

# Documentazione

Progetto realizzato da:

- Domenico Dell'Olio (mat. 699781)  
mail: [d.dellolio8@studenti.uniba.it](mailto:d.dellolio8@studenti.uniba.it) | [zeito99@gmail.com](mailto:zeito99@gmail.com)
- Giovanni Pio Delvecchio (mat. 697165)  
mail: [g.delvecchio35@studenti.uniba.it](mailto:g.delvecchio35@studenti.uniba.it) | [giovannidelvecchio4@gmail.com](mailto:giovannidelvecchio4@gmail.com)
- Giuseppe Lamantea (mat. 706593)  
mail: [g.lamantea@studenti.uniba.it](mailto:g.lamantea@studenti.uniba.it) | [lamantea.giuseppe@gmail.com](mailto:lamantea.giuseppe@gmail.com)

Il software presentato permette, a partire da un *dataset* memorizzato tramite il servizio *MySQL*, la predizione di un *attributo numerico target* per una certa porzione di esempi descritta dall'utente. Questa tecnica di *data mining* viene implementata tramite l'utilizzo di un albero di regressione costruito tramite le informazioni ricavate dalla base di dati a disposizione.

Un **albero di regressione** è una struttura ad albero utilizzata per la predizione di valori numerici, essa è formata da:

- dei **nodi interni**, detti anche *nodi di split*, che rappresentano gli attributi del *dataset*;
- degli **archi** fra nodi, che rappresentano il valore assunto dall'attributo associato al nodo da cui parte l'arco nella porzione di popolazione rappresentata dal nodo di arrivo;
- delle **foglie**, ovvero nodi senza figli, da cui si può generare una predizione del valore dell'*attributo target*. Tale predizione riguarda la porzione del *dataset* avente le caratteristiche descritte dal cammino dal nodo radice al nodo foglia in questione.

L'algoritmo che si occupa della costruzione dell'albero di regressione è così definito:

1. Determinare il nodo (attributo) in cui effettuare uno *split* genera porzioni del *dataset* con varianza dell'*attributo target* minima
2. Se un nodo generato dallo *split* rappresenta una porzione abbastanza piccola del *dataset*, esso sarà un nodo foglia
3. Ripetere il punto 1 sulle porzioni del *dataset* definite dai figli dello *split* effettuato, finché non sono state generate tutte le foglie dell'albero

## Progetto base

La versione base del progetto consiste in un'architettura *client/server* che permette all'utente di esplorare un albero di regressione creato dal server tramite varie opzioni.

Il server dovrà essere eseguito su di una macchina con un database *MySQL* in esecuzione. Le tabelle utilizzate per memorizzare i *dataset* non dovranno presentare tuple aventi attributi con valore *NULL*.

Il servizio sarà raggiungibile sulla porta 8080, e potrà comunicare con diversi client contemporaneamente.

I servizi offerti dal server (acceduti dall'utente tramite il client CLI) sono i seguenti:

- caricare un *dataset* fornendo il nome della tabella in cui è memorizzato nel database;
- creare ed esplorare un albero di regressione a partire dal *dataset* specificato;
- salvare l'albero di regressione appena creato nella macchina dove il server viene eseguito;
- esplorare un albero di regressione creato in precedenza.

Il server effettuerà dei log su console di quanto accade durante la sua esecuzione (ad esempio, connessione di un client o errore durante la creazione di un albero di regressione).

I messaggi di errore rilevanti per l'utente verranno inviati al client.

Il client da riga di comando permette di collegarsi ad una macchina che sta eseguendo un'istanza del server, la quale indirizzo IP e porta devono essere specificati all'avvio del client. L'utente dovrà specificare se creare un nuovo albero di regressione o caricarne uno creato precedentemente, e successivamente fornire il nome del *dataset* su cui lavorare.

Una volta effettuata la scelta, si avvierà l'esplorazione dell'albero, che viene presentata tramite diverse *query* testuali. L'utente quindi dovrà specificare il cammino da seguire tramite gli identificativi numerici dei nodi di split. Alla fine dell'esplorazione, all'utente sarà presentata la predizione del valore dell'*attributo target*. Sarà quindi possibile ricominciare l'esplorazione sullo stesso albero, o terminare l'esecuzione.

Il salvataggio dell'albero di regressione verrà effettuato in automatico alla creazione di un nuovo albero. L'utente sarà informato di eventuali errori sollevati dal server durante l'esecuzione delle operazioni richieste. I messaggi sono presentati in inglese, in quanto il codice fornito aveva già adottato questa scelta.

## Progetto esteso

L'estensione per il progetto consiste in un'interfaccia grafica per desktop.

Tale interfaccia è stata sviluppata tramite il framework *JavaFX*. Le funzionalità del client CLI sono state mantenute. Le feature aggiunte sono le seguenti:

- possibilità di memorizzare più server a cui connettersi (disponibili anche dopo la chiusura del programma);
- possibilità di esplorare diversi alberi su diversi server nella stessa sessione di utilizzo del programma;
- possibilità di visualizzare una schermata di aiuto;
- possibilità di lanciare il server su di una porta diversa dalla 8080;
- trascrizione dei messaggi di log del server su di un file testuale.

In generale, il client grafico è più tollerante agli errori da parte dell'utente. L'interfaccia del programma è presentata in italiano.

## Guida di installazione

*I file `.jar`, `.bat` e (dove necessario) `.sql` sono posti nella cartella `bin` di ogni progetto.*

### Installazione Server

Per eseguire correttamente il Server sulla propria macchina è necessario:

- aver installato la *Java Runtime Environment 8*
- aver installato *MySQL*
- eseguire il file `script.sql` per creare l'utente *MySQL*, il *Database* e le tabelle necessarie.

### Installazione Client (CLI e GUI)

Per eseguire correttamente il Client (sia testuale che grafico) è necessario:

- aver installato la *Java Runtime Environment 8*
- avere un server in ascolto

**NB:** il server in ascolto deve essere lo stesso presentato con il client utilizzato. Non è possibile, ad esempio, utilizzare il server presentato nel progetto base con il client grafico. Inoltre, anche gli alberi memorizzati da caricare devono essere stati creati dalla stessa versione del server che deve utilizzarli.

## Guida Utente

### Avvio del Server (progetto base)

Per avviare il Server è necessario eseguire il file `server.bat` nella stessa cartella di `server.jar`. Alternativamente è possibile avviarlo da riga di comando, tramite il comando

```
java -jar server.jar
```

### Avvio del Client (progetto base)

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 conatterà 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)

Similmente alla sua controparte nel progetto base, il server del progetto esteso è eseguibile avviando il file `server.bat` nella stessa cartella di `server.jar`. È possibile avviare il server specificando la porta dove sarà raggiungibile, eseguendo il comando tramite *shell*

```
java -jar server.jar [PORTA]
```

o sostituendo il comando presente nel file *batch* con quello sopra riportato. In caso non venga specificata alcuna porta, il server verrà eseguito sulla porta 8080.

Una volta avviato il server, verrà creato un file di *log* nella stessa cartella del file *jar*. Il file è testuale, e riporta varie informazioni sull'attività del server durante la sua esecuzione.

## Avvio del Client (progetto esteso)

Per avviare il client grafico è necessario avviare il file `client.bat` nella stessa cartella di `client.jar`. Sempre qui sarà memorizzato il file binario `servers.info` contenente le informazioni sui server memorizzati dall'utente.

## Guida di utilizzo

### Utilizzo del Client (progetto base)

Una volta avviato il Client, se la connessione con il server va a buon fine si specifica quale operazione si vuole effettuare.

```
Socket[addr=localhost/127.0.0.1,port=8080,localport=63307]
Learn Regression Tree from data [1]
Load Regression Tree from archive [2]
|
```

Una volta effettuata la scelta, si dovrà inserire il nome della tabella in cui è memorizzato il *dataset* su cui costruire l'albero.

Infine, l'utente potrà esplorare l'albero specificando le caratteristiche della porzione di dataset con cui effettuare la predizione sull'attributo target. Questo processo, all'atto pratico, si traduce nell'inserire gli identificativi numerici dei nodi di split su cui spostarsi (fino all'arrivo su di un nodo foglia).

```
File name:
servo
Starting data acquisition phase!
Starting learning phase!
Starting prediction phase!
0:pgain<=3.0
1:pgain>3.0

0
0:motor=A
1:motor=B
2:motor=C
3:motor=D
4:motor=E
```

Una volta finita un'esplorazione, si potrà avviarne una nuova sullo stesso albero.

```
2
0:vgain<=2.0
1:vgain>2.0

0
0:motor=A
1:motor=B
2:motor=C
3:motor=E

2
Predicted class:0.30000108999999997
Would you repeat ? (y/n)
|
```

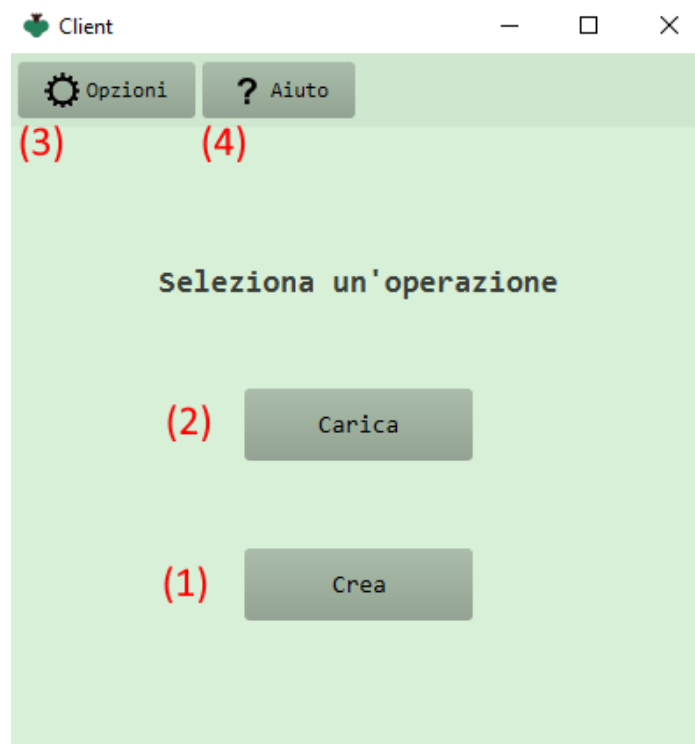
## Utilizzo del Client (progetto esteso)

### Schermata Home

Una volta avviato il Client grafico, ci si troverà sulla schermata principale, dalla quale possiamo scegliere se:

- creare un nuovo albero di regressione (1)
- caricare un albero di regressione creato in precedenza (2)

Accedendo alla schermata delle impostazioni (3) si potrà modificare la lista dei server conosciuti. Infine, è possibile accedere ad una schermata di aiuto (4) per avere informazioni sul programma.

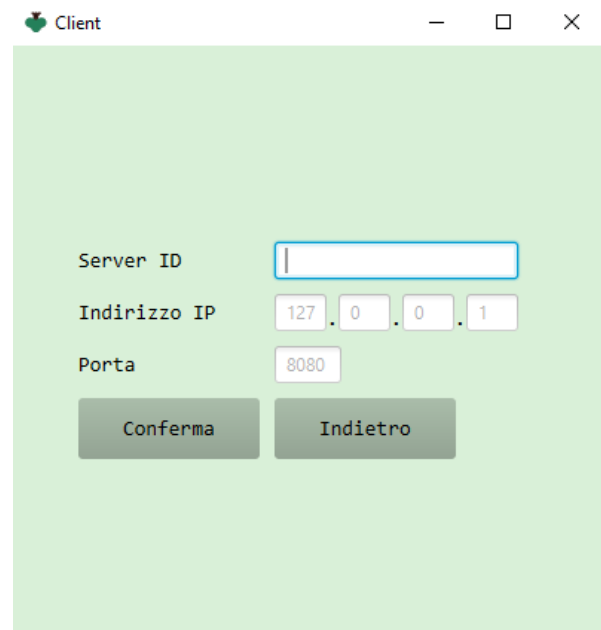
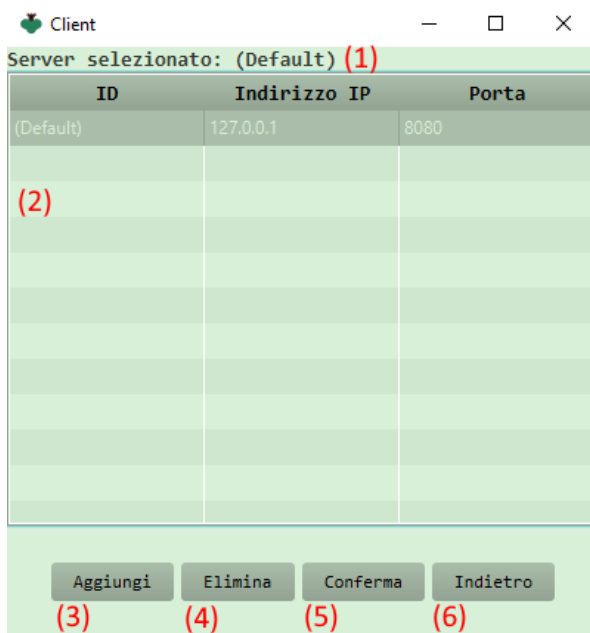


## Schermata delle Impostazioni

Dalla schermata delle impostazioni è possibile:

- visualizzare il server attualmente in uso (1)
- visualizzare i server memorizzati dall'utente (2)
- aggiungere un nuovo server (3)
- cancellare un server (4)
- impostare il server corrente (5).

È possibile tornare nella schermata della home tramite il pulsante Indietro (6)

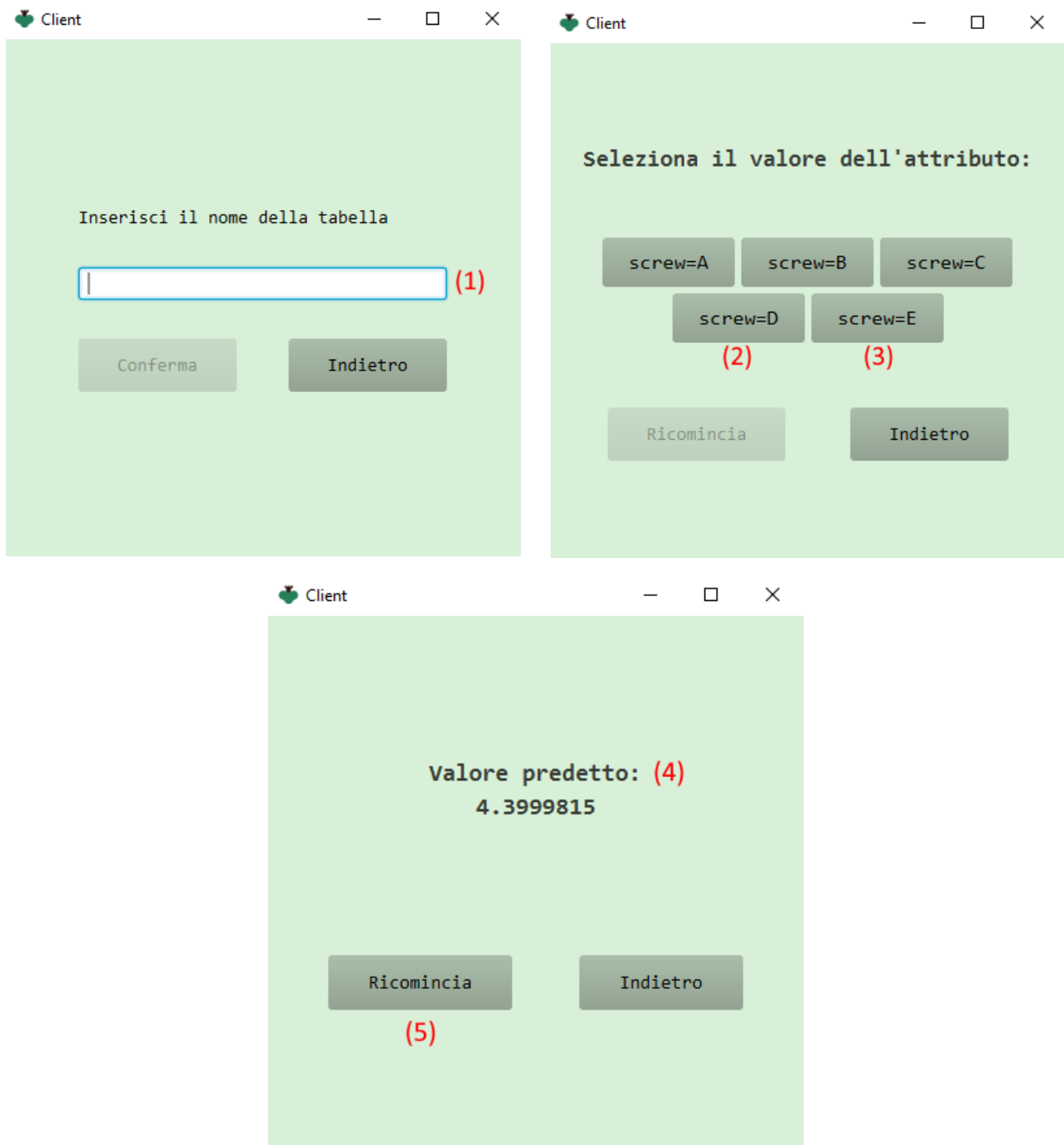


## Schermata di Predizione

Una volta selezionata l'operazione da effettuare, si dovrà inserire il nome della tabella contenente il *dataset* (1).

Fatto ciò, si passerà alla schermata di predizione, dove l'utente dovrà selezionare i valori degli attributi che descrivono la porzione di *dataset* su cui effettuare la predizione. Quest'operazione viene effettuata cliccando sui pulsanti che rappresentano le eventuali scelte possibili (2) (3).

Una volta terminata l'esplorazione dell'albero, sarà presentato il valore predetto (4), e sarà possibile ricominciare la predizione tramite il tasto Ricomincia (5).



# Esempi di Test

*Gli output numerici attesi sono stati ricavati dalla documentazione fornita insieme alle esercitazioni effettuate durante il corso.*

**Tabella di Input:** inesistente

**Output atteso:** messaggio di errore riguardante l'assenza della tabella specificata.

**Output:**

```
Socket[addr=localhost/127.0.0.1,port=8080,localport=52334]
Learn Regression Tree from data [1]
Load Regression Tree from archive [2]
1
File name:
motore
Starting data acquisition phase!
data.TrainingDataException: Table not found
```

**Tabella di Input:** "vuota"

**Output atteso:** messaggio di errore riguardante l'assenza di esempi nel dataset.

**Output:**

```
Socket[addr=localhost/127.0.0.1,port=8080,localport=52373]
Learn Regression Tree from data [1]
Load Regression Tree from archive [2]
1
File name:
vuota
Starting data acquisition phase!
data.TrainingDataException: Empty table
```

**Tabella di Input:** "notarget"

**Output atteso:** messaggio di errore riguardante l'assenza di un attributo target (deve essere sempre presente come ultimo attributo della tabella).

**Output:**

```
Socket[addr=localhost/127.0.0.1,port=8080,localport=52409]
Learn Regression Tree from data [1]
Load Regression Tree from archive [2]
1
File name:
notarget
Starting data acquisition phase!
data.TrainingDataException: Missing class attribute
```

**Tabella di Input:** "few\_attributes"

**Output atteso:** messaggio di errore riguardante la mancanza di un numero sufficiente di attributi nel *dataset*.

**Output:**

```
Socket[addr=localhost/127.0.0.1,port=8080,localport=52514]
Learn Regression Tree from data [1]
Load Regression Tree from archive [2]
1
File name:
few_attributes
Starting data acquisition phase!
data.TrainingDataException: Attributes must be at least 2
```



**Tabella di Input:** tabella non memorizzata (tramite caricamento)

**Output atteso:** messaggio di errore riguardante l'assenza di un albero memorizzato con tale nome.

**Output:**

```
Learn Regression Tree from data [1]
Load Regression Tree from archive [2]
2
File name:
motore
java.io.FileNotFoundException: motore.dmp (Impossibile trovare il file specificato)
```

**Tabella di Input:** "null\_values"

**Output atteso:** messaggio di errore riguardante la presenza di attributi nulli nel *dataset*.

**Output:**

```
Learn Regression Tree from data [1]
Load Regression Tree from archive [2]
1
File name:
null_values
Starting data acquisition phase!
data.TrainingDataException: The table has NULL values
```

**Tabella di Input:** "provaC"

**Scelte effettuate durante l'esplorazione dell'albero:** 0 – 2 (X = A, nodo non esistente)

**Output atteso:** messaggio di errore riguardante la scelta di un nodo non esistente.

**Output:**

```
Socket[addr=localhost/127.0.0.1,port=8080,localport=52553]
Learn Regression Tree from data [1]
Load Regression Tree from archive [2]
1
File name:
provac
Starting data acquisition phase!
Starting learning phase!
Starting prediction phase!
0:X=A
1:X=B

0
0:Y<=2.0
1:Y>2.0

2
server.UnknownValueException: The answer should be an integer between 0 and 1!
Would you repeat ? (y/n)
n
```

**Tabella di Input: “prova”**

**Scelte effettuate durante l’esplorazione dell’albero: 0 – 0 (X = A, Y = A)**

**Output atteso: 1.0**

**Output:**

```
Learn Regression Tree from data [1]
Load Regression Tree from archive [2]
1
File name:
prova
Starting data acquisition phase!
Starting learning phase!
Starting prediction phase!
0:X=A
1:X=B

0
0:Y=A
1:Y=B

0
Predicted class:1.0
Would you repeat ? (y/n)
n
```

**Tabella di Input: “provaC”**

**Scelte effettuate durante l’esplorazione dell’albero: 0 – 1 (X = A, Y > 2.0)**

**Output atteso: 1.5**

**Output:**

```
Learn Regression Tree from data [1]
Load Regression Tree from archive [2]
2
File name:
provac
Starting prediction phase!
0:X=A
1:X=B

0
0:Y<=2.0
1:Y>2.0

1
Predicted class:1.5
Would you repeat ? (y/n)
n
```

Non sono stati riportati esempi di test con la tabella `servo` poiché l’albero utilizzato come riferimento utilizza attributi discreti per rappresentare `pgain` e `vgain`, mentre la versione di `servo` utilizzata nel progetto impiega attributi continui.