

## GUIDA UTENTE E DESCRIZIONE DEL PROGETTO BASE

Lo scopo generale della costruzione di un albero di regressione a partire da un database è quella di poter estrarre automaticamente conoscenza da una base di dati e renderla disponibile e utilizzabile.

L'applicazione qui descritta consente quindi all'utente di poter creare un albero di regressione a partire da un insieme di esempi presenti in una tabella, a sua volta definita in un database. Sull'albero è inoltre possibile effettuare delle operazioni di predizione del valore di un attributo di classe continuo (numerico) associato a un esempio sulla base di valori osservati per altri attributi dell'esempio stesso.

L'applicazione è basata sul modello client-server perciò, per poterla utilizzare, bisogna per prima cosa avviare il server e successivamente avviare il/i client che comunicheranno con esso. Il client a sua volta fornisce un'interfaccia all'utente che gli permette di interagire con il sistema ed eseguire le operazioni che esso mette a disposizione. Se il client viene avviato prima del server il sistema comunica all'utente un errore di connessione:

[java.net.ConnectException](#): Connection refused: connect

### **GUIDA UTENTE**

Per poter avviare l'applicazione è necessario utilizzare i file batch presenti nella cartella '*Jar+Batch+ScriptSQL Base*'. La guida per l'avvio è allegata nella cartella sopracitata.

Una volta avviata l'applicazione, all'utente verrà mostrata a video la seguente schermata:

```
Learn Regression Tree from data [1]
Load Regression Tree from archive [2]
```

Digitando '1' all'utente verrà consentito di elaborare degli esempi presi da una nuova tabella del database, altrimenti digitando '2' l'utente potrà caricare un albero di regressione già presente all'interno dell'archivio. Se l'utente inserisce un comando diverso da 1 o 2, l'applicazione lo inviterà a reinserire un comando corretto.

Nel caso in cui l'utente digiti 1, l'applicazione gli chiederà di inserire il nome della tabella da utilizzare. Il client comunicherà al server la scelta dell'utente il quale accederà al database, preleverà da esso la tabella corrispondente a quella inserita dall'utente, ne estrarrà gli esempi e infine creerà l'albero di regressione con cui poter poi fare le predizioni. Se l'utente inserisce il nome di una tabella sbagliato, il sistema lancerà un'eccezione di tipo `TrainingDataException`, comunicando all'utente il problema con il messaggio:

The table doesn't exists!

e verrà chiusa l'applicazione.

Se il nome inserito è di una tabella effettivamente esistente, ma non è corretta strutturalmente per poter derivare da essa un albero di regressione, verranno mostrati all'utente i seguenti messaggi di errore e successivamente verrà chiusa l'applicazione:

- Se la tabella non è stata popolata verrà mostrato il messaggio(testare con tabella provaEmpty):  
The table has 0 rows
- Se la tabella ha meno di due colonne verrà mostrato il messaggio(testare con tabella provaD):  
The table has less than 2 columns!

- Se la tabella ha come ultimo attributo un valore discreto verrà mostrato il messaggio(testare con provaCont):  
The last column is not a continuous value
- Se la tabella ha un valore nullo in una delle sue colonne verrà mostrato il messaggio(testare con tabella provaNull):  
Found a null value in a row

Nel caso invece in cui l'utente digiti 2, gli verrà chiesto di inserire il nome del file, il quale sarà già presente nell'archivio. Il file specificato contiene l'albero di regressione desiderato, precedentemente generato dal database, che verrà caricato. Se l'utente invece inserisce il nome sbagliato di un file, il sistema comunicherà l'errore con il seguente messaggio:

nomeFile.dmp (Impossibile trovare il file specificato)

Se i passaggi precedentemente descritti vanno a buon fine, il sistema darà inizio alla fase di predizione: all'utente verrà mostrato l'elenco dei rami dell'albero che potrà scegliere per la predizione e potrà selezionarne uno tra quelli proposti attraverso l'inserimento del numero che si riferisce al ramo desiderato. Questo procedimento viene ripetuto finché non viene trovato il risultato della predizione.

Ecco un esempio del procedimento appena descritto:

Starting prediction phase!...

0:X=A

1:X=B

Choice: 0

0:Y<=2.0

1:Y>2.0

Choice: 1

Se l'utente inserisce un valore sbagliato nella scelta del ramo da percorrere, gli verrà mostrato un messaggio del tipo:

The answer should be an integer between 0 and 1!

Quando viene raggiunto un risultato, esso sarà comunicato all'utente con la successiva richiesta di ripetizione della predizione:

Predicted class:1.5

Would you repeat ? (y/n)

Se l'utente inserisce il comando 'y' verrà ripetuta l'intera procedura di predizione, altrimenti con qualsiasi altro comando verrà chiusa l'applicazione.

Ecco un esempio completo di utilizzo dell'applicazione:

Learn Regression Tree from data [1]

Load Regression Tree from archive [2]

1

Table name:

provaC

Starting data acquisition phase!...

Starting learning phase!...  
Starting prediction phase!...  
0:X=A  
1:X=B

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

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

y  
Starting prediction phase!...  
0:X=A  
1:X=B

Choice: 3  
The answer should be an integer between 0 and 1!  
Would you repeat ? (y/n)  
n

### Informazioni utili sull'applicazione:

- Il DBMS utilizzato per lo sviluppo dell'applicazione è MySQL tramite il quale accediamo al database MapDB fornito nello script SQL.
- Il client comunicherà con il server durante l'utilizzo dell'applicazione connettendosi alla porta 8080 del server. Se si tenta di avviare nuovamente l'applicazione senza aver terminato l'esecuzione precedente, potrebbero esserci degli errori di avvio del server poiché la porta risulterà occupata ([java.net.BindException: Address already in use: bind](#)). Sarà quindi necessario chiudere una delle istanze dell'applicazione per assicurare il corretto funzionamento di essa.
- Per leggere la documentazione relativa al progetto sono stati generati i Javadoc e si trovano nelle seguenti cartelle:

*MAP\_Citarella\_Iacobellis\_Ladisa\Base\Sorgente Base\mapClient\doc* per i javadoc del client e

*MAP\_Citarella\_Iacobellis\_Ladisa\Base\Sorgente Base\mapServer\doc* per i javadoc del server.

