



H-CLUS

DOCUMENTAZIONE PROGETTO D'ESAME MAP

Membri del progetto d'esame:

- Annarita Bruno, a.bruno113@studenti.uniba.it, mat: 766402
- Giovanni Cirigliano, g.cirigliano5@studenti.uniba.it, mat: 758295
- Maria Rita Losacco, m.losacco26@studenti.uniba.it, mat: 738053

INDICE

1.0 - INTRODUZIONE	3
1.1 - DESCRIZIONE	3
2.0 - GUIDA D'INSTALLAZIONE.....	4
3.0 - ESEGUIRE H-CLUS BASE	9
4.0 - TEST H-CLUS BASE	12
5.0 - ESEGUIRE H-CLUS ESTESO	15
6.0 - ISTRUZIONI PER L'USO H-CLUS ESTESO	17
7.0 - TEST H-CLUS ESTESO	20
8.0 - CAMBIAMENTI.....	22

1.0 - INTRODUZIONE

Nel periodo 2024/2025, il progetto di studio è stato incentrato su "H-CLUS", un sistema client-server per l'estrazione di dati basata su tecniche di clustering agglomerativo. Il server effettua il clustering utilizzando i metodi Single Link o Average Link per calcolare la distanza tra i cluster e generare un dendrogramma. Il client, sviluppato in Java, consente agli utenti di caricare o creare nuovi oggetti HierarchicalClusterMiner, visualizzare il dendrogramma e salvare i risultati per consultazioni future. Gli utenti hanno inoltre la possibilità di ricaricare file salvati in precedenza per visualizzare nuovamente i cluster e i dendrogrammi associati.

1.1 - DESCRIZIONE

L'algoritmo di clustering agglomerativo utilizzato nel progetto "H-CLUS" rappresenta uno strumento potente per individuare strutture nascoste nei dati. A differenza di altri algoritmi, come il k-means, l'approccio agglomerativo non richiede di specificare in anticipo il numero di cluster. Invece, l'algoritmo inizia trattando ciascun elemento come un cluster autonomo e unisce progressivamente i cluster più vicini, utilizzando criteri come:

- **Single Link:** distanza minima tra i punti dei cluster.

$$D(C1, C2) = \min_{(t1 \in C1, t2 \in C2)} dist(t1, t2)$$

- **Average Link:** distanza media tra i punti dei cluster.

$$D(C1, C2) = \frac{\sum_{t1 \in C1, t2 \in C2} d(t1, t2)}{\#(C1 \times C2)}$$

Ad esempio, dati i due cluster:

$$C_1 = \{t1 = [1,0,3], t2 = [0,0,0]\}, \quad C_2 = \{t3 = [5,0,2], t4 = [0,0,1], t5 = [1,1,1]\}$$

- SingleLink = 1
 - AverageLink = $\frac{avg(17+5+5+29+1+3)}{6} = 10.0$
-

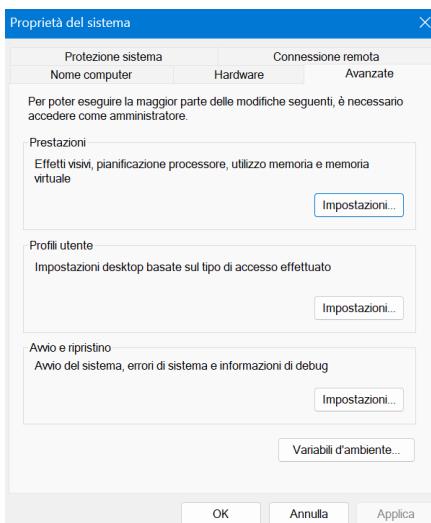
Con l'avanzare delle fusioni tra i cluster, viene generato un dendrogramma che rappresenta la gerarchia dei raggruppamenti. Il processo continua fino a ottenere un unico cluster o fino a un livello di profondità scelto dall'utente. La visualizzazione tramite dendrogramma rende l'algoritmo particolarmente intuitivo, permettendo di esplorare la struttura dei dati su vari livelli. Inoltre, questo approccio non dipende dalla scelta iniziale dei **centroidi**, riducendo così il rischio di risultati incoerenti e garantendo una visione più dettagliata della struttura interna dei dati.

2.0 - GUIDA D'INSTALLAZIONE

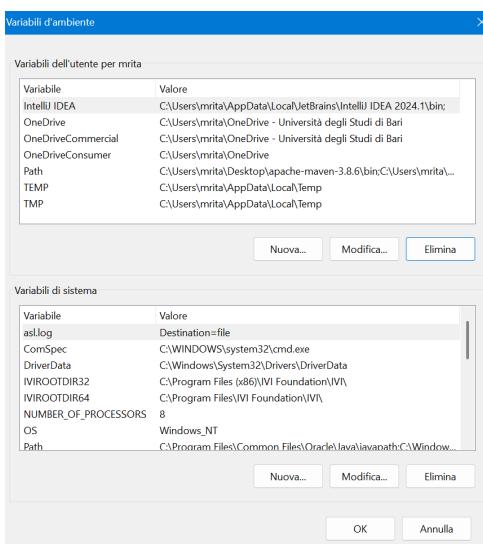
JAVA

Per eseguire il programma è essenziale avere Java installato sulla macchina:

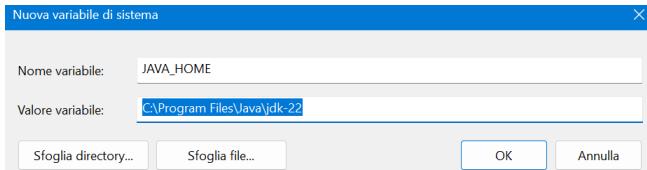
1. Installare la JRE (Java Runtime Environment) da [qui](#) (versione 8 per Windows) e la versione 22.0.1 del Java Development Kit (JDK), disponibile nella cartella “Risorse” o sul sito ufficiale [qui](#).
 2. Dopo aver eseguito l'installer del JDK, configurare la variabile d'ambiente di Java:
- Cercare “Proprietà di sistema”.



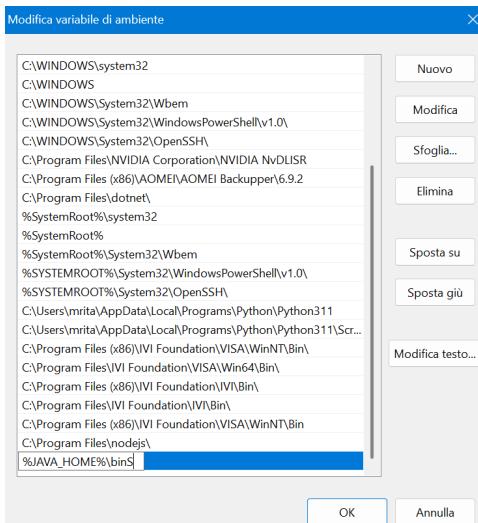
- Cliccare sulla voce “Variabili d’ambiente”.



- Nella sezione “Variabili di sistema” cliccare su **Nuovo**, nominare la variabile: “JAVA_HOME” e nel valore della variabile inserire il percorso di installazione di JDK ad esempio: C:\Program Files\Java\jdk-22, cliccare su **OK**.



- Sempre nella finestra “Variabili d’ambiente”, trovare la variabile Path nella sezione “Variabili di sistema” e cliccare su **Modifica**. Cliccare su **Nuovo** e inserire: %JAVA_HOME%\bin, quindi cliccare su **OK** per salvare tutte le modifiche.



3. Per verificare la corretta installazione, aprire il prompt dei comandi e scrivere java -version. Se tutto è stato configurato correttamente, apparirà la versione installata.

```
Prompt dei comandi
Microsoft Windows [Versione 10.0.26100.2161]
(c) Microsoft Corporation. Tutti i diritti riservati.

C:\Users\mrita>java -version
java version "22.0.1" 2024-04-16
Java(TM) SE Runtime Environment (build 22.0.1+8-16)
Java HotSpot(TM) 64-Bit Server VM (build 22.0.1+8-16, mixed mode, sharing)
```

Per maggiori dettagli, è possibile consultare la [documentazione ufficiale](#).

MySQL

Cosa è?

MySQL è un potente database relazionale open-source e multithreading, sviluppato nel 1996 da TcX, basato su un sistema di gestione di database precedente chiamato mSQL. La sua popolarità è dovuta principalmente alla natura gratuita e open-source, alle elevate prestazioni e alla flessibilità. La diffusione di Linux e l'ascesa di PHP hanno contribuito a rendere MySQL uno dei database non commerciali più utilizzati a livello globale.

Guida d'installazione

Nel progetto viene utilizzata la versione MySQL 8.0.39, insieme al MySQL Connector 8.0.17. Per il corretto funzionamento del server è necessario installare MySQL. Per procedere all'installazione, si può accedere alla cartella “Risorse” del progetto, dove è presente il file dell'installer, oppure visitare [questa pagina web](#), per scaricare la versione appropriata per il proprio sistema operativo.

- Selezionare il sistema operativo del proprio computer.
- Scaricare il file d'installazione (303.6 MB), seguendo l'immagine illustrativa fornita.

Select Version:

8.0.39

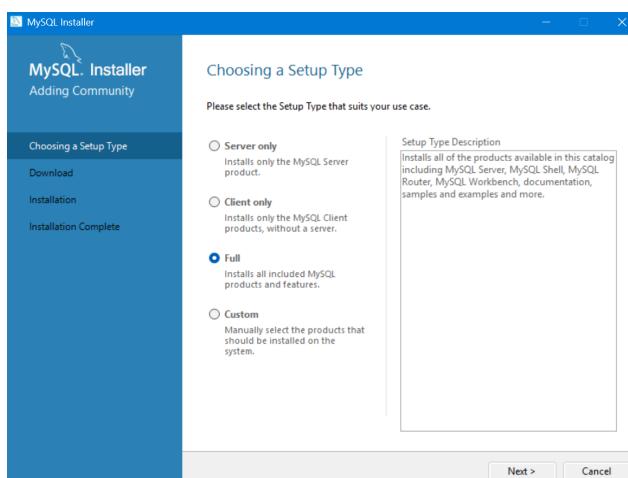
Select Operating System:

Microsoft Windows

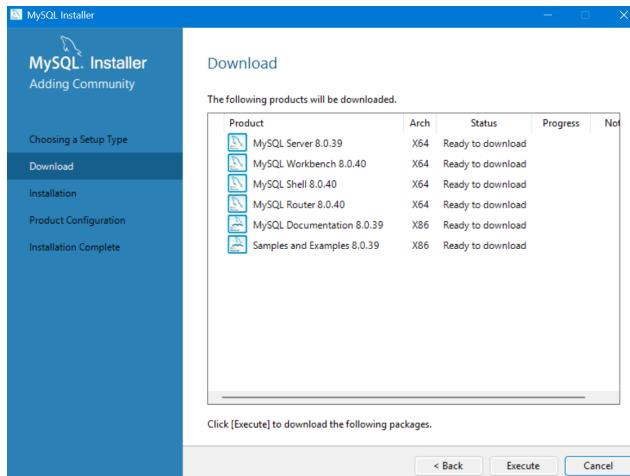
Windows (x86, 32-bit), MSI Installer	8.0.39	2.1M	Download
(mysql-installer-web-community-8.0.39.0.msi)	MD5: d8499da0b2c4b5dfa81a5c5185af9238 Signature		

Windows (x86, 32-bit), MSI Installer	8.0.39	303.6M	Download
(mysql-installer-community-8.0.39.0.msi)	MD5: 353c5e5ab9350d0e9ddcb42264229b5d Signature		

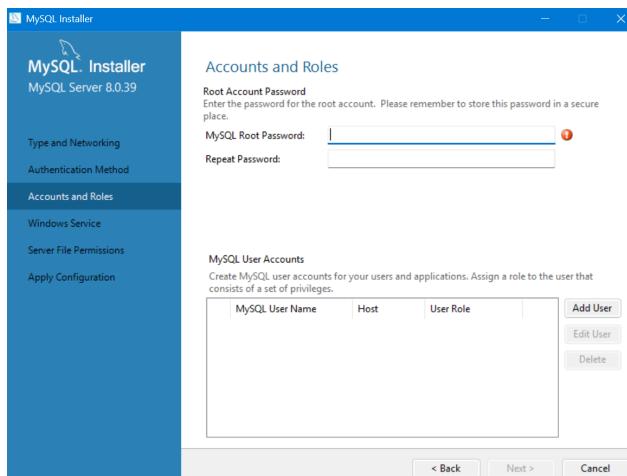
- Fare doppio clic sul file **mysql-installer-community-8.0.39.0.msi** per avviare l'installer. (*Se Windows chiede l'autorizzazione, cliccare su Sì per consentire all'installer di apportare modifiche*)
- Nella schermata “Choosing a Setup Type” selezionare la voce **Full** e cliccare su **Next**:



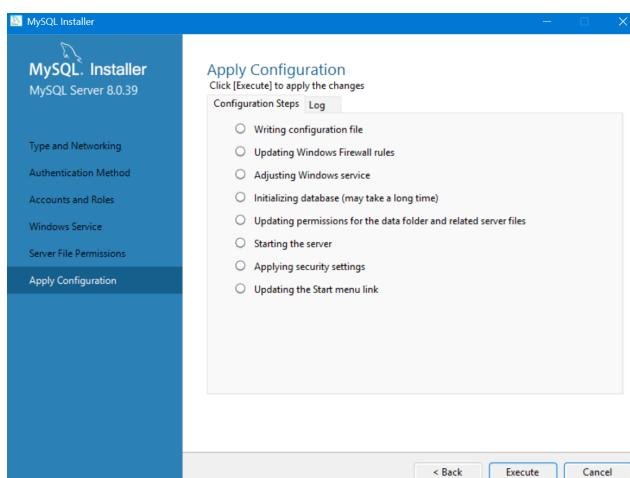
- Cliccare su **Execute** e attendere il completamento dell'installazione. Al termine, cliccare su **Next** (questa procedura dovrà essere eseguita due volte, una per **Download** e una per **Installation** nel pannello sulla sinistra).



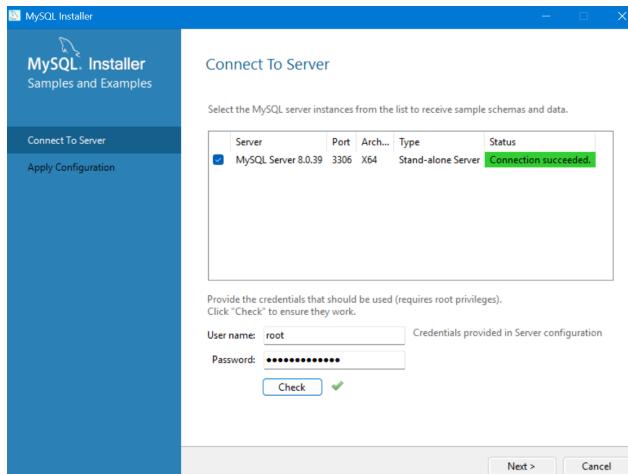
- Dopo le installazioni, cliccare su **Next** tre volte fino a raggiungere la schermata in cui è possibile impostare la password per l'account MySQL.



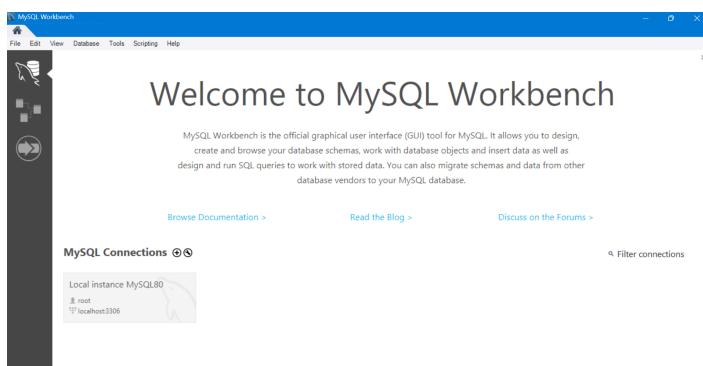
- Proseguire cliccando su **Next** fino a raggiungere la schermata successiva, quindi cliccare su **Execute** e, una volta completata l'operazione, su **Finish**.



- Cliccare su **Next** e poi su **Finish**. Si aprirà una schermata dove sarà possibile inserire l'utente e la password per verificare la connessione. Se il **Check** è verificato, cliccare su **Next**, quindi su **Execute** e infine su **Finish**.



- Se l'installazione è andata a buon fine, MySQL dovrebbe aprire automaticamente le schermate di configurazione.



Per effettuare un ulteriore controllo della corretta installazione, aprire **“MySQL 8.0 Command Line...”**, inserire la password del proprio account. Dovrebbe apparire un prompt di conferma dell'accesso a MySQL.

```
MySQL 8.0 Command Line Cli  x  +  v
Enter password: *****
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 20
Server version: 8.0.39 MySQL Community Server - GPL

Copyright (c) 2000, 2024, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.
```

3.0 - ESEGUIRE H-CLUS BASE

Per avviare correttamente la versione base di H-CLUS, seguire i passaggi nell'ordine indicato:

1. Spostarsi nella cartella “**H-CLUS_Base**” > “**Jar + Bat**”.
2. All’interno della cartella “Jar + Bat”, fare doppio clic sul file “**start_server.bat**” e lasciare il prompt dei comandi in esecuzione.
3. Successivamente, fare doppio clic sul file “**start_client.bat**”. Questo aprirà un secondo prompt dei comandi e, a questo punto, il progetto sarà pronto per essere utilizzato.

Nota: Lasciare il prompt del server in esecuzione per permettere la connessione tra client e server.

Per avviare e testare il progetto dall’IDE:

1. Aprire la cartella **Server**, inserire in “Libraries” il file “**mysql-connector-java-8.0.17**” che si trova nella cartella “Risorse” e avviare **MultiServer**.
2. Aprire la cartella **Client** e avviare **MainTest**.

-ISTRUZIONI PER L’USO H-CLUS BASE-

- **Schermata di avvio del programma:**

```
addr = /127.0.0.1
Socket[addr=/127.0.0.1,port=8080,localport=54119]
Nome tabella:
|
```

Una volta avviato il programma, apparirà una schermata con i seguenti dettagli:

- **addr:** indica l’indirizzo IP a cui si è connessi.
- **Socket:** contiene tre campi: addr: indirizzo locale, port: la porta del server e localport: la porta locale.

Alla richiesta “**Nome tabella**”, inserire il nome della tabella su cui si vuole lavorare. (es: exampleTab)

- **Inserimento della tabella:**

```
addr = /127.0.0.1
Socket[addr=/127.0.0.1,port=8080,localport=54119]
Nome tabella:
exampleTab
Scegli una opzione
(1) Carica Dendrogramma da File
(2) Apprendi Dendrogramma da Database
Risposta:
```

Dopo aver inserito il nome della tabella, apparirà un menu con due opzioni:

- 1) **Carica Dendrogramma da File:** consente di caricare un dendrogramma già salvato su un file.
- 2) **Apprendi Dendrogramma da Database:** permette di apprendere o generare un dendrogramma dai dati presenti nella tabella del database.

- **Scelta 2 (Apprendi Dendrogramma da Database):**

```
Scegli una opzione
(1) Carica Dendrogramma da File
(2) Apprendi Dendrogramma da Database
Risposta:2
Introdurre la profondita' del dendrogramma
|
```

Se si sceglie l'opzione 2, il programma richiederà di **inserire la profondità** del dendrogramma desiderata.

- **Inserimento della profondità:**

```
Introdurre la profondita' del dendrogramma
3
Distanza: single-link (1), average-link (2):
```

Dopo aver inserito la profondità, il programma offrirà due opzioni di calcolo:

- 1) **Single-link:** misura la distanza minima tra due cluster, collegando i punti più vicini tra i gruppi.
- 2) **Average-link:** calcola la distanza media tra tutti i punti di un cluster e quelli dell'altro.

- **Scelta Single-link (1):**

```
Distanza: single-link (1), average-link (2):
1
level0:
cluster0:<[1.0,2.0,0.0]>
cluster1:<[0.0,1.0,-1.0]>
cluster2:<[1.0,3.0,5.0]>
cluster3:<[1.0,3.0,4.0]>
cluster4:<[2.0,2.0,0.0]>

level1:
cluster0:<[1.0,2.0,0.0]><[2.0,2.0,0.0]>
cluster1:<[0.0,1.0,-1.0]>
cluster2:<[1.0,3.0,5.0]>
cluster3:<[1.0,3.0,4.0]>

level2:
cluster0:<[1.0,2.0,0.0]><[2.0,2.0,0.0]>
cluster1:<[0.0,1.0,-1.0]>
cluster2:<[1.0,3.0,5.0]><[1.0,3.0,4.0]>

Inserire il nome dell'archivio (comprensivo di estensione):
```

Se si seleziona il metodo **Single-link (1)**, verrà visualizzato il dendrogramma calcolato. Il programma chiederà quindi di **inserire il nome del file** in cui salvare il dendrogramma, comprensivo di estensione.

- **Inserimento nome file**

```
Inserire il nome dell'archivio (comprensivo di estensione):  
Example1.dat  
Dendrogramma salvato correttamente.  
Premere un tasto per continuare . . . |
```

Inserire il nome del file, ad esempio Example1.dat. Il dendrogramma sarà salvato all'interno della cartella “**saved**” nella directory “**Jar + Bat**” e il programma terminerà.

- **Scelta Average-link (2):**

```
Distanza: single-link (1), average-link (2):  
2  
level0:  
cluster0:<[1.0,2.0,0.0]>  
cluster1:<[0.0,1.0,-1.0]>  
cluster2:<[1.0,3.0,5.0]>  
cluster3:<[1.0,3.0,4.0]>  
cluster4:<[2.0,2.0,0.0]>  
  
level1:  
cluster0:<[1.0,2.0,0.0]><[2.0,2.0,0.0]>  
cluster1:<[0.0,1.0,-1.0]>  
cluster2:<[1.0,3.0,5.0]>  
cluster3:<[1.0,3.0,4.0]>  
  
level2:  
cluster0:<[1.0,2.0,0.0]><[2.0,2.0,0.0]>  
cluster1:<[0.0,1.0,-1.0]>  
cluster2:<[1.0,3.0,5.0]><[1.0,3.0,4.0]>  
  
Inserire il nome dell'archivio (comprensivo di estensione):  
|
```

Se si seleziona il metodo **Average-link (2)**, il programma calcolerà il dendrogramma utilizzando la distanza media e lo visualizzerà a schermo. Successivamente, sarà richiesto di **inserire il nome del file** in cui salvare il dendrogramma, comprensivo di estensione.

- **Scelta 1 (Carica Dendrogramma da File):**

```
Scegli una opzione  
(1) Carica Dendrogramma da File  
(2) Apprendi Dendrogramma da Database  
Risposta:1  
Inserire il nome dell'archivio (comprensivo di estensione):  
|
```

Se si sceglie l'opzione 1 per caricare un dendrogramma da file, sarà necessario **inserire il nome del file**, comprensivo di estensione, da cui si desidera caricare il dendrogramma. I file precedentemente salvati si trovano nella cartella “**saved**” all'interno di “**Jar + Bat**”.

- **Inserimento nome archivio con estensione:**

```
Scegli una opzione
(1) Carica Dendrogramma da File
(2) Apprendi Dendrogramma da Database
Risposta:1
Inserire il nome dell'archivio (comprensivo di estensione):
Example1.dat
level0:
cluster0:<[1.0,2.0,0.0]>
cluster1:<[0.0,1.0,-1.0]>
cluster2:<[1.0,3.0,5.0]>
cluster3:<[1.0,3.0,4.0]>
cluster4:<[2.0,2.0,0.0]>

level1:
cluster0:<[1.0,2.0,0.0]><[2.0,2.0,0.0]>
cluster1:<[0.0,1.0,-1.0]>
cluster2:<[1.0,3.0,5.0]>
cluster3:<[1.0,3.0,4.0]>

level2:
cluster0:<[1.0,2.0,0.0]><[2.0,2.0,0.0]>
cluster1:<[0.0,1.0,-1.0]>
cluster2:<[1.0,3.0,5.0]><[1.0,3.0,4.0]>

Premere un tasto per continuare . . .
```

Dopo aver inserito il nome dell'archivio, il programma mostrerà a schermo il dendrogramma caricato e terminerà l'esecuzione.

4.0 - TEST H-CLUS BASE

- **Inserimento nome della tabella errato:**

```
addr = /127.0.0.1
Socket[addr=/127.0.0.1,port=8080,localport=54675]
Nome tabella:
example
Errore nel recupero dei dati: Table 'mapdb.example' doesn't exist
Nome tabella:
```

Se si inserisce un nome di tabella non valido, il programma mostrerà un messaggio di errore, indicando che la tabella non esiste, e permetterà di inserire un nuovo nome.

- **Inserimento scelta tra Carica dendrogramma da File o Apprendi dendrogramma da database errata:**

```
Scegli una opzione
(1) Carica Dendrogramma da File
(2) Apprendi Dendrogramma da Database
Risposta:3
(1) Carica Dendrogramma da File
(2) Apprendi Dendrogramma da Database
Risposta:
```

Se si inserisce una scelta non valida (ad esempio, un numero diverso da 1 o 2), il programma permetterà di effettuare nuovamente la scelta.

- **Scelta Carica dendrogramma da file, inserimento del nome di un file inesistente:**

```
(1) Carica Dendrogramma da File
(2) Apprendi Dendrogramma da Database
Risposta:1
Inserire il nome dell'archivio (comprensivo di estensione):
example.dat
File non trovato: example.dat
Premere un tasto per continuare . . .
```

Se si sceglie di caricare un dendrogramma da un file e si inserisce il nome di un file non esistente, il programma mostrerà un avviso di errore e terminerà l'esecuzione.

- **Scelta Apprendi dendrogramma da database, inserimento della profondità errata:**

```
(2) Apprendi Dendrogramma da Database
Risposta:2
Introdurre la profondita' del dendrogramma
0
Distanza: single-link (1), average-link (2):
1
E' stato inserito un valore <=0 per la profondita' del Dendrogramma.
Premere un tasto per continuare . . .
```

```
(2) Apprendi Dendrogramma da Database
Risposta:2
Introdurre la profondita' del dendrogramma
-2
Distanza: single-link (1), average-link (2):
2
E' stato inserito un valore <=0 per la profondita' del Dendrogramma.
Premere un tasto per continuare . . .
```

Se si inserisce un valore di profondità non valido (ad esempio, un numero minore o uguale a 0), il programma mostrerà un avviso di errore. Verranno comunque proposte le opzioni per il calcolo della distanza, ma, una volta selezionata la distanza, il programma notificherà l'errore e terminerà l'esecuzione.

- **Inserimento scelta della distanza errata:**

```
Distanza: single-link (1), average-link (2):
3
Distanza: single-link (1), average-link (2):
```

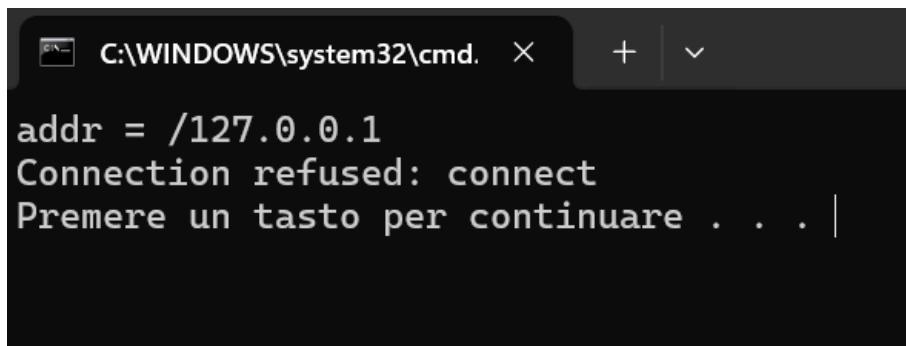
Se si inserisce una scelta di distanza non valida (ad esempio, un numero diverso da 1 o 2), il programma permetterà di effettuare nuovamente la scelta.

- **Test numero di esempi nella tabella caricata da file è minore degli esempi in cui è salvato un oggetto precedentemente serializzato.**

```
addr = /127.0.0.1
Socket[addr=/127.0.0.1,port=8080,localport=54135]
Nome tabella:
exampleTab1
Scegli una opzione
(1) Carica Dendrogramma da File
(2) Apprendi Dendrogramma da Database
Risposta:1
Inserire il nome dell'archivio (comprensivo di estensione):
esempio11.dat
Il numero di esempi nella tabella scelta e' minore del numero di esempi con cui e' stato salvato l'oggetto h-clus
precedentemente serializzato.
Premere un tasto per continuare . . .
```

Questo test verifica che il sistema sia in grado di rilevare e gestire correttamente il caso in cui il numero di esempi nella tabella caricata da un file è inferiore al numero di esempi con cui l'oggetto h-clus è stato salvato precedentemente. Il sistema mostrerà un messaggio di errore e il programma terminerà l'esecuzione.

- **Avvio del Client senza avviare il Server**



C:\WINDOWS\system32\cmd. × + ▾

```
addr = /127.0.0.1
Connection refused: connect
Premere un tasto per continuare . . . |
```

Questo test verifica il comportamento del sistema quando si tenta di avviare il client senza aver prima avviato il server. In questo scenario, il client cerca di stabilire una connessione all'indirizzo 127.0.0.1 (localhost), ma poiché il server non è in esecuzione, la connessione viene rifiutata.

- **Creazione del Clustering su una tabella vuota**

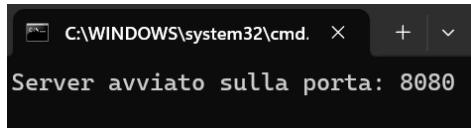
```
addr = /127.0.0.1
Socket[addr=/127.0.0.1,port=8080,localport=52441]
Nome tabella:
exampleTab1
Errore nel recupero dei dati: La tabella exampleTab1 è vuota.
Nome tabella:
```

Questo test verifica il comportamento del sistema quando l'utente tenta di selezionare una tabella vuota nel database. In questo scenario, il sistema prova a recuperare i dati dalla tabella indicata dall'utente (in questo caso exampleTab1), ma rileva che la tabella è vuota.

5.0 - ESEGUIRE H-CLUS ESTESO

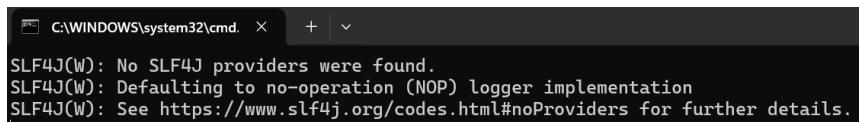
Per avviare correttamente H-CLUS, versione con estensione, seguire i passaggi indicati:

1. Spostarsi nella cartella “**H-CLUS_Esteso**” > “**Jar + Bat**”.
2. All'interno della cartella “**Jar + Bat**”, fare doppio clic prima sul file “**Start Server.bat**” e lasciare il prompt dei comandi in esecuzione.



```
C:\WINDOWS\system32\cmd. × + ▾
Server avviato sulla porta: 8080
```

3. Successivamente, fare doppio clic sul file “**Start Client.bat**”. Verrà avviato automaticamente un secondo prompt dei comandi.



```
C:\WINDOWS\system32\cmd. × + ▾
SLF4J(W): No SLF4J providers were found.
SLF4J(W): Defaulting to no-operation (NOP) logger implementation
SLF4J(W): See https://www.slf4j.org/codes.html#noProviders for further details.
```

4. Aprire **Telegram** e cercare “**Hierarchical Clustering Bot**” utilizzando il tag “**@HClusBot**” oppure cliccando il seguente [link](#).

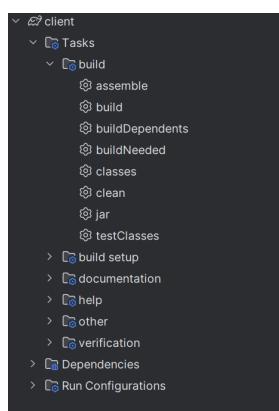


Nota: Lasciare in esecuzione i prompt dei comandi del server e del client per consentire la connessione.

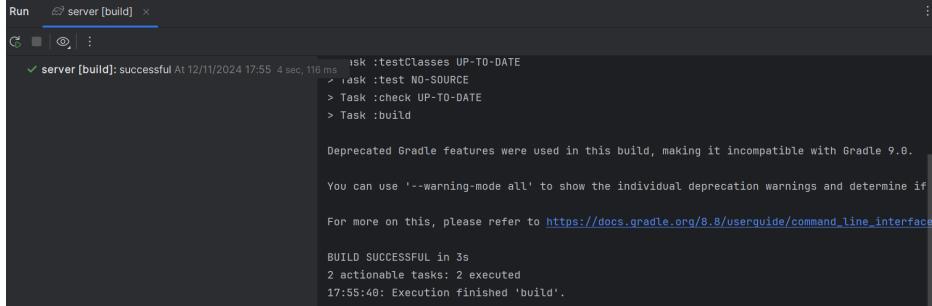
Avvio e test del progetto dall'IDE:

Per poter avviare e testare il progetto dall' IDE (bisogna aver installato il plugin Buildship Gradle Integration, [guida per l'installazione](#)):

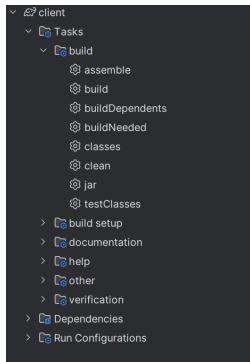
1. Aprire la cartella **Server**, fare clic sull'icona di **Gradle**, andare su **Task**, selezionare **Build** e fare doppio clic su **Build**.



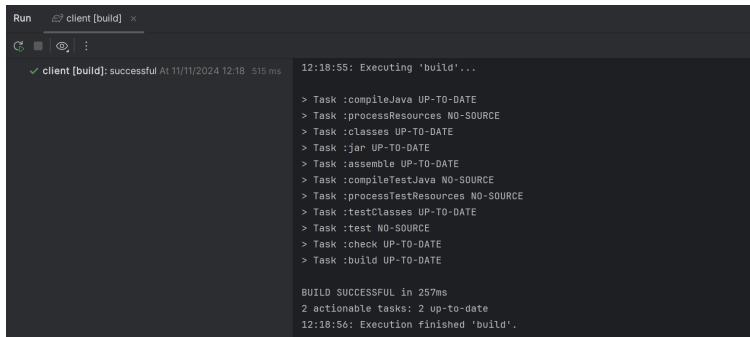
Se tutto fosse corretto, si dovrebbe visualizzare una schermata di conferma con lo stato "BUILD SUCCESSFUL".



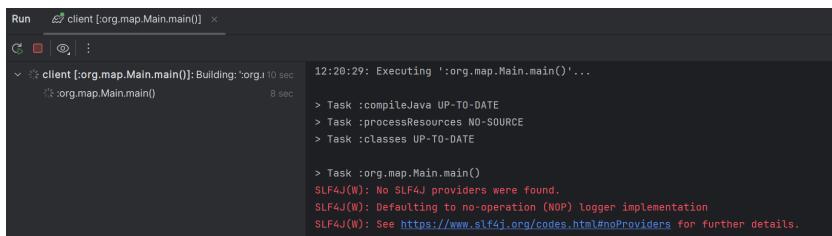
2. Aprire la cartella **Client**, fare clic sull'icona di **Gradle**, andare su **Task**, selezionare **Build** e fare doppio clic su **Build**.



Se tutto è corretto, si dovrebbe visualizzare una schermata di conferma con lo stato "BUILD SUCCESSFUL".



3. Successivamente, eseguire il **Main del Client**.



4. Infine, aprire **Telegram** e cercare nuovamente “**Hierarchical Clustering Bot**” con il tag “**@HClusBot**” per l’interazione con il bot.

6.0 - ISTRUZIONI PER L'USO H-CLUS ESTESO

▪ Schermata iniziale bot Telegram

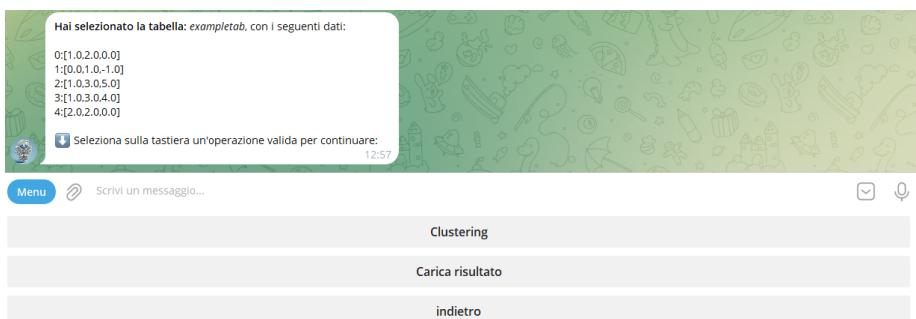


Cliccare su **Inizia** per avviare il bot o digitare **/start** per riavviarlo.

▪ Scelta della tabella



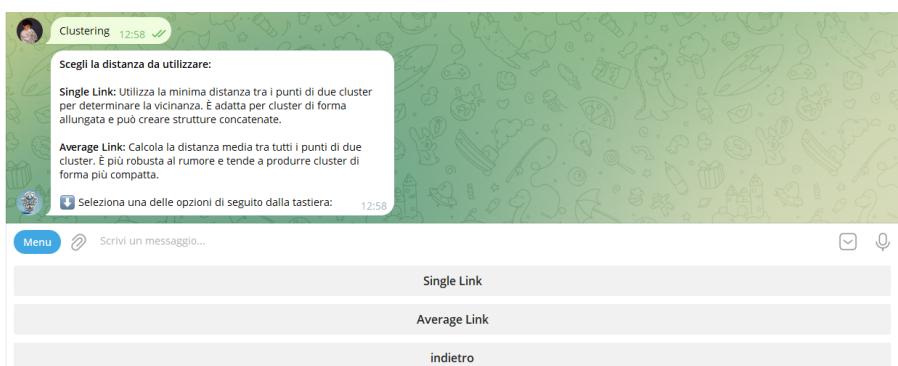
Dopo l'avvio, selezionare la tabella da usare. Ad esempio, utilizzare "exampletab" come tabella.



Una volta scelta la tabella, si avranno a disposizione le seguenti opzioni:

1. Creare il clustering.
1. Caricare risultato esistente.
2. Tornare indietro.

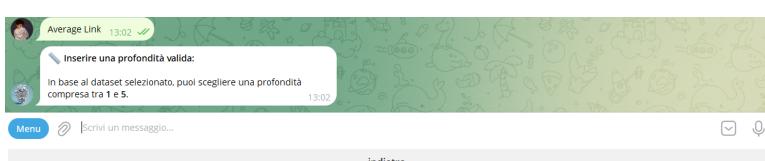
▪ Scelta 1. Clustering



Verranno mostrate due opzioni di clustering:

- 1) **Single-link**: metodo misura la distanza minima tra due cluster, collegando i punti più vicini tra i gruppi.
- 2) **Average-link**: metodo calcola la distanza media tra tutti i punti di un cluster e quelli dell'altro.
- 3) Tornare indietro.

▪ Dopo aver scelto la distanza



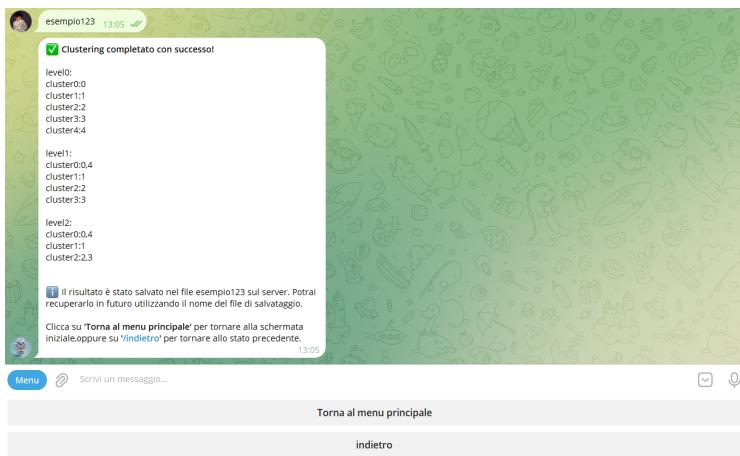
Inserire la **profondità** del dendrogramma desiderata oppure tornare indietro.

▪ Inserire la profondità desiderata



Dopo aver inserito la profondità, inserire il **nome del file di salvataggio** (senza estensione) dove verrà memorizzato il risultato del clustering.

▪ Inserimento nome file di salvataggio



Dopo aver inserito il nome del file, il dendrogramma sarà salvato nella cartella “**saved**” all'interno di “**Jar + Bat**”. A questo punto si può tornare al menu principale o indietro.

- **Scelta 2. Carica risultato**



Inserire il **nome del file** contenente il risultato del clustering che si trova nella cartella “**saved**” in “**Jar + Bat**”.

- **Inserimento nome file**



Dopo aver inserito il nome, il bot mostrerà il risultato del clustering. Sarà possibile scegliere se:

- Caricare un altro dendrogramma;
- Tornare al menu principale;
- Andare indietro.

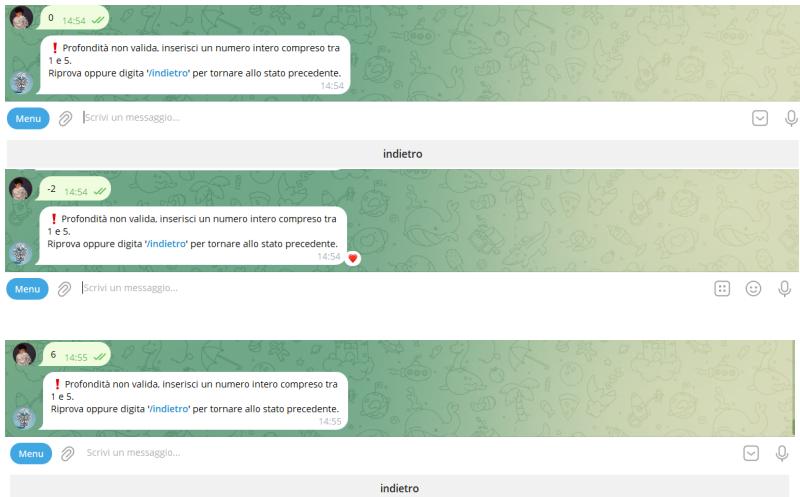
- **Scelta Carica un altro dendrogramma**



Selezionare un'altra tabella da cui partire e seguire le opzioni già descritte o tornare indietro.

7.0 - TEST H-CLUS ESTESO

▪ Inserimento profondità sbagliata



Se viene inserita una profondità non valida (minore o uguale a 0 oppure maggiore di 5), il programma mostrerà un messaggio di errore. Verrà data la possibilità di reinserire la profondità o di tornare indietro.

▪ Inserimento nome file errato da carica risultato



Se viene inserito un nome di file non valido durante il caricamento di un risultato, il programma mostrerà un messaggio di errore. Sarà possibile reinserire il nome corretto oppure tornare indietro.

▪ Test delle opzioni "Indietro"

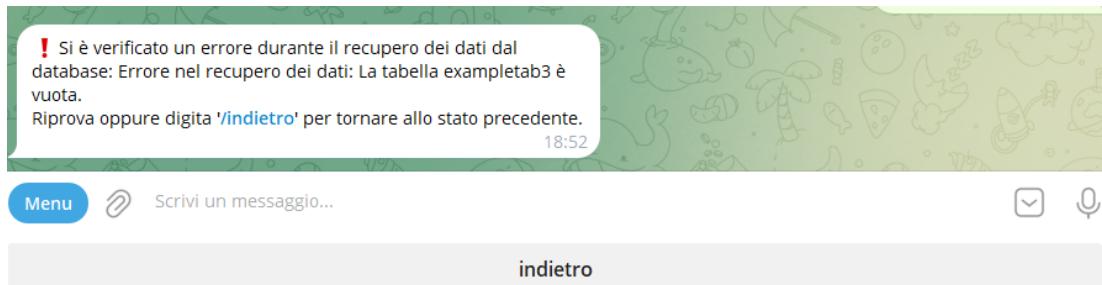
Tutte le opzioni "Indietro" sono state testate e funzionano correttamente, permettendo di tornare al menu precedente senza problemi.

▪ Avvio del Client senza avviare il Server



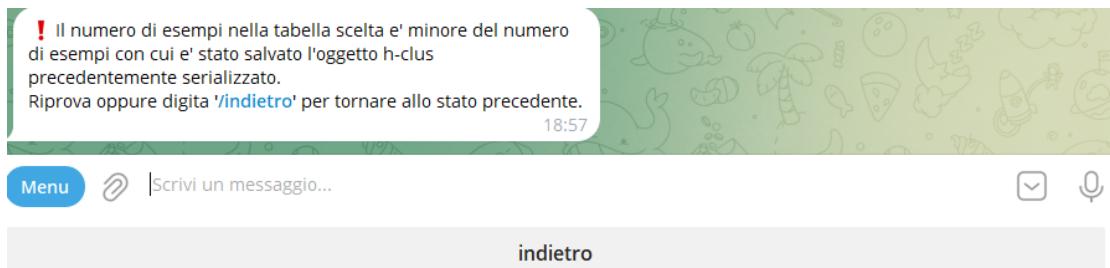
Questo test verifica il comportamento del sistema quando si tenta di avviare il client senza aver prima avviato il server. In questo scenario, il bot cerca di stabilire una connessione, ma poiché il server non è in esecuzione, la connessione viene rifiutata.

- **Creazione del Clustering su una tabella vuota**



Questo test verifica il comportamento del sistema quando l'utente tenta di selezionare una tabella vuota nel database. In questo scenario, il sistema prova a recuperare i dati dalla tabella indicata dall'utente (in questo caso exampleTab1), ma rileva che la tabella è vuota.

- **Test numero di esempi nella tabella caricata da file è minore degli esempi in cui è salvato un oggetto precedentemente serializzato.**



Questo test verifica che il sistema sia in grado di rilevare e gestire correttamente il caso in cui il numero di esempi nella tabella caricata da un file è inferiore al numero di esempi con cui l'oggetto h-clus è stato salvato precedentemente. Il sistema mostrerà un messaggio di errore e il programma darà la possibilità di tornare indietro.

8.0 - CAMBIAMENTI

CAMBIAMENTI MOTIVATI NELLA VERSIONE BASE RISPETTO I REQUISITI DEL PROGETTO

1. Metodo getLength() in ClusterSet / Metodo getLevel0Length() in HierarchicalClusterMiner:

- **Descrizione** `getLength()`: Questo metodo restituisce la lunghezza dell'array C in ClusterSet.
- **Descrizione** `getLevel0Length()`: Questo metodo restituisce la lunghezza del set di cluster al livello 0 del dendrogramma.
- **Motivazione:** Abbiamo aggiunto questi metodi per ottenere il numero di elementi presenti nel ClusterSet al livello 0 del dendrogramma e per passare tale informazione al metodo `loadDendrogramFromFile()` in `ServerOneClient`. Questo permette di verificare qui che il numero di esempi nella tabella scelta non sia minore del numero di esempi con cui è stato salvato il dendrogramma precedentemente.

2. Metodo getDepth () in HierarchicalClusterMiner:

- **Descrizione** `getDepth()`: Questo metodo restituisce la profondità del dendrogramma in `HierarchicalClusterMiner`.
- **Motivazione:** Abbiamo aggiunto questo metodo per verificare nel metodo `loadDendrogramFromFile()` in `ServerOneClient` che la profondità del dendrogramma caricato da file non sia maggiore del numero di esempi nella tabella scelta dall'utente.

CAMBIAMENTI MOTIVATI NELLA VERSIONE ESTESA RISPETTO ALLA BASE

1. Rimossa responsabilità gestione macchina a stati nel server.

Nella versione base del programma, il metodo `run` di `ServerOneClient` memorizzava l'oggetto `Data`, che rappresentava la tabella contenente il dataset scelto dall'utente. Questo oggetto era creato alla prima richiesta dal client, nel momento in cui l'utente specificava quale tabella del database utilizzare. I dati erano poi conservati all'interno del thread avviato per ciascuna connessione tra client e server, e mantenuti attivi grazie a un ciclo `while true`, che proseguiva fino alla disconnessione del client.

Nella versione estesa, invece, il client gestisce gli aggiornamenti (`Updates`) inviati dall'API di Telegram, che contengono i messaggi degli utenti diretti al bot. Teoricamente, il server della versione base potrebbe essere compatibile con il client della versione estesa, poiché è progettato per gestire connessioni multiple tramite multithreading: ogni aggiornamento del bot potrebbe quindi attivare un thread separato del server, avviando una nuova connessione `Socket`.

Tuttavia, questa soluzione è adatta solo se la connessione socket può rimanere aperta a lungo. Se, per esempio, un utente inviasse un messaggio ora e il successivo solo una settimana dopo, mantenere la connessione aperta risulterebbe inefficiente e costoso, poiché l'unico scopo sarebbe mantenere vivo l'oggetto `Data` nel thread.

Per questa ragione e per garantire al server una struttura generica, capace di interfacciarsi con client di natura diversa, abbiamo deciso di spostare l'oggetto `Data` nel client esteso, insieme ad altre informazioni necessarie per la gestione degli stati. Ad esempio, in futuro potrebbero esistere sia un client web sia un client bot Telegram. Supponiamo che un utente esegua due clustering uno sulla

versione web e una sulla versione bot Telegram; mantenere tutti i dati di stato sul server limiterebbe la possibilità di separare l'interazione su dispositivi diversi, oltre a imporre uno schema fisso basato sulla macchina a stati anche per i client che non lo necessitano. Un client senza macchina a stati potrebbe, ad esempio, essere una versione web in cui le operazioni “Carica tabella”, “Profondità”, “Distanza” e “Clustering” sono eseguite in un'unica richiesta, senza passare attraverso stati successivi.

Nel nuovo client quando riceve un aggiornamento da parte di un nuovo utente, crea un record nel dizionario memorizzando come chiave l'ID Telegram dell'utente e come valore un'istanza della classe StateContext, che gestisce la macchina a stati. Quando un utente invia un messaggio e la sua chiave è già presente nel dizionario, il sistema esegue le operazioni sul StateContext già esistente per quell'utente. StateContext è responsabile della gestione della macchina a stati e contiene uno Stack di stati, che memorizza la cronologia degli stati traversati dall'utente, più l'ultimo stato attivo in cui applicare gli aggiornamenti. Inoltre, StateContext include una classe Dati che funge da aggregatore di vari dati condivisi tra gli stati e include anche l'oggetto Data (dataset) originale della versione base del server. In questo modo, il sistema può gestire la persistenza e la transizione degli stati per ciascun utente, senza dover mantenere una connessione server-client attiva per tempi prolungati.

La classe StateContext contiene i seguenti attributi di istanza:

- **Stack<State> StateHistory:** rappresenta la cronologia degli stati traversati dall'utente. Di default, lo stack contiene lo stato iniziale passato nel costruttore e viene poi popolato aggiungendo ogni nuovo stato tramite operazioni di push. Questo approccio facilita il rollback degli stati. Ad esempio, se l'utente è nello stato 3 e lo stack contiene gli stati [3, 2, 1], un rollback eseguirà un pop dello stato 3, riportando automaticamente l'utente allo stato 2, senza la necessità di gestire manualmente ogni singolo caso di retrocessione.

Ogni stato ha la proprietà **allowBack**. Se impostata su false (valore di default), consente il rollback dallo stato corrente. Se invece è impostata su true, non è possibile tornare indietro da quello stato. Questo comportamento è stato applicato allo stato iniziale, poiché non è consentito fare rollback una volta entrati nello stato di avvio. Ad esempio, gli stati Clustering e ShowSavedClustering offrono opzioni per tornare allo stato principale (ossia Start), ma una volta raggiunto lo stato Start, il rollback non è consentito poiché segna l'inizio di una nuova iterazione.

Ogni **stato** è suddiviso in due fasi di operazione:

- **Pre-Operation:** viene eseguita una sola volta quando lo stato viene creato o resettato. Nel flusso di lavoro attuale, ogni stato fornisce all'utente una serie di operazioni da eseguire, attende la risposta dell'utente e la valida. La fase di preOperation corrisponde all'operazione di chiedere all'utente cosa desidera fare.
- **Post-Operation:** valida la risposta dell'utente e gestisce le transizioni verso altri stati. Passando StateContext come parametro al metodo postOperation, è possibile richiamare il metodo changeState per indicare il nuovo stato da eseguire. Una volta cambiato lo stato, viene eseguita automaticamente la preOperation del nuovo stato, permettendo al sistema di rispondere immediatamente all'utente con il nuovo set di operazioni disponibili per quello stato.

Durante l'esecuzione di una delle due operazioni possono verificarsi errori dovuti a input non validi, dati incorretti o problemi nelle risposte socket. Il comportamento in caso di errore varia in base alla fase dell'operazione:

- Pre-Operation: un errore in questa fase causa un rollback automatico allo stato precedente. Ad esempio, nella fase di pre-operazione del Clustering, può verificarsi un errore se l'utente sceglie un nome per il file di salvataggio del clustering che esiste già dallo stato precedente. In tal caso, il blocco catch esegue un rollback allo stato precedente (con apposito log d'errore), ovvero lo stato ChooseName della post-operazione, permettendo all'utente di inserire un nuovo nome per il file di salvataggio.
- Post-Operation: in questa fase, un eventuale errore non altera il flusso del programma, ma comporta la ripetizione della post-operazione. Ad esempio, nel metodo ChooseDepth, la post-operazione convalida la profondità scelta dall'utente; se questa profondità non è valida o fuori range, viene generata un'eccezione che porta al riavvio della post-operazione (con apposito log d'errore), consentendo all'utente di inserire una nuova profondità valida.

Ecco lo schema degli stati della macchina a stati:

