

Documentazione

Il progetto è stato realizzato da:

- Colavito Giuseppe (675824)
mail: giuseppecolavito999@gmail.com | g.colavito2@studenti.uniba.it
- Gasparro Paolo (677711)
mail : gasparro.paolo@gmail.com | p.gasparro4@studenti.uniba.it
- Grisulli Antonio (682779)
mail : antoniogrisulli23@gmail.com | a.grisulli@studenti.uniba.it

Il software realizzato utilizza l'algoritmo QualityThreshold elaborando dati estratti da tabelle di database utilizzando il servizio MySQL.

L'algoritmo del QualityThreshold (QT) è un metodo alternativo di partizionamento dei dati, inventato per il clustering genico. Richiede più potenza di calcolo rispetto a k-means, ma non richiede di specificare il numero di cluster a priori e restituisce sempre lo stesso risultato quando viene eseguito più volte.

L'algoritmo è così definito:

1. Scegliere un raggio massimo per i cluster (operazione effettuata dall'utente).
2. Costruire un cluster candidato per ogni punto includendo i punto con distanza inferiore al raggio.
3. Salvare il cluster candidato con il maggior numero di punti come primo vero cluster e rimuovere tutti i punti del cluster per evitare che vengano riesaminati. Si stabilisce cosa succede se più di un cluster ha il numero massimo di punti.
4. Ripetere dal punto 2 con il set ridotto di punti se il set non è vuoto

Il progetto base consiste in una applicazione distribuita client/server.

Il server deve essere lanciato specificando una porta, oppure verrà usata di default la 8080.

Il server si occupa di ricevere le richieste di uno o più client, i quali possono effettuare le seguenti operazioni:

1. Richiedere di caricare una tabella da database, indicandone il nome
2. Eseguire il clustering sulla tabella caricata e visualizzarne i risultati
3. Salvare il risultato del clustering su file (lato server)
4. Leggere da file un clustering effettuato in precedenza

5. Chiudere la connessione

È presente anche una operazione per chiedere al server i nomi delle tabelle disponibili che, tuttavia, non viene utilizzata nella versione base in cui invece l'utente deve conoscere a priori il nome delle tabelle che vuole utilizzare.

Il Client deve essere lanciato specificando indirizzo e porta del server.

Una volta stabilita la connessione, l'utente può decidere tra:

1. Caricare un clustering effettuato in precedenza da file
2. Chiedere al server di effettuare un nuovo clustering

In entrambi i casi dovrà specificare il nome della tabella e il raggio.

Il client chiederà al server in automatico di salvare su file i nuovi clustering effettuati.

Se dovesse verificarsi un errore durante l'esecuzione delle operazioni, l'applicazione notificherà all'utente l'errore avvenuto, dando eventualmente la possibilità di ripetere l'inserimento dei dati.

Al termine dell'operazione e della visualizzazione dei risultati, il client chiede all'utente se vuole inserire un nuovo raggio sulla stessa tabella e, in caso di risposta negativa, permette di selezionare una nuova operazione dal menu iniziale oppure di chiudere l'applicazione.

L'estensione del progetto base è una interfaccia grafica per smartphone android.

Il client potrà quindi essere avviato sul dispositivo mobile per inviare richieste al server.

Il client android, a differenza di quello base, può essere lanciato anche senza specificare indirizzo e porta, perché in questo caso si collegherà di default all'indirizzo "paologas91.ddns.net" alla porta "8080", che corrisponde all'indirizzo di un server di proprietà del gruppo di sviluppo. Se si vuole modificare questa opzione, si può accedere alle impostazioni e specificare indirizzo e porta alternativi.

Un'altra differenza consiste nella scelta delle tabelle: mentre nel client base l'utente doveva conoscere a priori il nome della tabella, sull'app android viene visualizzato uno spinner dal quale selezionare, tra le tabelle esistenti sul database, quella desiderata.

Guida di installazione

Requisiti server

Per utilizzare il server è necessario:

- installare [MySQL](#) sul proprio computer
- eseguire lo script sql (script.sql) presente nel percorso “\QTServer\sql”
Esempio : \. C:\Users\giuse\Desktop\QTServer\sql\script.sql
Digitando questo comando sulla shell di MySQL sarà creato il database e le tabelle necessarie.
- Installare [JRE](#) 8

Requisiti client

Per utilizzare il client è necessario:

- Installare [JRE](#) 8
- Server in ascolto

Requisiti app-client

Per utilizzare l'app è necessario:

- Un dispositivo android con API>15 (Android 4.1 JellyBean) ma è consigliato avere un dispositivo con API>22 (Android 5.1 Lollipop). (Controlla la tua versione in Impostazioni > Informazioni Dispositivo > Versione Android oppure verifica se ci sono nuovi aggiornamenti.
- Server in ascolto
- Connessione ad internet

Installazione apk

Trasferire l'apk sullo smartphone (\QTAndroid\app\release\app-release.apk). Dalle Impostazioni>Sicurezza attivare l'opzione Origini Sconosciute.

Cliccare sull'apk e scegliere Installa. Nel caso in cui l'installazione sia bloccata da Play Protect cliccare su “Installa comunque”.

Guida utente

Avvio server

Per avviare il server è necessario eseguire il file batch (.bat) il cui percorso è:

“\QTServer\build\distributions\QTServer\bin\QTServer.bat”

Si può sia lanciarlo cliccandolo o da riga di comando senza parametri (in questo caso la connessione sarà avviata di default sulla porta 8080), sia lanciarlo da riga di comando inserendo come parametro la porta desiderata.

Avvio client

Per avviare il client è necessario che il server sia attivo. Va quindi eseguito il file batch al percorso: “\QTClient\build\distributions\QTClient\bin\QTClient.bat”

Da riga di comando bisogna specificare IP e PORTA, oppure modificando il .bat a riga 85, inserendo IP e PORTA al posto della stringa “%CMD_LINE_ARGS%”.

```
82 set CLASSPATH=%APP_HOME%\lib\QTClient.jar;%APP_HOME%\lib\guava-27.0.1-jre.jar;%APP_HOME%\lib\failureaccess-1.0.1.jar;%AP
83
84 @rem Execute QTClient
85 "%JAVA_EXE%" %DEFAULT_JVM_OPTS% %JAVA_OPTS% %QT_CLIENT_OPTS% -classpath "%CLASSPATH%" Client %CMD_LINE_ARGS%
86
87 :end
88 @rem End local scope for the variables with windows NT shell
```

Guida utilizzo client-app

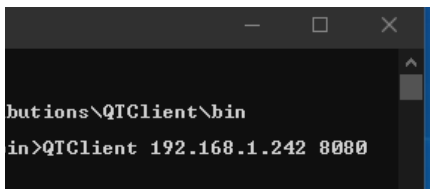
Una volta avviato il client e stabilita la connessione, si potrà selezionare tra:

- 1) Caricare clusters da file
- 2) Creare clusters dal database

In entrambi i casi bisognerà inserire nome della tabella e raggio.

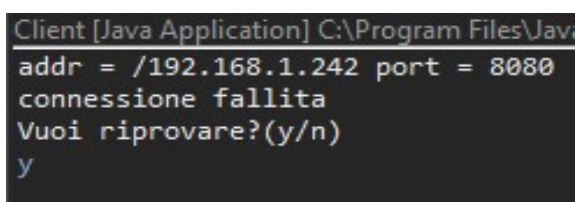
Una volta selezionati un nome di tabella esistente e un raggio non troppo grande e diverso da zero, si potranno visualizzare i cluster.

Esempi di uso Client



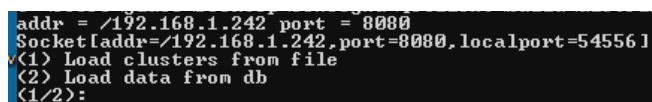
```
hutions\QTClient\bin
in>QTClient 192.168.1.242 8080
```

1) Ecco un possibile comando per lanciare il client.



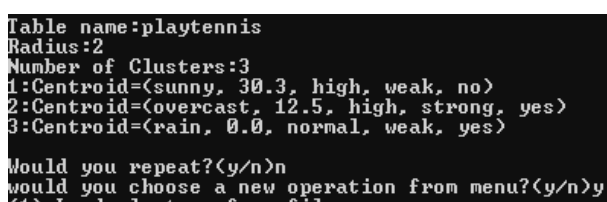
```
Client [Java Application] C:\Program Files\Java
addr = /192.168.1.242 port = 8080
connessione fallita
Vuoi riprovare?(y/n)
y
```

2) se la connessione fallisce, il client ci chiederà se vogliamo riprovare a connetterci al server.



```
addr = /192.168.1.242 port = 8080
Socket[addr=/192.168.1.242,port=8080,localport=545561]
(1) Load clusters from file
(2) Load data from db
(1/2):
```

3) se invece la connessione va a buon fine visualizzeremo una schermata come questa.



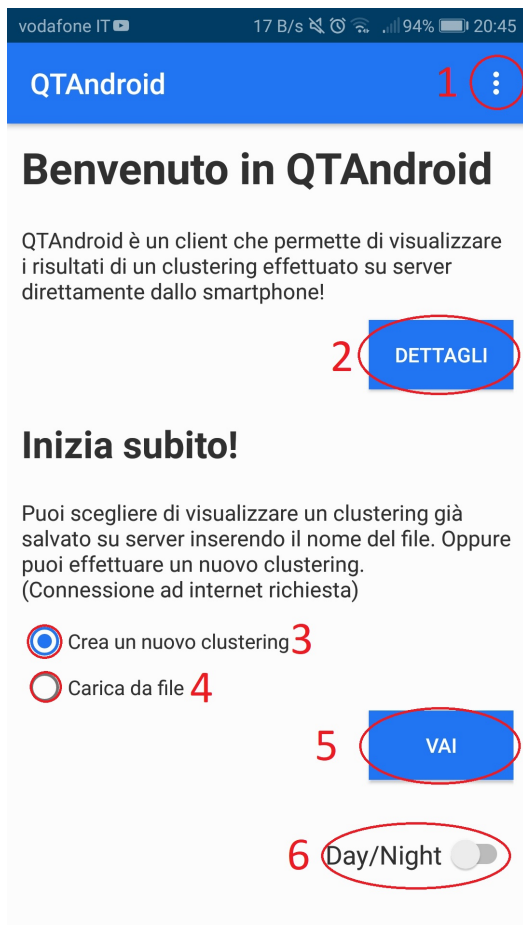
```
Table name:playtennis
Radius:2
Number of Clusters:3
1:Centroid=(sunny, 30.3, high, weak, no)
2:Centroid=(overcast, 12.5, high, strong, yes)
3:Centroid=(rain, 0.0, normal, weak, yes)
Would you repeat?(y/n)
would you choose a new operation from menu?(y/n)y
(1)
```

4) selezioniamo una operazione e inseriamo nome tabella e raggio. Otteniamo quindi un risultato come quello in figura se la tabella esiste e il raggio non è troppo grande. Ci verrà chiesto dunque se vogliamo ripetere un’operazione sulla stessa tabella. In caso negativo potremo decidere

se selezionare una nuova operazione dal menu.

Esempi di uso App android

Aprendo l'app ci si ritroverà davanti la seguente schermata.



1) Un menu a tendina dal quale si può accedere a:

- la sezione “Contatti”, in cui si possono trovare i profili dei realizzatori dell’app
- la sezione “Impostazioni”, che permette di settare indirizzo e porta del server.

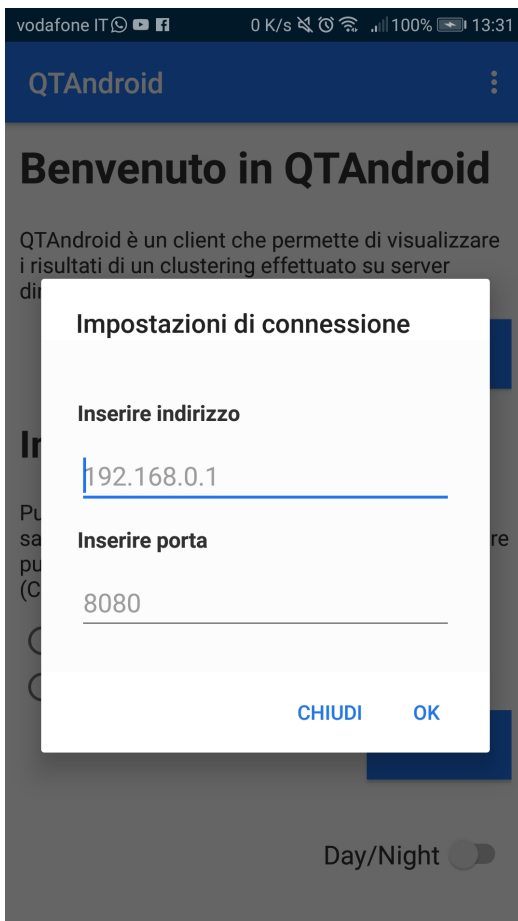
2) Toccando sul bottone dettagli si possono avere maggiori informazioni sul funzionamento del clustering su server.

3) Premendo su “Crea un nuovo clustering” si specifica che si vuole accedere ai dati del database per creare un nuovo clustering.

4) Toccando su “Carica da file” si specifica che si vuole visualizzare un clustering già effettuato in precedenza

5) Premendo “VAI” si conferma la scelta (è richiesta connessione ad internet per procedere)

6) Tramite questo switch si può passare alla DarkMode.

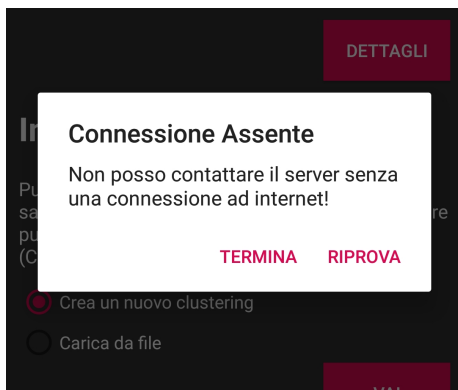


Toccando su “Impostazioni” si accede ad un dialog di questo tipo.

Se nessuno dei campi è riempito, o uno dei due campi è vuoto o non valido (nel caso della porta), prima di effettuare la prima connessione (che sarà stabilita alla pressione del pulsante “VAI”), sarà settato di default l’indirizzo e la porta del nostro server.

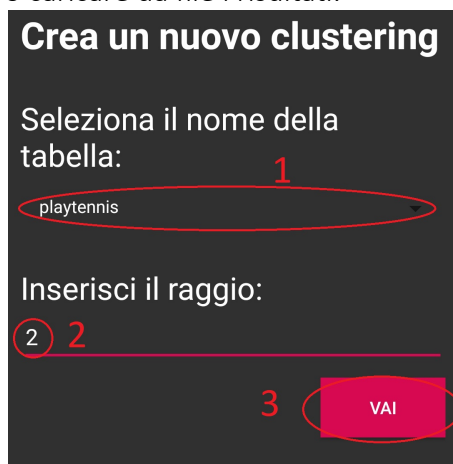
Sarà comunque possibile modificarlo in seguito.

Premendo su “VAI”, in assenza di connessione, si riceve il seguente messaggio:



Con “Termina” si chiude l’applicazione, con “Riprova” l’app effettua un nuovo check sulla connessione.

Toccando su “VAI”, si accede a questa schermata in cui si seleziona tabella e raggio per clusterizzare o caricare da file i risultati.



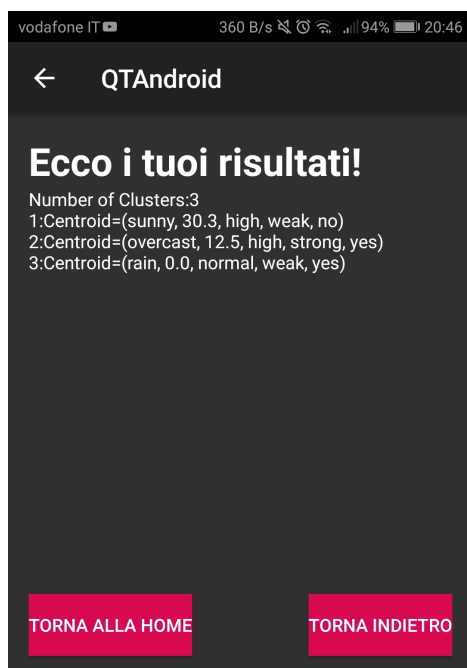
(non è possibile lasciare campi vuoti)

1) Spinner per la selezione della tabella da utilizzare

2) EditText per inserire il raggio. (N.B: non è consentito inserire 0 o numeri negativi)

3) Conferma la scelta.

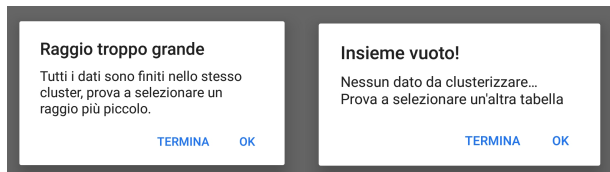
Una volta confermata la scelta si entra in una attività per mostrare i risultati:



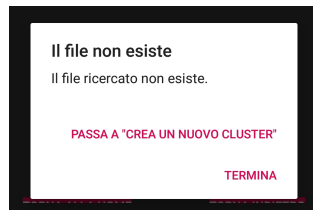
Nel caso migliore, visualizzeremo una schermata come questa con i cluster creati.

Ci sono inoltre due pulsanti:

Uno per tornare alla home (la connessione con il server verrà interrotta) e uno per tornare indietro e rieffettuare un clustering. (nel caso in cui riutilizzassimo la stessa tabella, quest’ultima non verrà ricaricata dal server).



Se invece la tabella selezionata è vuota, oppure tutte le tuple sono finite nello stesso cluster (raggio troppo grande), verranno visualizzati dei Dialog di errore.



Se avevamo selezionato un file inesistente, l'app ci suggerirà di passare alla sezione “crea un nuovo clustering”.

Esempi di test

Possibili casi di test:

Input:

table:playtennis

radius:0

Output:

L' applicazione chiederà un nuovo raggio mostrando il seguente messaggio:

"Insert a radius greater than 0"

Input:

table:tabellaInesistente

Output:

L' applicazione chiederà una nuova tabella mostrando il seguente messaggio:

"The selected table does not exists"

Input:

table:emptyTable

Output:

L' applicazione chiederà una nuova tabella mostrando il seguente messaggio:

"The selected table is empty"

Input:

table:playtennis

radius:2

Output:

L' applicazione mostrerà il seguente risultato:

Number of Clusters:3

1:Centroid=(sunny 30.3 high weak no)

Examples:

[sunny 30.3 high weak no] dist=0.0

[sunny 30.3 high strong no] dist=1.0

[sunny 13.0 high weak no] dist=0.5709571100859968

AvgDistance=0.5236523700286656

2:Centroid=(overcast 12.5 high strong yes)

Examples:

[overcast 30.0 high weak yes] dist=1.5775577703182049

[overcast 0.1 normal strong yes] dist=1.4092409343968995

[sunny 12.5 normal strong yes] dist=2.0

[overcast 12.5 high strong yes] dist=0.0

[rain 12.5 high strong no] dist=2.0

AvgDistance=1.397359740943021

3:Centroid=(rain 0.0 normal weak yes)

Examples:

[rain 13.0 high weak yes] dist=1.4290429150935235

[rain 0.0 normal weak yes] dist=0.0

[rain 0.0 normal strong no] dist=2.0

[sunny 0.1 normal weak yes] dist=1.003300330116104

[rain 12.0 normal weak yes] dist=0.3960396139324834

[overcast 29.21 normal weak yes] dist=1.9640264269139867

AvgDistance=1.132068214342683

Input:

table:playtennis

radius:6

Output:

L' applicazione mostrera il seguente risultato:

"The radius is too big"

per poi chiedere se si vuole inserire un altro raggio