

# **REGRESSION TREE MINER**

## **ESTENSIONE**

**Caso di Studio del corso Metodi Avanzati di Programmazione (corso A)**

**Anno 2019-2020**

**DONATO LERARIO**

**Matricola: 683240**

### **Indice**

1. Introduzione
2. Installazione
3. Casi di test
4. Materiale

### **1. Introduzione**

Per estendere il progetto Base (presente nella cartella Regression Tree Miner) si è scelto di implementare un'interfaccia grafica grazie a JavaFX.

L'interfaccia è stata, prima di tutto, costruita utilizzando JavaFX Scene Builder 8.5.0.

Quest'estensione, lato Client, è stata sviluppata per permettere una fruizione più intuitiva e piacevole dell'applicazione all'utente. Le funzionalità del programma sono rimaste le stesse del progetto base, ossia lato Server abbiamo un Server che assegna una ServerSocket per ogni Client che si collega e viene gestito in un apposito thread.

JavaFX è una tecnologia software, utilizzabile combinata a Java, che permette lo sviluppo e la distribuzione di applicazioni all'avanguardia.

Inoltre, oltre ad avere una crescente libreria di funzionalità grafiche, include anche un linguaggio di scripting indipendente da Java, ossia JavaFX script.

Questo linguaggio ha la caratteristica di essere fortemente orientato alla programmazione grafica.

### **2. Installazione**

Come per il progetto base, per poter avviare il programma è richiesta l'installazione sulla macchina di MySQL Server e almeno Java SE Development Kit 14, poiché gli applicativi sono stati compilati con questa specifica JDK.

Assicurarsi di avere, oltre all'installazione di una versione recente di MySQL Server, anche il servizio in esecuzione (tramite Gestione delle Attività -> Servizi). Inoltre, in questa estensione è richiesta l'installazione di una versione recente di JavaFX e la sua cartella deve trovarsi all'interno del percorso "C:\Program Files\Java". In caso di dubbi, questa estensione è stata sviluppata utilizzando la versione 11.0.2.

A questo punto, il procedimento è molto simile al progetto base: nella cartella Script, troviamo l'SQLScript DropAndCreateDatabase.sql che permette di eliminare versioni precedenti sulla macchina del database MapDB e lo ricrea creando anche un utente MapUser con password "map" che sarà utilizzato per accedere al database. Una volta eseguito questo script SQL, il database viene popolato con le tabelle servo, provaC e prova per poter effettuare dei test con queste tabelle (già piene) nel programma. All'interno della cartella otherSQLscript sono presenti degli script SQL con il nome delle tre tabelle in caso si desideri creare le singole tabelle all'interno del database MapDB senza doverlo eliminare e ricreare.

Una volta fatto questo, è possibile cliccare sul file mapServerRun.bat che avvia l'applicativo Server del programma presente nel file jar mapServer. Nel caso in cui la porta 8080 è per qualche ragione già occupata, è possibile avviare il file FreePort8080.bat che libera questa porta e l'applicativo Server non avrà problemi ad avviarsi.

A questo punto è possibile avviare il file mapClientJFXRun.bat che avvia l'interfaccia grafica dell'applicativo Client del programma presente nel file jar mapClientJFX.

In caso di problemi con l'avvio dell'applicativo Client, fare tasto destro sul file bat mapClientJFXRun e cliccare su "Modifica". Ritroveremo scritto: --module-path "C:\Program Files\Java\javafx-sdk-11.0.2\lib". Verificare il percorso in cui si trovano le librerie JavaFX sul vostro pc, copiare il percorso e incollarlo tra le virgolette. Facendo così l'applicativo Client non avrà più problemi all'avviarsi.

### **3. Casi di test**

Nel caso in cui la porta 8080 è già utilizzata da qualche processo, viene visualizzato il seguente messaggio di errore nell'applicativo Server:

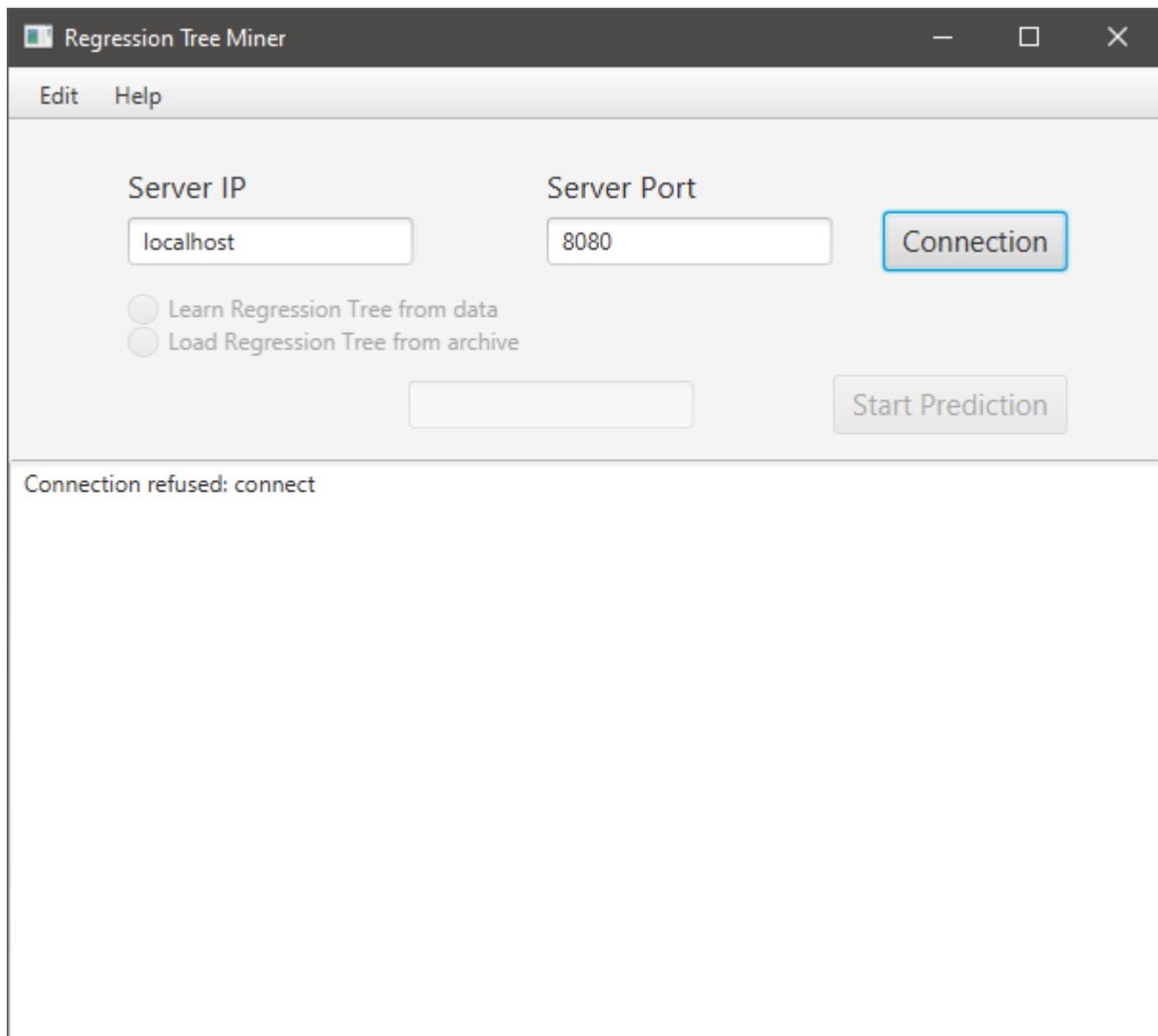
```
C:\Windows\system32\cmd.exe
Exception in thread "main" java.net.BindException: Address already in use: bind
    at java.base/sun.nio.ch.Net.bind0(Native Method)
    at java.base/sun.nio.ch.Net.bind(Net.java:550)
    at java.base/sun.nio.ch.Net.bind(Net.java:539)
    at java.base/sun.nio.ch.NioSocketImpl.bind(NioSocketImpl.java:643)
    at java.base/java.net.ServerSocket.bind(ServerSocket.java:396)
    at java.base/java.net.ServerSocket.bind(ServerSocket.java:350)
    at server.MultiServer.run(MultiServer.java:35)
    at server.MultiServer.<init>(MultiServer.java:24)
    at MainTest.main(MainTest.java:14)
```

In questo caso avviare il file Free8080.bat nella cartella Script per liberare la porta e riavviare l'applicativo Server.

A questo punto avviare l'applicativo Client e ci troveremo di fronte alla seguente schermata:

The screenshot shows the 'Regression Tree Miner' application window. It has a title bar with standard Windows controls (minimize, maximize, close). Below the title bar is a menu bar with 'Edit' and 'Help' options. The main area contains two input fields: 'Server IP' with the value 'localhost' and 'Server Port' with the value '8080'. To the right of these fields is a 'Connection' button. Below the input fields are two radio buttons: 'Learn Regression Tree from data' (selected) and 'Load Regression Tree from archive'. To the right of the radio buttons is a 'Start Prediction' button. The bottom half of the window is a large empty rectangular area, likely for displaying results or logs.

Se l'applicativo Server non venisse avviato e si cliccasse su Connection, avremmo il seguente messaggio:



Mentre, avendo l'applicativo Server regolarmente avviato e cliccando su Connection, ci conatteremo con il Server presente nel localhost. A questo punto visualizzeremo la seguente schermata e avremo ulteriori opzioni disponibili:

Regression Tree Miner

Edit Help

Server IP: localhost

Server Port: 8080

Connection

☒ Learn Regression Tree from data

☐ Load Regression Tree from archive

Choose the Table:

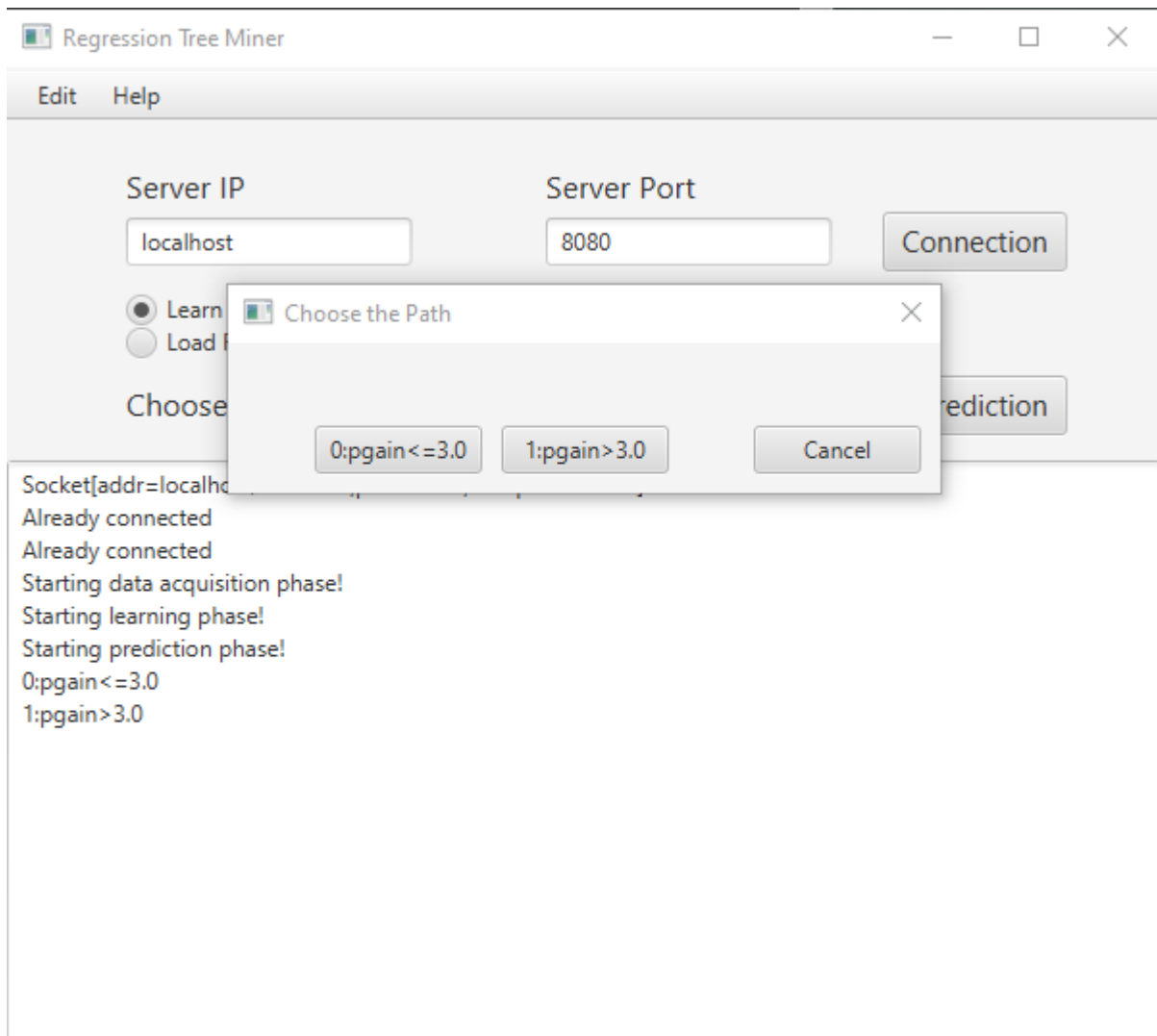
Start Prediction

Socket[addr=localhost/127.0.0.1,port=8080,localport=58544]

Nel caso in cui ricliccassimo su Connection, nell'area di testo verremo avvisati che siamo già connessi ad un Server:

The screenshot shows a window titled "Regression Tree Miner" with a menu bar containing "Edit" and "Help". The main interface has two input fields: "Server IP" with the value "localhost" and "Server Port" with the value "8080". To the right of these fields is a blue button labeled "Connection". Below the input fields are two radio buttons: "Learn Regression Tree from data" (which is selected) and "Load Regression Tree from archive". Below the radio buttons is a label "Choose the Table" followed by an empty text input field. To the right of this field is a button labeled "Start Prediction". At the bottom of the window, there is a text area containing the message: "Socket[addr=localhost/127.0.0.1,port=8080,localport=58544]" followed by "Already connected".

A questo punto, lasciando spuntata la RadioBox dove vi è scritto “Learn Regression Tree from data” (già spuntata di default), indicheremo di apprendere un albero di regressione dal trainingSet presente nella tabella che andremo ad indicare. Quindi scrivendo, ad esempio, “servo” nel campo a destra della scritta “Choose the Table” e cliccando poi il pulsante Start Prediction, viene visualizzata la seguente schermata che ci porterà subito alla scelta dei nodi per definire il cammino nell’albero di predizione:



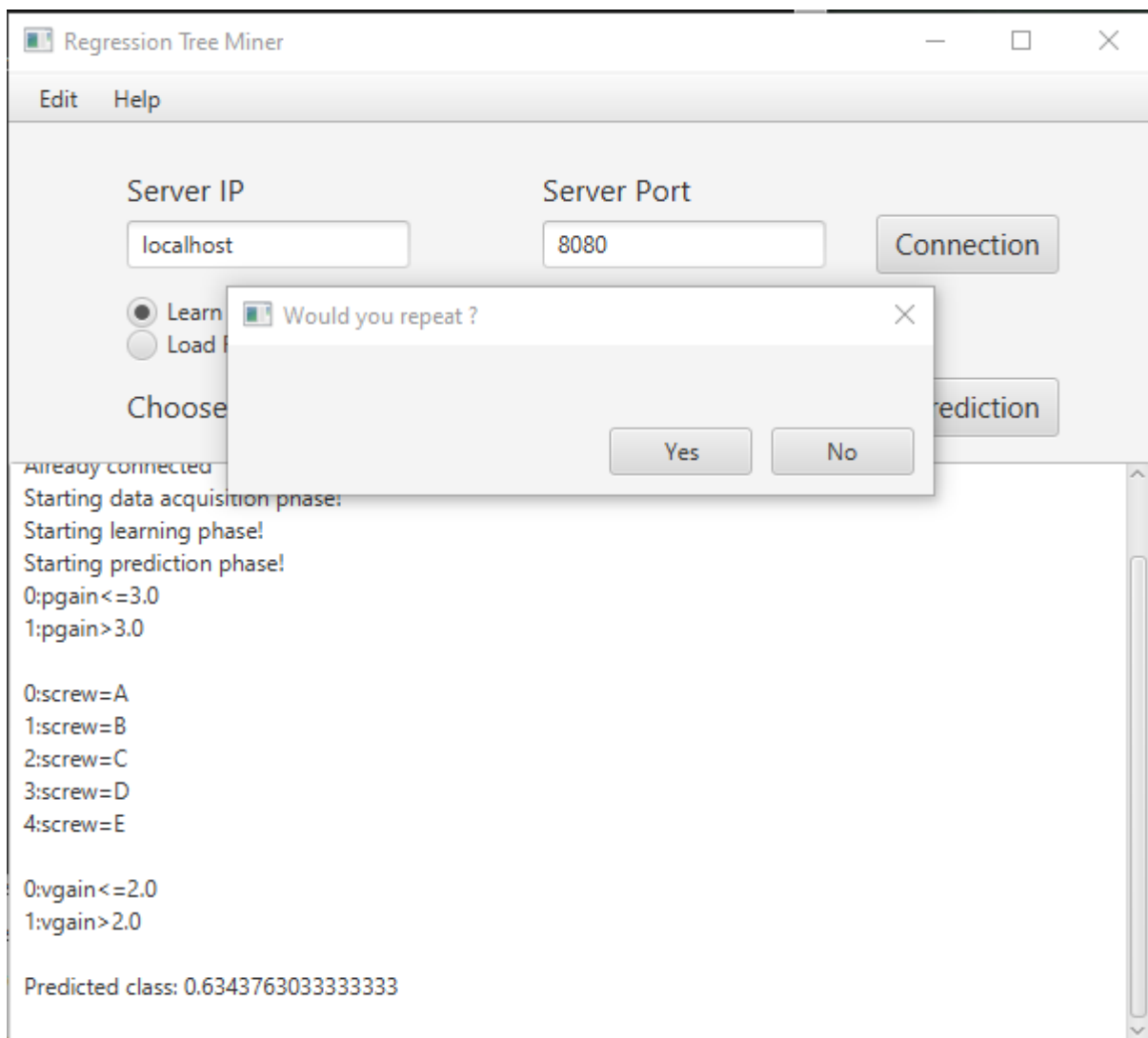
A questo punto verrà creato, nella cartella Script, il file servo.dat contenente l'albero di regressione.

Se, durante la scelta del cammino, l'applicativo Server dovesse chiudersi, visualizzeremmo la seguente schermata:

The screenshot shows a window titled "Regression Tree Miner" with a menu bar containing "Edit" and "Help". The main interface has two input fields: "Server IP" with the value "localhost" and "Server Port" with the value "8080". To the right of these fields is a "Connection" button. Below the "Server IP" field are two radio buttons: "Learn Regression Tree from data" (which is selected) and "Load Regression Tree from archive". Below the "Server Port" field is a text input field containing the word "servo". To the right of this field is a "Start Prediction" button. At the bottom of the window is a large text area containing the following text: "Socket[addr=localhost/127.0.0.1,port=8080,localport=50796]", "Starting data acquisition phase!", "Starting learning phase!", "Starting prediction phase!", "0:pgain<=3.0", "1:pgain>3.0", and "Connection reset by peer".

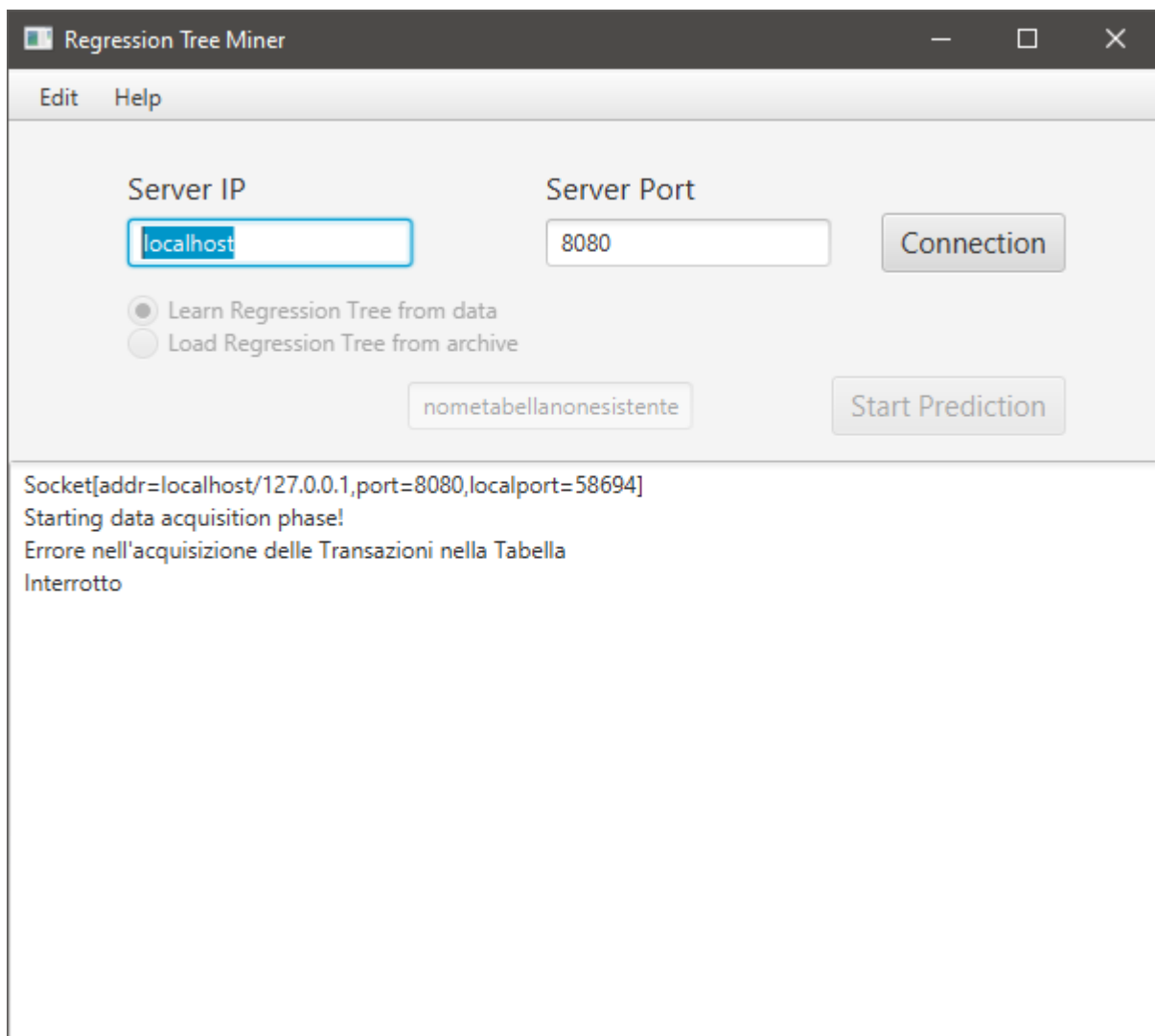
Continuando con la scelta del cammino, porteremo a termine il programma e visualizzeremo la seguente schermata:



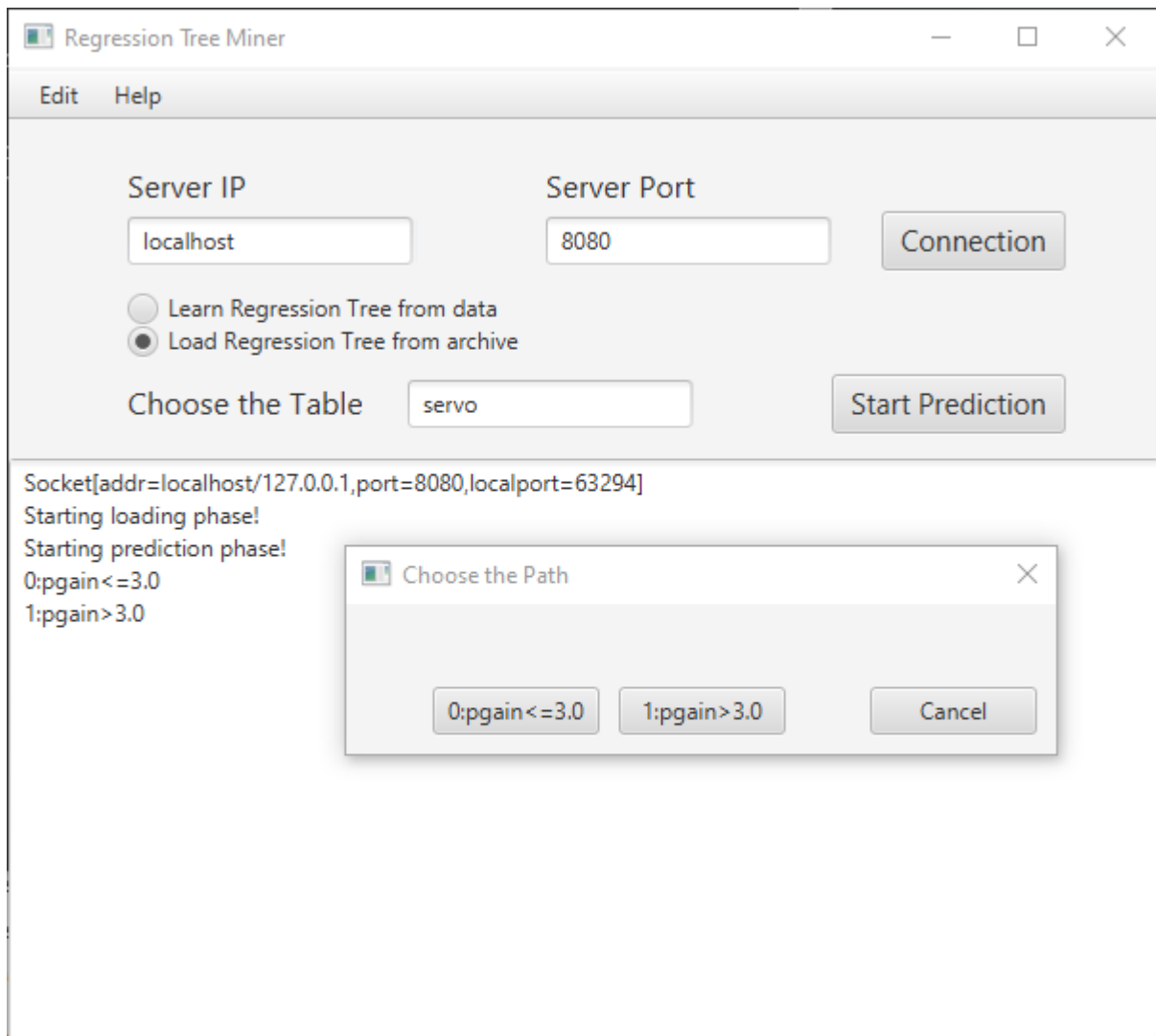


Quindi avremo il risultato della predizione (in basso a sinistra) e ci verrà chiesto se continuare oppure no. Premendo Yes ripartirà il programma sullo stesso trainingSet, mentre premendo No il programma si fermerà, la Socket Client verrà chiusa e sarà possibile ricollegarsi al Server oppure collegarsi ad un altro.

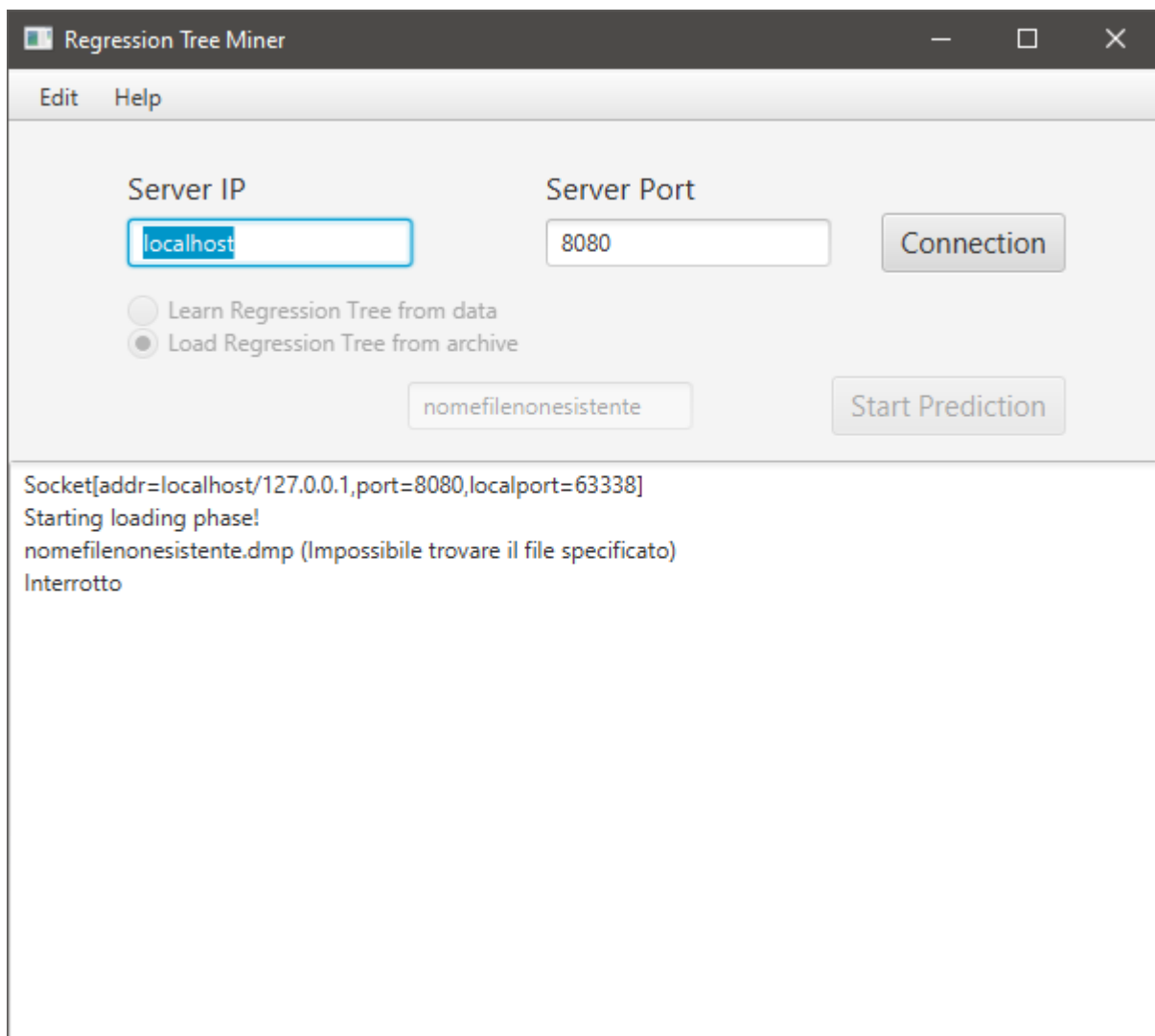
Nel caso in cui scriviamo il nome di una tabella non esistente nel database MapDB, il processo client termina e viene visualizzata la seguente schermata:



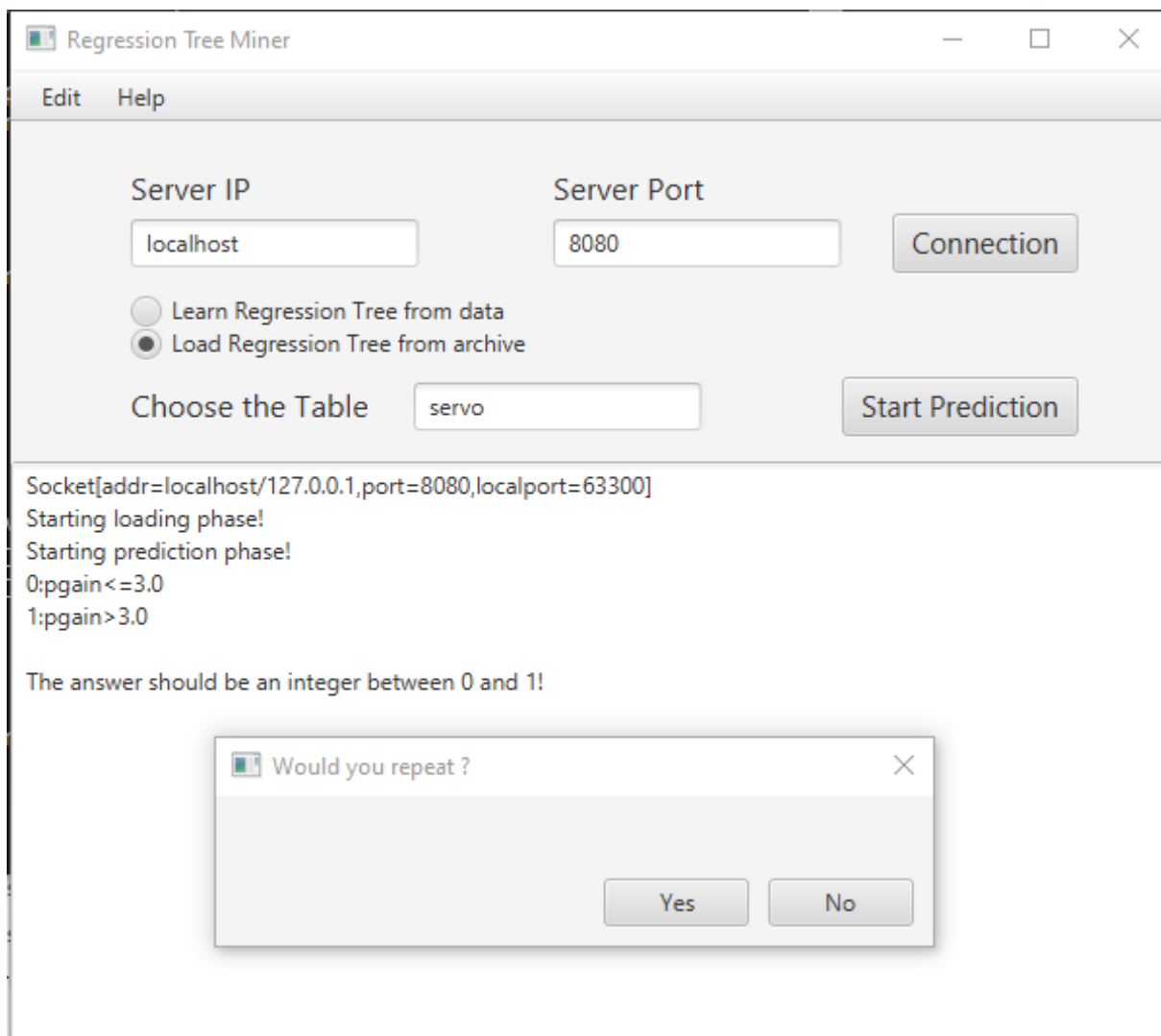
Eseguito almeno una volta con successo il programma e avendo creato il file .dmp nella cartella Script, è possibile provare il programma caricando un albero di regressione precedentemente salvato su file. In questo caso, dopo il collegamento tra applicativo Client e applicativo Server, arriviamo alla schermata principale con le varie opzioni sbloccate, ma a questo punto spuntiamo sul Radio Box dove vi è scritto "Load Regression Tree from archive" e inseriamo il nome del file presente nella cartella, in questo caso "servo" e clicchiamo su Start Prediction. Ci troveremo di fronte a questa schermata:



Scrivendo il nome di un file non presente nella cartella Script, verrà visualizzato il seguente messaggio:

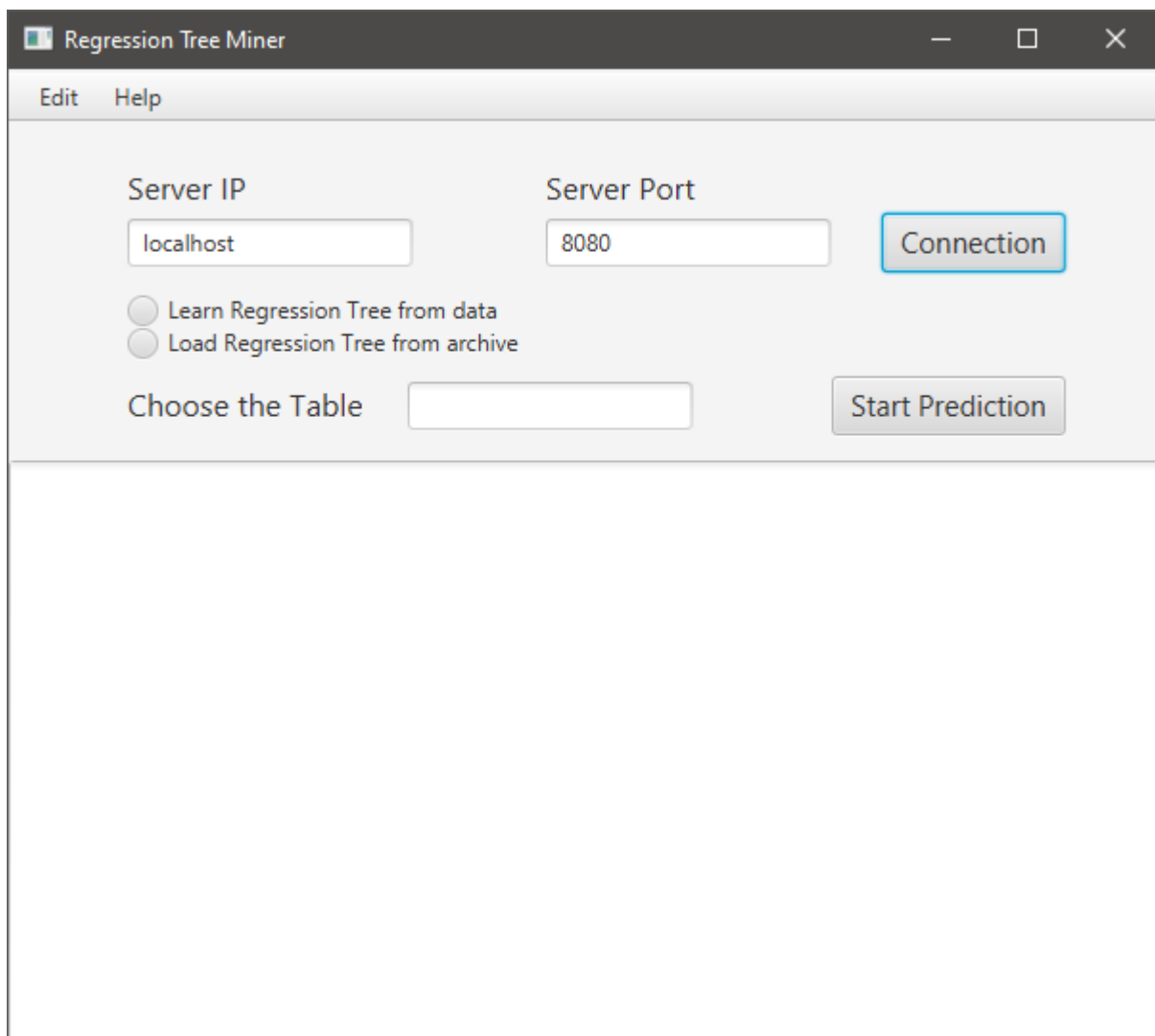


Se, durante la scelta del cammino, clicchiamo su Cancel, verrà visualizzato il seguente messaggio nell'area di testo e ci verrà chiesto se ripetere la fase di predizione:



Completando con successo la computazione, otterremo la Predicted class e lo scenario sarà tale a quale a quello visualizzato precedentemente al completamento della computazione nel caso di apprendimento di albero di regressione da tabella contenente trainingSet.

In alto a sinistra, cliccando sul Pulsante "Edit" e poi "Delete" è possibile svuotare tutti i campi ad eccezione del campo dove inserire il Server IP e il campo dove inserire il Server Port, che continueranno ad avere di default "localhost" e "8080" per velocizzare i casi di test:



The image shows a software window titled "Regression Tree Miner". It has a menu bar with "Edit" and "Help". The main area contains several controls: "Server IP" with a text box containing "localhost", "Server Port" with a text box containing "8080", and a "Connection" button. Below these are two radio buttons: "Learn Regression Tree from data" (selected) and "Load Regression Tree from archive". At the bottom, there is a "Choose the Table" label, an empty text box, and a "Start Prediction" button. The window has standard OS window controls (minimize, maximize, close) in the top right corner.

Regression Tree Miner

Edit Help

Server IP localhost Server Port 8080 Connection

☒ Learn Regression Tree from data  
☐ Load Regression Tree from archive

Choose the Table Start Prediction

Cliccando invece sul pulsante "Help" e poi "Author" verrà visualizzato il nome dell'Autore e alcune informazioni.

#### **4. Materiale**

- Codice Sorgente (cartelle MapClientGui e MapServer)
- Javadoc
- Script SQL per la creazione del database MapDB, delle tabelle e per il loro riempimento (DropAndCreateDatabase nella cartella Script)
- Singoli script SQL solo per la creazione delle tabelle all'interno di MapDB (cartella Script -> cartella otherSQLscript)
- File Jar contenente gli applicativi Client e Server (mapClientJFX.jar e mapServer.jar, nella cartella Script)
- File bat per avviare gli applicativi Client e Server (mapClientJFXRun.bat e mapServerRun.bat, nella cartella Script)
- Diagramma delle classi UML (cartella UML)