PROGETTO KMEANS (2023/2024)

Indice

[da completare]

TeamProgetto [ALTRE INFO? MAGARI IN TABELLA]

Cirilli Davide

Curri Mattia

Fontana Emanuele

DESCRIZIONE DEL 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 dato 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 dovra' essere eseguito su una macchina con un database MySQL in esecuzione.

Il servizio sara' raggiungibile sulla porta 8080, e potra' 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 salverà informazioni relativi agli errori nel file di logging mentre nell’interfaccia CLI scriverà quando la comunicazione con client si interrompe. Il client da riga di comando permette di collegarsi ad una macchina che sta eseguendo un'istanza del server. L'utente nel menu' avra le due opzioni di lettura o scoperta di cluster

**Progetto esteso**

**Guida di installazione [da rivedere, l ho copiata]**

I file .jar , .bat e dove necessario .sql sono posti nella cartella out\artifacts\esegubili 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)**

Per eseguire correttamente il client e' necessario:

aver installato la Java Runtime Environment 8 o superiore

avere un server in ascolto

**Avvio del Server (progetto base) [da vedere]**

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) [da vedere]**

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 connetterà 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) [da vedere]**

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)**

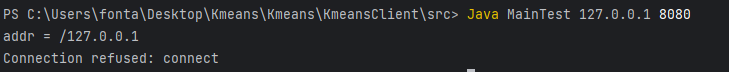
MANUALE UTENTE

//PARTE RELATIVA ALL’AVVIO DEL PROGRAMMA MEDIANTE IL .bat, esecuzione script sql per creazione del db, etc…

//PROOBABILMENTE GLI SCREEN SONO DA RIFARE E MATTIA LO VORRA’ FARE IN LATEX, MA ALMENO DO UNA BASE, MEGLIO DI NIENTE

Esempi di test

1. Esecuzione del client con server offline



La classe Socket in Java usa il protocollo di trasporto TCP (Transmission Control Protocol) il quale cerca di connettersi a un socket entro un certo tempo, se non ci riesce rifiuta la connessione. In questo caso questo accade in quanto il server è offline

1. Esecuzione senza parametri

Immagine che contiene testo, schermata, Carattere, linea

Descrizione generata automaticamente

Il programma deve essere avviato fornendo come parametro il socket del processo server. In particolare, bisogna fornire l’indirizzo IP/DNS del server e la porta logica. Se si lavora con la stessa macchina si può inserire come prima parametro localhost, 127.0.0.1 (indirizzo IPv4 locale) oppure ::1 (indirizzo IPv6 locale)

1. Esecuzione con porta errata



La porta logica è un numero a 16 bit, dunque in decimale compreso tra 0 e 65.535

1. Esecuzione con IP/DNS inesistente



In questo caso di esecuzione il server non esiste e dunque il client non riesce a connettersi. In particolare, in questo caso, il programma si connette a un server DNS per convertire l’indirizzo fornito in un indirizzo IP ma tale indirizzo non è registrato e dunque si ha errore

1. Esecuzione con parametri corretti

Immagine che contiene testo, Carattere, schermata

Descrizione generata automaticamente

Quando il programma viene eseguito nelle condizioni funzionali (quindi server avviato e parametri validi) vengono stampati a schermo indirizzo IP e porta del server, seguiti dalla porta che sta usando il processo per la comunicazione. Viene poi mostrato all’utente un menù con due opzioni. L’opzione numero (1) permetterà al client di caricare un cluster di dati che è stato serializzato sul server come un file, mentre l’opzione numero (2) consente di creare un nuovo cluster di dati mediante l’algoritmo del KMeans.

Immagine che contiene testo, Carattere, schermata

Descrizione generata automaticamente

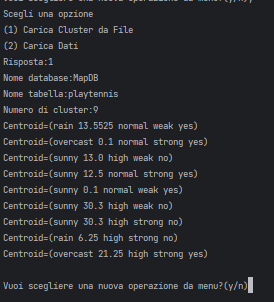
In caso di input non validi viene chiesto all’utente di reinserire finchè non si avrà un’opzione valida.

CASO (1)

Immagine che contiene testo, schermata, Carattere

Descrizione generata automaticamente

Viene chiesto all’utente di inserire il nome del database, della tabella e del numero di cluster creati. Il server andrà dunque a cercare il file corrispondente a tali informazioni e, se non trovato, manderà un messaggio di errore all’utente (come nell’esempio). Viene chiesto all’utente se vuole tornare al menù oppure terminare l’esecuzione del programma. Se viene premuto “n” il programma termina.



Nel caso in cui il file corrispondente alle richieste esista viene inviato al client il contenuto del file, ovvero i centroidi dei cluster.

CASO (2)

Immagine che contiene testo, schermata, Carattere

Descrizione generata automaticamente

Viene chiesto all’utente se vuole usare i valori di default o meno per effettuare una prova. I valori di default sono:

* Localhost -> server database
* 3306 -> porta dabasase
* MapDB -> nome database
* Playtennis -> nome tabella
* MapUser -> nome utente
* Map -> password

Immagine che contiene testo, schermata, menu, Carattere

Descrizione generata automaticamente

Nel caso di risposta affermativa viene chiesto all’utente il numero dei cluster. Una volta confermati (se questi sono validi) il server eseguirà l’algoritmo di KMeans e invierà il risultato al client. Viene poi chiesto se si vuole riprendere l’esecuzione sullo stesso dataset o meno.

Immagine che contiene testo, schermata, software, Carattere

Descrizione generata automaticamente

Esempio di numero di cluster non validi (in particolare il numero massimo indica quanti sono i dati distinti presenti nel database)

Se l’utente non vuole ripetere l’esecuzione sul dataset ha due scelte:

1. Tornare al menu
2. Terminare l’esecuzione

Immagine che contiene testo, schermata, Carattere

Descrizione generata automaticamente

L’utente può scegliere di non usare i valori di default. In quel caso deve inserire tutte le informazioni necessarie da terminale.

Immagine che contiene testo, schermata

Descrizione generata automaticamente

In caso di valori non validi verrà segnalato l’errore all’utente, il quale sceglierà se proseguire con i valori di default oppure se riprovare a inserire dei valori personalizzati.

L’uso di valori personalizzati permette di utilizzare dataset già presenti nel server del database.

ESEMPIO

Immagine che contiene testo, schermata, software

Descrizione generata automaticamente

Ogni volta che l’utente richiede la creazione di un dataset al server, quest’ultimo serializzerà i cluster in un file con nome del tipo: NomeserverNometabellaNumerocluster.dat

GUIDA UTENTE ESTENSIONE