



UNIVERSITÀ
DEGLI STUDI DI BARI
ALDO MORO

GUIDA UTENTE & CASI DI TEST



Quality threshold

Realizzazione: Lorusso Claudia , Dileo Angela

Caso di studio di Metodi Avanzati di Programmazione A.A. :

2018/2019

Prof.ssa Annalisa Appice

SOMMARIO

Introduzione	3
Obbiettivo	4
Architettura dell'applicazione.....	4
Descrizione dell'Algoritmo	5
Avvio del Software	6
1. Avvio del server:.....	6
2. Avvio del client:	7
Potenzialità del Programma	8
1. Generazione del CLuster Set (TAB From DB):	8
2. Copia e salvataggio del Cluster Set (TAB From DB):.....	10
3. Estrapolazione di informazioni da file (TAB From File):	11
Black Box Tests	12
4. Test – “login”	12
5. Test – “From DB”	16
6. Test – “From File”	21
GLOSSARIO	23
Riferimenti	24

INTRODUZIONE

Il **Quality Threshold** (QT) è un algoritmo di **clustering** che rientra nell'area di ricerca del **Data Mining**.

Lo scopo del *Data Mining* è l'estrazione automatica di conoscenza nascosta in voluminose basi di dati al fine di renderla disponibile e direttamente utilizzabile.

Le tecniche di *clustering* si basano sul raggruppamento di una popolazione di elementi, anche detti **tuple**, presenti in quello che nel corso della guida è definito come **dataset**, ovvero una molteplicità, un insieme di elementi presenti in tabelle prelevabili da un **database**.

Ogni tupla è, dunque, smistata all'interno di determinati insiemi, detti **cluster**.

L'*intersezione* tra i vari cluster è pari all'insieme vuoto: una tupla, infatti, può appartenere ad un uno ed un solo cluster.

L'insieme risultante dalla loro *unione*, invece, è definito con il termine **cluster set**, letteralmente 'insieme di cluster', e la sua cardinalità è pari alla totalità delle tuple contenute nel dataset di partenza.

Per determinare il cluster di appartenenza di ogni tupla ci si basa sulla 'somiglianza' delle varie tuple; quest'ultima è calcolata in base alla loro *distanza*.

Pertanto, si può dire che *l'appartenenza, o meno, di una tupla ad un cluster dipende da quanto la tupla presa in esame sia distante dal cluster preso in considerazione*.

Nello specifico, per determinare la distanza sopracitata si fa fede alla *tupla rappresentativa* del cluster considerato che è detta **centroide**.

Il requisito fondamentale affinché una tupla faccia parte di quel cluster è che la distanza tra il centroide e la tupla presa in esame sia minore o uguale ad un certo valore chiamato **radius** (raggio) il quale risulta essere un valore indispensabile ai fini della generazione del cluster set risultante.

Il radius è scelto dall'utente e, se selezionato opportunamente, è in grado di fornire informazioni di grande rilevanza sul dataset considerato.

Può però accadere che l'algoritmo non riesca a dare informazioni di alcuna importanza: ciò avviene quando l'utente sceglie un raggio talmente alto da far risultare tutte le tuple del dataset all'interno di un unico cluster - risultato che, per l'appunto, non risulta essere di alcuna utilità.

OBIETTIVO

Progettazione e realizzazione di un sistema **client - server** QT.

Il **server** include funzionalità di data mining per la scoperta di clusters di dati.

Il **client** è un **applet Java** che consente di usufruire del servizio di scoperta in remoto e visualizza i cluster identificati.

ARCHITETTURA DELL'APPLICAZIONE

Il **server** è di tipo *multiclient* il che vuol dire che è in grado di gestire le richieste da parte di più di un client alla volta.

Il suo compito è quello di connettersi al database ed eseguire l'algoritmo di clustering (QT) per la generazione del Cluster Set; si occupa inoltre di salvare e caricare i centroidi dei clusters scoperti e memorizzati all'interno di un file il cui nome e percorso sono forniti dall'utente.

Se questo non inserisce alcun percorso, il file viene salvato (o acquisito) automaticamente nella stessa cartella in cui sono contenuti i file .jar e .bat del client e del server.

Il **client** ha per lo più il compito di mostrare a video i risultati delle computazioni del server; ha un ruolo fondamentale per l'acquisizione di informazioni che solo l'utente può fornirgli.

Interagisce infatti in modo diretto con quest'ultimo acquisendo:

- il nome della tabella, da cui reperire le informazioni necessarie al server per le varie computazioni, ed il radius;
- il percorso (facoltativo) ed il nome del file su cui salvare il Cluster Set;
- il percorso (facoltativo) ed il nome del file da cui caricare il Cluster Set.

Inoltre permette all'utente di copiare, tramite l'apposito bottone, il contenuto della box in cui sono contenuti i risultati delle varie computazioni.

DESCRIZIONE DELL'ALGORITMO

Si considerino un *dataset* ed un *radius* che, come già anticipato, è la distanza massima che ogni *tupla* presa in esame deve avere dal *centroide* del *cluster* di cui si vuole verificare l'appartenenza.

L'algoritmo *QT* si prefigge di scoprire i cluster di appartenenza di ogni tupla appartenente al dataset e per farlo si compone dei seguenti passi:

1. Per ogni tupla del dataset, creare un cluster candidato includendo la tupla più vicina, la prossima più vicina e così via, fino a che non si raggiunge la soglia massima indicata dal radius; l'unione dei vari clusters darà origine ad un cluster set temporaneo.
2. Selezionare tra i vari cluster rilevati quello più popoloso ovvero con il maggior numero di tuple¹.
3. Ripetere la procedura escludendo le tuple già clusterizzate finché non rimangono più tuple da clusterizzare; quando tutte le tuple risulteranno clusterizzate, l'insieme dei cluster risultanti corrisponderà al cluster set desiderato.

¹ Si è fatto in modo che il cluster più popoloso venga posto in automatico in ultima posizione, all'interno del cluster set temporaneo, in modo da semplificare la sua ricerca.

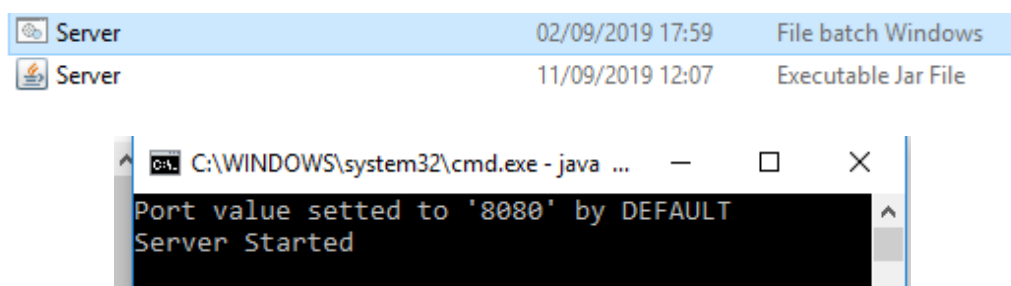
AVVIO DEL SOFTWARE

Per fare in modo che il programma si avvii, il server deve essere connesso sulla stessa porta su cui si vuole far connettere il client. Se ciò non avviene non sarà possibile utilizzare le varie funzionalità del programma.

Durante l'esecuzione del programma, il server deve restare sempre in ascolto: attenzione dunque a *non* chiudere la finestra del server altrimenti si giungerà ad un arrestamento forzato del client.

IMPORTANTE: Avviare prima il Server ed in seguito il Client per una corretta esecuzione del programma.

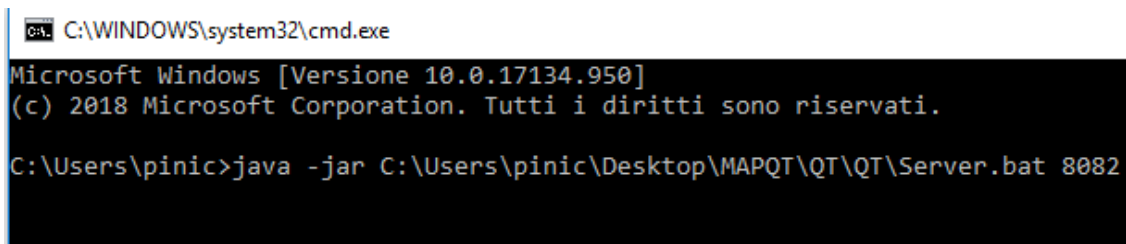
- 1. AVVIO DEL SERVER:** per poter avviare il server fare doppio click sul file batch denominato '*Server.bat*'.



La “schermata nera” della **command line** comunica all’utente che il valore della porta è stato settato automaticamente al valore di default ‘8080’.



Se invece si vuole scegliere un valore di porta differente aprire la command line, digitare `java -jar` seguito dal percorso del jar + `Server.jar` - o, più semplicemente, trascinare il file jar all’interno della finestra² ed, infine, specificare il numero di porta su cui ci si vuole connettere.

Di seguito un’immagine esplicativa per rendere più intuitive le azioni che permettono l’avviamento del server da linea di comando:

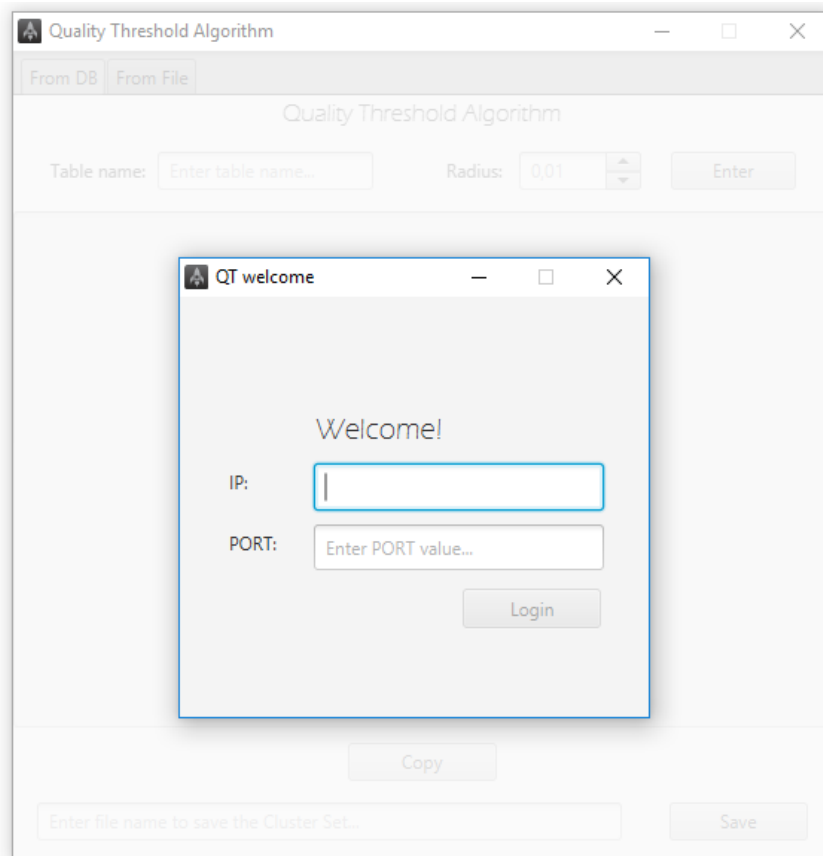


² L’azione permette il riempimento automatico della stringa contenente il percorso ed il nome del jar.

- 2. AVVIO DEL CLIENT:** per poter avviare il client fare doppio click sul file batch denominato 'AppClient.bat'.

	AppClient	11/09/2019 12:26	File batch Windows
	AppClient	11/09/2019 19:46	Executable Jar File

Appare una schermata di login che si sovrappone a quella principale del programma, inizialmente disabilitata:



Le due caselle di testo permettono la digitazione dell'ip e del numero di porta su cui il server è in esecuzione.

Nel caso in cui il server sia in esecuzione sulla stessa macchina su cui si ha intenzione di eseguire il client, è possibile inserire come indirizzo ip '127.0.0.1' o, in alternativa, 'localhost'.

Se i valori sono stati inseriti correttamente, rispettando la sintassi richiesta, il pulsante di login si abilita automaticamente permettendo di tentare la procedura di connessione con il server.

Se la connessione è andata a buon fine, la finestra di login scompare lasciando spazio alla finestra sottostante che nel frattempo si è abilitata.

POTENZIALITÀ DEL PROGRAMMA**1. GENERAZIONE DEL CLUSTER SET (TAB FROM DB):**

Stabilita una connessione tra client e server appare una schermata in cui è possibile inserire il nome di una tabella presente nel database preesistente³ nonché il radius che si preferisce.

Quality Threshold Algorithm

From DB From File

Quality Threshold Algorithm

Table name: Radius:

Type the name of the table you would like to get the data from
(no spaces nor any other symbol, different from [A-Za-z0-9] allowed)
& set your radius value in a range from 0.01 to 100000000.
WARNING: if your choice is equal to '0', the radius value will
automatically be set to 0.01;
press ENTER to validate your choice

³ Per la creazione del database si consulti la *guida all'installazione*.

Dopo aver inserito il nome della tabella ed il valore del radius, vengono stampate, nel box sottostante, le informazioni desiderate:

- Il nome della tabella.
- Il radius selezionato.
- Le tuple presenti nella tabella.
- Il numero di cluster generati.
- Il cluster set definitivo, i cui cluster sono rappresentati da:
 - il centroide, con la distanza media tra le varie tuple;
 - la lista delle tuple contenute in essi con la relativa distanza dal centroide.

Supponendo di inserire, ad esempio, i valori “example1”⁴ e “0.5”, rispettivamente per il nome della tabella e per il radius, dopo aver premuto sul bottone enter, si visualizza nel box sottostante la risposta del server con le informazioni ottenute in seguito alle varie computazioni (si noti che sono stati rilevati tre clusters):

The screenshot shows a window titled "Quality Threshold Algorithm". It has two tabs: "From DB" (selected) and "From File". Below the tabs, there are input fields for "Table name:" (containing "example1") and "Radius:" (containing "0,5"), followed by an "Enter" button. The main area of the window displays the results of the algorithm:

Number of Clusters: 3

0:Centroid=(14.0 12.5)

Examples:

- [11.0 12.0] dist=0.21875
- [11.0 13.0] dist=0.21875
- [13.0 13.0] dist=0.09375
- [12.0 8.5] dist=0.375
- [13.0 8.0] dist=0.34375
- [13.0 9.0] dist=0.28125
- [13.0 7.0] dist=0.40625
- [7.0 13.0] dist=0.46875
- [16.0 16.0] dist=0.34375
- [11.5 8.0] dist=0.4375
- [13.0 10.0] dist=0.21875
- [12.0 13.0] dist=0.15625
- [14.5 11.5] dist=0.09375
- [15.0 10.5] dist=0.1875
- [15.0 9.5] dist=0.25

At the bottom of the window, there is a "Copy" button and a text input field labeled "Enter file name to save the Cluster Set..." with a "Save" button next to it.

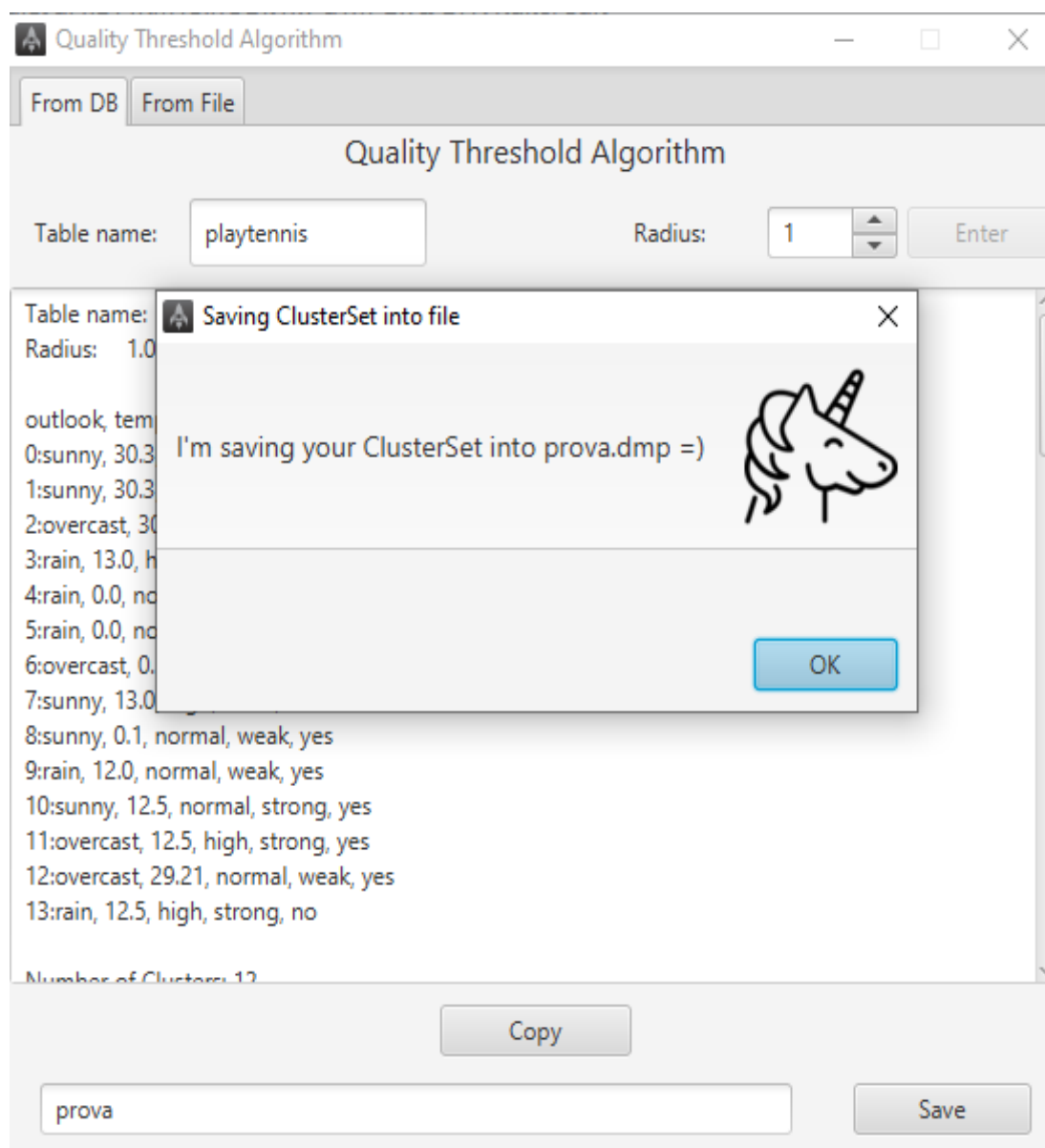
⁴ Per maggiori dettagli si rimanda alla sezione *Riferimenti*.

2. COPIA E SALVATAGGIO DEL CLUSTER SET (TAB FROM DB):

A seguito della generazione del Cluster Set è possibile copiare, per mezzo del tasto copy, o salvare le informazioni su di un file inserendo un percorso (esistente!) ed il nome del file di salvataggio.

Se non si specifica il percorso, il file viene salvato automaticamente nella stessa cartella in cui sono contenuti il file .bat ed il .jar del client e del server.

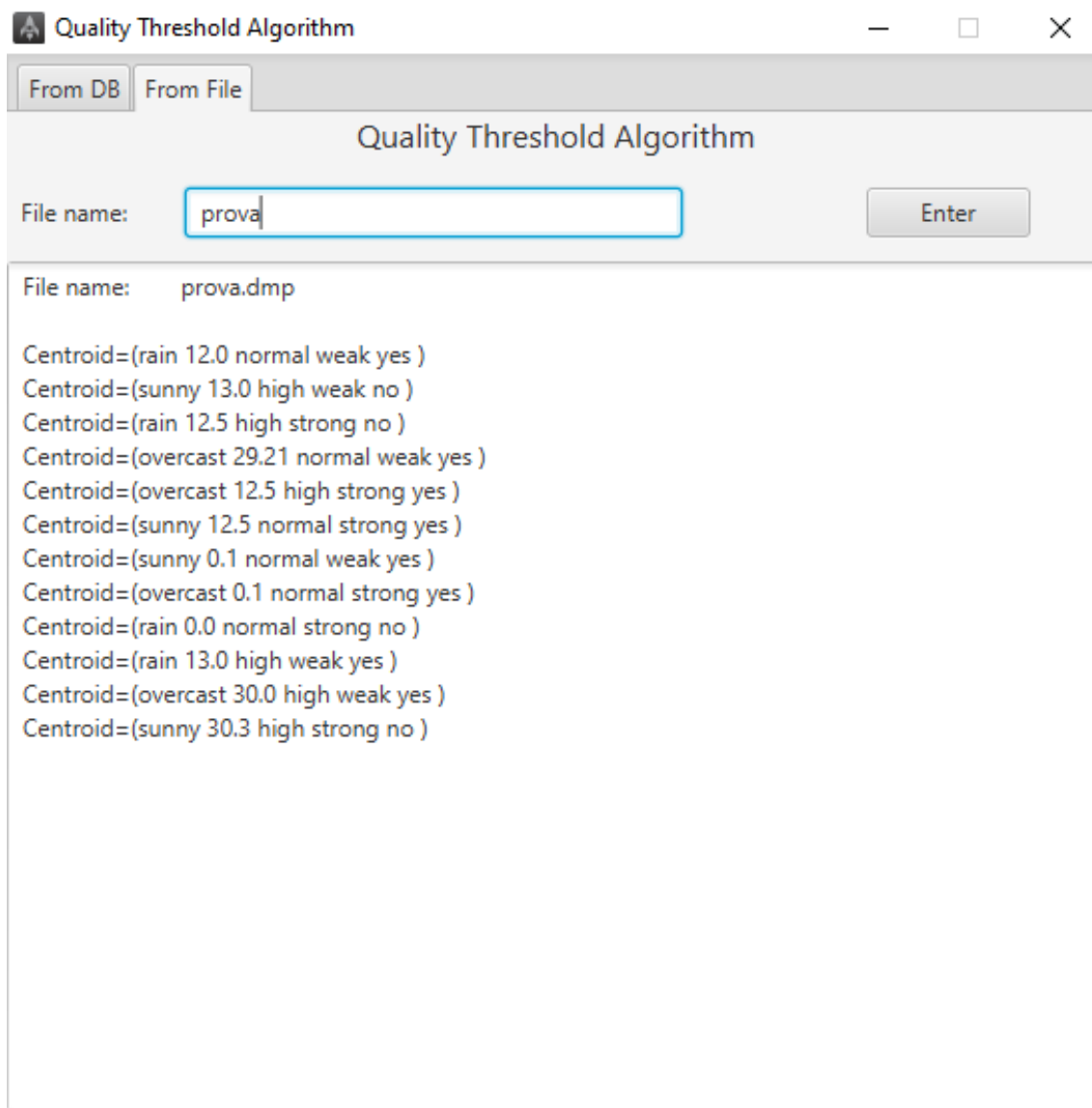
Premendo sul tasto Save una finestra informa l'utente della buona riuscita del salvataggio, come illustrato nella figura seguente.



3. ESTRAPOLAZIONE DI INFORMAZIONI DA FILE (TAB FROM FILE):

Per visualizzare il contenuto di un file precedentemente creato:

- Cliccare in alto a sinistra sulla dicitura *From File*.
- Digitare il percorso (facoltativo) ed il nome del file che si vuole caricare.
- Cliccare sul tasto *Enter* o in alternativa premere invio sulla tastiera.



Quality Threshold Algorithm

From DB From File

Quality Threshold Algorithm

File name: Enter

File name: prova.dmp

Centroid=(rain 12.0 normal weak yes)
Centroid=(sunny 13.0 high weak no)
Centroid=(rain 12.5 high strong no)
Centroid=(overcast 29.21 normal weak yes)
Centroid=(overcast 12.5 high strong yes)
Centroid=(sunny 12.5 normal strong yes)
Centroid=(sunny 0.1 normal weak yes)
Centroid=(overcast 0.1 normal strong yes)
Centroid=(rain 0.0 normal strong no)
Centroid=(rain 13.0 high weak yes)
Centroid=(overcast 30.0 high weak yes)
Centroid=(sunny 30.3 high strong no)

BLACK BOX TESTS

I test di tipo *Black Box* consistono nel dare in input al software determinati valori per verificare la correttezza del programma che si sta testando.

In questa sezione sono presenti i test più significativi al fine di verificare la validità del software.

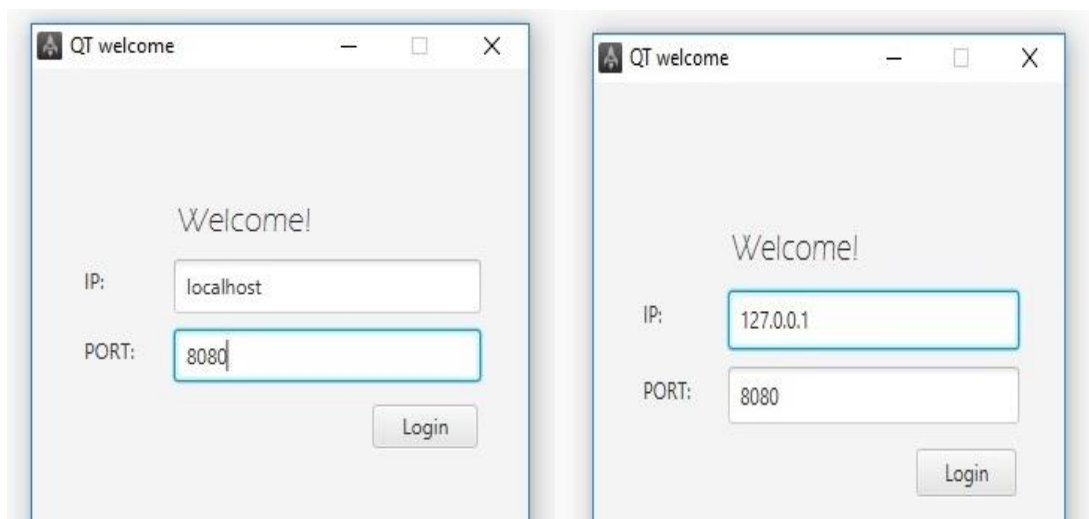
4. TEST – “LOGIN”

Di seguito sono riportati i test inerenti alla parte di login.

Il primo stage che si presenta all'utente è quello di login in cui è possibile inserire un indirizzo ip ed un numero di porta.

La prima finestra ne sovrasta un'altra che contiene il cuore del software ed in cui si effettuano le varie computazioni ed operazioni; questa è disabilitata e non è possibile accedervi finché non viene stabilita una connessione tra il client ed il server.

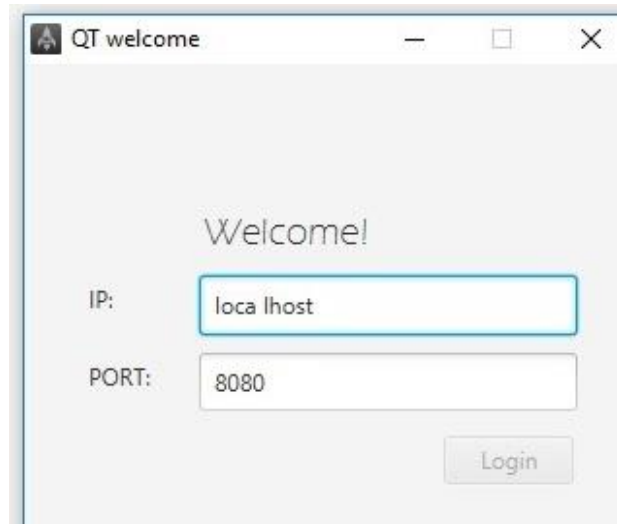
TEST 1-2:



Result = input accepted

Inserendo correttamente l'indirizzo Ip ed il numero di porta il tasto per il login si abilita automaticamente.

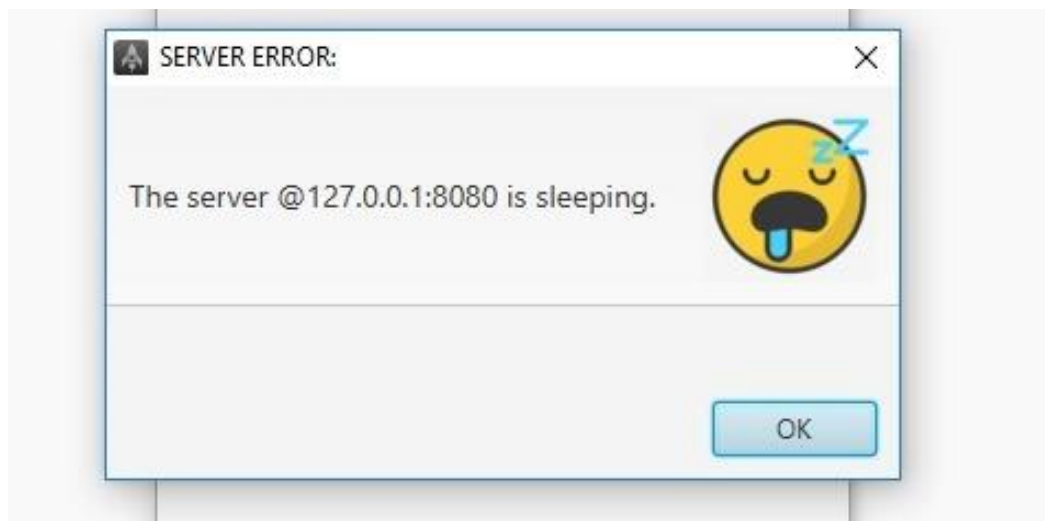
TEST 3:



Result = input rejected

Inserendo in modo scorretto l'indirizzo Ip o/e il numero di porta il tasto per il login resta disabilitato.

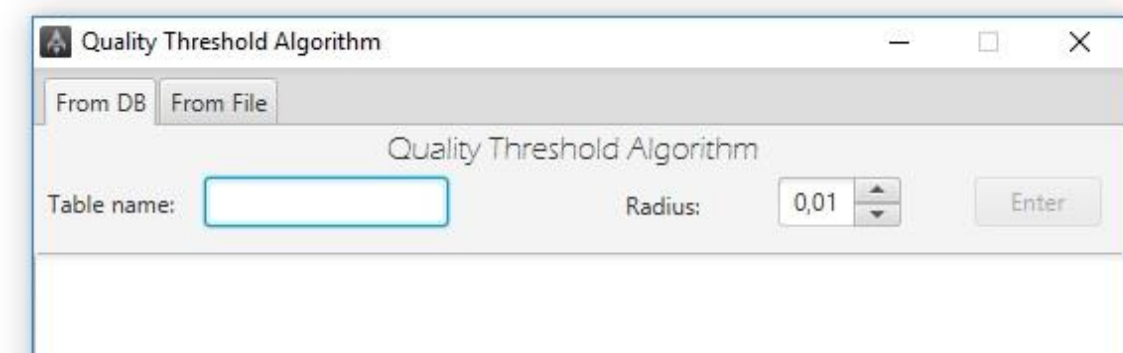
TEST 4:



Result = no connection available

Inserendo un indirizzo ip ed una porta su cui non è connesso alcun server il client non stabilisce alcuna connessione e mostra a video una finestra che avverte l'utente che il server non è connesso.

TEST 5:

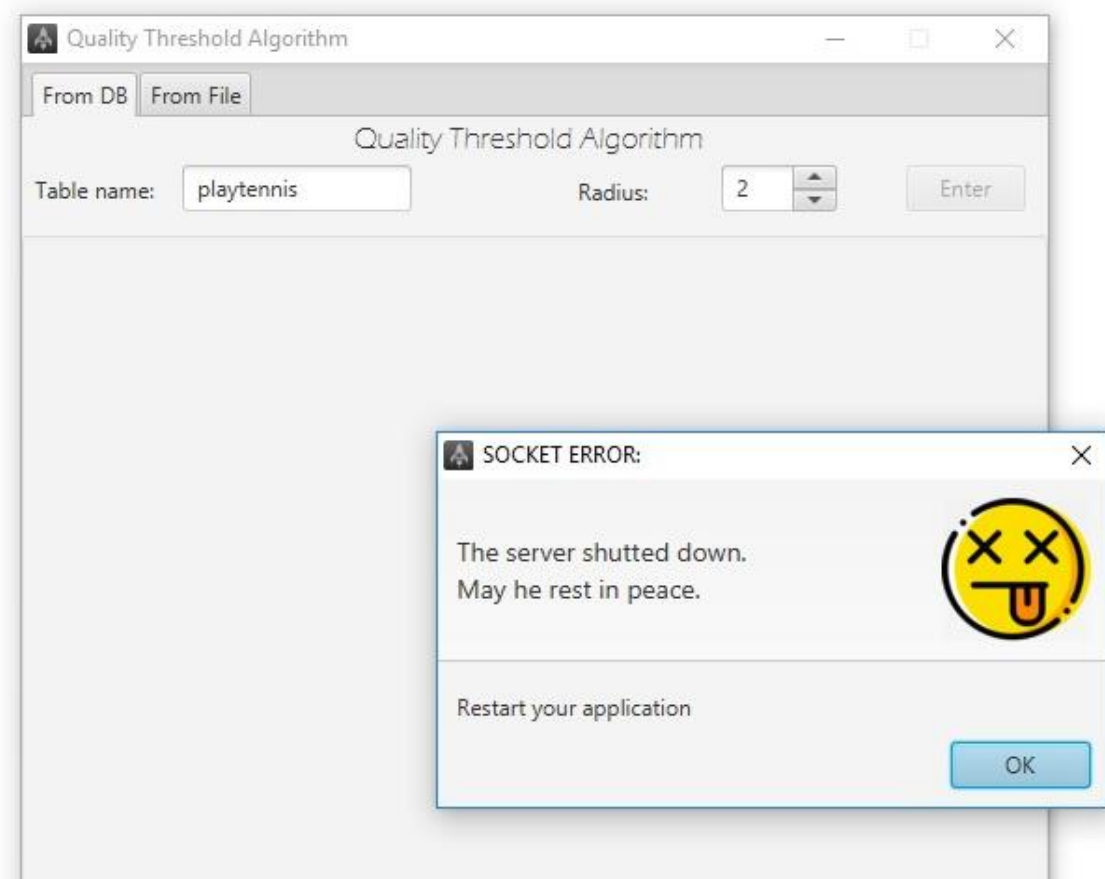


Result = input accepted

Inserendo in modo corretto un indirizzo ed una porta su cui è connesso il server viene stabilita la connessione tra quest'ultimo ed il client: la finestra di login scompare lasciando il posto a quella in cui si effettuano le varie computazioni.

Quest'ultima, infatti, si abilita rendendosi disponibile all'utilizzo dell'utente.

TEST 6:



Result = client shutdown

Effettuato il login, se dovessero sorgere problemi con il server, come ad esempio una sua chiusura anomala, non risulta possibile procedere con il normale utilizzo del software: compare, infatti, una finestra che avvisa l'utente che la connessione è stata interrotta e gli consiglia di provare a riavviare il programma.

Premendo su "ok" o sul tasto "X" il programma si chiude.

5. TEST – “FROM DB”

In questa sezione sono illustrati alcuni test relativi al prelievo ed all’uso delle informazioni presenti nel database inserendo il nome della tabella da cui prelevare i dati ed il raggio.

TEST 1:

Quality Threshold Algorithm

From DB From File

Quality Threshold Algorithm

Table name: playtennis Radius: 2 Enter

Radius: 2.0

outlook, temperature, umidity, wind, play

0:sunny, 30.3, high, weak, no

1:sunny, 30.3, high, strong, no

2:overcast, 30.0, high, weak, yes

3:rain, 13.0, high, weak, yes

4:rain, 0.0, normal, weak, yes

5:rain, 0.0, normal, strong, no

6:overcast, 0.1, normal, strong, yes

7:sunny, 13.0, high, weak, no

8:sunny, 0.1, normal, weak, yes

9:rain, 12.0, normal, weak, yes

10:sunny, 12.5, normal, strong, yes

11:overcast, 12.5, high, strong, yes

12:overcast, 29.21, normal, weak, yes

13:rain, 12.5, high, strong, no

Number of Clusters: 4

Copy

Enter file name to save the Cluster Set...

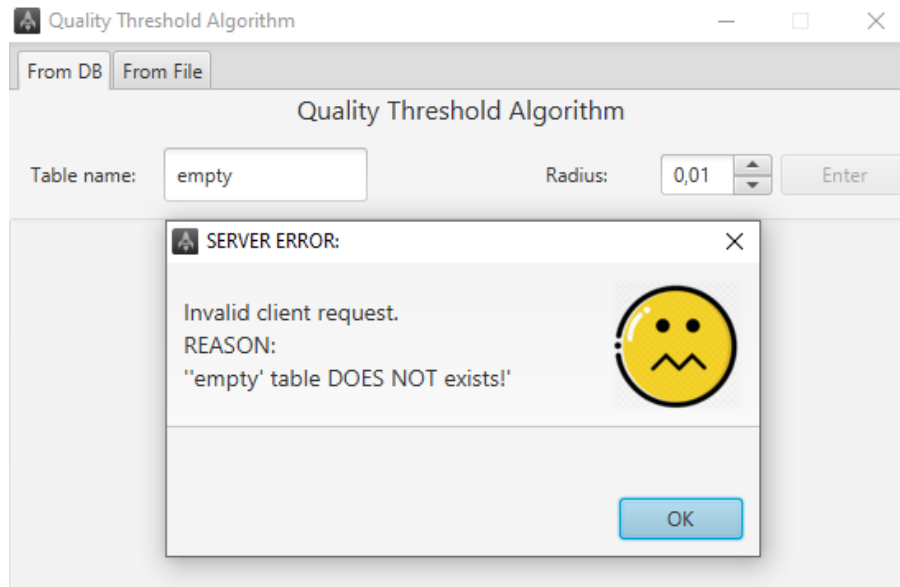
Save

Result = input accepted

Inserendo input corretti il tasto “Enter” si abilita automaticamente.

Una volta premuto questo pulsante o, in alternativa, premendo sul tasto enter/invio della tastiera, il software effettua le varie computazioni mostrando a video i dati presenti nella tabella ed i cluster generati tenendo conto del raggio specificato dall’utente.

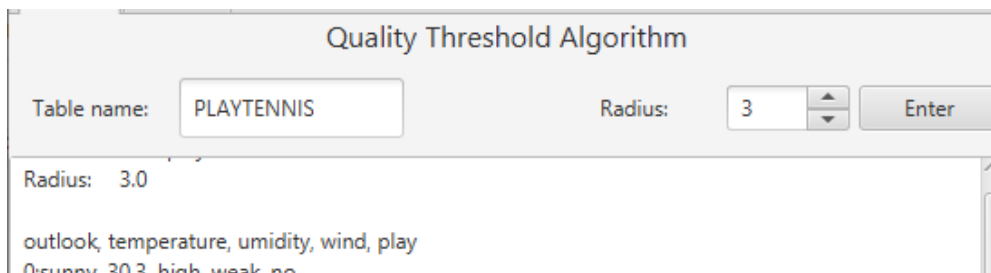
TEST 2:



Result = Input rejected

Inserendo il nome di una tabella non presente nel database viene mostrata a video una finestra che avverte l'utente della sua inesistenza.

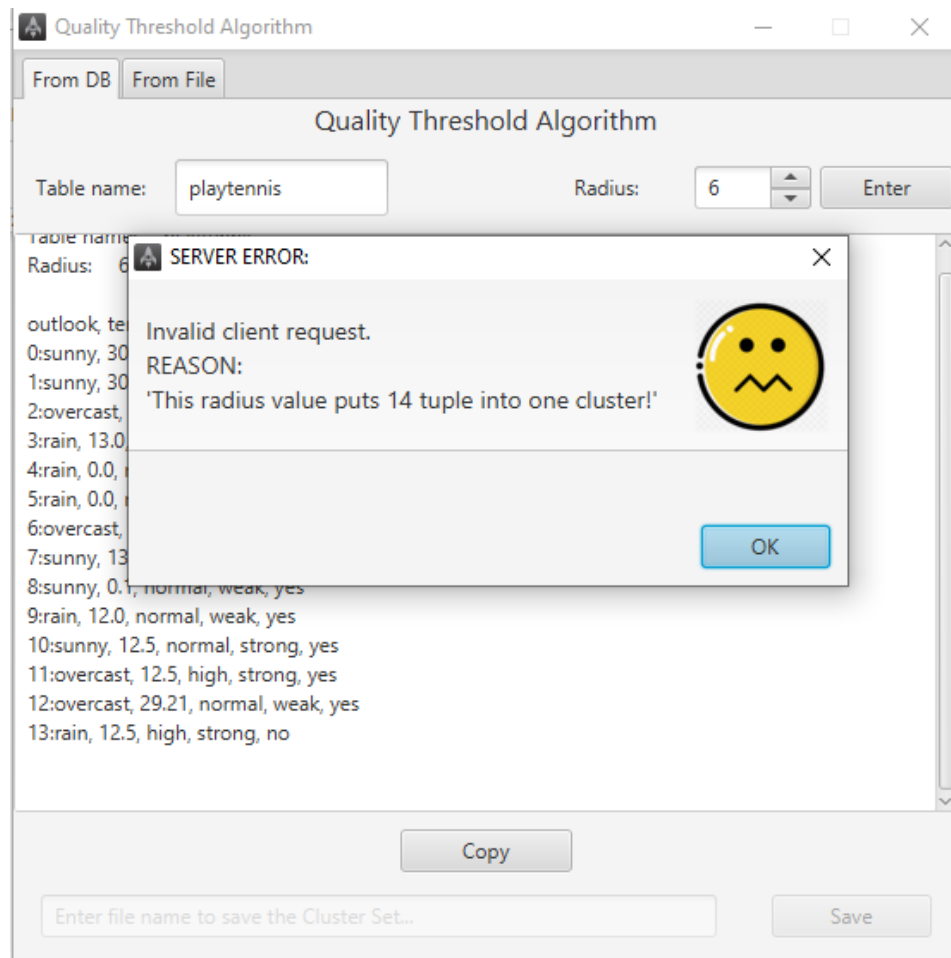
TEST 3:



Result = Input accepted

Non ha nessuna importanza se il nome della tabella sia scritto in maiuscolo o in minuscolo.

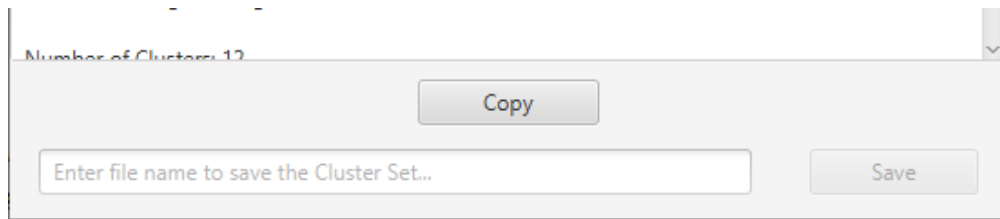
TEST 4:



Result = Input rejected

Se si immette un valore del raggio troppo grande, tutte le tuple presenti nella tabella vengono incluse all'interno di un unico cluster - superfluo specificare che non è possibile estrarre nulla di significativo da un risultato simile!

Compare dunque una finestra che comunica all'utente l'impossibilità di effettuare alcuna computazione.

TEST 5:

Premendo il pulsante “copy” viene attivata la funzione di copia verificabile in qualsiasi editor di testo, come mostrato nella schermata sottostante:

```

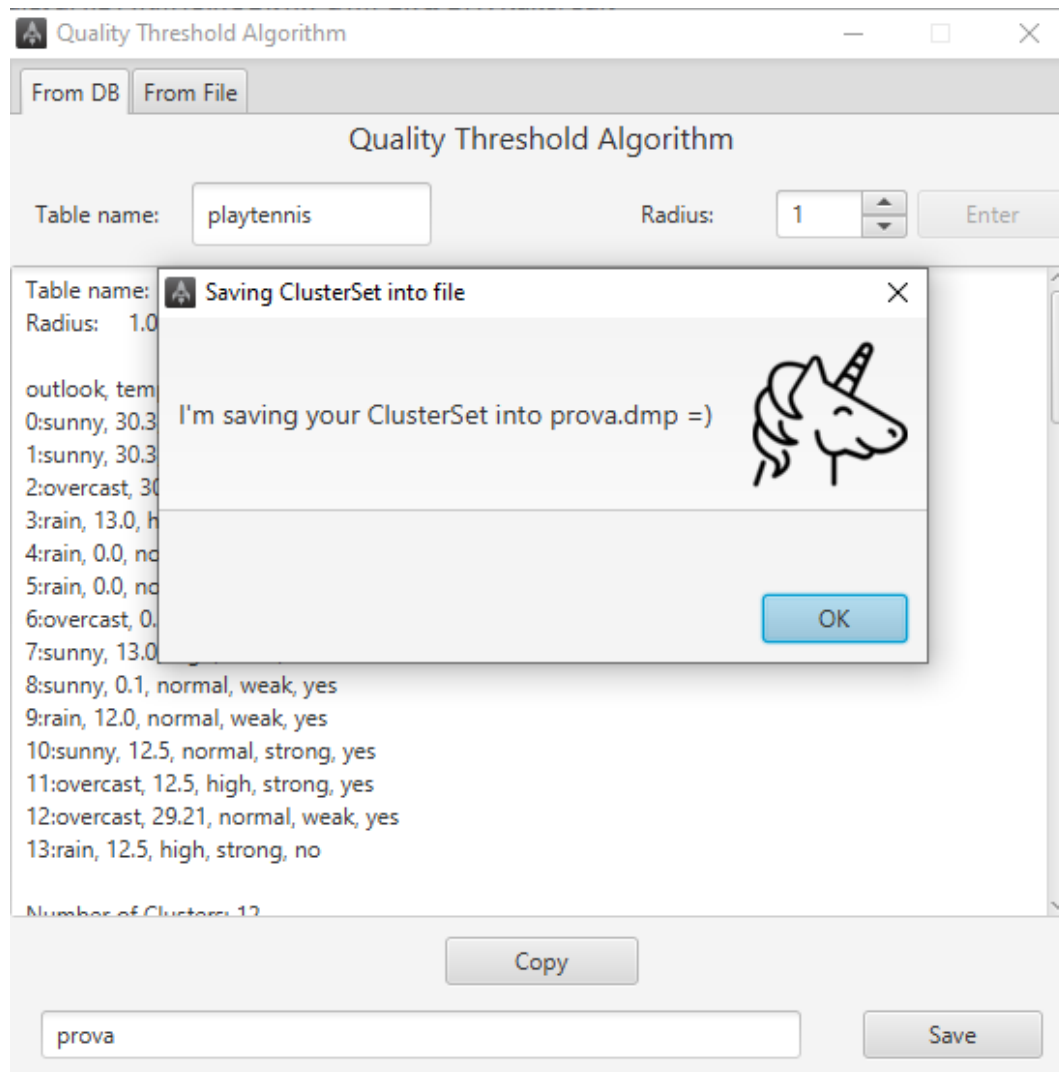
*Senza nome - Blocco note di Windows
File Modifica Formato Visualizza ?
Table name:    playtennis
Radius: 1.0

outlook, temperature, umidity, wind, play
0:sunny, 30.3, high, weak, no
1:sunny, 30.3, high, strong, no
2:overcast, 30.0, high, weak, yes
3:rain, 13.0, high, weak, yes
4:rain, 0.0, normal, weak, yes
5:rain, 0.0, normal, strong, no
6:overcast, 0.1, normal, strong, yes
7:sunny, 13.0, high, weak, no
8:sunny, 0.1, normal, weak, yes
9:rain, 12.0, normal, weak, yes
10:sunny, 12.5, normal, strong, yes
11:overcast, 12.5, high, strong, yes
12:overcast, 29.21, normal, weak, yes
13:rain, 12.5, high, strong, no

Number of Clusters: 12

0:Centroid=(rain 12.0 normal weak yes )
Examples:
[rain 0.0 normal weak yes ] dist=0.3960396139324834
[rain 12.0 normal weak yes ] dist=0.0
AvgDistance=0.1980198069662417
  
```

Result = success

TEST 6:**Result = success**

È possibile specificare il percorso ed il nome del file su cui si vogliono salvare i risultati della computazione.

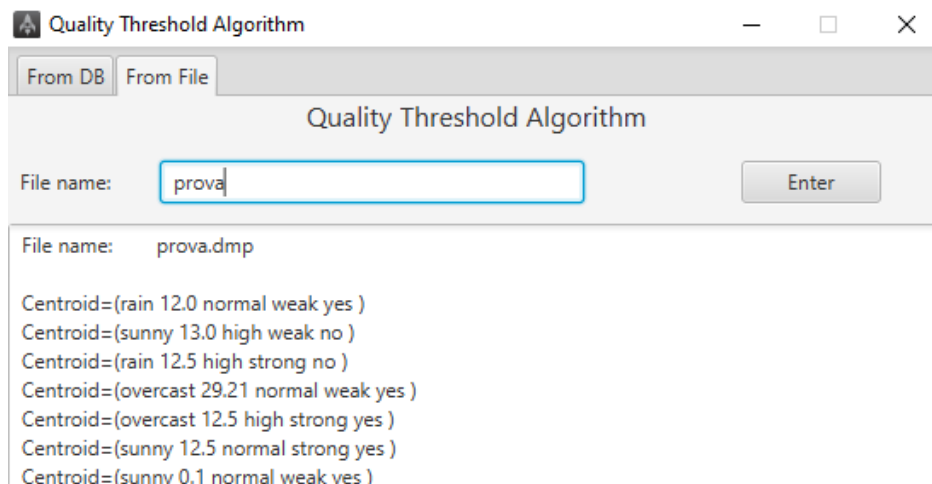
Nell'esempio si è deciso di denominare il file "prova" a cui il software attribuisce automaticamente l'estensione '.dmp' che non è modificabile.

Se non si specifica alcun percorso, il file viene inserito nella cartella in cui sono contenuti i file con estensione '.bat' e '.jar'.

6. TEST – “FROM FILE”

In questa sezione sono mostrati i test sulla parte inerente al prelievo delle informazioni da un file già esistente.

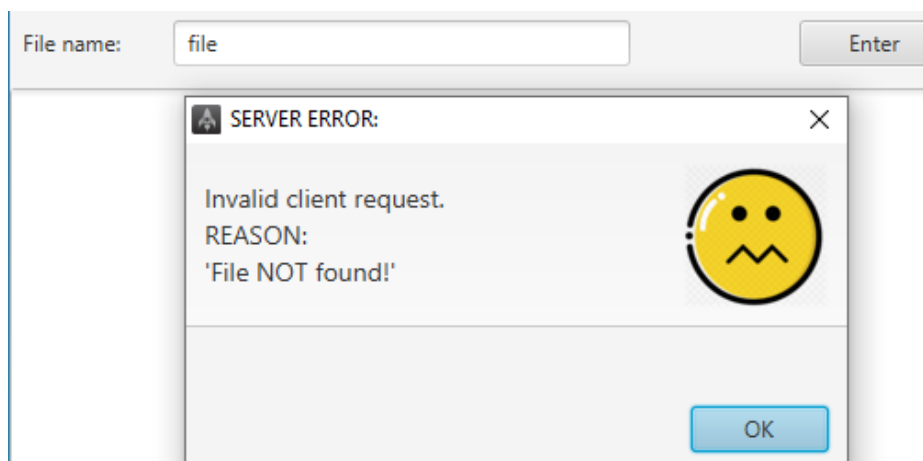
TEST 1:



Result = success

Inserendo il percorso (facoltativo) ed il nome di un file già esistente vengono mostrati a video i centroidi dei cluster rilevati e salvati nel file specificato.

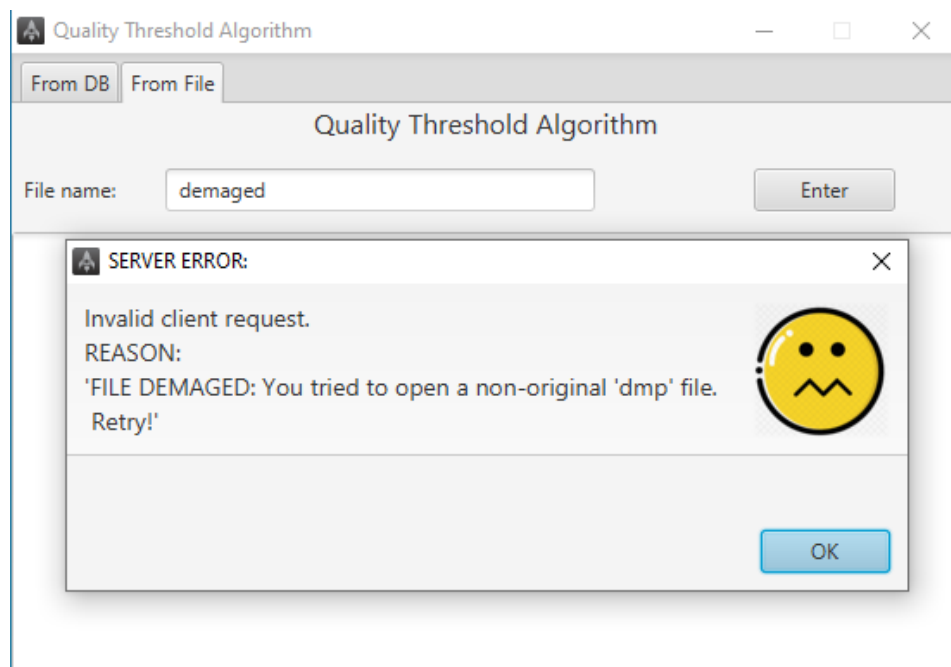
TEST 2:



Result = input rejected

Il test non è andato a buon fine perché il file da cui prelevare i dati non esiste.

TEST 3:



Result = input rejected

Il file "demaged.dmp" è stato creato con un semplice editor di testo e non per mezzo del software in analisi. Se l'utente tenta di aprire tale file, il software lo informa che ciò non è possibile.

Ciononostante il server continua a funzionare normalmente.

GLOSSARIO

1. **Algoritmo:** sequenza di istruzioni da eseguire per risolvere un problema o raggiungere un determinato obiettivo.
2. **Applet Java:** programma scritto in linguaggio Java che può essere eseguito da un web browser (elaborazione lato client).
3. **Centroide:** corrisponde alla tupla rappresentativa del cluster considerato.
4. **Client:** utente, computer o software che richiede l'esecuzione di un servizio.
5. **Cluster:** insieme di elementi.
6. **Cluster set:** insieme di cluster.
7. **Clustering:** insieme di tecniche volte alla selezione ed al raggruppamento di elementi omogenei in un insieme di dati.
8. **Command line:** interfaccia utente a riga di comando (o console) caratterizzata da un'interazione testuale tra utente ed elaboratore.
9. **Data mining:** insieme di tecniche e metodologie che estrapolano informazioni utili da grandi quantità di dati (es. database) attraverso metodi (semi) automatici.
10. **Database:** collezione di dati, tra loro correlati, utilizzati per rappresentare una porzione del mondo reale.
11. **Dataset:** insieme di dati strutturati in forma relazionale corrispondente al contenuto di una singola tabella di un database in cui ogni colonna rappresenta una particolare variabile ed ogni riga corrisponde ad un determinato membro del dataset in questione.
12. **File batch:** file di testo che contiene una sequenza di comandi per l'interprete di comandi del sistema.
13. **File jar:** archivio dati compresso (ZIP) usato per distribuire raccolte di classi Java. Tali file sono concettualmente e praticamente assimilabili a package e quindi talvolta associabili al concetto di libreria.
14. **Ip:** *Internet Protocol*, è un'etichetta numerica che identifica univocamente un dispositivo.
15. **Java:** In informatica la piattaforma Java è una piattaforma software, sviluppata su specifiche e implementazioni di Sun Microsystems, acquisita nel gennaio 2010 dalla Oracle Corporation, ovvero l'ambiente di esecuzione necessario per l'esecuzione di programmi scritti in linguaggio java.
16. **MySql:** è un *database management system* relazionale composto da un client a riga di comando ed un server.
17. **Porta:** punto fisico (hardware) o logico (software) sul quale instaurare delle connessioni, cioè il canale attraverso il quale i dati vengono trasferiti da un dispositivo all'altro.
18. **Quality Threshold:** algoritmo di clustering.
19. **Radius:** raggio, valore di riferimento indispensabile ai fini della generazione del cluster set.
20. **Server:** componente di elaborazione che a livello logico o fisico fornisce un qualunque tipo di servizio ad altre componenti (i clients) che ne fanno richiesta.
21. **Tupla:** record, elemento di un database relazionale caratterizzato da uno o più attributi.

RIFERIMENTI

- ¹ La tabella “example1”, contenente 30 tuple bidimensionali, è stata prelevata dal seguente blog di Data Mining: <http://data-mining.philippe-fournier-viger.com/introduction-clustering-k-means-java-code/>. In tale sito il professor *Philippe Fournier Viger* illustra l'applicazione di un algoritmo di clustering alternativo al Quality Threshold, il **K-Means**: in alternativa all'inserimento dell'ampiezza del radius, per poter clusterizzare le tuple presenti nel dataset, l'utente deve precisare un numero intero positivo corrispondente al numero di cluster di cui desidera essere composto il cluster set finale.

Si noti che assegnando il valore 0.5 al radius si ottengono gli stessi tre cluster che vengono generati per mezzo dell'algoritmo K-Means sopracitato; utilizzando un grafico a due dimensioni si ottiene la seguente rappresentazione:

