

Progetto Kmeans (2023/2024)

Indice

- [Team progetto](#)
- [Descrizione Kmeans](#)
- [Progetto base](#)
- [Progetto esteso](#)
- [Guida di installazione](#)
 - [Installazione Server](#)
 - [Installazione Client \(CLI e GUI\)](#)
- [Guida utente](#)
 - [Avvio del Server \(progetto base\)](#)
 - [Avvio del Client \(progetto base\)](#)
 - [Avvio del Server \(progetto esteso\)](#)
 - [Avvio del Client \(progetto esteso\)](#)
- [Guida di utilizzo](#)
 - [Utilizzo del Client \(progetto base\)](#)
 - [Utilizzo del Client \(progetto esteso\)](#)
 - [Schermata Credenziali Server](#)
 - [Schermata Menu](#)
 - [Schermata risultato lettura cluster da file](#)
 - [Schermata risultato scoperta cluster](#)
- [Casi di Test](#)

Team progetto

Nome	Cognome	Username GitHub
Leonardo	Colucci	xDrokra
Gianfranco	De Vincenzo	GianfrancoDev18

Descrizione Kmeans

Il K-means è un algoritmo di **clustering**, una tecnica di apprendimento non supervisionato utilizzata per suddividere un insieme di dati in gruppi omogenei chiamati cluster. L'obiettivo del K-means è di assegnare ogni punto dati al **cluster più vicino**, in modo che i punti all'interno di ciascun cluster siano simili tra loro e i punti tra cluster diversi siano diversi. Ecco come funziona l'algoritmo K-means:

- **Inizializzazione:** Si inizia scegliendo il numero desiderato di cluster, chiamato K, e si selezionano casualmente K punti come centroidi iniziali. Un centroide rappresenta il centro del cluster.
- **Assegnazione:** Per ogni punto dati, viene calcolata la distanza tra il punto e i centroidi. Il punto viene assegnato al cluster rappresentato dal centroide più vicino in base alla distanza.
- **Aggiornamento dei centroidi:** Una volta assegnati tutti i punti ai cluster, i centroidi vengono aggiornati calcolando la media delle posizioni dei punti all'interno di ciascun cluster. Questa media diventa il nuovo centroide per il cluster corrispondente.
- **Ripetizione:** I passi 2 e 3 vengono ripetuti fino a quando i centroidi smettono di cambiare o si raggiunge un numero massimo di iterazioni. In generale, l'algoritmo converge verso una soluzione stabile, anche se la soluzione ottenuta può essere un minimo locale invece del minimo globale.
- **Risultato:** Alla fine delle iterazioni, si ottiene un insieme di K centroidi e i punti dati assegnati a ciascun cluster. Questi cluster possono essere utilizzati per analizzare i dati, identificare pattern o raggruppare elementi simili insieme

Progetto base

La versione base del progetto consiste in un'architettura *client/server* che permette all'utente la scoperta/lettura di cluster tramite le varie opzioni offerte nel menu'.

Il **server** dovrà essere eseguito su una macchina con un database **MySQL** in esecuzione.

Il servizio sarà raggiungibile sulla porta *8080*, e potrà comunicare con diversi client contemporaneamente.

I servizi offerti dal server (*offerti all'utente tramite il client CLI*) sono i seguenti:

- lettura di cluster fornendo al server il path del file in cui sono serializzati i cluster da recuperare
- scoperta di cluster fornendo al server il nome della tabella presente nel database ed il numero di cluster da scoprire

- salvare i centroidi generati dalla scoperta nella macchina dove il server viene eseguito, dove potranno essere successivamente letti.

Il server effettuerà dei log su console di quanto accade durante la sua esecuzione (ad esempio, connessione di un client o errore durante la creazione di un albero di regressione).

I messaggi di errore rilevanti per l'utente verranno inviati al client.

Il client da riga di comando permette di collegarsi ad una macchina che sta eseguendo un'istanza del server, l'indirizzo IP e la porta sono settati di default. L'utente nel menu' avrà le due opzioni di lettura o scoperta di cluster

Progetto esteso

L'estensione per il progetto consiste in un'interfaccia grafica per desktop. Tale interfaccia e' stata sviluppata tramite il framework **Swing**

Guida di installazione

I file `.jar`, `.bat` e dove necessario `.sql` sono posti nella cartella **out\artifacts\eseguibili** di *ogni progetto*.

Mentre i **JavaDocs** si trovano all'interno della cartella **out\artifacts\javadocs**

Installazione Server

Per *eseguire correttamente* il server sulla propria macchina e' necessario:

- aver installato la *Java Runtime Environment 8 o superiore*
- aver installato *MySQL*
- eseguire il file `script.sql` per creare l'utente MySQL, il *Database* e la *tabella* necessaria

Installazione Client (CLI e GUI)

Per eseguire correttamente il client (sia testuale che grafico) e' necessario:

- aver installato la *Java Runtime Environment 8 o superiore*
- avere un *server in ascolto*

NB: Il server in ascolto deve essere lo stesso presentato con il client utilizzato, in quanto non e' possibile, ad esempio, utilizzare il server presentato nel progetto base con il client grafico. Inoltre, anche i file coi cluster memorizzati da caricare devono essere stati creati dalla stessa versione del server che deve utilizzarli.

Guida utente

Avvio del Server (progetto base)

Per avviare il server e' necessario eseguire il file `server.bat` nella stessa cartella di `server.jar` . In alternativa e' possibile avviarlo da riga di comando tramite il comando `java -jar server.jar`

Avvio del Client (progetto base)

Per avviare il Client da riga di comando, è necessario eseguire il file `client.bat` nella stessa cartella del file `client.jar` . Questa modalità di avvio conatterà il Client ad un server in esecuzione sulla propria macchina, sulla porta 8080. Per specificare un altro server a cui connettersi, è necessario avviare il Client da una shell tramite il comando `java -jar client.jar [INDIRIZZO_IP] [PORTA]` o inserendo tale comando nel file batch di avvio.

Avvio del Server (progetto esteso)

Simile al progetto base, il server del progetto esteso e' eseguibile avviando il file `server.bat` nella stessa cartella di `server.jar` . In alternativa e' possibile avviarlo da riga di comando tramite il comando `java -jar server.jar` . La porta di default sara' *8080*

Avvio del Client (progetto esteso)

Per avviare il client grafico è necessario avviare il file `clientWithArgs.bat` nella stessa cartella di `client.jar`. Questa modalità di avvio conatterà il Client ad un server in esecuzione sulla propria macchina, sulla porta 8080. Per specificare un altro server a cui connettersi, è necessario avviare l'altro file `clientWithoutArgs.bat` che chiederà direttamente al client di inserire la porta e l'indirizzo ip del server.

Guida di utilizzo

Utilizzo del Client (progetto base)

Una volta avviato il Client, se la connessione con il server va a buon fine si specifica quale operazione si vuole effettuare.

```
Socket[addr=localhost/127.0.0.1,port=8080,localport=55992]
Scegli una opzione
(1) Carica Cluster da File
(2) Carica Dati
Risposta: |
```

- Se scelta l'opzione **1** verrà richiesto il path del file, banalmente il nome del path sarà il nome della tabella all'interno del db, più il **k** che indica il numero di centroidi. Successivamente si può ritornare al menu per effettuare altre operazioni.

```
Scegli una opzione
(1) Carica Cluster da File
(2) Carica Dati
Risposta: 1
Nome tabella: playtennis
Numero iterate: 2
Centroid=(rain, 10.1125, high, weak, no)
Centroid=(overcast, 19.10166666666667, high, strong, yes)
Vuoi scegliere una nuova operazione da menu? (y/n):
```

- Se scelta l'opzione **2** verrà richiesto il nome della tabella da trovare nel db e il numero di cluster in modo tale per effettuare l'algoritmo del Kmeans. Successivamente se tutto va buon fine i centroidi generati verranno salvati automaticamente.

```
Scegli una opzione
(1) Carica Cluster da File
(2) Carica Dati
Risposta: 2
Nome tabella: playtennis
Numero di cluster: 2
```

- Una volta finita l'operazione verra' richiesto all'utente di ripetere l'opzione **2** con un possibile numero differente di cluster. Altrimenti l'utente puo' ritornare alla schermata del menu' per effettuare altre operazioni.

```
Vuoi ripetere l'esecuzione? (y/n): y
Numero di cluster: |
```

Utilizzo del Client (progetto esteso)

Schermata Credenziali Server

Se il client viene avviato senza argomenti verra' chiesto direttamente all'utente di inserirli, oppure puo' premere il pulsante di default che impostera' la porta ed l'indirizzo ip ad un valore di default

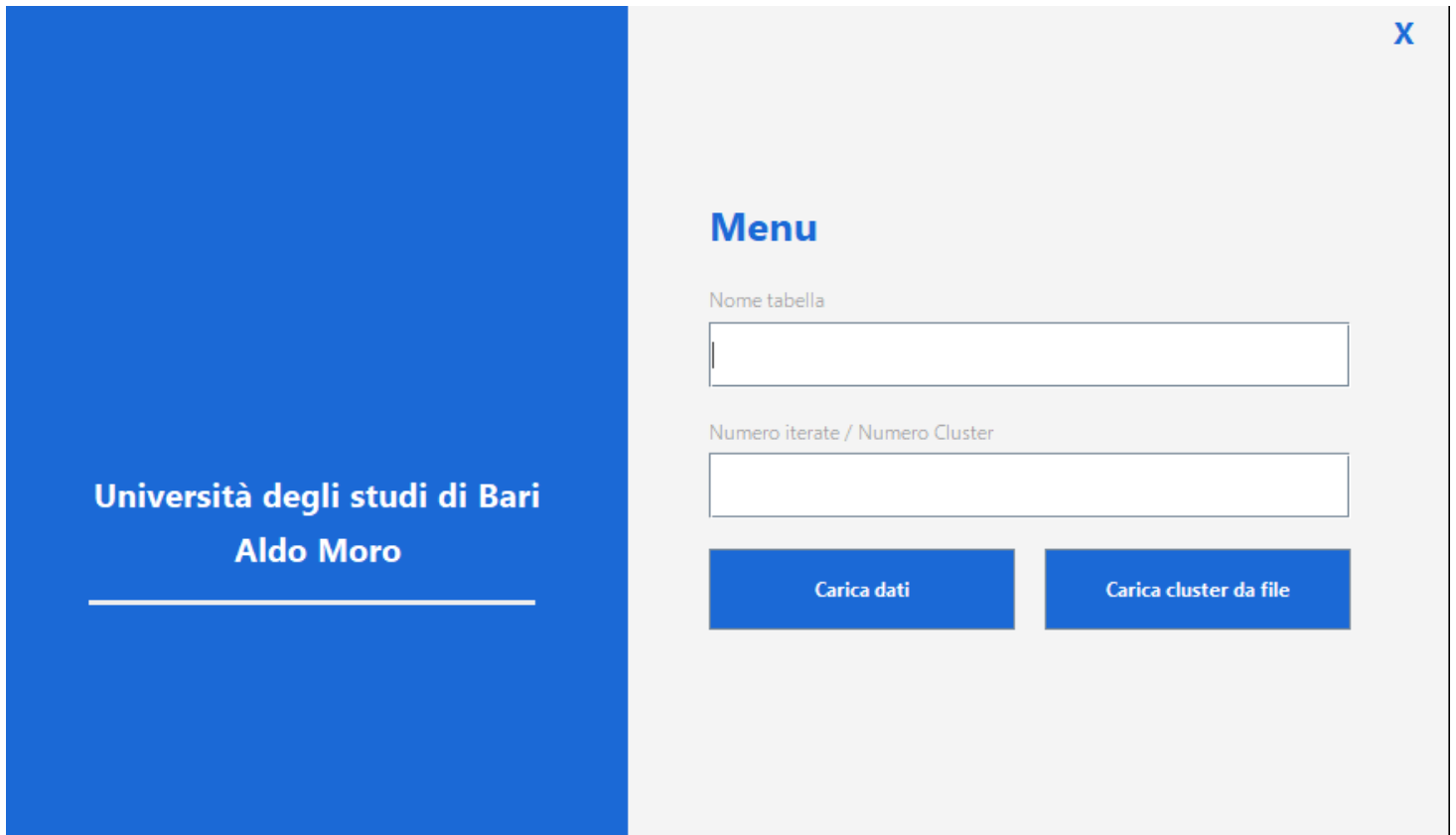


The screenshot shows a window titled "Credenziali Server" with a close button (X) in the top right corner. The window has a light gray background and blue borders on the left and right sides. Inside the window, the title "Credenziali Server" is displayed in blue text. Below the title, there are two input fields: "Indirizzo Ip" and "Porta". The "Indirizzo Ip" field is currently empty, and the "Porta" field is also empty. At the bottom of the window, there are two blue buttons: "Collegati" and "Default".

Schermata Menu

Una volta avviato il Client grafico, ci si trovera' sulla schermata del menu da quale possiamo scegliere le due opzioni che vi erano precedentemente nel client del progetto. Nei due input text viene aggiunto

il nome della tabella e il k. Questi due input text serviranno per svolgere le due operazioni citate prima di lettura/scoperta di cluster.



Università degli studi di Bari
Aldo Moro

Menu

Nome tabella

Numero iterate / Numero Cluster

Carica dati

Carica cluster da file

Schermata risultato lettura cluster da file

In caso di scelta di **"lettura da cluster da file"** che sarebbe la precedente **1** opzione si aprirà un finestra che indica i centroidi salvati sul file specificato.



Risultato

Centroid=(overcast, 8.30125, normal, strong, yes)
Centroid=(sunny, 21.516666666666666, high, weak, no)

[Torna indietro](#)

Schermata risultato scoperta cluster

In caso di scelta dell'opzione "**Carica dati**" che sarebbe la precedente **2** opzione si aprirà sempre una finestra che indicherà il risultato ottenuto applicando l'algoritmo e si potrà ripetere con un numero differente di cluster, specificato in basso a sinistra.

Risultato

```
[sunny 30.3 high strong no ] dist=3.7929042904290426
[overcast 30.0 high weak yes ] dist=2.7830033003300327
[rain 13.0 high weak yes ] dist=3.2219471947194718

AvgDistance=1.2054455445544554
2:Centroid=(rain 14.051666666666668 normal weak yes )
Examples:
[sunny 30.3 high weak no ] dist=3.5362486248624863
[sunny 30.3 high strong no ] dist=4.536248624862486
[overcast 30.0 high weak yes ] dist=2.5263476347634763
[rain 13.0 high weak yes ] dist=1.0347084708470848
[rain 0.0 normal weak yes ] dist=0.4637513751375138
[rain 0.0 normal strong no ] dist=2.4637513751375137

AvgDistance=1.1755408874220756
```

Ripeti con iterate

Torna indietro

Casi di Test

Scelta menu: 1

Output atteso: Nome tabella seguito da numero iterate

Output:

```
Socket[addr=localhost/127.0.0.1,port=8080,localport=58428]
Scegli una opzione
(1) Carica Cluster da File
(2) Carica Dati
Risposta: 1
Nome tabella: playtennis
Numero iterate: 2
```

Scelta menu: 3

Output atteso: Ristampa del menu

Output:

```
Socket[addr=localhost/127.0.0.1,port=8080,localport=58349]  
Scegli una opzione  
(1) Carica Cluster da File  
(2) Carica Dati  
Risposta: 3  
(1) Carica Cluster da File  
(2) Carica Dati  
Risposta:
```

Scelta menu: 1

Nome tabella: nome tabella esistente nel file

Numero iterate: numero iterate esistente nel file

Output atteso: centroidi nel path "nome tabella_numero Iterate"

Output:

```
Scegli una opzione  
(1) Carica Cluster da File  
(2) Carica Dati  
Risposta: 1  
Nome tabella: playtennis  
Numero iterate: 2  
Centroid=(sunny, 20.844285714285714, high, weak, yes)  
Centroid=(rain, 7.085714285714286, normal, strong, yes)
```

Scelta menu: 1

Nome tabella: nome tabella inesistente nel file

Numero iterate: numero iterate inesistente nel file

Output atteso: Errore nel path

Output:

```
Scegli una opzione  
(1) Carica Cluster da File  
(2) Carica Dati  
Risposta: 1  
Nome tabella: prova  
Numero iterate: 120  
java.rmi.ServerException:  
[Error]: .\clustersetsaved\prova_120 (Impossibile trovare il file specificato)
```

Scelta menu: 2

Nome tabella: nome tabella esistente nel db

Numero cluster: numero compreso tra > 0 e <= delle tuple distinte

Output atteso: algoritmo del kmeans effettuato correttamente con il numero di cluster inserito

Output:

```

Scegli una opzione
(1) Carica Cluster da File
(2) Carica Dati
Risposta: 2
Nome tabella: playtennis
Numero di cluster: 3
Clustering output:
Numero di iterazione: 3
0:Centroid=(overcast 8.985 normal weak yes )
Examples:
[sunny 30.3 high weak no ] dist=3.7034653465346534
[sunny 30.3 high strong no ] dist=4.703465346534653
[overcast 30.0 high weak yes ] dist=1.6935643564356435
[rain 13.0 high weak yes ] dist=2.1325082508250826
[rain 0.0 normal weak yes ] dist=1.2965346534653466
[rain 0.0 normal strong no ] dist=3.2965346534653466

AvgDistance=1.2943344334433444
1:Centroid=(overcast 18.5 high weak yes )
Examples:
[sunny 30.3 high weak no ] dist=2.3894389438943895
[sunny 30.3 high strong no ] dist=3.3894389438943895
[overcast 30.0 high weak yes ] dist=0.3795379537953796

AvgDistance=0.9196919691969198
2:Centroid=(sunny 17.22 high strong no )
Examples:
[sunny 30.3 high weak no ] dist=1.4316831683168316
[sunny 30.3 high strong no ] dist=0.43168316831683173
[overcast 30.0 high weak yes ] dist=3.4217821782178217
[rain 13.0 high weak yes ] dist=3.1392739273927392
[rain 0.0 normal weak yes ] dist=4.568316831683168

AvgDistance=1.3453465346534654

```

Scelta menu: 2

Nome tabella: nome tabella inesistente nel db

Numero cluster: numero compreso tra > 0 e <= delle tuple distinte

Output atteso: tabella inesistente nel db

Output:

```
Socket[addr=localhost/127.0.0.1,port=8080,localport=58605]
Scegli una opzione
(1) Carica Cluster da File
(2) Carica Dati
Risposta: 2
Nome tabella: prova
java.rmi.ServerException:
[Error]: tabella inserita inesistente
```

Scelta menu: 2

Nome tabella: nome tabella esistente nel db

Numero cluster: numero compreso tra < 0

Output atteso: numero di cluster inserito errato

Output:

```
Scegli una opzione
(1) Carica Cluster da File
(2) Carica Dati
Risposta: 2
Nome tabella: playtennis
Numero di cluster: 0
java.rmi.ServerException:
[Error]: k inserito errato
```

Scelta menu: 2

Nome tabella: nome tabella esistente nel db

Numero cluster: numero compreso tra $>$ delle tuple distinte

Output atteso: numero di cluster inserito errato

Output:

```
Socket[addr=localhost/127.0.0.1,port=8080,localport=58663]
Scegli una opzione
(1) Carica Cluster da File
(2) Carica Dati
Risposta: 2
Nome tabella: playtennis
Numero di cluster: 20
java.rmi.ServerException:
[Error]: Il numero inserito è minore di 0 oppure maggiore di 14
```