

A thick vertical olive-green bar is positioned on the left side of the page. Below it, several thin, curved lines in shades of olive and grey sweep upwards from the bottom left corner.

Documentazione

Regression Tree Miner

Metodi Avanzati di
Programmazione

A.A. 2019/2020

Docente: Prof.ssa Appice
Annalisa

Francesco Lavecchia
MATRICOLA 670654

INTRODUZIONE

Il progetto realizzato crea alberi di regressione sulla base di dataset estraibili da tabelle all'interno di un database MySQL.

La regressione è una delle aree di applicazione nel campo del Data Mining. Lo scopo del Data Mining è l'estrazione semi-automatica di conoscenza, nascosta in voluminose basi di dati al fine di renderla disponibile e direttamente fruibile. La regressione, in particolare, si occupa della predizione del valore di un attributo numerico associato a un esempio sulla base di valori osservati per altri attributi dell'esempio medesimo.

Come funziona l'algoritmo?

L'algoritmo costruisce una struttura ad albero partendo da esempi descritti come vettori di coppie attributo-valore, per i quali è noto l'attributo numerico di target. Nelle funzioni di regressione, ogni nodo interno rappresenta una variabile, ogni arco verso un nodo figlio rappresenta un possibile valore per quella proprietà, e ogni nodo foglia rappresenta il valore predetto per quella classe a partire dai valori delle altre proprietà.

Data una collezione di esempi di apprendimento, che chiamiamo *training set*, ciascun esempio è una tupla di valori per un prefissato insieme di attributi, incluso un attributo di classe numerico (*target*). Un attributo può essere descritto come continuo o discreto a seconda che i suoi valori siano rispettivamente numerici o nominali. L'attributo di classe è continuo e ha valori nell'insieme dei numeri reali.

Nella tabella sottostante, un esempio di dataset avente due attributi X1 e X2 discreti e un attributo di target Y:

X1	X2	Y
A	A	1
A	A	1
A	A	1
A	A	1
A	B	1,5
A	B	1,5
A	B	1,5
B	B	10
A	B	1,5
A	B	1,5
B	C	10
B	B	10
B	C	10
B	C	10
A	A	1

Come determinare quale attributo permette di costruire la migliore funzione di regressione? La risposta sta nel calcolare l'errore quadratico medio (SSE). Sia:

- S la porzione di esempi di training correttamente analizzati
- Y la variabile di classe

La varianza di Y in S è calcolata come:

$$SSE(S) = \sum_{i \in S} (Y(i) - \bar{Y})^2 = \sum_{i \in S} Y(i)^2 - \frac{(\sum_{i \in S} Y(i))^2}{size(S)}$$

SSE(S) è una misura della variabilità contenuta in S. Assume 0 se solo tutti gli eventi sono associati allo stesso valore di Y.

Sia S_1, \dots, S_t il partizionamento di S per il test t sull'attributo A_i :

$$SSE(S, t) = \sum_i SSE(S_i)$$

Il criterio basato sulla varianza sceglie il test t che minimizza $SSE(S, t)$.

L'esempio all'interno della tabella mostrata in figura contiene 15 valori di apprendimento:

Y: 1 – 1 – 1 – 1 – 1,5 – 1,5 – 1,5 – 10 – 1,5 – 1,5 – 10 – 10 – 10 – 10 – 1

SSE(S) = **255,833**

Varianza per X1:

X1: A (1 – 1 – 1 – 1 – 1,5 – 1,5 – 1,5 – 1,5 – 1), B (10 – 10 – 10 – 10 – 10)

Il test su X1 partiziona S come segue:

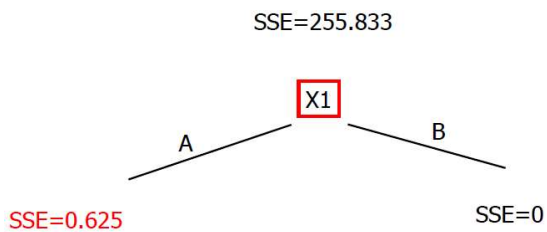
$SSE(S, X1) = SSE(S, X1 = A) + SSE(S, X1 = B) = 0,625 + 0 = \mathbf{0,625}$

Varianza per X2:

X2: A (1 – 1 – 1 – 1 – 1), B (1,5 – 1,5 – 1,5 – 10 – 1,5 – 1,5 – 10), C (10 – 10 – 10)

Il test su X2 partiziona S come segue:

$$SSE(S, X_2) = SSE(S, X_2 = A) + SSE(S, X_2 = B) + SSE(S, X_3 = C) = 0 + 103,21 + 0 = \mathbf{103,21}$$



Come identificare le possibili soglie Θ per l'attributo continuo A? Si ordinano gli esempi sulla base dei valori dell'attributo A (es. tramite l'algoritmo di quicksort). In secondo luogo, per ciascun valore distinto risultante dall'ordinamento, si considera una possibile soglia x per un test $A \leq x$ vs $A > x$.

X1	X2	Y
A	2	1
A	2	1
A	1	1
A	2	1
A	5	1,5
A	5	1,5
A	6	1,5
B	6	10
A	6	1,5
A	6	1,5
B	10	10
B	5	10
B	12	10
B	14	10
A	1	1

Per l'esempio di training a sinistra:

$$1 \rightarrow SSE(S, X_2 \leq 1, X_2 > 1) = 232,69$$

1

$$2 \rightarrow SSE(S, X_2 \leq 2, X_2 > 2) = 180,625$$

2

$$5 \rightarrow SSE(S, X_2 \leq 5, X_2 > 5) = 192,85$$

5

$$6 \rightarrow SSE(S, X_2 \leq 6, X_2 > 6) = 128,22$$

6

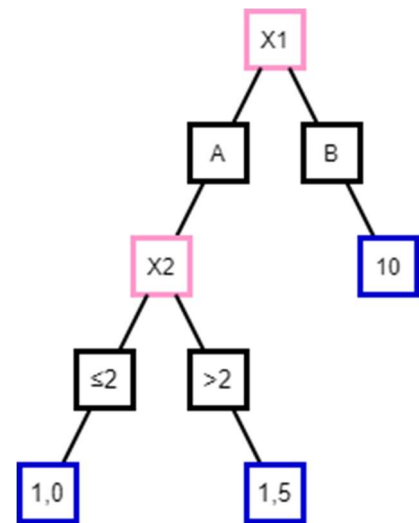
6

6

$$10 \rightarrow SSE(S, X_2 \leq 10, X_2 > 10) = 177,30$$

$$12 \rightarrow SSE(S, X_2 \leq 12, X_2 > 12) = 219,375$$

14



L'algoritmo sarà:

```
learnTree(Table S, int begin, int end) {
    if (isLeaf(S, begin, end)) {
        root = new LeafNode(S, begin, end);
    } else {
        root = determineBestSplitNode(S, begin, end);
        childTree = new RegressionTree[root.getNumberOfChildren()];
        for (int i=0; i<root.getNumberOfChildren(); i++) {
            childTree[i] = new RegressionTree();
            childTree[i].learnTree(S, root.begin, root.end);
        }
    }
}
```

DESCRIZIONE DEL PROGETTO

Il progetto consiste in un sistema Client-Server denominato "Regression Tree Miner". Il server include funzionalità di data mining per l'apprendimento di alberi di regressione e uso degli stessi come strumento di previsione. Il client è un applicativo Java che consente di effettuare previsioni usufruendo del servizio di predizione remoto. Il server deve essere lanciato sulla porta 8080 di default. Una volta lanciato, resterà in ascolto affinché uno o più client tenteranno di connettersi. Una volta stabilita la connessione, l'utente, tramite il client, può chiedere al server di effettuare una o più delle seguenti operazioni:

- Caricare un training set da una tabella del database, indicando il nome della tabella stessa;
- Caricare un albero già appreso in precedenza da un file all'interno dell'archivio, indicando il nome del file stesso;
- Apprendere il training set per generare un albero di regressione;
- Salvare il risultato dell'apprendimento su file lato server;
- Stampare a video sul client l'albero appreso;
- Avviare la fase di predizione che consente all'utente di scegliere, a ogni iterazione, il nodo figlio affinché, una volta arrivati al nodo foglia, viene stampato a video sul client il valore della predizione per quella classe.
- Chiudere la connessione.

Il Client deve essere lanciato specificando indirizzo IP e porta del server. Una volta lanciato, tenterà di stabilire una connessione con il server, che in caso di successo, inviterà l'utente ad effettuare una scelta tra:

1. Apprendere l'albero dal database;
2. Caricare l'albero dall'archivio.

Caso 1:

Il client chiede al server di fornirgli i nomi delle tabelle presenti nel database in quel momento, permettendo all'utente di poter effettuare una scelta tra le tabelle mostrate. Una volta effettuata la scelta e appreso l'albero, questo verrà serializzato e salvato su file all'interno dell'archivio del server.

Caso 2:

Il client chiede al server di fornirgli i nomi dei file presenti nell'archivio del server in quel momento, permettendo all'utente di poter effettuare una scelta tra i file mostrati.

Una volta effettuata una delle due scelte descritte precedentemente, l'utente può scegliere di poter visualizzare l'albero in formato testuale oppure di avviare la fase di predizione. In caso venga selezionata la prima scelta, verrà mostrato su video lato client una rappresentazione testuale dell'albero appreso. Nel caso in cui si voglia avviare la fase di predizione invece, verrà scandito l'albero e verrà data la possibilità all'utente di scegliere quale ramo dell'albero percorrere affinché si raggiunga un nodo foglia. Una volta giunti al nodo foglia, verrà mostrato a video il valore di predizione per quella classe.

L'estensione del progetto consiste in un applicativo per Android, il quale fornisce un'interfaccia grafica molto più user friendly rispetto all'interfaccia a riga di comando del client originale. Il client Android contiene le funzioni descritte precedentemente. È in grado di gestire le eccezioni come la perdita di connessione, la mancanza di tabelle all'interno del database, ecc. mostrando a video un Dialog che informa l'utente dell'errore riscontrato. Inoltre, all'interno delle impostazioni dell'applicazione, è possibile cambiare il tema scegliendo tra *Day Mode* (dove i colori dell'interfaccia sono più chiari) e *Night Mode* (dove i colori dell'interfaccia sono più scuri).

GUIDA ALL'INSTALLAZIONE

Requisiti:

Server:

- Installare JRE 8 o versioni successive;
- Installare MySQL server;
- Connessione ad Internet;
- Accedere al database con l'utente root ed eseguire lo script SQL contenuto nel file "setupDB.sql" all'interno della cartella "setup". Il percorso sarà "/setup/setupDB.sql".

Client:

- Installare JRE 8 o versioni successive;
- Connessione ad Internet sulla stessa LAN del server;
- Server in ascolto.

Client Android:

- Dispositivo con a bordo Android 4.1 Jelly Bean o versioni successive;
- Connessione ad Internet sulla stessa LAN del server;
- Server in ascolto.

Avviare il server eseguendo il file batch il cui percorso è "/setup/runServer.bat": runServer avvierà l'eseguibile JAR raggiungibile all'indirizzo "/Server/mapServer.jar". Una volta avviato il server, esso rimarrà in ascolto aspettando la connessione di uno o più client.

Per il client JAR:

Avviare il client eseguendo il file batch il cui percorso è "/setup/runClient.bat": runClient avvierà l'eseguibile JAR raggiungibile all'indirizzo "/Client/mapClient.bat".

ATTENZIONE: Per poter avviare correttamente il client occorre impostare l'indirizzo del server identificato da IP e porta. Per fare ciò, aprire il file runClient.bat con un editor di testo e modificare il penultimo valore con l'IP del server e l'ultimo valore con la porta del server.

Modello: `start cmd /K java -jar ../Client/mapClient.jar IP PORTA`

Ad esempio: `start cmd /K java -jar ../Client/mapClient.jar 192.168.43.106 8080`

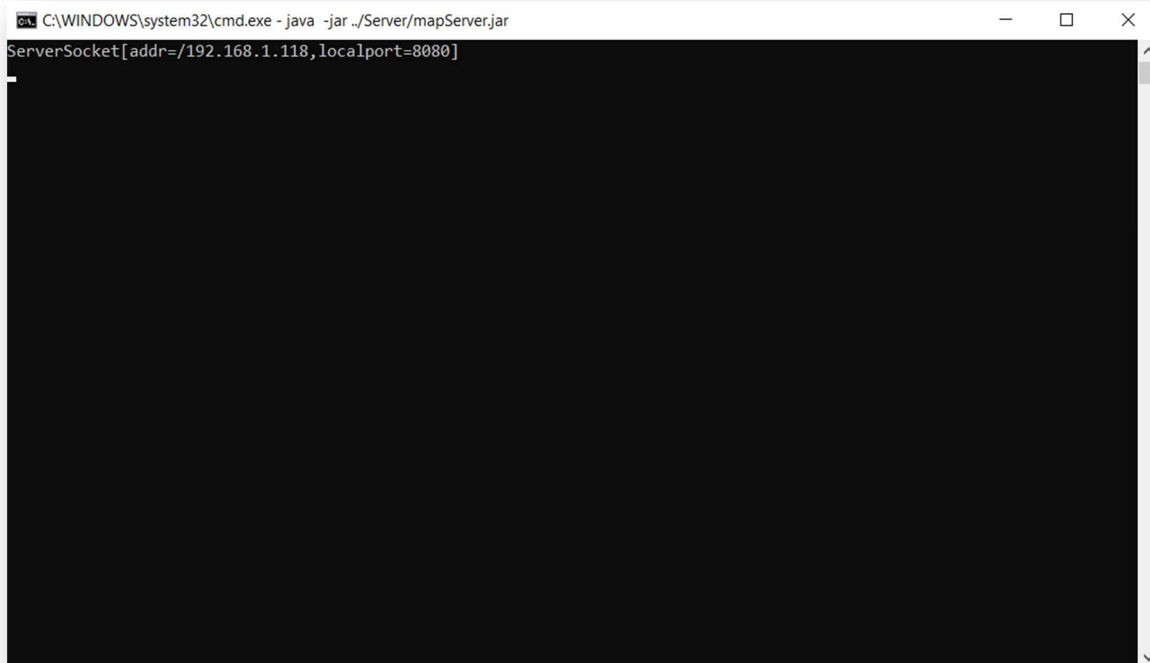
Per il client Android:

Installare sul dispositivo l'apk denominato "app-release.apk" il cui percorso è "/Android Client/apk-release.apk" e avviarlo. Specificare indirizzo IP e porta del server all'interno delle impostazioni raggiungibili dal menu principale.

GUIDA ALL'USO

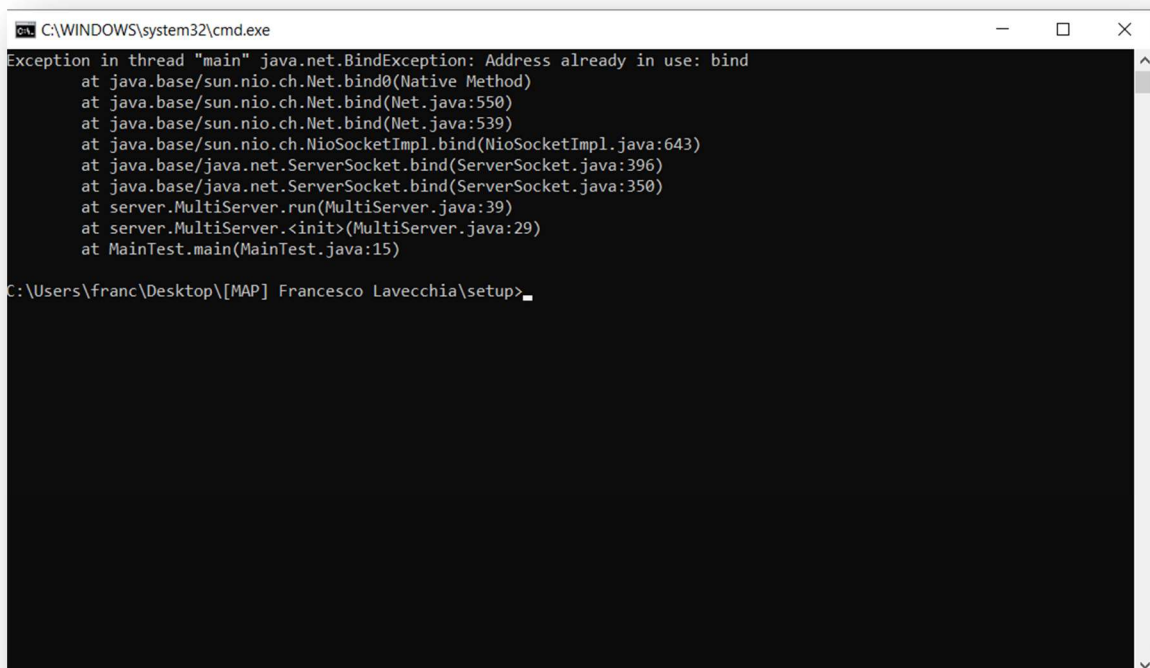
Server

Una volta avviato il server, esso ci mostrerà il proprio IP e la porta dove resterà in ascolto:



```
C:\WINDOWS\system32\cmd.exe - java -jar ../Server/mapServer.jar
ServerSocket[addr=/192.168.1.118,localport=8080]
```

Nel caso in cui la porta 8080 sia già stata occupata, verrà lanciata un'eccezione. SOLUZIONE: chiudere qualunque processo o server che occupa la porta 8080 prima di procedere.



```
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:39)
    at server.MultiServer.<init>(MultiServer.java:29)
    at MainTest.main(MainTest.java:15)

C:\Users\franc\Desktop\[MAP] Francesco Lavecchia\setup>
```

Ogni volta che un client si connette o effettua qualche altra operazione, il server mostra l'indirizzo del Client, l'orario in cui ha effettuato l'operazione, e una descrizione dell'operazione stessa:

```
C:\WINDOWS\system32\cmd.exe - java -jar ./Server/mapServer.jar
ServerSocket[addr=/192.168.1.118,localport=8080]
[192.168.1.118 | 11/12/2020 11:31:54] - Connesso.
[192.168.1.118 | 11/12/2020 11:31:54] - Sto ascoltando...
[192.168.1.118 | 11/12/2020 11:31:59] - Restituisco al client la lista delle tabelle.
[192.168.1.118 | 11/12/2020 11:32:00] - Sto ascoltando...
[192.168.1.112 | 11/12/2020 11:32:18] - Connesso.
[192.168.1.112 | 11/12/2020 11:32:18] - Sto ascoltando...
[192.168.1.112 | 11/12/2020 11:32:18] - Restituisco al client la lista delle tabelle.
[192.168.1.112 | 11/12/2020 11:32:18] - Sto ascoltando...
[192.168.1.112 | 11/12/2020 11:32:23] - Disconnesso.
[192.168.1.118 | 11/12/2020 11:32:29] - Disconnesso.
```

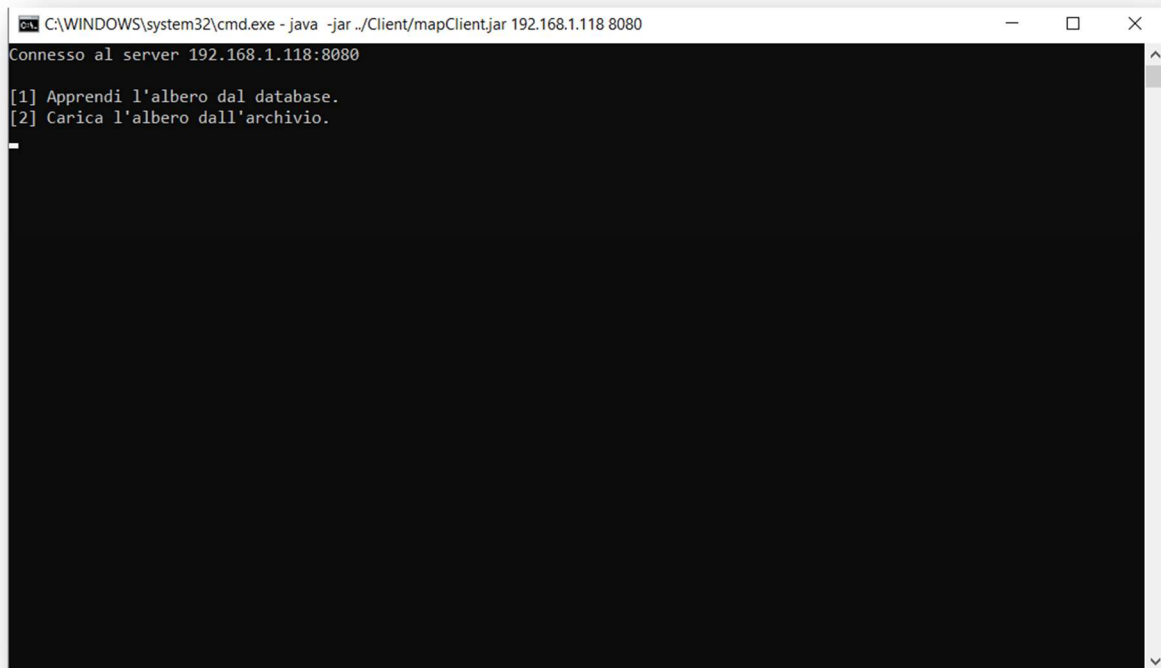
Il server rimarrà sempre in ascolto affinché non venga chiuso l'applicativo runServer.bat.

Client JAR

Una volta avviato il server, avviare il client con i corretti argomenti (IP e porta del server). Se l'indirizzo IP e la porta non corrispondono a nessun server in ascolto, il client ci comunicherà che il tentativo di connessione è fallito. SOLUZIONE: Inserire gli argomenti IP e porta corretti o verificare la connessione.

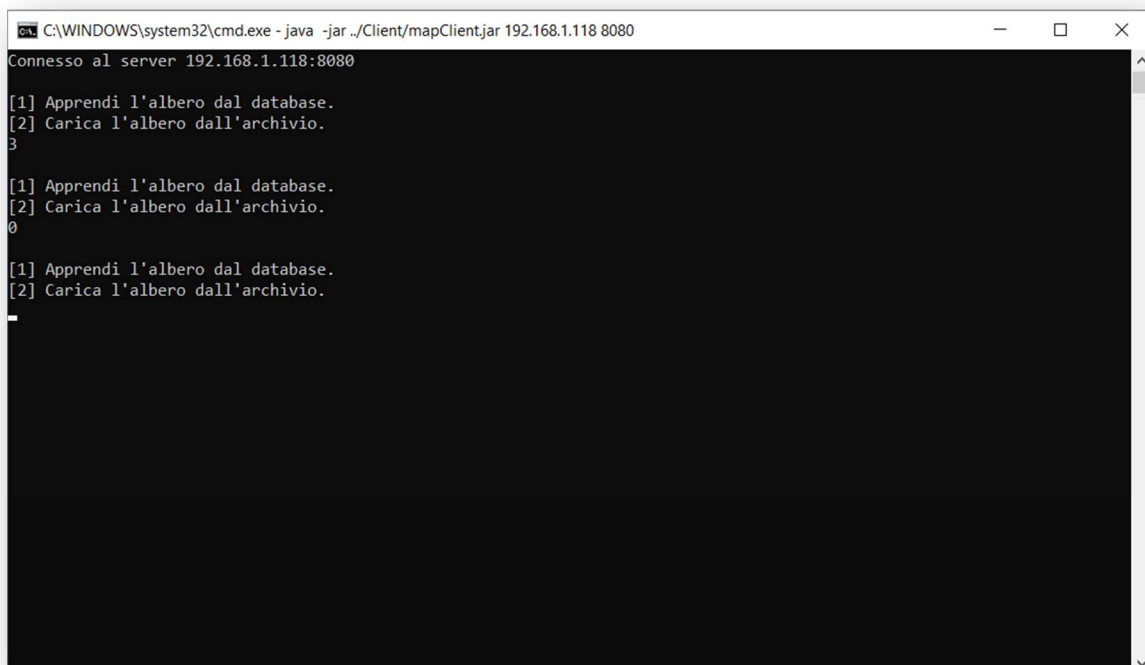
```
C:\WINDOWS\system32\cmd.exe
[!] Connessione fallita.
C:\Users\franc\Desktop\[MAP] Francesco Lavecchia\setup>
```

In caso di connessione avvenuta con successo, ci verrà chiesto di effettuare una scelta:



```
C:\WINDOWS\system32\cmd.exe - java -jar ./Client/mapClient.jar 192.168.1.118 8080
Connesso al server 192.168.1.118:8080
[1] Apprendi l'albero dal database.
[2] Carica l'albero dall'archivio.
```

Il client accetta come input soltanto le scelte "1" e "2". In caso contrario, non è possibile proseguire finché non inseriamo un valore tra quelli suggeriti (in questo caso, 1 o 2).



```
C:\WINDOWS\system32\cmd.exe - java -jar ./Client/mapClient.jar 192.168.1.118 8080
Connesso al server 192.168.1.118:8080
[1] Apprendi l'albero dal database.
[2] Carica l'albero dall'archivio.
3
[1] Apprendi l'albero dal database.
[2] Carica l'albero dall'archivio.
0
[1] Apprendi l'albero dal database.
[2] Carica l'albero dall'archivio.
```

- Scelta 1: Si avvierà la fase di apprendimento dell'albero partendo da un training set all'interno del database. Va inserito il nome della tabella tra quelle elencate. Una volta effettuata la scelta, verrà prelevato un training set dal database e appreso.

- Scelta 2: Si avvierà la fase di deserializzazione partendo da un albero precedentemente serializzato su un file. Va inserito il nome del file tra quelli elencati. Una volta effettuata la scelta, verrà caricato l'albero che era salvato su file.

```
C:\WINDOWS\system32\cmd.exe - java -jar ../Client/mapClient.jar 192.168.1.118 8080
Connesso al server 192.168.1.118:8080

[1] Apprendi l'albero dal database.
[2] Carica l'albero dall'archivio.
1

Scegli una delle tabelle elencate qui sotto:
[prova, provac, servo, servoc]

Scegli il nome della tabella:
-
```

Nel caso in cui il database (scelta 1) o l'archivio (scelta 2) fossero vuoti, il client ce lo comunicherebbe, invitandoci ad effettuare di nuovo la scelta.

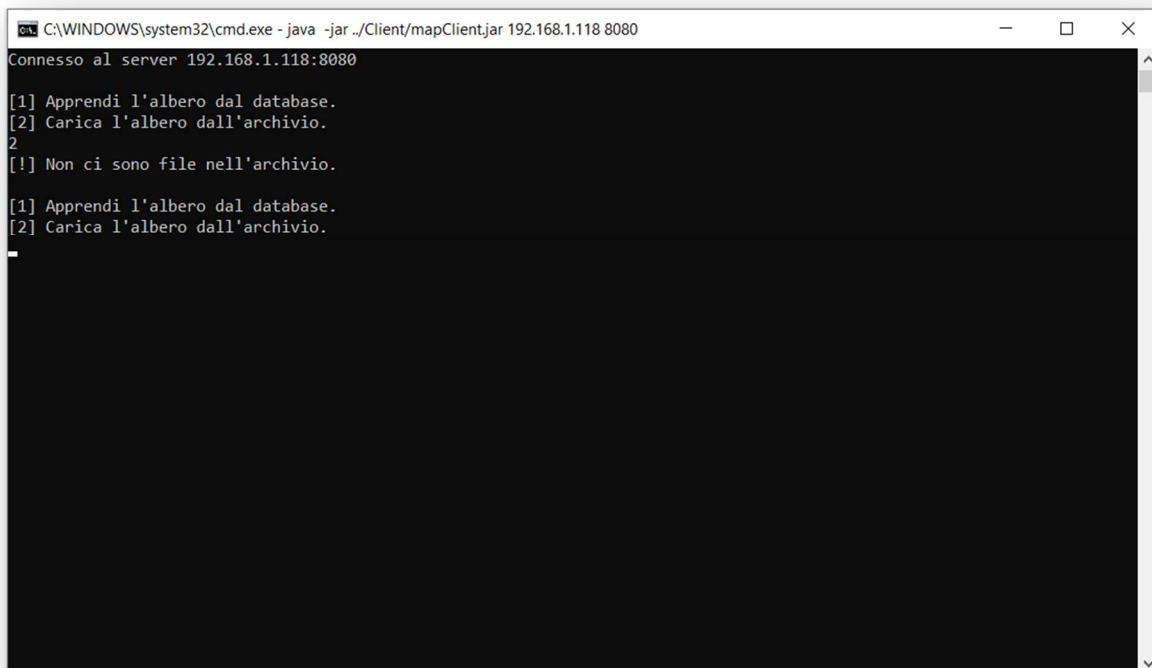
Nessuna tabella:

```
C:\WINDOWS\system32\cmd.exe - java -jar ../Client/mapClient.jar 192.168.1.118 8080
Connesso al server 192.168.1.118:8080

[1] Apprendi l'albero dal database.
[2] Carica l'albero dall'archivio.
1
[!] Non ci sono tabelle nel database.

[1] Apprendi l'albero dal database.
[2] Carica l'albero dall'archivio.
-
```

Nessun file:



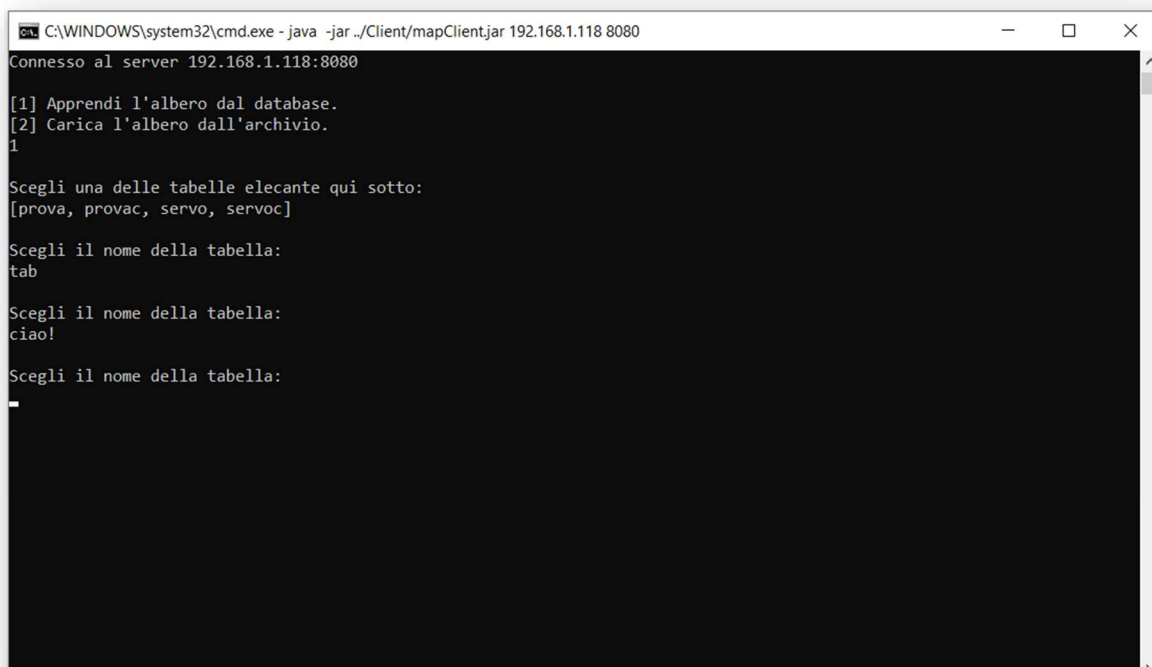
```
C:\WINDOWS\system32\cmd.exe - java -jar ./Client/mapClient.jar 192.168.1.118 8080
Connesso al server 192.168.1.118:8080

[1] Apprendi l'albero dal database.
[2] Carica l'albero dall'archivio.
2
[!] Non ci sono file nell'archivio.

[1] Apprendi l'albero dal database.
[2] Carica l'albero dall'archivio.
_
```

Durante l'inserimento del nome della tabella o del file, va effettuata la scelta tra i nomi mostrati. Non potremmo proseguire in caso contrario.

Tabella non esistente:



```
C:\WINDOWS\system32\cmd.exe - java -jar ./Client/mapClient.jar 192.168.1.118 8080
Connesso al server 192.168.1.118:8080

[1] Apprendi l'albero dal database.
[2] Carica l'albero dall'archivio.
1

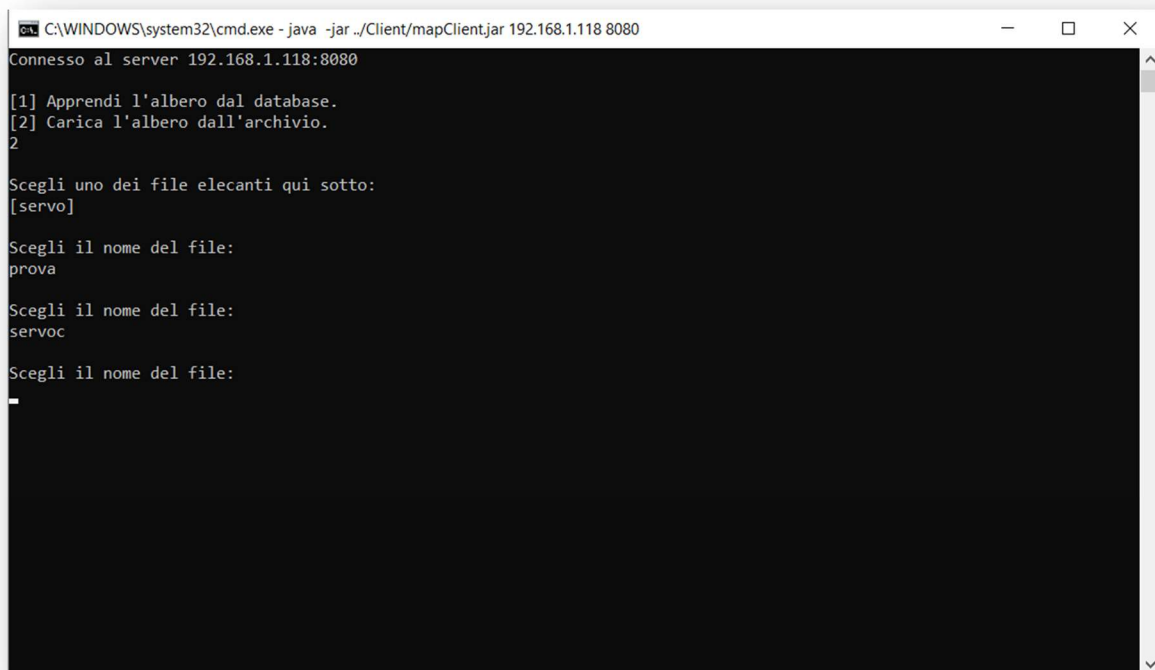
Scegli una delle tabelle elencate qui sotto:
[prova, provac, servo, servoc]

Scegli il nome della tabella:
tab

Scegli il nome della tabella:
ciao!

Scegli il nome della tabella:
_
```

File non esistente:



```
C:\WINDOWS\system32\cmd.exe - java -jar ./Client/mapClient.jar 192.168.1.118 8080
Connesso al server 192.168.1.118:8080

[1] Apprendi l'albero dal database.
[2] Carica l'albero dall'archivio.
2

Scegli uno dei file elencati qui sotto:
[servo]

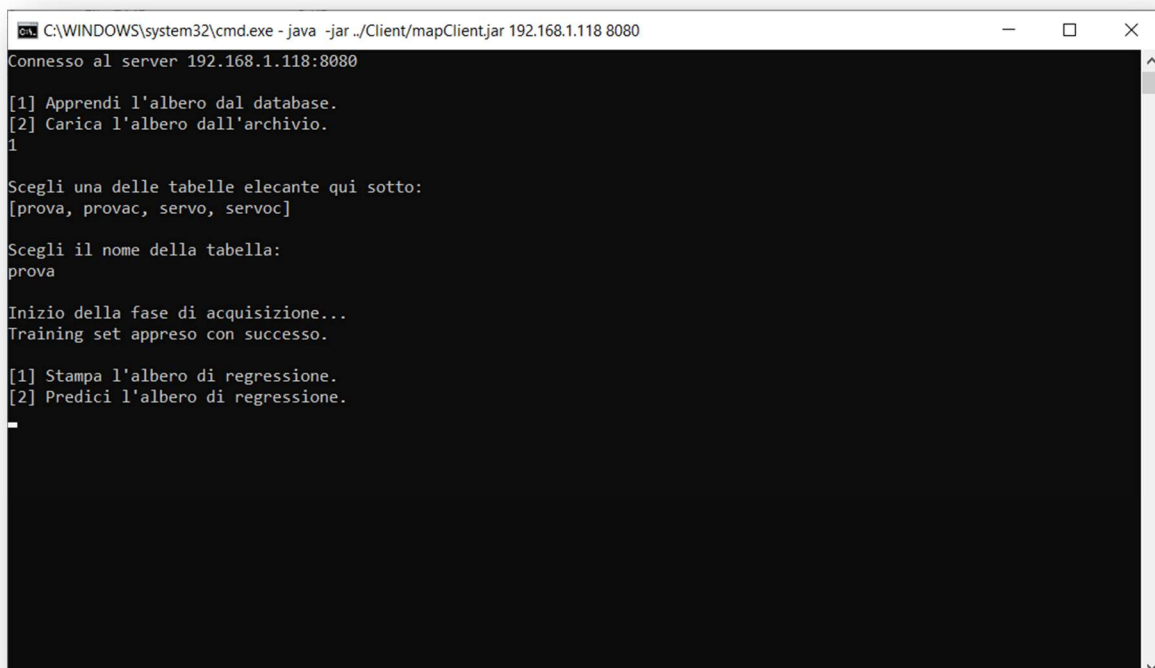
Scegli il nome del file:
prova

Scegli il nome del file:
servoc

Scegli il nome del file:

```

Entrambe le scelte descritte precedentemente, ci portano ad un'altra scelta: la stampa a video dell'albero in formato stringa, oppure l'avvio della fase di predizione.



```
C:\WINDOWS\system32\cmd.exe - java -jar ./Client/mapClient.jar 192.168.1.118 8080
Connesso al server 192.168.1.118:8080

[1] Apprendi l'albero dal database.
[2] Carica l'albero dall'archivio.
1

Scegli una delle tabelle elencate qui sotto:
[prova, provac, servo, servoc]

Scegli il nome della tabella:
prova

Inizio della fase di acquisizione...
Training set appreso con successo.

[1] Stampa l'albero di regressione.
[2] Predici l'albero di regressione.
1

```

Anche in questo caso siamo costretti a effettuare una scelta forzata tra 1 e 2. Se inseriamo un valore diverso tra quelli mostrati, non potremmo continuare.

```
C:\WINDOWS\system32\cmd.exe - java -jar ./Client/mapClient.jar 192.168.1.118 8080
Connesso al server 192.168.1.118:8080

[1] Apprendi l'albero dal database.
[2] Carica l'albero dall'archivio.
1

Scegli una delle tabelle elencate qui sotto:
[prova, provac, servo, servoc]

Scegli il nome della tabella:
prova

Inizio della fase di acquisizione...
Training set appreso con successo.

[1] Stampa l'albero di regressione.
[2] Predici l'albero di regressione.
3

[1] Stampa l'albero di regressione.
[2] Predici l'albero di regressione.
```

- Scelta 1 (es. con training set *prova*):

```
C:\WINDOWS\system32\cmd.exe - java -jar ./Client/mapClient.jar 192.168.1.118 8080
Scegli una delle tabelle elencate qui sotto:
[prova, provac, servo, servoc]

Scegli il nome della tabella:
prova

Inizio della fase di acquisizione...
Training set appreso con successo.

[1] Stampa l'albero di regressione.
[2] Predici l'albero di regressione.
1

Albero di regressione del training set 'prova'.
DISCRETE SPLIT: attribute=X Nodo: [Examples: 0-14] variance: 255.83333333333331 Split Variance: 0.625
  child 0, split value=A [Examples: 0-9]
  child 1, split value=B [Examples: 10-14]

DISCRETE SPLIT: attribute=Y Nodo: [Examples: 0-9] variance: 0.625 Split Variance: 0.0
  child 0, split value=A [Examples: 0-4]
  child 1, split value=B [Examples: 5-9]

LEAF: Class=1.0 Nodo: [Examples: 0-4] variance: 0.0
LEAF: Class=1.5 Nodo: [Examples: 5-9] variance: 0.0
LEAF: Class=10.0 Nodo: [Examples: 10-14] variance: 0.0

Vuoi continuare? (y/n)
```

- Scelta 2 (es. con training *prova*):

Anche in questo caso siamo costretti a fornire una scelta tra quelle elencate, altrimenti non potremmo continuare, come per gli esempi precedenti.

```
C:\WINDOWS\system32\cmd.exe - java -jar ./Client/mapClient.jar 192.168.1.118 8080
[2] Carica l'albero dall'archivio.
1

Scegli una delle tabelle elencate qui sotto:
[prova, provac, servo, servoc]

Scegli il nome della tabella:
prova

Inizio della fase di acquisizione...
Training set appreso con successo.

[1] Stampa l'albero di regressione.
[2] Predici l'albero di regressione.
2

Predizione dell'albero di regressione del training set 'prova'.
0:X=A
1:X=B

0
0:Y=A
1:Y=B

1
Valore di predizione: 1.5.

Vuoi continuare? (y/n)
```

In questo caso il valore di predizione deriva dall'esempio $X=A, Y=B \rightarrow 1,5$.

A questo punto il ciclo del programma giunge al termine. È possibile scegliere se continuare con una nuova iterazione o chiudere l'esecuzione del programma essendo sempre costretti a scegliere tra "y" o "n". Nel caso in cui si effettui la scelta "n", il client verrà disconnesso dal server.

```
C:\WINDOWS\system32\cmd.exe - java -jar ./Client/mapClient.jar 192.168.1.118 8080

Albero di regressione del training set 'prova'.
DISCRETE SPLIT: attribute=X Nodo: [Examples: 0-14] variance: 255.8333333333331 Split Variance: 0.625
  child 0, split value=A [Examples: 0-9]
  child 1, split value=B [Examples: 10-14]

DISCRETE SPLIT: attribute=Y Nodo: [Examples: 0-9] variance: 0.625 Split Variance: 0.0
  child 0, split value=A [Examples: 0-4]
  child 1, split value=B [Examples: 5-9]

LEAF: Class=1.0 Nodo: [Examples: 0-4] variance: 0.0
LEAF: Class=1.5 Nodo: [Examples: 5-9] variance: 0.0
LEAF: Class=10.0 Nodo: [Examples: 10-14] variance: 0.0

Vuoi continuare? (y/n)
n

Vuoi continuare? (y/n)
r

Vuoi continuare? (y/n)
y

[1] Apprendi l'albero dal database.
[2] Carica l'albero dall'archivio.
```

Client Android

Una volta avviato il server, avviare l'applicazione RTMiner sul proprio smartphone Android e in seguito verrà mostrata la prima schermata [figura 1]. È possibile aprire il menu [figura 2] in alto a destra per accedere alle informazioni circa l'autore del progetto, una breve descrizione del progetto o per configurare le impostazioni, tra cui le impostazioni di connessione e l'attivazione/disattivazione della *Dark Mode* [figura 3]. All'interno delle impostazioni di connessione, inserire l'IP e la porta del server.



FIGURA 1



FIGURA 2



FIGURA 3

Una volta inseriti IP e porta del server, premere su "OK" per confermare i dati inseriti. Solo a questo punto è possibile andare avanti a utilizzare l'applicazione. Premendo su "Carica da database" (o "Carica da file"), si aprirà una nuova schermata [figura 4] e l'app si conatterà al server fornendo, tramite uno *spinner collapse*, i nomi delle tabelle che popolano il database [figura 5]. Effettuata la scelta, è possibile scegliere se stampare l'albero del training set selezionato oppure avviare la fase di predizione. Se, invece, dalla schermata principale selezionassimo la voce "Carica da file", la schermata sarà uguale a quella precedentemente descritta, ma specializzata per il caricamento di alberi pre-serializzati su file [figura 6]. In tal caso verranno mostrati solo i file già serializzati.



FIGURA 4



FIGURA 5



FIGURA 6

Scegliamo, per esempio, il training set *prova*. Selezionando “Stampa l'albero”, si aprirà una schermata con la rappresentazione testuale dell'albero risultante dall'apprendimento del training set *prova* [figura 7]. Selezionando invece “Predici l'albero”, si aprirà un'altra schermata [figura 8] dove l'utente può scegliere i nodi da percorrere fino ad arrivare a un nodo foglia. In questo caso verrà mostrato il valore di predizione per quell'esempio [figura 9]. A questo punto, selezionando su “Chiudi”, torniamo alla schermata precedente, mentre selezionando su “Ripeti”, verrà avviata un'altra volta la fase di predizione per il training set scelto.



FIGURA 7



FIGURA 8



FIGURA 9

L'applicazione, inoltre, vanta della modalità *Night Mode* che permette all'utente di utilizzare un'interfaccia grafica con colori più scuri, seguendo le necessità del momento e permettendo a display AMOLED di consumare meno batteria. In [figura 10] e [figura 11] sono mostrati alcuni esempi. Inoltre, se la lingua dello smartphone è impostata su “Inglese”, l'applicazione avrà testi in inglese [figura 12].



FIGURA 10



FIGURA 11

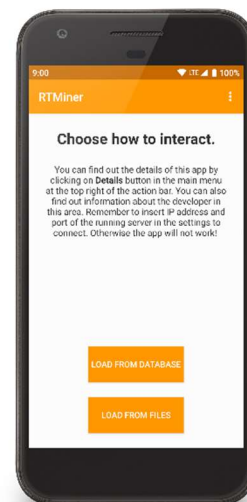


FIGURA 12

Toast mostrato nel caso in cui si inserisce un indirizzo IP con formato non valido:

- SOLUZIONE: Inserire un indirizzo IP con formato valido "x.x.x.x" con $x \in [0, 255]$.

ERRORE: Inserisci un indirizzo IP valido.

Toast mostrato nel caso in cui si inserisce una porta con formato non valido:

- SOLUZIONE: Inserire una porta compresa tra 1024 e 65535, estremi compresi.

ERRORE: Inserisci una porta valida.

Toast mostrato nel caso in cui si inserisce un indirizzo IP e una porta entrambi con formato non valido:

- SOLUZIONE: Inserire un indirizzo IP e una porta con formati validi precedentemente descritti.

ERRORE: Prima di poter procedere, inserisci un indirizzo IP e una porta validi.

Dialog mostrato nel caso in cui il server non risulta essere raggiungibile perché nessun server è in ascolto all'indirizzo IP e alla porta selezionati:

- SOLUZIONE: Avviare il server nel caso in cui esso non sia stato avviato, altrimenti inserire il corretto indirizzo IP e la corretta porta.

Server non raggiungibile

E' possibile che sia stato inserito un indirizzo IP o una porta errati.

CHIUDI

Dialog mostrato nel caso in cui nessun file è presente nell'archivio del server:

- SOLUZIONE: Prelevare uno o più dataset dal database per serializzare gli alberi su file.

Errore

Non ci sono file all'interno dell'archivio.

CHIUDI

Dialog mostrato nel caso in cui nessuna tabella è presente nel database:

- SOLUZIONE: Eseguire lo script setupDB.sql prima di procedere o aggiungere al database un training set custom.

Errore

Non ci sono tabelle all'interno del database.

CHIUDI

Nel caso in cui lo smartphone non è connesso a Internet, verrà mostrato un Toast che descrive l'errore:

- SOLUZIONE: Connettere lo smartphone a Internet.

ERRORE: Per poter funzionare, l'app ha bisogno di una connessione a Internet. Accendi il Wi-Fi o la rete dati per poter continuare.