

K-Means Clustering

a simple way to cluster data

D. Boccuzzi, M. Capone, P. Di Grassi

Indice

1. Introduzione

- 1.1. Clusterizzazione dati: l'algoritmo K-Means
- 1.2. Vantaggi e Svantaggi dell'algoritmo

2. K-Means Clustering: introduzione al progetto

- 2.1. Estensione sviluppata
- 2.2. Progettazione

3. Diagrammi UML

- 3.1. Server
- 3.2. Client Console
- 3.3. UI Client

4. Manuale utente

- 4.1. Requisiti
- 4.2. Versione console
- 4.3. Versione grafica

1. Introduzione

Al giorno d'oggi, i dati hanno acquisito un ruolo di straordinaria importanza e si sono trasformati in un prezioso tesoro da sfruttare. Ogni giorno, miliardi di dati vengono generati da una varietà di fonti, tra cui dispositivi connessi, piattaforme online, sensori, transazioni commerciali e molto altro. Le aziende, le organizzazioni e in generale i mercati, usano questi dati raccolti per elaborare informazioni e utilizzarle per rimanere operativi e soprattutto competitivi. Al mancare dell'informazione si possono verificare situazioni di incertezza in quanto non sarà noto a priori l'esito di una qualsiasi operazione. L'informazione giusta, nel momento giusto è quindi alla base dei successi aziendali. Tuttavia, l'abbondanza di dati non garantisce automaticamente una maggiore comprensione o una presa di decisioni più informata. Affinché la grande mole sia gestita efficientemente, risulta necessaria l'analisi dei dati.

Uno degli strumenti utilizzati nell'analisi dei dati è la clusterizzazione dei dati stessi. Questo approccio si basa sulla capacità di identificare e categorizzare pattern simili all'interno dei dati, raggruppandoli in cluster o gruppi omogenei. L'analisi dei cluster non solo semplifica l'interpretazione dei dati molto complessi, ma svela anche relazioni e tendenze che potrebbero rimanere altrimenti inosservate.

1.1 Clusterizzazione dati: l'algoritmo K-Means

L'algoritmo K-Means è uno dei metodi di clustering più ampiamente utilizzati nell'analisi dei dati, in grado di identificare K cluster, dove K è un valore arbitrario specificato a priori dall'utente. I cluster formati dall'algoritmo sono bilanciati rispetto alla forma, proprietà che rende i cluster tipicamente di forma globulare, e rispetto alla densità, cioè il numero medio di punti presenti in una determinata area è costante.

Il suo funzionamento è piuttosto semplice: inizialmente, vengono scelti casualmente K punti come centroidi iniziali dei cluster. Successivamente, ogni tupla viene assegnata al cluster il cui centroide è più vicino in base a una misura di distanza, come ad esempio la distanza euclidea. Una volta che tutti i punti sono stati sistemati, i centroidi vengono aggiornati come la media dei punti assegnati a ciascun cluster. Questo ciclo di assegnazione e aggiornamento viene ripetuto finché i centroidi convergono a posizioni stabili.

Va specificato che K-Means richiede di scegliere in anticipo il numero di cluster. Questa scelta, però, implica il modo in cui i dati saranno raggruppati e interpretati. Una selezione imprecisa di K può portare a risultati meno significativi o addirittura fuorvianti. Tuttavia, l'algoritmo K-Means è una potente tecnica di clustering che trova applicazione in molte aree grazie alla sua efficacia e semplicità.

1.2 Vantaggi e Svantaggi dell'algoritmo

Come qualsiasi algoritmo, K-Means presenta dei vantaggi e delle limitazioni. Per quanto riguarda i vantaggi è possibile dire che K-Means è:

- **Efficiente:** in quanto computazionalmente è poco oneroso e converge velocemente;
- È adatto per **grandi dataset**;
- **Semplice** da Implementare: in quanto l'idea di assegnazione dei punti a cluster e aggiornamento dei centroidi è concettualmente semplice.

D'altra parte, però, le limitazioni sono le seguenti:

- K-Means è **sensibile all'inizializzazione**: la scelta iniziale dei centroidi può influenzare il risultato finale; quindi, l'algoritmo potrebbe convergere in diversi cluster in base alla scelta iniziale.
- L'algoritmo **richiede il numero di Cluster**: l'utente deve specificare il numero di cluster K in anticipo, il che potrebbe non essere sempre ovvio.

2. K-Means Clustering: Introduzione al progetto

Affinché l'algoritmo K-Means possa essere studiato e utilizzato, è stato creato un software in Java, che permette la clusterizzazione di esempi di dati a partire da una tabella di un database. Il software è composto da due applicativi: un client e un server. In particolare, il client consente agli utenti di interagire con il server attraverso il terminale, in maniera tale da ottenere i risultati delle operazioni di clustering e caricamento file. Il client, quindi, offre funzionalità intuitive per selezionare le impostazioni desiderate, come il numero di cluster K da definire e la tabella da cui estrarre i dati. Il server è il motore vero e proprio del software in quanto può eseguire diverse richieste per i diversi client connessi. In particolare, può estrapolare dati da una tabella di un database, eseguire il clustering su quei dati, salvarli e di conseguenza leggerli da un file. Inoltre, lo stato del server è consultabile da terminale.

2.1 Estensione sviluppata

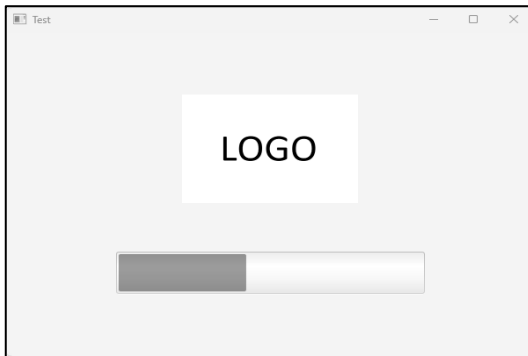


Per quanto la versione base sia completa e funzionante, abbiamo deciso che, come estensione del progetto, sarebbe stato interessante sviluppare un'interfaccia grafica per il client. La motivazione di questa scelta risiede in diversi fattori. Innanzitutto, si è deciso di avvantaggiare la semplicità d'uso: senza dubbio un'interfaccia grafica risulta notevolmente più user-friendly rispetto a un'applicazione che richiede l'interazione attraverso un terminale.

Purché l'applicazione sviluppata sia a solo uso interno, non va dimenticato che l'aggiunta dell'interfaccia grafica rappresenta un'evoluzione del progetto in risposta alle esigenze degli utenti moderni. Seguendo proprio le esigenze degli utenti, va specificato che l'interfaccia è stata sviluppata seguendo alcune delle euristiche di Nielsen: abbiamo cercato di dare all'interfaccia un aspetto semplice e pulito, fornendo all'utente controllo e libertà di movimento tra le diverse finestre senza possibilità di cadere in errore.

Inoltre, l'implementazione di un'interfaccia grafica ci ha permesso di esplorare un campo differente dalle nostre competenze, mettendoci nei panni di un UI designer. Questa esperienza si è dimostrata altamente formativa, spingendoci a sviluppare abilità creative e a immergerci nell'approfondimento del design dell'applicazione. Attraverso questa sfida abbiamo ampliato la nostra prospettiva, precedentemente limitata al solo utilizzo del terminale, sperimentando così nuovi processi creativi.

2.2 Progettazione



L'interfaccia grafica è stata creata attraverso SceneBuilder e JavaFX, utilizzando gli appositi file *.fxml*, i relativi controller e file *.css* per introdurre concetti grafici non supportati nativamente da SceneBuilder.



Dopo aver definito con chiarezza i requisiti e valutato possibili fonti di ispirazione per la struttura dell'interfaccia grafica da sviluppare, si è proceduto alla realizzazione dei mockup. Questi mockup hanno lo scopo di fornire una guida visiva per l'implementazione pratica dell'interfaccia, delineando il layout e l'organizzazione degli elementi in modo da rispettare le esigenze identificate in precedenza.



Il progetto dell'interfaccia ha coinvolto la realizzazione di quattro schermate principali. La prima schermata è stata dedicata alla fase di caricamento iniziale in cui avviene l'avvio del programma e la connessione al server, seguita da una seconda schermata che include spiegazioni dettagliate insieme a due opzioni per l'utente. A seconda della scelta, la pagina può indirizzarlo verso la lettura di dati da un file oppure verso l'individuazione dei cluster da un database.



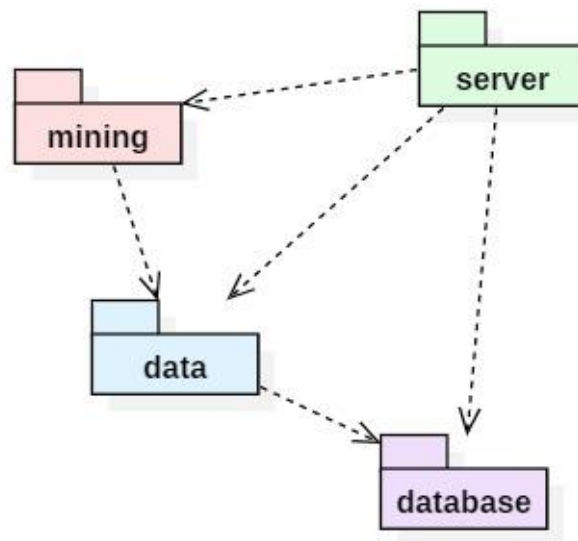
L'utilizzo dei mockup ha facilitato, innanzitutto, la comunicazione tra i membri del team e ha fornito un punto di riferimento iniziale per l'implementazione dell'interfaccia. Inoltre, l'utilizzo dei modelli ha permesso di lavorare contemporaneamente sui due problemi principali nella realizzazione di un'interfaccia grafica: il suo funzionamento e il suo aspetto. Di conseguenza, questa metodologia ha contribuito a ridurre il rischio di equivoci durante lo sviluppo e ha aiutato a mantenere una visione chiara dell'aspetto finale dell'applicazione. La realizzazione finale è visibile nella guida utente di questo documento.

3. Diagrammi UML

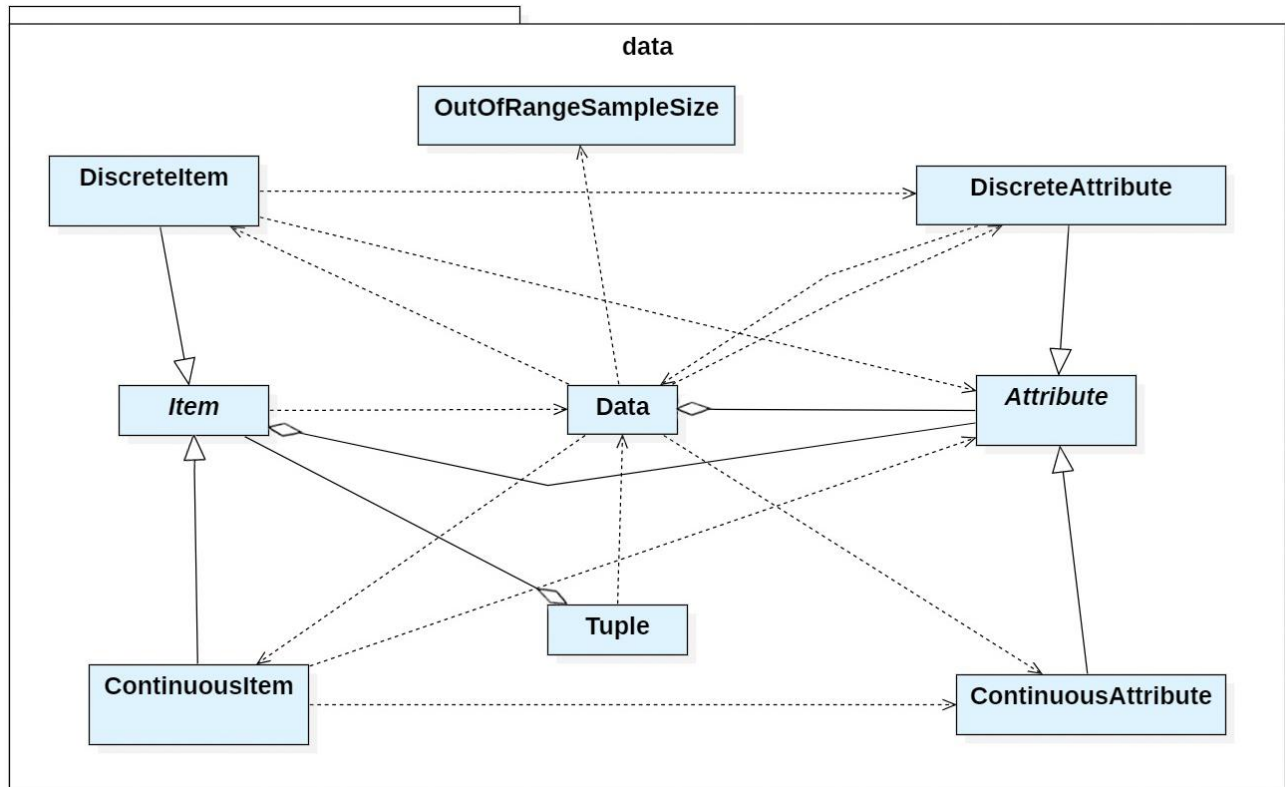
In questa sezione del documento vengono mostrati i diagrammi UML delle classi e dei package. Tutti i diagrammi sono stati realizzati utilizzando il software StarUML. Abbiamo tratto ispirazione dal libro *Java Modeling Color with UML* e quindi deciso di utilizzare i colori per identificare le classi di uno stesso package.

3.1 Server

Diagramma dei package



Nello specifico:



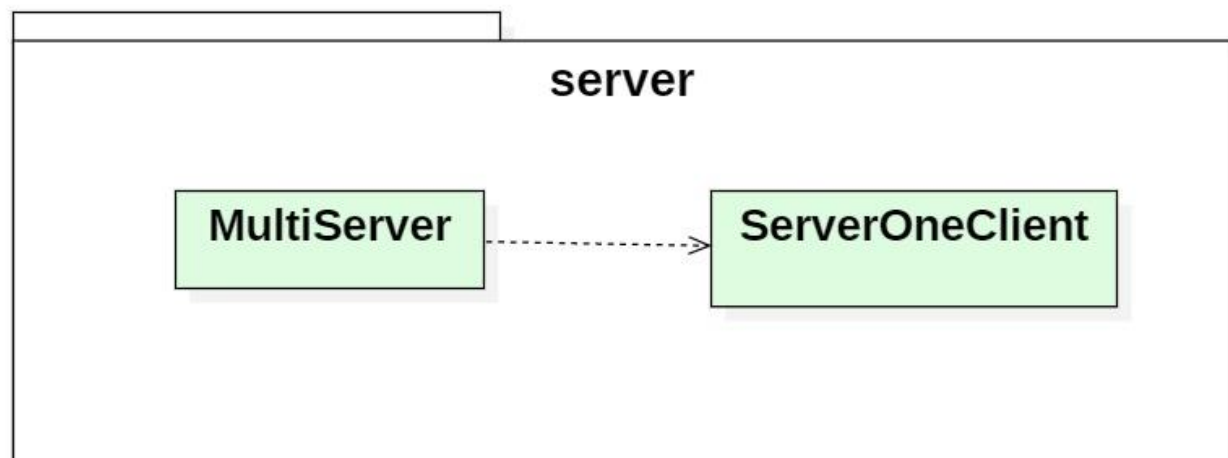
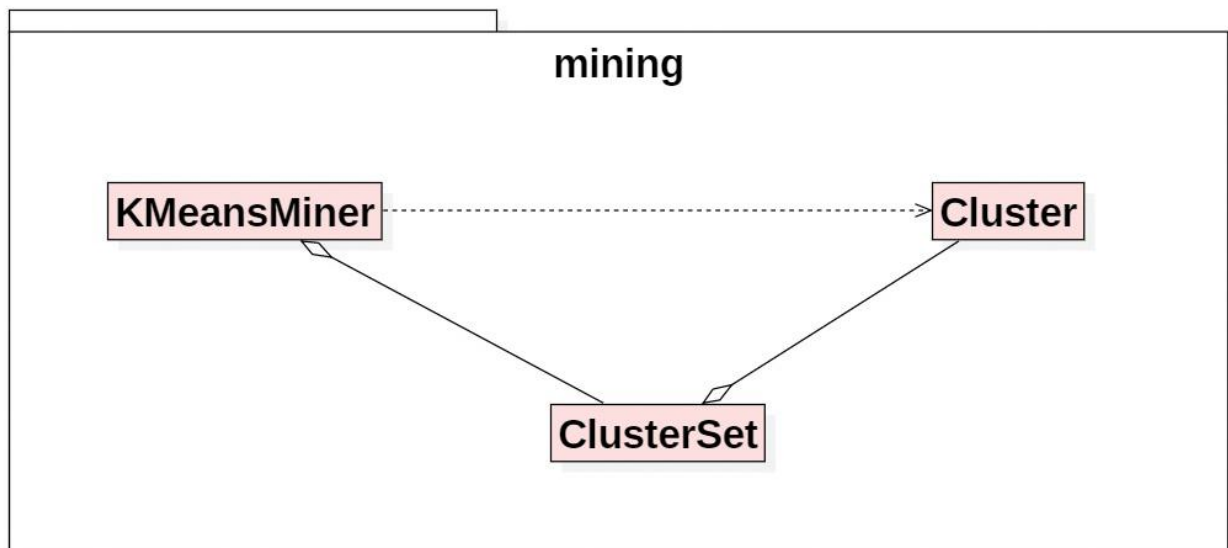
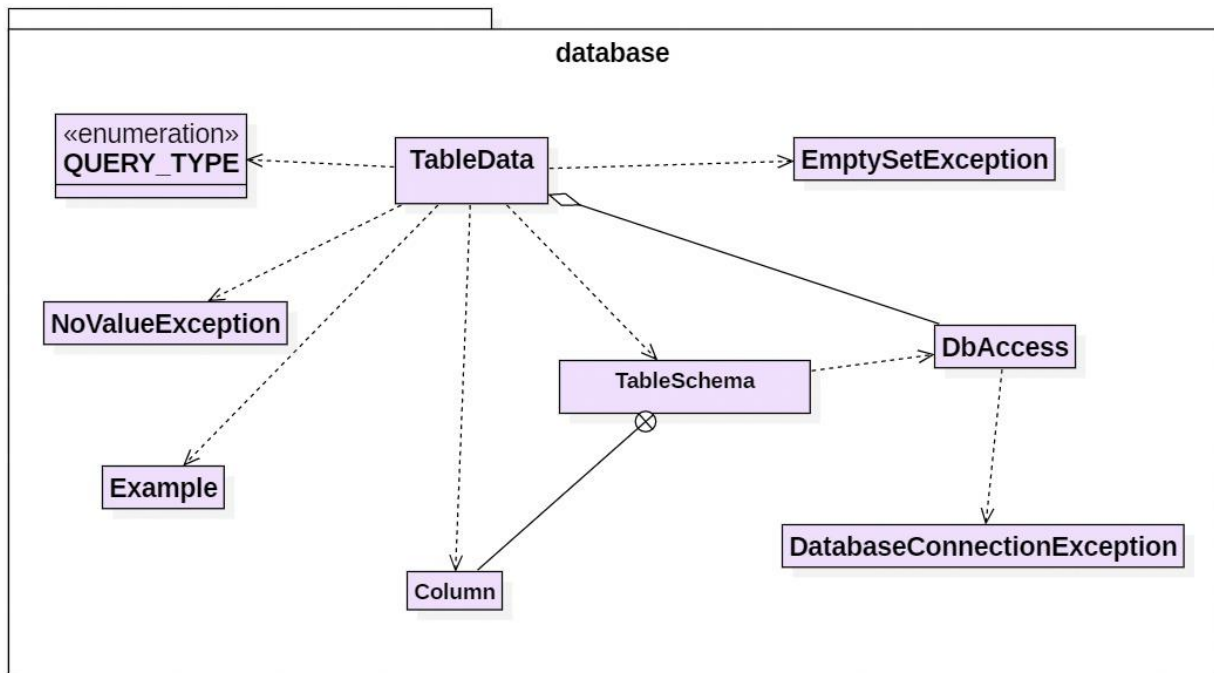
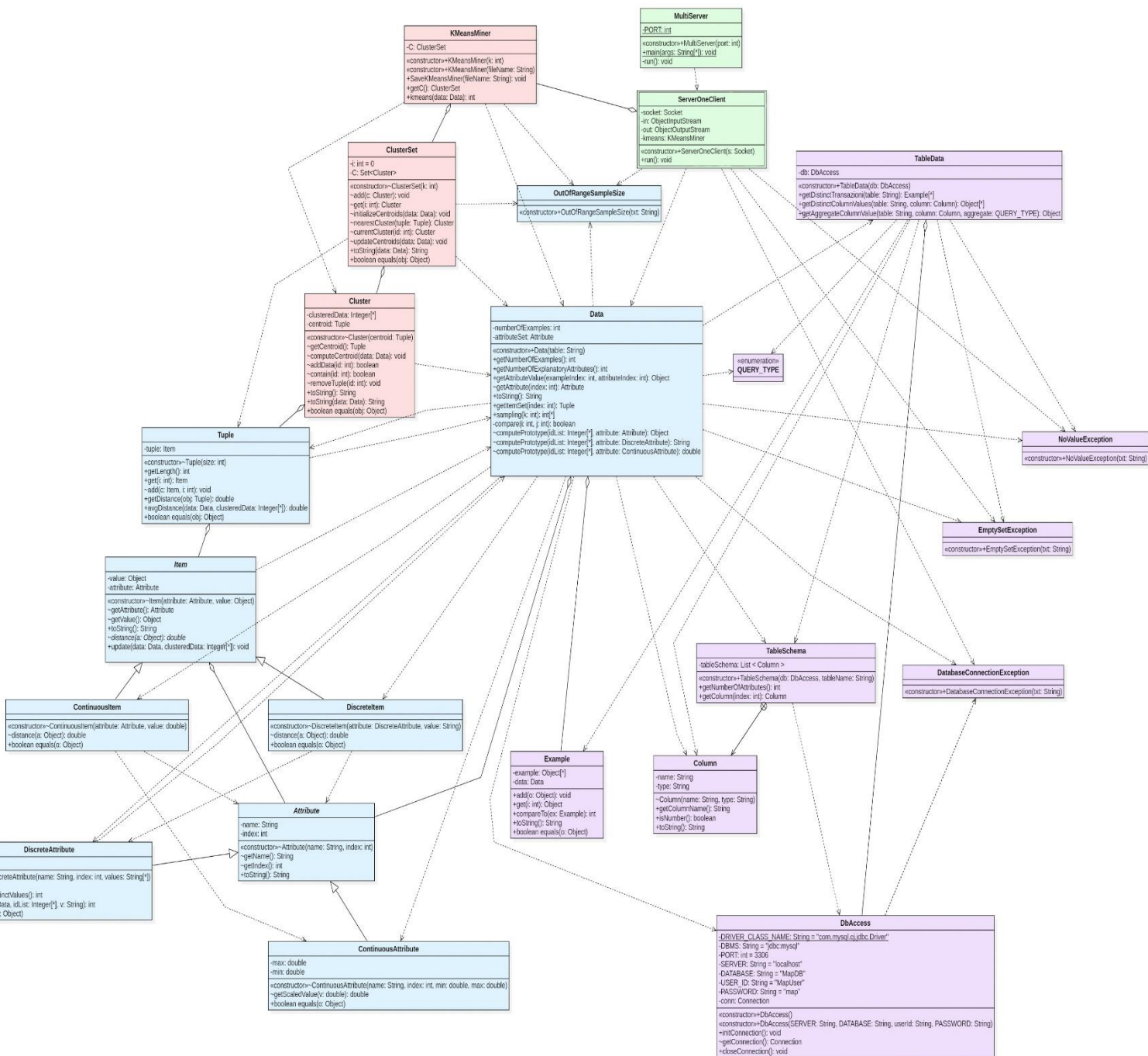


Diagramma delle classi

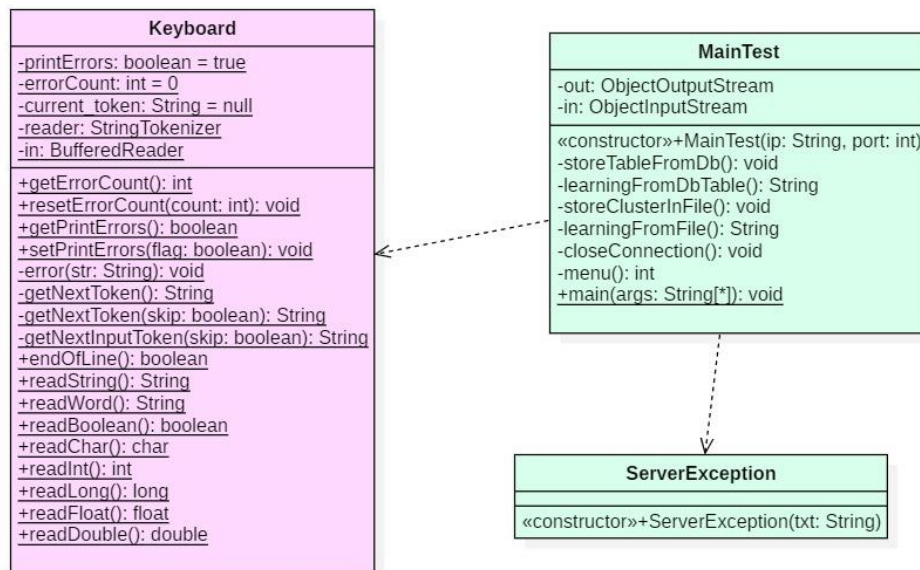


3.2 Client Console

Diagramma dei package



Diagramma delle classi

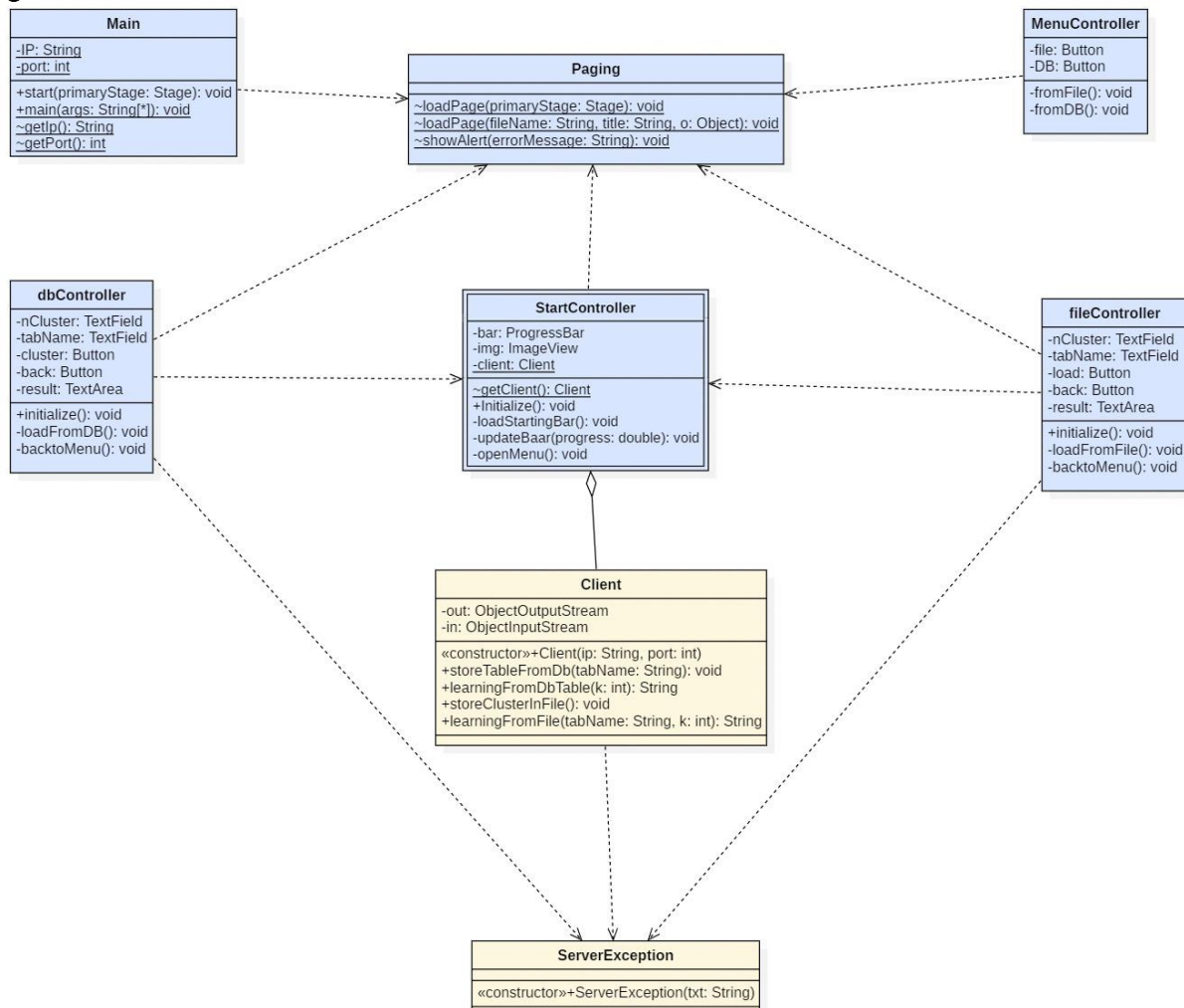


3.3 UI Client

Diagramma dei package



Diagramma delle classi



4. Manuale utente

Il software è composto da tre applicativi: un server che esegue le operazioni di clustering, salvataggio e caricamento file e due client per interagire con il server. Per entrambi i client (console o interfaccia grafica) è possibile usare lo stesso server. Per semplicità d'uso, client e server utilizzano un indirizzo IP e una porta preimpostati, è possibile modificarli aprendo il file *.bat*.

4.1 Requisiti

Per poter eseguire il software non sono necessarie procedure complesse. Tuttavia, per il corretto funzionamento è necessario che:

- Sia installata una versione aggiornata di Java SDK;
- Sia installato e attivo il servizio MySQL (versione 5.7 o successive);
- Sia eseguito lo script "createDB.sql" nel caso in cui non fosse già stato eseguito. Va specificato che è possibile utilizzare altre tabelle purché vengano inserite in mapDB;

4.2 Versione console

Guida di installazione

Prima di utilizzare il software, assicurarsi di aver rispettato i requisiti del punto precedente. È necessario che venga avviato il server prima di avviare il client. In caso contrario verrà visualizzato un messaggio di errore, interrompendo l'esecuzione del programma client. Per poter eseguire server e client in maniera automatica è possibile utilizzare i file **K-Means_Server.bat** e **K-Means_Client.bat** facendo doppio click. In alternativa, per gli utenti più esperti, è possibile utilizzare il comando da terminale *java -jar nomeFile.jar* recandosi nella cartella dei JAR.

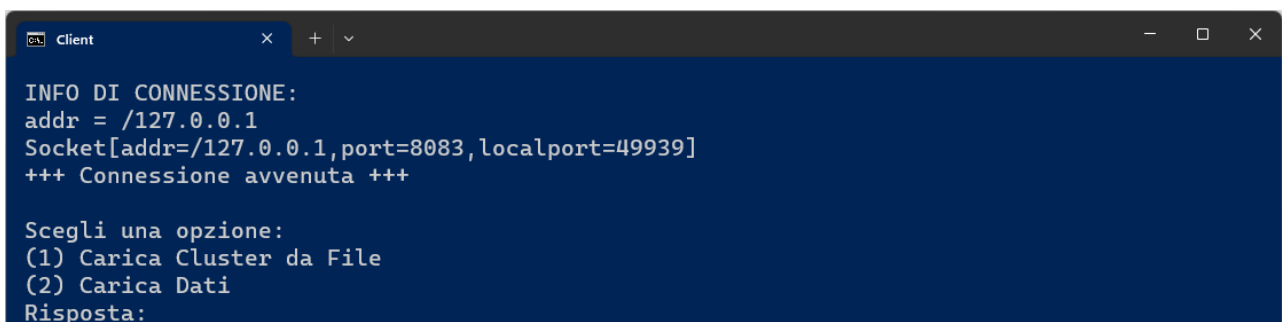
Guida Utente

Aprendo K-Means_Server.bat, avviene l'avvio del server. Visualizzando il messaggio in figura, il server è pronto a ricevere richieste dai client.



```
+++ Server ON +++
```

Aprendo K-Means_Client.bat, avviene l'avvio del client. Vengono visualizzate le informazioni di connessione e il menu di scelta:



```
INFO DI CONNESSIONE:  
addr = /127.0.0.1  
Socket[addr=/127.0.0.1,port=8083,localport=49939]  
+++ Connessione avvenuta +++  
  
Scegli una opzione:  
(1) Carica Cluster da File  
(2) Carica Dati  
Risposta:
```

L'utente quindi può scegliere tra due opzioni: caricare i dati da un file oppure scoprire dei nuovi cluster da una tabella di un database. Scegliendo l'operazione 2 dal menu sarà quindi possibile eseguire nuove clusterizzazioni, salvando in un file i risultati. Affinché i dati vengano clusterizzati sarà necessario indicare il nome della tabella da cui estrapolare i dati e quanti cluster generare:

```
Client
Nome tabella: playtennis
Numero di cluster: 5

Numero iterazioni: 3

1: 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

2: Centroid = (overcast 29.605 high weak yes)
Examples:
[overcast 30.0 high weak yes] dist=0.013036303630363011
[overcast 29.21 normal weak yes] dist=1.0130363036303631
AvgDistance=0.5130363036303631

3: Centroid = (overcast 8.366666666666667 normal strong yes)
Examples:
[overcast 0.1 normal strong yes] dist=0.27282728272827284
[sunny 12.5 normal strong yes] dist=1.1364136413641364
[overcast 12.5 high strong yes] dist=1.1364136413641364
AvgDistance=0.8485515218188485

4: Centroid = (rain 6.275 normal weak yes)
Examples:
[rain 13.0 high weak yes] dist=1.221947194719472
[rain 0.0 normal weak yes] dist=0.2070957095709571
[sunny 0.1 normal weak yes] dist=1.203795379537954
[rain 12.0 normal weak yes] dist=0.1889438943894389
AvgDistance=0.7054455445544555

5: 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

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

Dopo la visualizzazione dei risultati, viene segnalato all'utente che il file contenente la clusterizzazione è stato salvato. L'utente quindi può ripetere il processo scegliendo nuovamente il numero di cluster da generare.

Tornando alla schermata del server è possibile consultare lo stato in base alle informazioni che vengono visualizzate. La lettura delle stesse risulta immediatamente chiara: nel thread-0 è connesso il client identificato dall'IP 127.0.0.1 che richiede dei servizi che il server offre.

```
Server
+++ Server ON +++

[Thread-0: 127.0.0.1] Connessione in: jdbc:mysql://localhost:3306/MapDB?user=MapUser&password=map&serverTimezone=UTC

[Thread-0: 127.0.0.1] Lettura tabella da DB

[Thread-0: 127.0.0.1] Clusterizzazione dati

[Thread-0: 127.0.0.1] playtennis_5.dat: file salvato
```

Invece, selezionando l'opzione 1 dal menù precedente, sarà possibile leggere i dati da un file. È necessario inserire il nome della tabella e il numero di cluster in precedenza calcolati in quanto il file viene salvato sottoforma di *nomeTabella_numeroCluster.dat*:

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

Nome tabella: playtennis
Numero di cluster:3

1: 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

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

3: 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

Vuoi scegliere una nuova operazione da menu?(y/n)
```

Come è possibile vedere, il programma permette di tornare al menù principale per una nuova operazione. Anche in questo caso è possibile consultare lo stato del server:

```
Server

+++ Server ON +++

[Thread-0: 127.0.0.1] Connessione in: jdbc:mysql://localhost:3306/MapDB?user=MapUser&password=map&serverTimezone=UTC

[Thread-0: 127.0.0.1] playtennis_3.dat: file letto
```

Va specificato che durante l'esecuzione del client, se l'utente dovesse inserire informazioni sbagliate, tipo tabelle di mapDB vuote, inesistenti oppure un numero di cluster non valido, verranno visualizzati messaggi di errore. Ad ogni errore, sarà necessario reinserire i dati corretti per poter procedere con i calcoli e le letture dei file. Qualora il sever venga spento durante l'esecuzione del client, quest'ultimo verrà chiuso.

Esempi di test del client console

Descrizione	Input	Output effettivo
Avvio del client dopo aver avviato il server	∅	Informazioni di connessione + scelta dal menu
Avvio del client senza aver avviato il server	∅	Informazioni di connessione + messaggio di errore: connessione al server fallita, assicurati che sia online

Inserimento nel menu di un valore accettato	2	Inserimento del nome della tabella del database
Inserimento nel menu di un valore non accettato (intero)	3	Nuovo inserimento nel menu di un valore
Inserimento nel menu di un valore non accettato (stringa)	"text"	[MIN_VALUE value returned]: nuovo inserimento nel menu di un valore
Inserimento di una tabella esistente e popolata in mapDB	"playtennis"	Prossimo inserimento: numero cluster
Inserimento di una tabella (diversa da playtennis) esistente e popolata in mapDB	"salesadata"	Prossimo inserimento: numero cluster
Inserimento di una tabella non esistente in mapDB	"nonesiste"	Messaggio di errore: Tabella non trovata, sceglierne un'altra
Inserimento di una tabella esistente e non popolata in mapDB	"prova"	Messaggio di errore: tabella vuota, sceglierne un'altra
Inserimento di un numero di cluster compreso nel range [1- tuple playtennis]	5	Numero iterazioni + Risultato della clusterizzazione
Inserimento di un numero di cluster compreso nel range [1- tuple playtennis] – CASO LIMITE	14	Numero iterazioni + Risultato della clusterizzazione
Inserimento di un numero di cluster compreso nel range [1- tuple playtennis] – CASO LIMITE	1	Numero iterazioni + Risultato della clusterizzazione
Inserimento di un numero di cluster compreso non nel range [1- tuple playtennis]	-9	Messaggio di errore: valore inserito non valido, non può essere negativo
Inserimento di un numero di cluster compreso non nel range [1- tuple playtennis]	0	Messaggio di errore: valore inserito non valido, deve essere compreso tra 1 e 14
Inserimento di una stringa nel campo cluster (richiede un intero)	"text"	Messaggio di errore [MIN_VALUE value returned]: valore inserito non valido, deve essere compreso tra 1 e 14
Inserimento di un numero di cluster compreso non nel range [1- tuple playtennis]	20	Messaggio di errore: valore inserito non valido, deve essere compreso tra 1 e 14
Inserimento di una tabella esistente e popolata in mapDB + numero cluster (file esistente)	playtennis + 5	Risultato lettura da file playtennis_5
Inserimento di una tabella (diversa da playtennis) esistente e popolata in mapDB + numero cluster	salesadata + 9	Risultato lettura da file salesdata_9
Inserimento di una tabella non esistente in mapDB + numero cluster	Nonesiste + 8	Messaggio di errore: Tabella non trovata, sceglierne un'altra
Inserimento di una tabella esistente in mapDB + numero cluster (file non esistente)	Playtennis + 6	Messaggio di errore: File non trovato

4.3 Versione interfaccia grafica

Guida di installazione

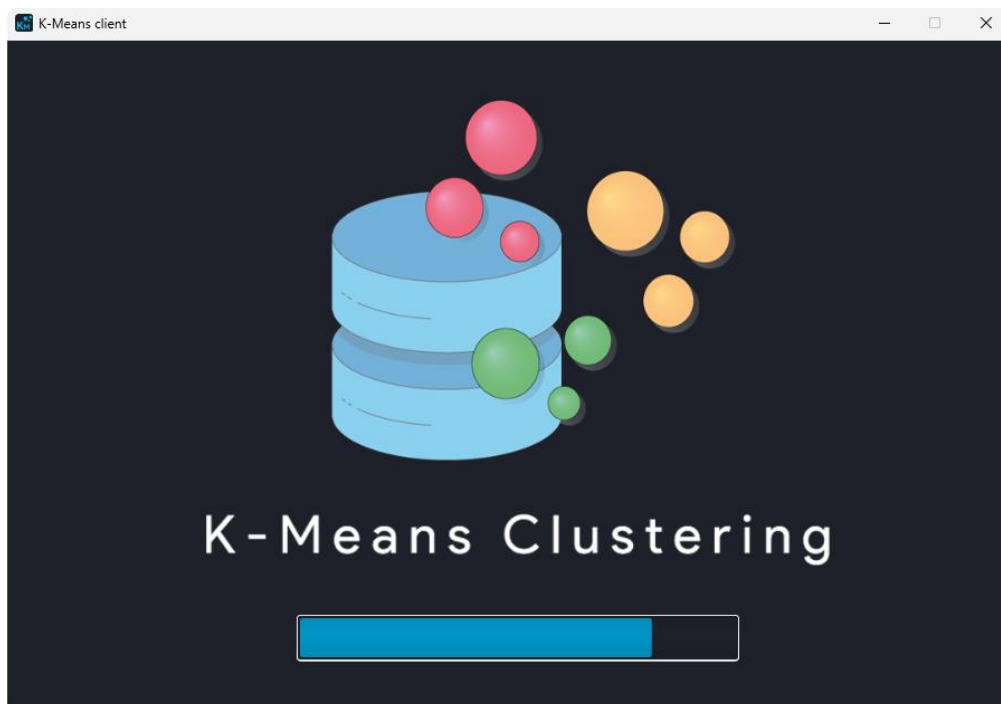
Anche in questo caso è necessario rispettare i requisiti al punto 4.1 e eseguire il server prima di aprire l'interfaccia. È possibile avviare automaticamente il programma facendo doppio click sul file **K-Means_UIClient.bat**. Si sconsiglia l'avvio manualmente attraverso il comando da terminale.

Guida Utente

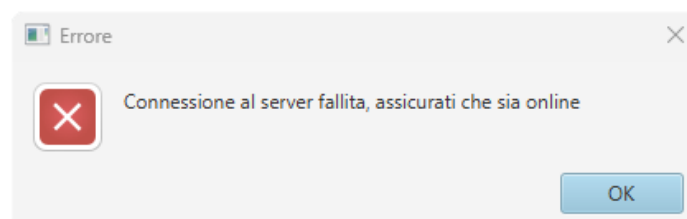
Proprio come la versione console, è necessario aprire **K-Means_Server.bat** per avviare il server che si metterà in ascolto di richieste dei client:



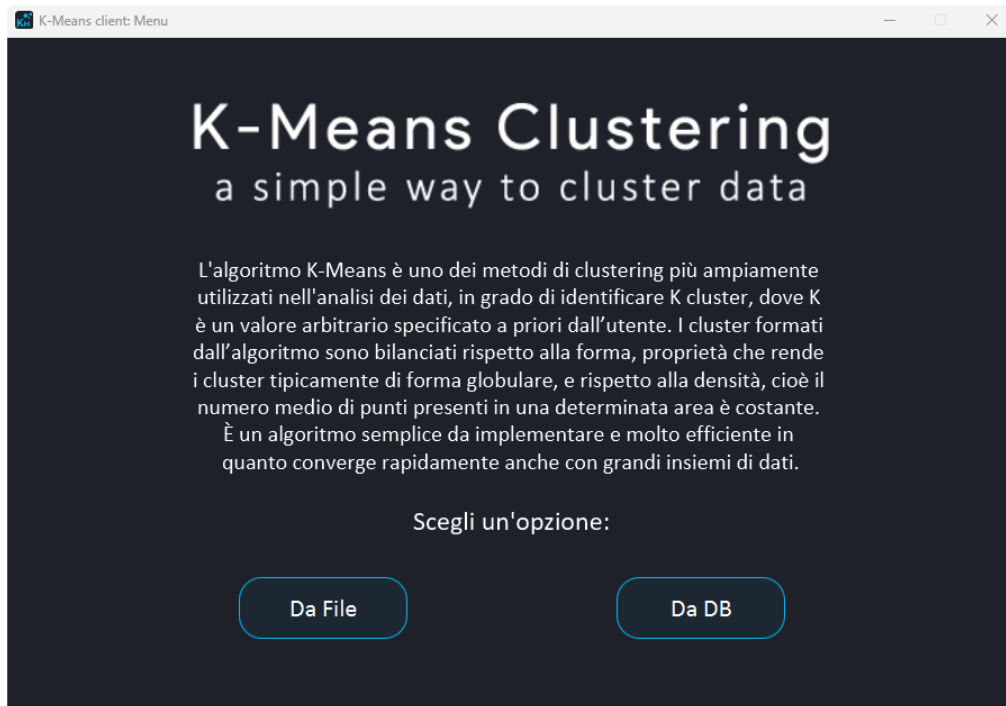
All'avvio del client verrà aperta una finestra in cui viene visualizzato il logo del progetto con la barra di caricamento:



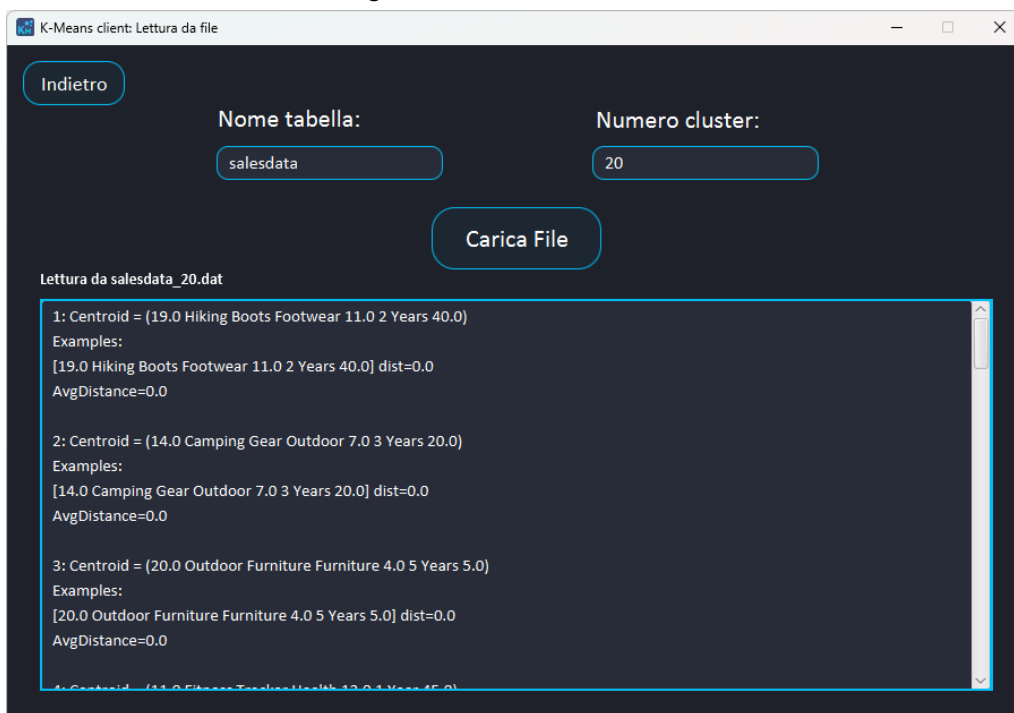
Nel caso in cui la connessione al server non vada a buon fine verrà visualizzato un messaggio di errore, prima della schermata di caricamento, che avvertirà della connessione fallita, bloccando l'esecuzione del programma:



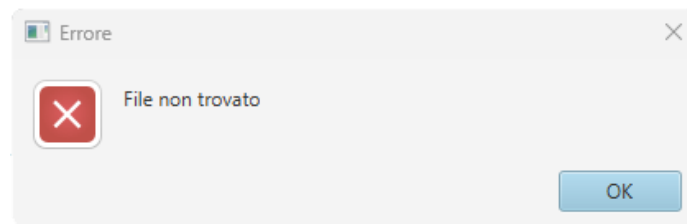
Al termine del caricamento verrà visualizzata la finestra del menù, composta da una breve descrizione dell'algoritmo applicato e da due tasti che portano alle pagine dedicate alle funzionalità del programma:



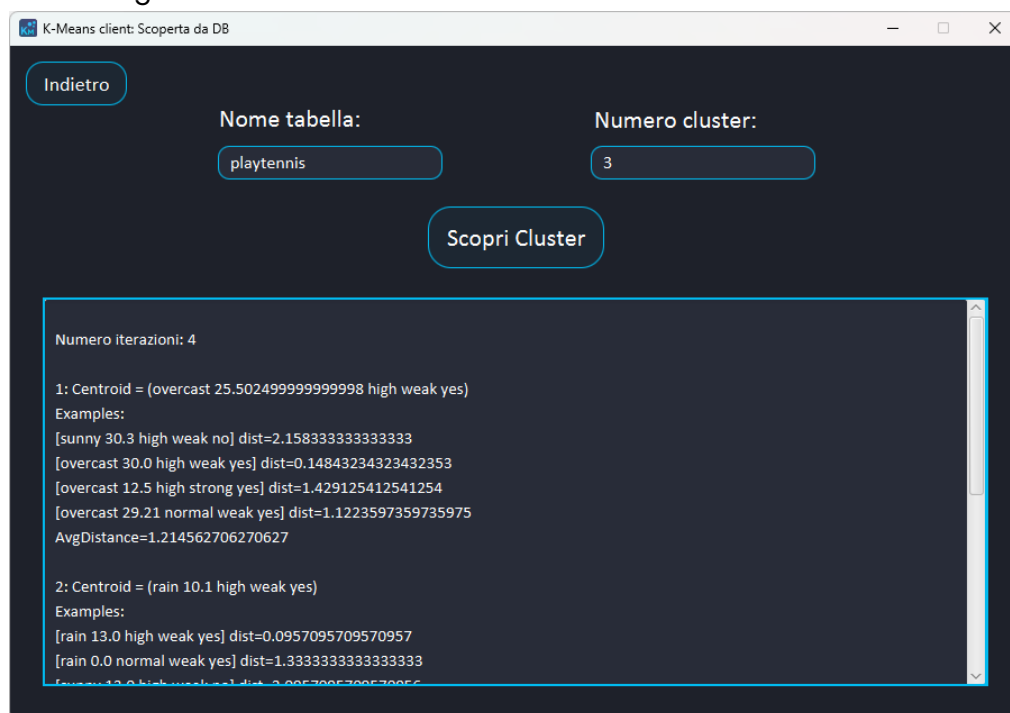
Se nella finestra iniziale si seleziona l'opzione "Da File" si accederà alla schermata dedicata alla lettura dei file salvati dalle precedenti clusterizzazioni. È necessario fornire al programma il nome della tabella e il numero di cluster affinché possa essere rintracciato il file. Una volta riempiti i campi necessari, basterà premere il tasto "Carica File" per visualizzare il risultato. Inoltre, sarà possibile tornare alla finestra iniziale grazie al tasto "Indietro".



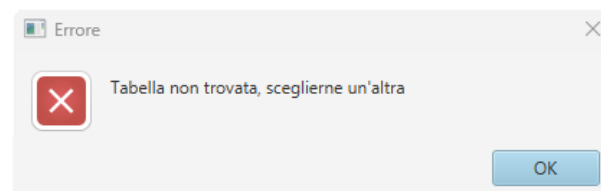
Se il file non viene trovato si verificherà un errore:



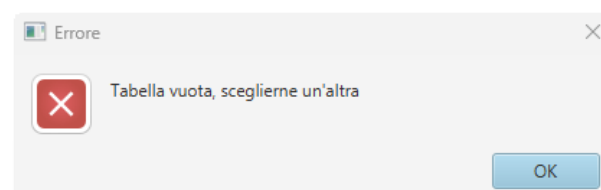
Se si seleziona l'opzione "Da DB" si accederà alla schermata dedicata alla clusterizzazione dei dati estrapolati da una tabella del DB. Anche in questo caso è necessario fornire al programma il nome della tabella e il numero di cluster da generare. Una volta riempiti i campi necessari, basterà premere il tasto "Scopri Cluster" per generare il risultato. Inoltre, sarà possibile tornare alla finestra iniziale grazie al tasto "Indietro".



Se la tabella non viene trovata si verificherà un errore:



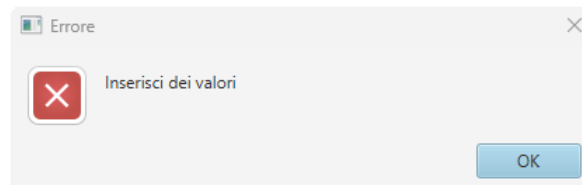
Se la tabella specificata non contiene dati si verificherà un errore:



Errori comuni ad entrambe le opzioni

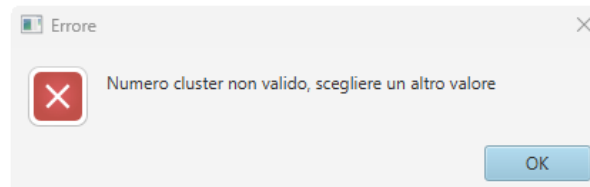
Nella scelta del nome della tabella e del numero cluster è possibile incorrere in errori sia nell'opzione "Da File" che "Da DB".

Se i campi contenenti le informazioni necessarie per generare il risultato non vengono compilati si verificherà un errore:

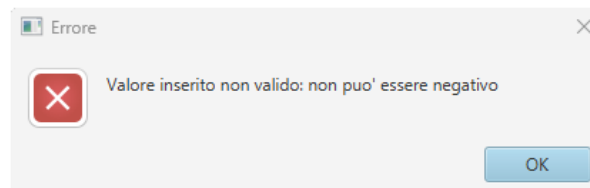


Quando si specifica il numero di cluster da generare è possibile incorrere in tre possibili errori:

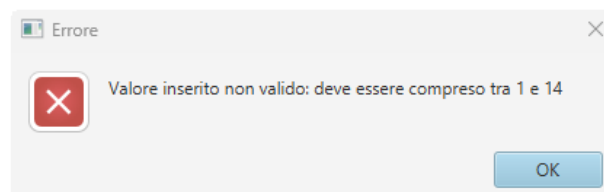
- Se nel campo numero di cluster viene inserita una stringa si verificherà un errore:



- Qualora il numero di cluster inserito dovesse essere negativo si verificherà un errore:



- Infine, se il valore inserito non dovesse essere compreso tra 1 e il numero di attributi della tabella scelta si verificherà un errore:



Esempi di test del client console

Descrizione	Input	Output
Avvio del client con il server spento	∅	Visualizzazione del messaggio di errore per la fallita connessione al server con conseguente chiusura del client
Avvio del client con il server acceso	∅	Corretta esecuzione del client
Click di una delle due opzioni nella schermata iniziale	Click su "Da File"	Visualizzazione della schermata dedicata alla selezione da locale del file

Click di una delle due opzioni nella schermata iniziale	Click su "Da DB"	Visualizzazione della schermata dedicata alla clusterizzazione da DB
Click del tasto in alto a sinistra per tornare indietro	Click su "Indietro"	Visualizzazione della finestra iniziale
Inserimento del nome della tabella non valido "Da File"	playerror	Visualizzazione del messaggio di errore per l'errato inserimento dei dati (file non trovato)
Inserimento del nome della tabella non valido "Da DB"	playerror	Visualizzazione del messaggio di errore per l'errato inserimento dei dati (tabella non trovata)
Inserimento del nome di una tabella vuota "Da DB"	playtennis(vuota)	Visualizzazione del messaggio di errore (tabella vuota)
Inserimento del nome della tabella non valido + numero cluster nell'opzione "Da File"	playerror + 3	Visualizzazione del messaggio di errore per l'errato inserimento dei dati (file non trovato)
Inserimento del nome della tabella corretto in una delle due opzioni scelte ("Da File" o "Da DB")	playtennis	Visualizzazione del messaggio di errore per il mancato inserimento dei dati (numero cluster mancante)
Inserimento del numero di cluster in una delle due opzioni scelte ("Da File" o "Da DB")	5	Visualizzazione del messaggio di errore per il mancato inserimento dei dati (nome tabella mancante)
Inserimento del nome della tabella non valido + numero cluster nell'opzione "Da DB"	playerror + 3	Visualizzazione del messaggio di errore per l'errato inserimento dei dati (tabella non trovata)
Inserimento del nome della tabella corretto + numero cluster non valido in una delle due opzioni scelte ("Da File" o "Da DB")	playtennis + str	Visualizzazione del messaggio di errore per l'errato inserimento dei dati (numero cluster non valido)
Click del tasto per la generazione del risultato in una delle due opzioni scelte ("Da File" o "Da DB")	Click sul tasto per generazione del risultato	Visualizzazione del messaggio di errore per non aver inserito i dati necessari
Inserimento del nome della tabella corretto + numero cluster valido in una delle due opzioni scelte ("Da File" o "Da DB")	playtennis + 5	Generazione corretta del risultato