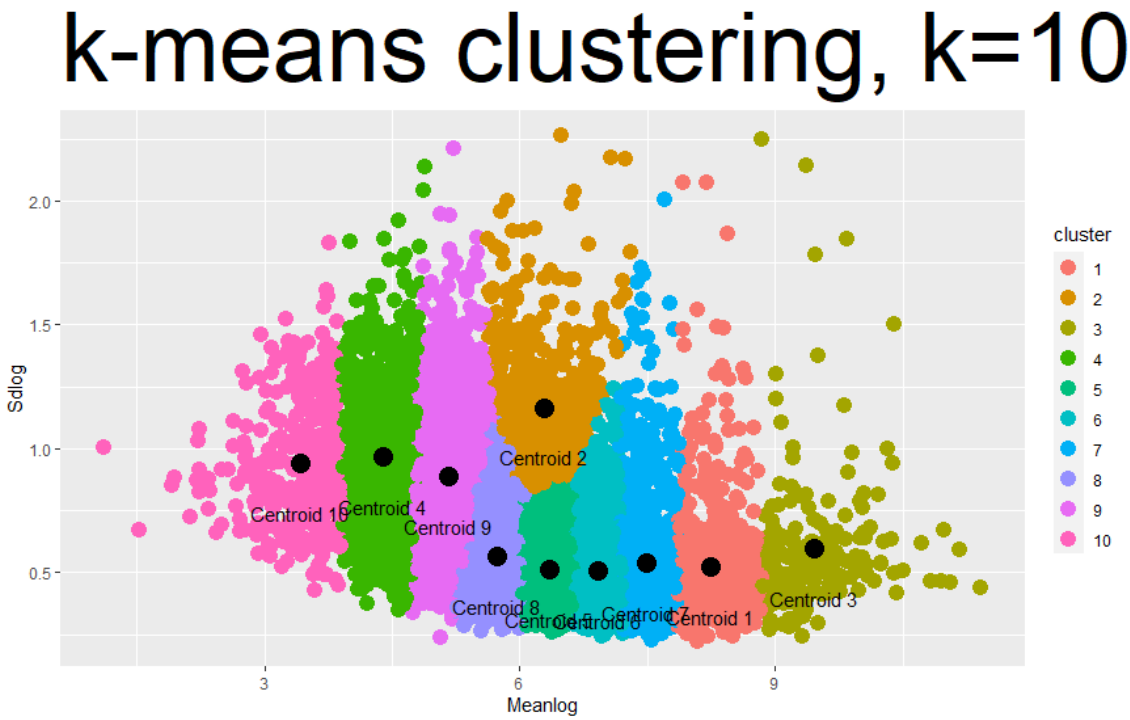

Figura 19: Plot del risultato dell'applicazione dell'algoritmo K-means con K = 10.

Conclusioni

Dopo aver applicato i vari algoritmi proposti a entrambi i dataset è stata riscontrata un'alta efficacia del k-means sui dati proposti, si evincono dei cluster distinti, risultato non ottenuto dagli altri due algoritmi utilizzati.