K-MEANS

Progetto MAP 2022/2023

Author: Michele Fraccalvieri – MAT: 724204 – m.fraccalvieri8@studenti.uniba.it

1 Introduzione

K-Means è un'applicazione che sfrutta l'algoritmo K-Means per creare cluster di dati a partire da una tabella in un database SQL. I cluster sono gruppi di dati omogenei che condividono caratteristiche simili, mentre i centroidi rappresentano i punti centrali di ciascun cluster. L'obiettivo dell'algoritmo K-means è suddividere un insieme di dati non etichettati in sottogruppi (cluster) basati sulle somiglianze tra le loro caratteristiche.

2 Algoritmo K-Means

L'algoritmo K-Means è uno degli algoritmi di clustering più utilizzati nell'ambito del data mining e dell'apprendimento automatico. Esso ci permette di suddividere un insieme di dati non etichettati in gruppi omogenei chiamati cluster, facilitando l'identificazione di pattern nascosti e la comprensione delle relazioni tra i dati.

L'algoritmo K-Means si basa su un processo iterativo che coinvolge due fasi principali: l'assegnazione dei dati ai cluster e l'aggiornamento dei centroidi. Il termine "K" nell'algoritmo K-Means indica il numero desiderato di cluster in cui si desidera suddividere i dati.

Fase 1: Inizializzazione dei Centroidi

Selezionare casualmente K punti dal dataset come centroidi iniziali. Ogni punto rappresenterà il centro del suo cluster.

Fase 2: Assegnazione dei Dati ai Cluster

Per ogni punto del dataset, calcolare la distanza rispetto a ciascun centroide. Assegnare il punto al cluster del centroide più vicino.

Fase 3: Aggiornamento dei Centroidi

Calcolare la media delle coordinate di tutti i punti appartenenti a ciascun cluster. Utilizzare la media calcolata come nuovo centroide del cluster.

Fase 4: Iterazione

Ripetere le Fasi 2 e 3 finché i centroidi non cambiano significativamente o per un numero massimo di iterazioni prestabilito.

L'algoritmo K-Means è un potente strumento per il clustering dei dati che ci consente di suddividere un insieme di dati non etichettati in gruppi omogenei. Attraverso il processo iterativo di assegnazione dei dati ai cluster e l'aggiornamento dei centroidi, K-Means raggiunge una convergenza verso cluster ottimali.

3 Architettura del progetto

Il progetto K-MEANS è stato sviluppato quasi interamente con l'IDE Intellij IDEA, e con Eclipse per le fasi iniziali di progettazione. Esso presenta un sistema client-server. Il server include funzionalità di data mining per la scoperta di cluster di dati. Il client è un applicativo Java che consente di usufruire del servizio di scoperta remoto e visualizza la conoscenza (cluster) scoperta.

4 Istruzioni per l'installazione

Per eseguire il progetto K-MEANS sul proprio ambiente locale, seguire le seguenti istruzioni:

1. Installare la Java Runtime Environment, versione 16 o superiore
2. Installare il DBMS MySql, versione 5.7 o superiore
3. Eseguire il server MySql e successivamente lo script "createDB" presente nel seguente percorso: "K-MEANS\VersioneBase\Server\createDB.sql"

5 Guida utente

5.1 Avvio del server

Per garantire il corretto funzionamento del programma, è necessario avviare prima il server. Per fare ciò, è sufficiente eseguire il file "ServerLauncher.bat" presente nel percorso "K-MEANS\VersioneBase\Server\ServerLauncher.bat utilizzando uno dei seguenti metodi:

- Doppio click sul file
- Eseguire il file tramite riga di comando

In questo modo il server verrà correttamente avviato sulla porta predefinita 8080.

5.2 Avvio del client

L'avvio del client è molto simile a quello del server. Bisogna eseguire il file "ClientLauncher.bat" presente nel percorso "K-MEANS\VersioneBase\Client\ClientLauncher.bat" utilizzando uno dei seguenti metodi:

- Doppio click sul file

- Eseguire il file tramite riga di comando

In questo modo il client verrà correttamente avviato con i parametri di default (localhost e 8080).

5.3 Esempi di test

Se tutti i passi precedentemente descritti sono stati correttamente seguiti, nel client verrà visualizzata la seguente schermata:

```
Server is listening on port 8080
New client connected
```

Nella schermata del client invece, sarà possibile effettuare l'operazione desiderata:

```
addr = localhost/127.0.0.1
Socket[addr=localhost/127.0.0.1,port=8080,localport=64390]
Scegli una opzione
(1) Carica Cluster da File
(2) Carica Dati
Risposta:
```

Esempio di test per l'opzione "Carica dati":

```
Scegli una opzione
(1) Carica Cluster da File
(2) Carica Dati
Risposta:2
Nome tabella:playtennis
Numero di cluster:3
Clustering output:Numero di iterazioni: 3

0:Centroid=(rain 5.0200000000000005 normal weak yes )
Examples:
[rain 13.0 high weak yes ] dist=1.2633663366336634
[rain 0.0 normal weak yes ] dist=0.1656765676567657
[rain 0.0 normal strong no ] dist=2.1656765676567655
[sunny 0.1 normal weak yes ] dist=1.1623762376237623
[rain 12.0 normal weak yes ] dist=0.2303630363036303
AvgDistance=0.9974917491749175

1:Centroid=(overcast 13.5775 normal strong yes )
Examples:
[overcast 0.1 normal strong yes ] dist=0.4448019801980198
[sunny 12.5 normal strong yes ] dist=1.0355610561056106
[overcast 12.5 high strong yes ] dist=1.0355610561056106
[overcast 29.21 normal weak yes ] dist=1.515924092409241
AvgDistance=1.0079620462046206

2:Centroid=(sunny 23.22 high weak no )
Examples:
[sunny 30.3 high weak no ] dist=0.23366336633663376
[sunny 30.3 high strong no ] dist=1.2336633663366339
[overcast 30.0 high weak yes ] dist=2.223762376237624
[sunny 13.0 high weak no ] dist=0.3372937293729372
[rain 12.5 high strong no ] dist=2.353795379537954
AvgDistance=1.2764356435643565
```

Il relativo salvataggio del cluster scoperto:

```
Inserire il nome del file in cui salvare i cluster trovati:
prova
File salvato.
Vuoi ripetere l'esecuzione?(y/n)
```

Esempio di test per l'opzione "Carica cluster da file":

```
Scegli una opzione
(1) Carica Cluster da File
(2) Carica Dati
Risposta:1
Nome file:prova
0:Centroid=(rain 5.0200000000000005 normal weak yes )
1:Centroid=(overcast 13.5775 normal strong yes )
2:Centroid=(sunny 23.22 high weak no )

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

6 Diagrammi UML

Diagramma delle classi nel package Data:

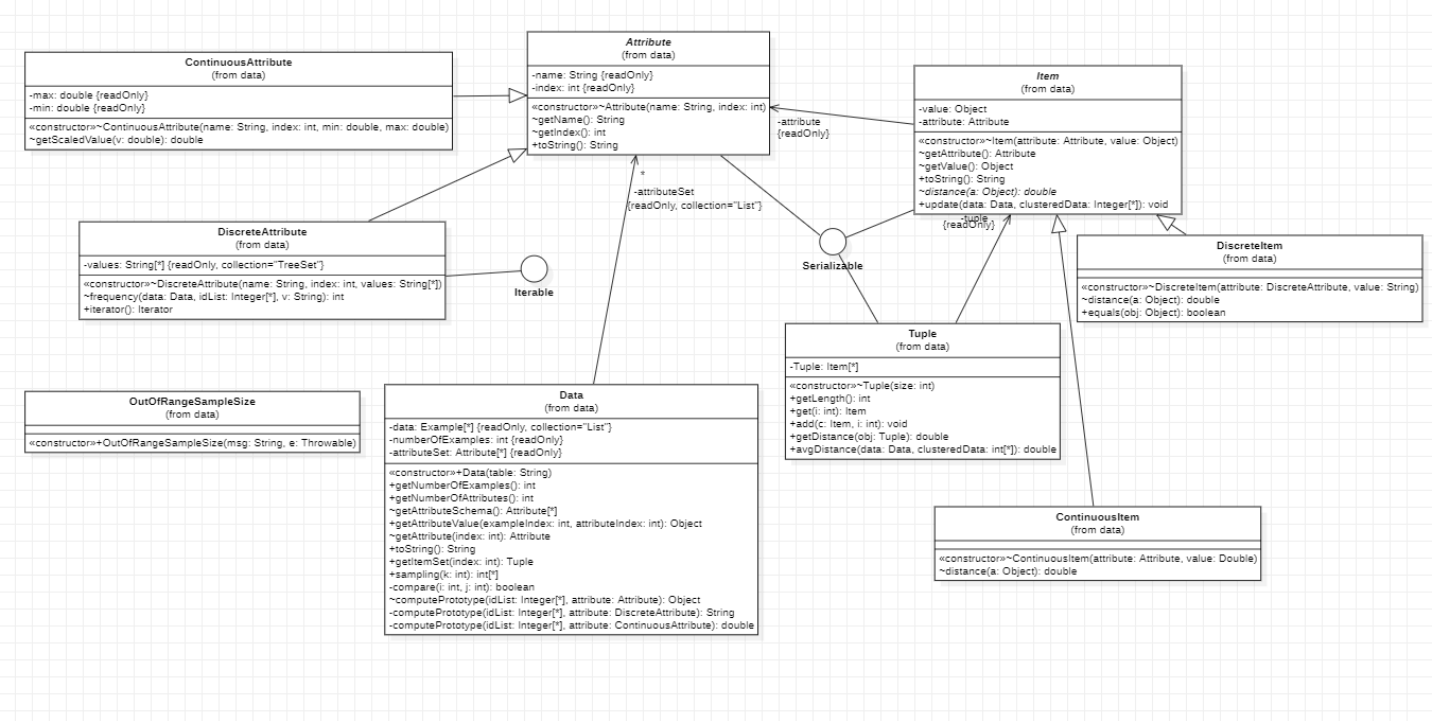


Diagramma delle classi nel package Database:

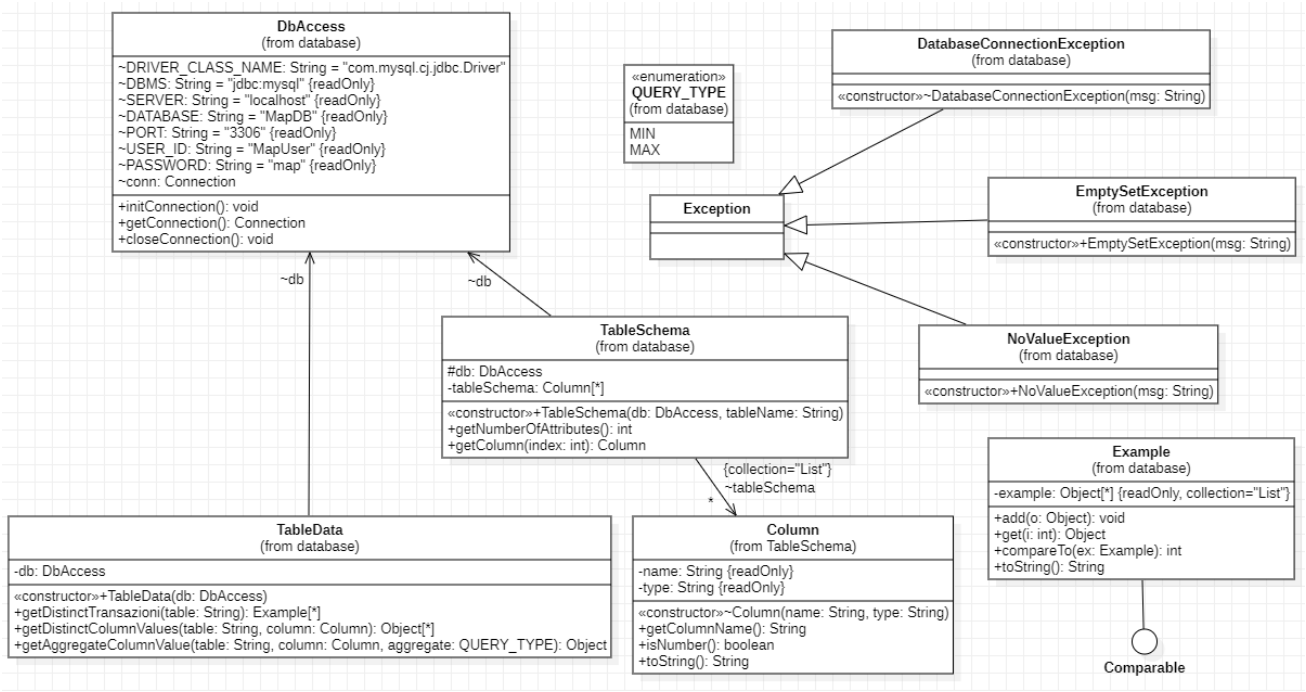


Diagramma delle classi nel package Mining:

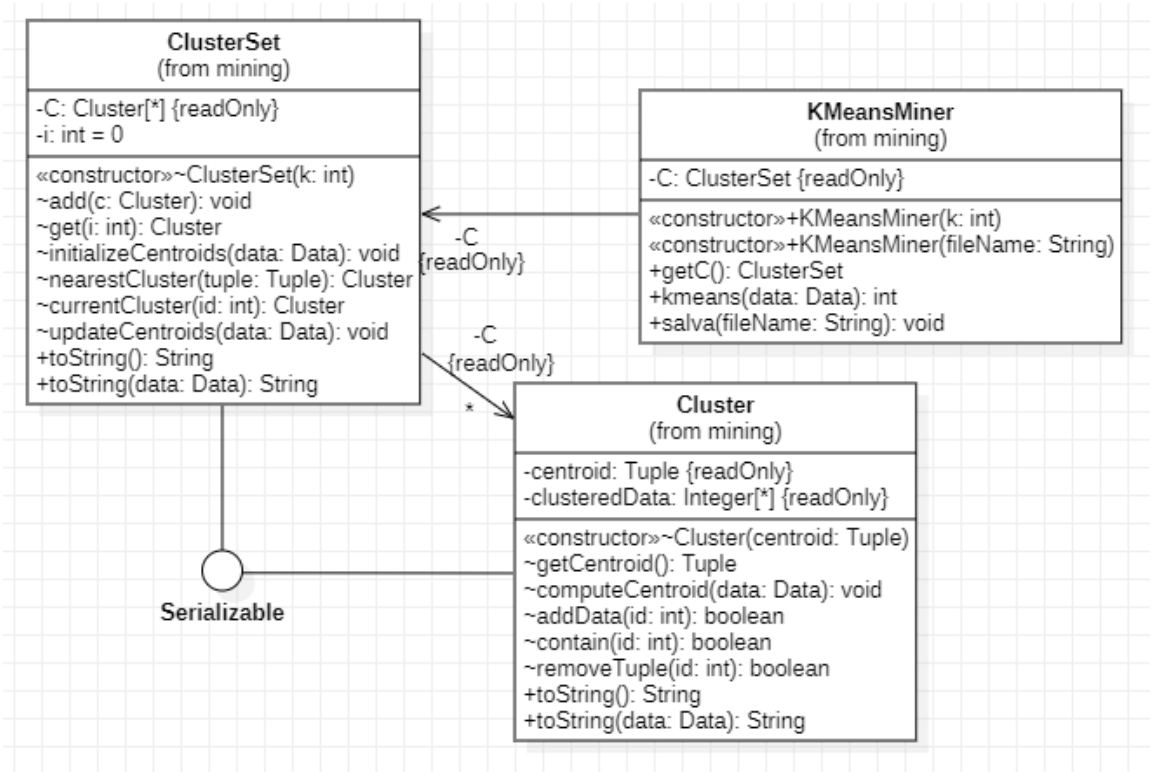


Diagramma delle classi Client:

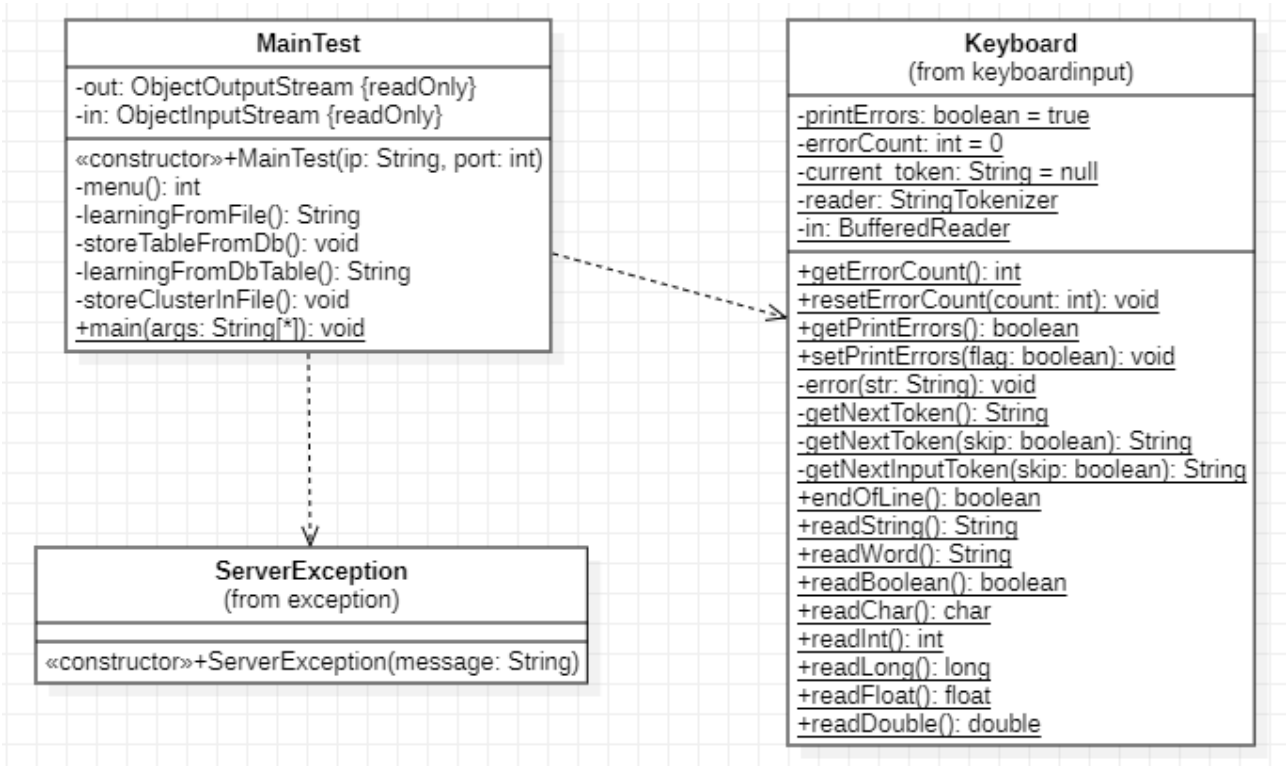
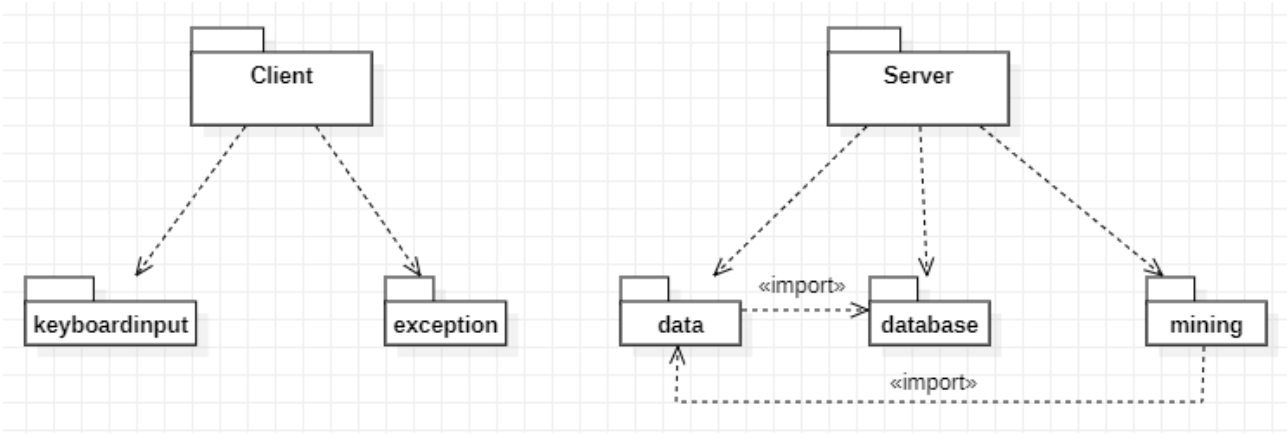


Diagramma dei package:



7 Javadoc

La documentazione del client e del server è presente nei seguenti percorsi:

- “K-MEANS\VersioneBase\Client\build\docs\javadocs\allclasses-index”
- “K-MEANS\VersioneBase\Server\build\docs\javadocs\allclasses-index”