

K-MEANS

Membri del gruppo:

-Belvito Ivan.

-Furetto Matteo.

Indice

1. Introduzione al clustering e al k-means

2. Realizzazioni

- a. Sorgente del sistema
- b. Estensione

3. Installazione e utilizzo

- a. Server
- b. Client

4. Test del software

- a. Client CLI
- b. Client GUI

1. INTRODUZIONE

Il clustering è una tecnica di analisi dei dati utilizzata per raggruppare insieme dati simili in un insieme di cluster o gruppi, in modo che gli oggetti all'interno di un cluster siano più simili tra loro rispetto a quelli in cluster diversi. Si definiscono i dati come una collezione D di transazioni dove, ogni transazione è un vettore di coppie attributo-valore (detto item) e un valore intero k ; Lo scopo è partizionare D in k insiemi di transazioni D_1, \dots, D_k , tale che:

- D_i ($i=1, \dots, k$) è un segmento (selezione) omogenea di D ;
- $D = \bigcup D_i$ and $D_i \cap D_j = \emptyset$.

Per svolgere il clustering il programma fa uso dell'algoritmo K-means con il quale i dati vengono analizzati a partire da delle tabelle che li memorizzano su un database. Il K-means è uno degli algoritmi di clustering più utilizzati; si basa su una tecnica di apprendimento non supervisionata con la quale i dati in possesso verranno suddivisi in gruppi omogenei detti cluster. L'obiettivo è di assegnare ogni transazione al cluster più vicino, in modo che i punti all'interno dei cluster siano omogenei tra loro e i punti tra cluster diversi siano diversi. Il suo funzionamento può essere diviso in 5 passi:

PASSO 1: si inizia scegliendo un numero k di cluster. L'algoritmo crea k partizioni e assegna casualmente k punti come centroidi (un centroide rappresenta il centro di un cluster) iniziali, uno per partizione.

PASSO 2: ogni transazione viene assegnata al cluster il cui centroide è più vicino in base alla distanza (solitamente la distanza euclidea) tra la transazione stessa e i centroidi dei cluster.

PASSO 3: dopo aver assegnato tutti i punti ai cluster, i centroidi vengono ricalcolati come la media delle posizioni dei punti appartenenti a ciascun cluster.

PASSO 4: i passaggi 2-3 dell'algoritmo K-Means vengono ripetuti fino a quando uno dei seguenti criteri viene soddisfatto:

- Si raggiunge un numero massimo di iterazioni prestabilite. L'algoritmo eseguirà un numero fissato di iterazioni, indipendentemente da ciò che accade durante queste iterazioni.
- Non ci sono più transazioni che cambiano cluster di appartenenza. Questo significa che durante un'iterazione, se tutti i punti mantengono lo stesso cluster a cui sono stati assegnati durante l'iterazione precedente, l'algoritmo smette di iterare.

PASSO 5: visione dei k cluster in cui ogni transazione è assegnata al cluster più vicino.

2. REALIZZAZIONI

2.a Sorgente del sistema

La versione 'base' del progetto è formata da un architettura client-server che consente ad un utente di effettuare il clustering. La parte del server deve essere eseguita su un calcolatore che possiede un database MySQL in esecuzione. Il server si troverà sulla porta 8080 e sarà in grado di gestire comunicazioni con client diversi contemporaneamente. Il Client in questo caso comunica con il server utilizzando un menu mostrato tramite CLI, potendo scegliere di eseguire 2 servizi:

-lettura di cluster (digitando 1): il client invia al Server il nome del file in cui sono serializzati i cluster da recuperare;

-scoperta di cluster (digitando 2): in questo caso il client invia al server il numero di cluster da scoprire, il nome della tabella di database, il nome del file in cui serializzare (memorizzare) i cluster scoperti.

In entrambe le attività il client acquisisce il risultato trasmesso dal server e lo visualizza a video tramite CLI. Se durante le operazioni specificate si verificano errori (sul server o a causa di un input errato del client) vengono visualizzati tramite CLI

2.b Estensione

L'estensione del progetto prevede la realizzazione di un'interfaccia grafica lato client. L'interfaccia utente (ClientGUI) consente all'utente di interagire con l'applicazione client-server. Tramite questa interfaccia il client può:

- **caricare dati da file:** l'utente può premere il pulsante "Carica da File" per caricare dati da un file esistente nel server. Dopo aver cliccato sul pulsante, l'utente viene invitato a inserire il nome della tabella contenente i dati da caricare. L'utente specifica il numero di cluster desiderato. Una volta forniti i dati, l'applicazione invia una richiesta al server per visualizzare i cluster memorizzati su questo file. Tali cluster vengono visualizzati nell'area di testo dell'applicazione.

-**scoprire cluster:** l'utente può premere il pulsante "Carica da Database" per caricare dati da un database esistente. L'applicazione richiede all'utente il numero di cluster desiderato per il clustering e il nome della tabella da utilizzare. Successivamente, l'applicazione invia una richiesta al server per eseguire il clustering KMeans sui dati del database. I risultati del clustering vengono quindi visualizzati nell'area di testo dell'applicazione.

(Prima di inviare richieste al server, l'applicazione effettua una validazione dell'input per garantire che l'utente abbia inserito informazioni corrette e complete. Ad esempio, verifica se sono stati forniti nomi di tabelle o numeri di cluster validi. L'utente inoltre ha sempre la possibilità di annullare in un qualsiasi momento l'operazione precedentemente richiesta tornando alla schermata iniziale).

-**visualizzare informazioni:** l'utente può premere sul pulsante "help" per ottenere informazioni sull'uso dell'applicazione e sulle opzioni disponibili.

-**visualizzazione i risultati:** tutti i risultati delle operazioni, compresi i messaggi di errore, vengono visualizzati nell'area di testo dell'applicazione.

-**chiudere l'applicazione:** l'utente può chiudere l'applicazione quando ha terminato l'utilizzo.

3. INSTALLAZIONE E UTILIZZO

3.a Server

Per utilizzare il server è necessario:

- installare MySQL sul sistema;

- installare il JDK versione 19.0.2

- eseguire lo script sql 'createDB.sql' sul terminale sql per creare il database e la tabella da utilizzare per l'applicazione. Lo script si trova in `Kmeans-Base\KmeansServer\out\production` (o rispettivamente per la versione con estensione si trova in `Kmeans-estensione\KmeansServer\out\production`).

Sia nella versione base che in quella con l'estensione, il server viene utilizzato allo stesso modo. Per l'avvio del server è necessario eseguire il file ***server.bat*** presente nella cartella ***Kmeans-Base\KmeansServer*** (rispettivamente ***Kmeans-estensione\KmeansServer*** per la versione con estensione). In alternativa è possibile avviarlo da riga di comando tramite il comando ***java -jar server.jar*** eseguito nella cartella dove si trova il *jar* stesso (occorre spostarsi nella directory ***kmeansServer*** ed utilizzare il comando). E' necessario avviare il server per poter utilizzare l'applicazione.

3.b Client

Per utilizzare il client (sia nella versione base che in quella con l'estensione) è necessario che il server sia in ascolto. Nel caso della versione base, se l'utilizzo viene fatto eseguendo il file ***clientBase.bat*** posto nella cartella ***Kmeans-base\kmeansClientBase\out\artifacts\kmeansClientBase_jar***, non è necessaria nessun'altra operazione. Altrimenti se l'esecuzione avviene da riga di comando è necessario spostarsi nella directory e utilizzare il comando ***java -jar kmeansclientBase.jar <indirizzo IP> <porta>*** dove in questo caso (essendo il server collocato in locale) i valori da assegnare sono 127.0.0.1 e 8080.

Nel caso della versione con estensione, è necessario o eseguire il file ***Client.bat*** posto nella cartella ***kmeansestensione\kmeansClient\out\artifacts\kmeansClient_jar*** (senza ulteriori operazioni); altrimenti da riga di comando è necessario spostarsi nella directory corretta e successivamente bisogna utilizzare lo stesso comando della versione base: ***java -jar kmeansClient.jar <indirizzo IP> <porta>*** (assegnando i valori 127.0.0.1 e 8080).

4. TEST DEL SOFTWARE

4.a Test del client base

1. Esecuzione del client con il server non attivo:

```
C:\Users\Lenovo Gaming\Desktop\intellij-work\kmeansClientBase\out\artifacts\kmeansClientBase_jar>
java -jar kmeansClientBase.jar 127.0.0.1 8080
addr = /127.0.0.1
java.net.ConnectException: Connection refused: connect
```

2. Esecuzione del client tramite jar senza utilizzare <indirizzo IP> e <porta>:

```
C:\Users\Lenovo Gaming\Desktop\intellij-work\kmeansClientBase\out\artifacts\kmeansClientBase_jar>
java -jar kmeansClientBase.jar
Inserire indirizzo IP e numero di porta.
```

3. Esecuzione con indirizzo ip errato:

```
C:\Users\Lenovo Gaming\Desktop\intellij-work\kmeansClientBase\out\artifacts\kmeansClientBase_jar>
java -jar kmeansClientBase.jar 127.121.4.6 8080
addr = /127.121.4.6
java.net.ConnectException: Connection refused: connect
```

4. Esecuzione con porta errata:

```
C:\Users\Lenovo Gaming\Desktop\intellij-work\kmeansClientBase\out\artifacts\kmeansClientBase_jar>
java -jar kmeansClientBase.jar 127.0.0.1 jofab
Il numero di porta non è valido.
```

5. Esecuzione con parametri corretti: indirizzo ip 127.0.0.1 e porta 8080 (in questo caso il server è in ascolto). Viene mostrato all'utente un menù con due opzioni: (1) permette al client di caricare dei dati serializzati sul server come un file, mentre l'opzione (2) consente di creare un nuovo cluster di dati mediante l'algoritmo K-means. In caso di input non valido viene chiesto all'utente di reinserirlo:

```
C:\Users\Lenovo Gaming\Desktop\intellij-work\kmeansClientBase\out\artifacts\kmeansClientBase_jar>
java -jar kmeansClientBase.jar 127.0.0.1 8080
addr = /127.0.0.1
Socket[addr=/127.0.0.1,port=8080,localport=63233]

Scegli una opzione:
(1) Carica Cluster da File.
(2) Carica Dati.

Inserisci una risposta valida (1/2):
```

```
Scegli una opzione:
(1) Carica Cluster da File.
(2) Carica Dati.

Inserisci una risposta valida (1/2): 5
Input non valido. Riprova.

Inserisci una risposta valida (1/2): 1
Input non valido. Riprova.

Inserisci una risposta valida (1/2):
```

Test delle opzioni

Opzione 1: viene chiesto all'utente di inserire il nome della tabella e il numero di cluster. Il server ricerca un file il cui nome è del tipo "nomeTabella" + "numerocluster" + "k" (dove k è aggiunto dal server) dove sono serializzati dei cluster. Se il file esiste, essi vengono mostrati al client, altrimenti visualizza un errore.

-Caso in cui viene inserito il nome errato della tabella:

```
Inserisci una risposta valida (1/2): 1
Ricerca un file inserendo...
Nome tabella: play
Numero cluster: 12
File non trovato.
Vuoi scegliere una nuova operazione da menu?(y/n) y

Scegli una opzione:
(1) Carica Cluster da File.
(2) Carica Dati.

Inserisci una risposta valida (1/2): 1
Ricerca un file inserendo...
Nome tabella: 0
Numero cluster: 5
File non trovato.
```

-Caso in cui viene inserito un numero di cluster errato:

```
Inserisci una risposta valida (1/2): 1
Ricerca un file inserendo...
Nome tabella: playtennis
Numero cluster: 16
File non trovato.
Vuoi scegliere una nuova operazione da menu?(y/n) y

Scegli una opzione:
(1) Carica Cluster da File.
(2) Carica Dati.

Inserisci una risposta valida (1/2): 1
Ricerca un file inserendo...
Nome tabella: playtennis
Numero cluster: 1
File non trovato.
```

-Caso in cui l'utente inserisce dati corretti: nome di tabella "playtennis" e numero di cluster compreso tra 1 e 14 (estremi inclusi).


```

Inserisci una risposta valida (1/2): 1
Ricerca un file inserendo...
Nome tabella: playtennis
Numero cluster: 3
0:Centroid=(sunny 21.525 high strong no )
Examples:
[sunny 30.3 high weak no ] dist=1.2896039603960396
[sunny 30.3 high strong no ] dist=0.28960396039603964
[rain 12.5 high strong no ] dist=1.2978547854785478
[sunny 13.0 high weak no ] dist=1.2813531353135312
AvgDistance=1.0396039603960396

1:Centroid=(rain 9.051666666666668 normal weak yes )
Examples:
[sunny 0.1 normal weak yes ] dist=1.2954345434543455
[rain 12.0 normal weak yes ] dist=0.09730473047304722
[rain 13.0 high weak yes ] dist=1.1303080308030802
[rain 0.0 normal weak yes ] dist=0.2987348734873488
[overcast 29.21 normal weak yes ] dist=1.6652915291529153
[rain 0.0 normal strong no ] dist=2.2987348734873487
AvgDistance=1.130968096809681

2:Centroid=(overcast 13.775 high strong yes )
Examples:
[overcast 30.0 high weak yes ] dist=1.5354785478547854
[sunny 12.5 normal strong yes ] dist=2.042079207920792
[overcast 12.5 high strong yes ] dist=0.04207920792079206
[overcast 0.1 normal strong yes ] dist=1.4513201320132012
AvgDistance=1.2677392739273925

```

Quando l'operazione termina (sia se il file viene trovato o meno) l'utente può sceglierne un'altra o uscire:

```

Inserisci una risposta valida (1/2): 1
Ricerca un file inserendo...
Nome tabella: play
Numero cluster: 7
File non trovato.
Vuoi scegliere una nuova operazione da menu?(y/n) y

Scegli una opzione:
(1) Carica Cluster da File.
(2) Carica Dati.

Inserisci una risposta valida (1/2):

```

Opzione 2: viene chiesto all'utente di inserire il nome della tabella e il numero di cluster. In questo caso il server si connette al database "MapDB" in cui si trova la tabella "playtennis", scopre dei nuovi cluster mediante l'algoritmo k-means (se ad esempio l'utente inserisce 5 al numero di cluster, verranno generati 5 cluster), li fa visualizzare all'utente e li serializza in un file (il cui nome è "playtennis" + "numerocluster" + "k") sostituendo salvati precedentemente.

- Caso in cui viene inserito il nome errato della tabella:

```
Inserisci una risposta valida (1/2): 2
Nome tabella: play
Tabella non trovata.
Nome tabella:
```

-Caso in cui viene inserito un numero di cluster errato:

```
Inserisci una risposta valida (1/2): 2
Nome tabella: playtennis
Numero di cluster: 15
Non ci sono centroidi a sufficienza poichè il numero di tuple distinte e': 14.
Vuoi ripetere l'esecuzione? (y/n) y
Numero di cluster: 1
Numero cluster non valido, inserisci un numero positivo.
Vuoi ripetere l'esecuzione? (y/n)
```

-Caso in cui l'utente inserisce dati corretti: nome di tabella "playtennis" e numero di cluster compreso tra 1 e 14 (estremi inclusi).

```
Inserisci una risposta valida (1/2): 2
Nome tabella: playtennis
Numero di cluster: 3
Numero di iterazioni: 2
0:Centroid=(rain 16.516666666666666 high strong no )
Examples:
[sunny 30.3 high weak no ] dist=2.4548954895489548
[sunny 30.3 high strong no ] dist=1.454895489548955
[rain 13.0 high weak yes ] dist=2.116061606160616
[rain 0.0 normal strong no ] dist=1.545104510451045
[sunny 13.0 high weak no ] dist=2.116061606160616
[rain 12.5 high strong no ] dist=0.13256325632563248
AvgDistance=1.6365969930326365

1:Centroid=(overcast 8.985 normal weak yes )
Examples:
[rain 0.0 normal weak yes ] dist=1.2965346534653466
[overcast 0.1 normal strong yes ] dist=1.2932343234323431
[sunny 0.1 normal weak yes ] dist=1.2932343234323431
[rain 12.0 normal weak yes ] dist=1.0995049504950494
[sunny 12.5 normal strong yes ] dist=2.1160606060606066
[overcast 29.21 normal weak yes ] dist=0.6674917491749175
AvgDistance=1.2943344334433444

2:Centroid=(overcast 21.25 high strong yes )
Examples:
[overcast 30.0 high weak yes ] dist=1.2887788778877889
[overcast 12.5 high strong yes ] dist=0.2887788778877888
AvgDistance=0.7887788778877889

Vuoi ripetere l'esecuzione? (y/n)
```

Quando l'operazione termina l'utente può scegliere se ripetere l'esecuzione con un numero di cluster diverso, effettuare un'altra operazione o uscire.


```

Inserisci una risposta valida (1/2): 2
Nome tabella: playtennis
Numero di cluster: 1
Numero di iterazioni: 2
0:Centroid=(rain 13.965 high weak yes )
Examples:
[sunny 30.3 high weak no ] dist=2.5391089108910894
[sunny 30.3 high strong no ] dist=3.5391089108910894
[overcast 30.0 high weak yes ] dist=1.5292079207920792
[rain 13.0 high weak yes ] dist=0.03184818481848184
[rain 0.0 normal weak yes ] dist=1.4608910891089109
[rain 0.0 normal strong no ] dist=3.4608910891089106
[overcast 0.1 normal strong yes ] dist=3.4575907590759076
[sunny 13.0 high weak no ] dist=2.031848184818482
[sunny 0.1 normal weak yes ] dist=2.4575907590759076
[rain 12.0 normal weak yes ] dist=1.064851485148515
[sunny 12.5 normal strong yes ] dist=3.0483498349834983
[overcast 12.5 high strong yes ] dist=2.0483498349834983
[overcast 29.21 normal weak yes ] dist=2.503135313531353
[rain 12.5 high strong no ] dist=2.0483498349834983
AvgDistance=2.23008015087223

Vuoi ripetere l'esecuzione? (y/n) n
Vuoi scegliere una nuova operazione da menu?(y/n) y

Scegli una opzione:
(1) Carica Cluster da File.
(2) Carica Dati.

Inserisci una risposta valida (1/2):

```

4.b Test del client con estensione

1. Esecuzione del client con il server non attivo:

```

C:\Users\Lenovo Gaming\Desktop\intellij-work\kmeansClient\out\artifacts\kmeansClient_jar>
java -jar KmeansClient.jar 127.0.0.1 8080
java.net.ConnectException: Connection refused: connect

```

2. Esecuzione del client tramite jar senza utilizzare <indirizzo IP> e <porta>:

```

C:\Users\Lenovo Gaming\Desktop\intellij-work\kmeansClient\out\artifacts\kmeansClient_jar
java -jar KmeansClient.jar
Inserire indirizzo IP e numero di porta.

```

3. Esecuzione con indirizzo ip errato:

```

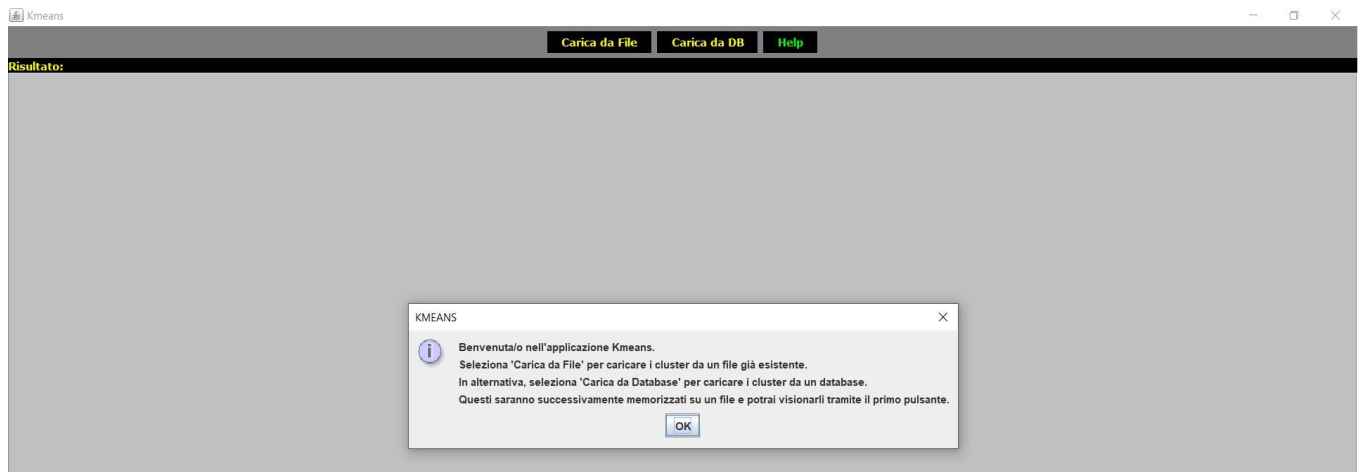
C:\Users\Lenovo Gaming\Desktop\intellij-work\kmeansClient\out\artifacts\kmeansClient_jar
>java -jar KmeansClient.jar 127.6.45.7 8080
java.net.ConnectException: Connection refused: connect

```

4. Esecuzione con porta errata:

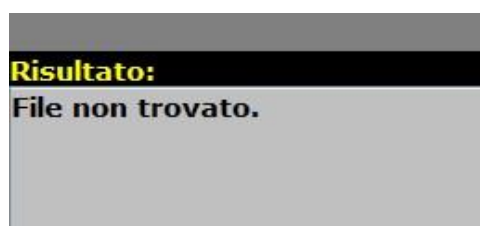
```
C:\Users\Lenovo Gaming\Desktop\intellij-work\kmeansClient\out\artifacts\kmeansClient_jar
>java -jar KmeansClient.jar 127.0.0.1 toyota
Il numero di porta non è valido.
```

5. Esecuzione con parametri corretti: indirizzo ip 127.0.0.1 e porta 8080 (in questo caso il server è in ascolto). Viene mostrato all'utente un primo messaggio di benvenuto. Successivamente, l'utente può interagire con l'applicazione tramite dei pulsanti: "Carica da file" permette al client di caricare dei dati serializzati sul server come un file, mentre "Carica da db" consente di creare un nuovo cluster di dati mediante l'algoritmo K-means. Il pulsante "help" consente di visualizzare informazioni sul funzionamento dell'applicazione.



Test opzione 1:

- **Opzione 1:** l'utente può premere il pulsante "Carica da File" per caricare dati da un file esistente nel server. Dopo aver cliccato sul pulsante, l'utente viene invitato a inserire il nome della tabella contenente i dati da caricare. L'utente specifica il numero di cluster desiderato. Una volta forniti i dati, l'applicazione invia una richiesta al server per visualizzare i cluster memorizzati su questo file. Tali cluster vengono visualizzati nell'area di testo dell'applicazione. L'utente può decidere in un qualsiasi momento di annullare l'operazione per uscire o effettuarne un'altra.
- Caso in cui viene inserito il nome errato della tabella:



- Caso in cui viene inserito un numero di cluster errato:



Input

Nome tabella:

playtennis

OK Cancel



Input

Numero cluster:

16

OK Cancel



Input

Nome tabella:

playtennis

OK Cancel

-Caso in cui l'utente inserisce dati corretti: nome di tabella "playtennis" e numero di cluster compreso tra 1 e 14 (estremi inclusi).



Input

Nome tabella:

playtennis

OK Cancel



Input

Numero cluster:

5

OK Cancel

Risultato:

0:Centroid=(rain 13.95 high strong no)

Examples:

[sunny 30.3 high strong no] dist=1.5396039603960396

[rain 0.0 normal strong no] dist=1.4603960396039604

[rain 12.5 high strong no] dist=0.04785478547854782

[sunny 13.0 high weak no] dist=2.0313531353135312

AvgDistance=1.2698019801980198

1:Centroid=(overcast 29.836666666666662 high weak yes)

Examples:

[sunny 30.3 high weak no] dist=2.0152915291529157

[overcast 29.21 normal weak yes] dist=1.0206820682068205

[overcast 30.0 high weak yes] dist=0.005390539053905519

AvgDistance=1.013788045471214

2:Centroid=(rain 6.275 normal weak yes)

Examples:

[sunny 0.1 normal weak yes] dist=1.203795379537954

[rain 12.0 normal weak yes] dist=0.1889438943894389

[rain 13.0 high weak yes] dist=1.221947194719472

[rain 0.0 normal weak yes] dist=0.2070957095709571

AvgDistance=0.7054455445544554

3:Centroid=(overcast 12.5 high strong yes)

Examples:

[overcast 12.5 high strong yes] dist=0.0

AvgDistance=0.0

4:Centroid=(overcast 6.3 normal strong yes)

Examples:

[overcast 0.1 normal strong yes] dist=0.2046204620462046

[sunny 12.5 normal strong yes] dist=1.2046204620462047

AvgDistance=0.7046204620462047

-**Opzione 2:** l'utente può premere il pulsante "Carica da Database" per caricare dati da un database esistente. L'applicazione richiede all'utente il numero di cluster desiderato per il clustering e il nome della tabella da utilizzare. Successivamente, l'applicazione invia una richiesta al server per eseguire il clustering

KMeans sui dati del database. I risultati del clustering vengono quindi visualizzati nell'area di testo dell'applicazione. L'utente può decidere in un qualsiasi momento di annullare l'operazione per uscire o effettuarne un'altra.

- Caso in cui viene inserito il nome errato della tabella:



The image shows a small dialog box titled "Input" with a close button (X) in the top right corner. On the left is a green square icon with a white question mark. To its right is the label "Nome tabella:" followed by a text input field containing the word "play". Below the input field are two buttons: "OK" and "Cancel".

Risultato:
Tabella non trovata.

-Caso in cui viene inserito un numero di cluster errato:



The image shows a small dialog box titled "Input" with a close button (X) in the top right corner. On the left is a green square icon with a white question mark. To its right is the label "Nome tabella:" followed by a text input field containing the word "playtennis". Below the input field are two buttons: "OK" and "Cancel".



The image shows a small dialog box titled "Input" with a close button (X) in the top right corner. On the left is a green square icon with a white question mark. To its right is the label "Numero cluster:" followed by a text input field containing the number "18". Below the input field are two buttons: "OK" and "Cancel".

Risultato:
Non ci sono centroidi a sufficienza poichè il numero di tuple distinte e': 14.

-Caso in cui l'utente inserisce dati corretti: nome di tabella "playtennis" e numero di cluster compreso tra 1 e 14 (estremi inclusi).

Input

×

?

Nome tabella:

playtennis

OK

Cancel

Input

×

?

Numero cluster:

5

OK

Cancel

Risultato:

Numero di iterazioni: 3

0:Centroid=(sunny 4.23333333333333 normal strong yes)

Examples:

[overcast 0.1 normal strong yes] dist=1.1364136413641364

[sunny 0.1 normal weak yes] dist=1.1364136413641364

[sunny 12.5 normal strong yes] dist=0.27282728272827284

AvgDistance=0.8485515218188485

1:Centroid=(overcast 21.25 high strong yes)

Examples:

[overcast 30.0 high weak yes] dist=1.2887788778877889

[overcast 12.5 high strong yes] dist=0.2887788778877888

AvgDistance=0.7887788778877889

2:Centroid=(sunny 24.53333333333333 high weak no)

Examples:

[sunny 30.3 high weak no] dist=0.19031903190319044

[sunny 30.3 high strong no] dist=1.1903190319031904

[sunny 13.0 high weak no] dist=0.38063806380638054

AvgDistance=0.5870920425375871

3:Centroid=(rain 6.25 high strong no)

Examples:

[rain 0.0 normal strong no] dist=1.2062706270627062

[rain 12.5 high strong no] dist=0.20627062706270627

AvgDistance=0.7062706270627062

4:Centroid=(rain 13.5525 normal weak yes)

Examples:

[rain 13.0 high weak yes] dist=1.0182343234323432

[rain 0.0 normal weak yes] dist=0.4472772277227723

[rain 12.0 normal weak yes] dist=0.05123762376237628

[overcast 29.21 normal weak yes] dist=1.5167491749174917

AvgDistance=0.758374587458746