**Regression Tree Miner - Estensione**

*Metodi avanzati di Programmazione - Caso di studio a.a. 2019-2020*

Progetto realizzato da: **Gianfranco Demarco** (mat. 708795) - gianfranco.demarco26@studenti.uniba.it

**Indice**

1. Introduzione
2. React e React Native
3. Guida d'installazione
4. Riepilogo funzionalità aggiuntive
5. Casi di test
6. Javadoc
7. Link Utili

**1. Introduzione**

La documentazione del progetto base è disponibile al percorso *[****progettoBase****]/mapServer/doc/Guida utente*

Come estensione si è scelto di realizzare un'applicazione Android, sviluppata sfruttando React Native, che sostituise l'applicativo client.

Inoltre sono presenti delle estensioni minori, sviluppate per migliorare l'esperienza utente o per ampliare la gamma di funzionalità dell'applicativo.

La comunicazione avviene tramite il meccanismo delle socket; si è scelto di mantenere questo meccanismo sebbene l'impiego di altre architetture (ad esempio l'esposizione di API Rest) avrebbe semplificato notevolmente la fase di sviluppo, con l’obiettivo di non snaturare il progetto.

**2. React e React Native**

**React**

**React** (noto anche come React.js o ReactJS) è una libreria JavaScript per la creazione di interfacce utente. È mantenuto da Facebook e da una comunità di singoli sviluppatori e aziende.

React può essere utilizzato come base nello sviluppo di applicazioni a pagina singola o mobile. Tuttavia, React si occupa solo del rendering dei dati sul DOM, pertanto la creazione di applicazioni React richiede generalmente l'uso di librerie aggiuntive per lo state management e il routing. Redux e React Router sono i rispettivi esempi di tali librerie.

(Wikipedia)

React nasce per velocizzare lo sviluppo di applicativi di Front-End e per massimizzare il riutilizzo di codice.

Concetti cardine di React sono:

* JSX (JavaScript eXtended): permette di trattare blocchi di codici HTML come variabili;
* componente: unità base dello sviluppo in React; rappresenta un componente riutilizzabile e personalizzabile

**React Native**

React Native è un framework per lo sviluppo di applicazioni mobile sviluppato da Facebook e basato sull'ecosistema di React.

Con esso condivide molti concetti ed approcci, sebbene non sia possibile utilizzare JSX e HTML con React Native.

Lo scopo di React Native è quello di velocizzare lo sviluppo di applicativi mobile, nonché fornire un metodo agli sviluppatori per far sì che la stessa codebase (o comunque con modifiche minime) possa essere utilizzata per compilare gli applicativi per diversi sistemi operativi (Android/iOS).

Con React Native, lo sviluppatore costruisce le interfacce e le logiche attraverso i componenti messi a disposizione della libreria (potendone comunque creare di nuovi partendo da essi).

Il compilatore si occupa poi di tradurre questi componenti standard nel codice nativo della piattaforma scelta.

Questo fornisce alle applicazioni sviluppate in React Native delle prestazioni del tutto equiparabili a quelle native.

**3. Guida di installazione**

**Server**

La guida d'installazione per l'applicativo server è analoga a quella della versione base del progetto.

Si riportano di seguito i dettagli.

*Pre-requisiti:*

- MySQL; JRE 8+ installati sulla macchina.

- esistenza di un'utenza "root" per il server MySQL. Se la password è diversa da "root", modificare il file "setup.bat" alla riga 2 sostituendo la password corretta.

1. Eseguire lo script *setup.bat*.

Esso si occupa di creare l'utenza per l'applicativo sul database; creare e popolare le tabelle per 2 dataset di test; eseguire l'applicativo server e l'applicativo client.

Per le esecuzioni successive, eseguire gli script *runServer.bat* e *runClient.bat* in questo ordine.

**Client**

Come anticipato, l'applicativo client per l'estensione è rappresentato da un'applicazione per sistema operativo mobile Android.

*Pre-requisiti:*

* Dispositivo Android versione Android 4.4W o superiore
* Connessione ad Internet sul dispositivo Android

E' necessario trasferire il file compilato (*.apk*) sul dispositivo (ad esempio tramite Cloud o connessione USB da un PC).

Per poter testare agevolmente tutte le funzionalità dell'applicativo, trasferire anche i file *boston*.*dat* e *boston*.*sql* che si trovano nella cartella *src/script*.

1. Eseguire l'apk ed installare l'applicativo.
2. Il sistema operativo Android potrebbe richiedere di abilitare un'impostazione per installare applicativi da “origini sconosciute”; confermare.
3. Sarà inoltre richiesto il permesso di accedere alla connessione internet; confermare anche in questo caso.

**Troubleshooting**

Perché la comunicazione tra i 2 dispositivi funzioni, è necessario che essi siano connessi alla medesima rete wifi/locale.

Inoltre, è necessario disattivare il firewall (o aggiungere una regola specifica) sul PC su cui viene eseguito l'applicativo server (o anche sul dispositivo mobile, se presente) per permettere le connessioni in ingresso/uscita.

**4. Riepilogo funzionalità aggiuntive**

* *Recupero della lista dei dataset memorizzati sul database*
* *Recupero della lista dei dataset di cui è stato effettuato il dump*
* *Upload di file .dat/.sql (max 64kb)*
* *Esecuzione dei file .sql caricati*
* *Creazione di script SQL a partire da file .dat con dataset in formato standard*
* *Dataset “Boston”*

1. **Casi di test**

**UC1**: Creazione della connessione client-server

***Pre-condizioni:***

1. ​E' stato eseguito almeno una volta con successo lo script setup.bat
2. E’ stata installata correttamente l’applicazione “Regression Tree Miner”

***Flusso di esecuzione:***

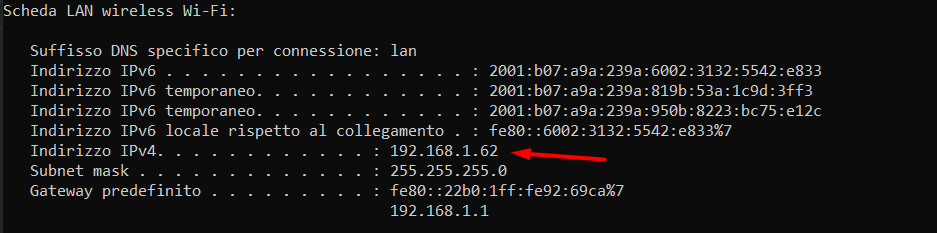
1. Eseguire il file runServer.bat
2. Avviare l’applicazione
3. Nella schermata di connessione, inserire l’indirizzo IP del server e la porta su cui è in ascolto



Per conoscere l’indirizzo ip del server eseguire sul PC dove è in esecuzione:

*cmd.exe > ipconfig (Windows)*

Il valore ricercato è indicato come “Indirizzo IPv4”



1. Premere “Connect”. Verrà mostrata una schermata di caricamento al termine della quale verrà visualizzata la schermata successiva.

L’avvenuta connessione è anche riscontrabile dall’output “Connection accepted!” visualizzato sull’applicativo server.

***Post-condizioni:***

1. Entrambi gli applicativi sono stati avviati correttamente.
2. L’applicazione Android visualizza la schermata successiva

***Scenari alternativi:***

1. Se durante il tentativo di connessione o comunque durante tutta la sessione di utilizzo dell’applicazione dovessero riscontrarsi dei problemi di comunicazione tra l’applicativo client e server, l’app mostrerà la seguente schermata con un collegamento alla prima view (schermata di connessione).

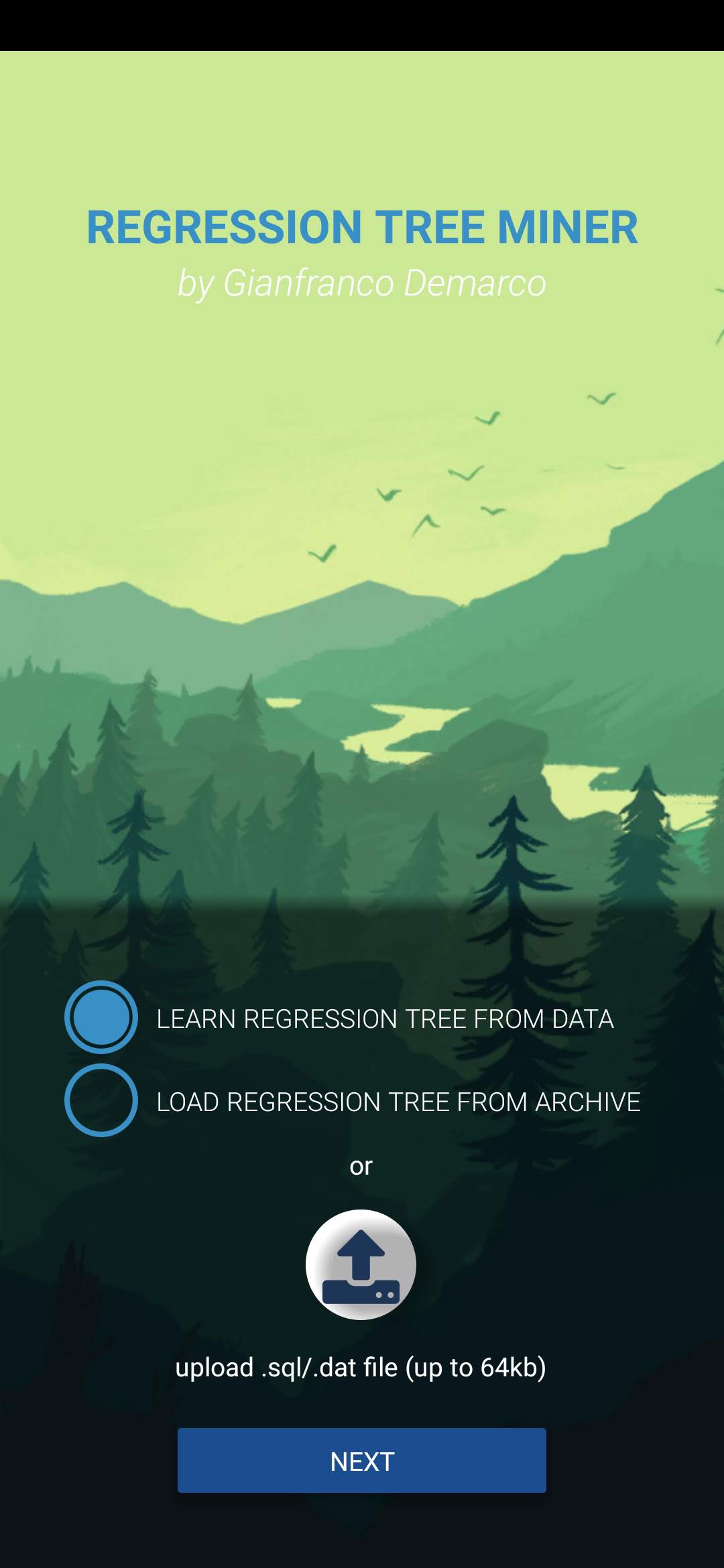


**UC2**: Regression Tree da database

***Pre-condizioni:***

* 1. ​E' stato eseguito lo UC1

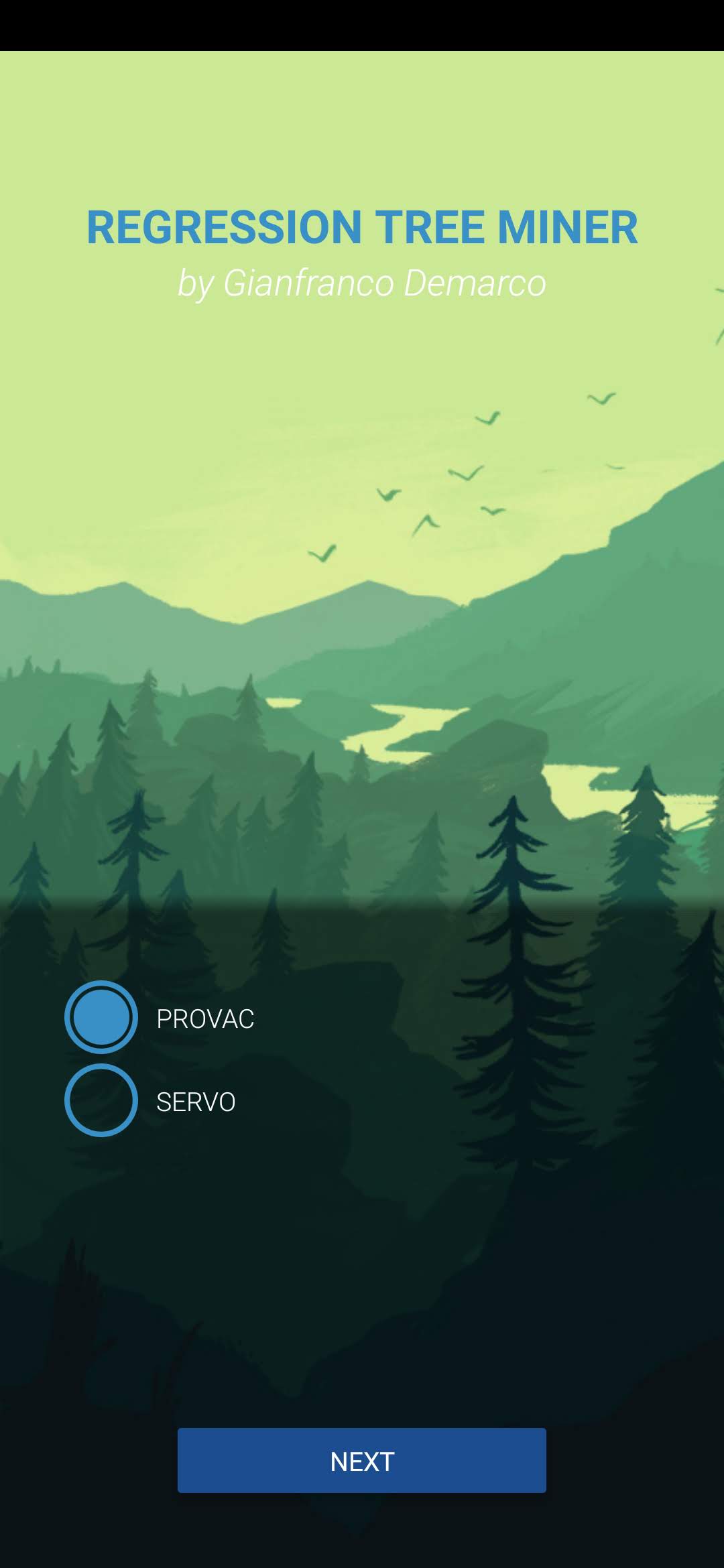
1. L’app visualizza la seguente schermata:



***Flusso di esecuzione:***

* 1. Selezionare “LEARN REGRESSION TREE FROM DATA” (selezionato di default)
  2. Premere il pulsante “NEXT”

1. L’app mostra un elenco con tutte le tabelle (datasets) memorizzate sul database



1. Selezionare una delle tabelle elencate e premere “NEXT”

***Post-condizioni:***

* 1. E’ stato creato nel path dove è stato avviato il server il file *./resources/dumps****/[nome\_dataset\_scelto]****.dmp*
  2. L’app visualizza la schermata del Regression Tree

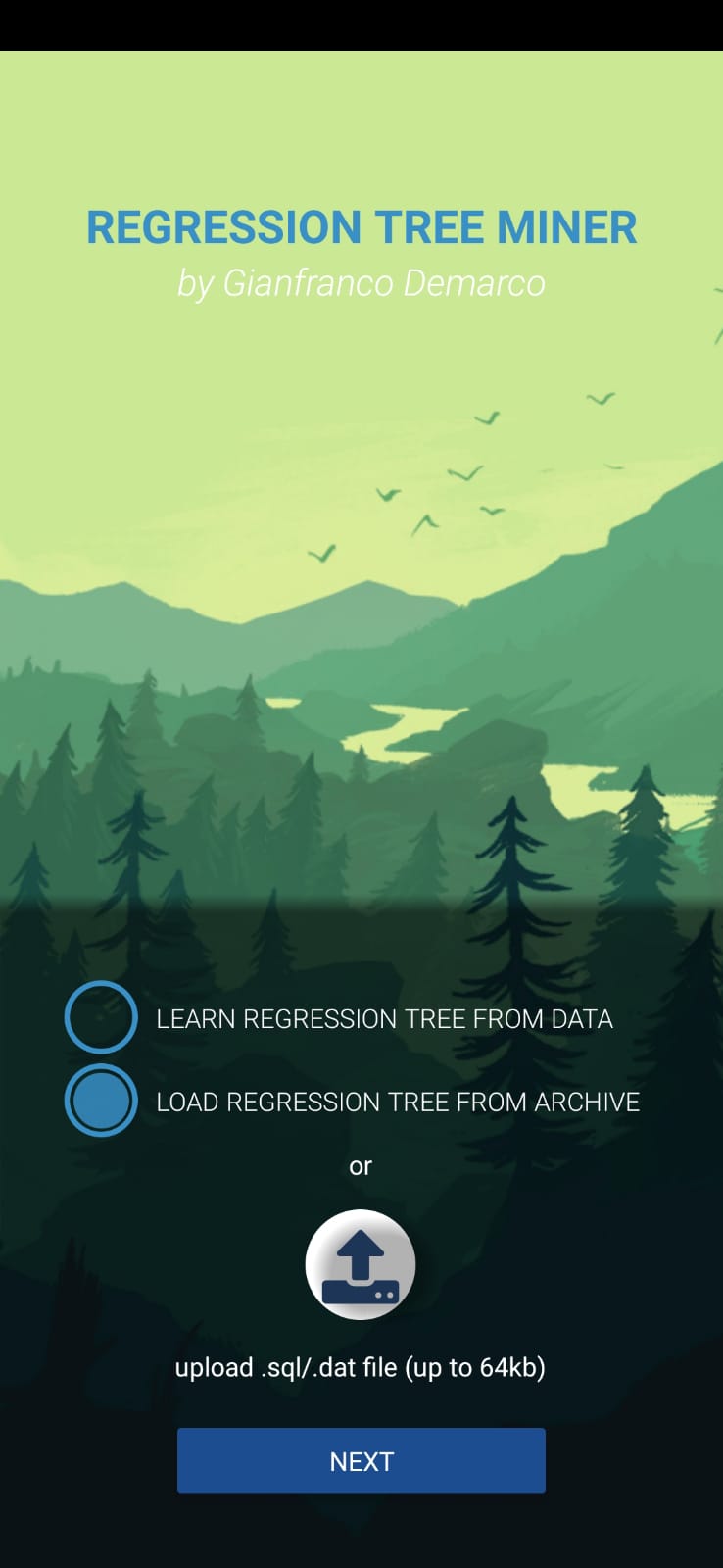
**UC3**: Regression Tree da dump

***Pre-condizioni:***

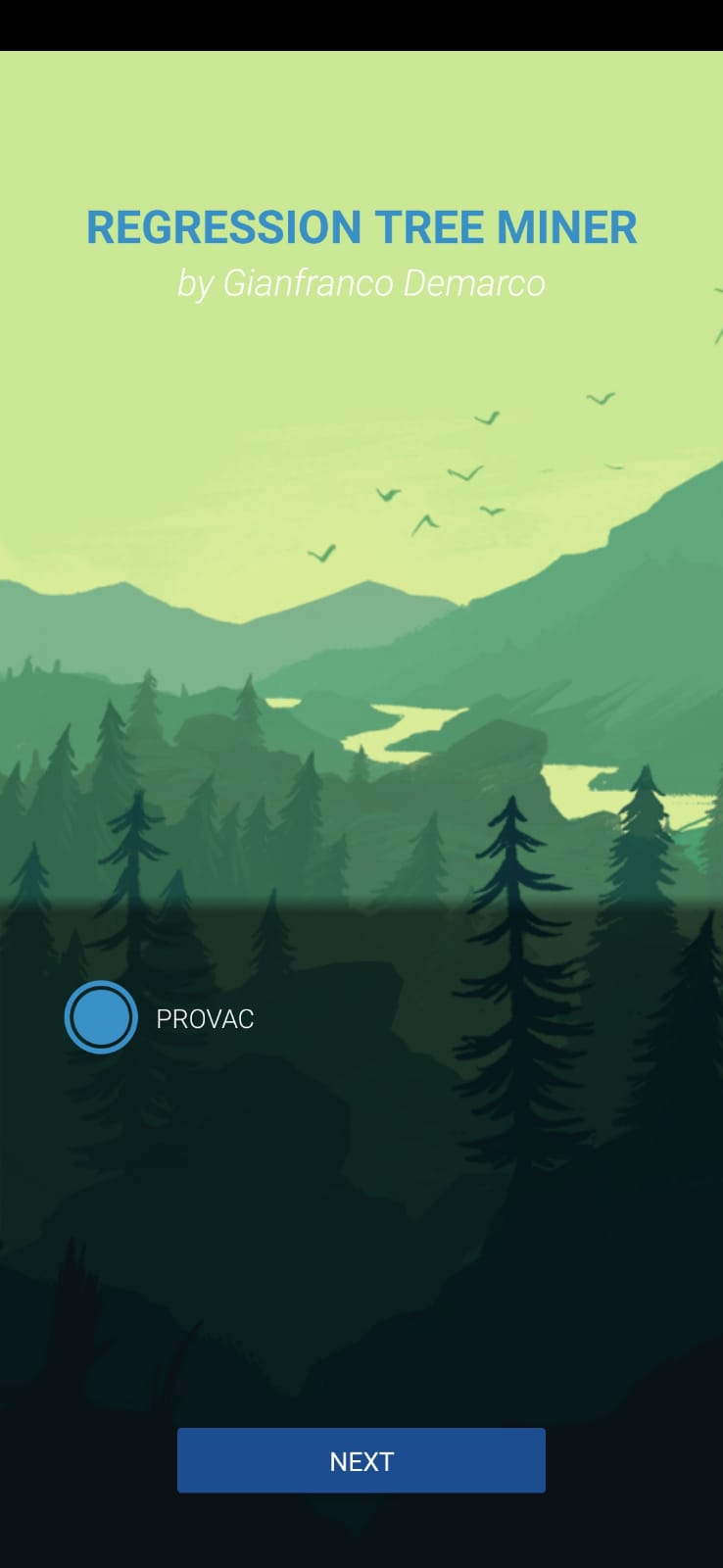
* 1. *Analoghe a UC2*
  2. *E’ stato eseguito almeno una volta UC2*

***Flusso di esecuzione:***

1. Selezionare “LEARN REGRESSION TREE FROM ARCHIVE”



1. Premere il pulsante “NEXT”
2. L’app mostra un elenco con tutti i dumps disponibili (datasets che sono stati selezionati almeno una volta in una precedente esecuzione dell’UC2)



1. Selezionare uno dei dump elencati e premere “NEXT”

***Post-condizioni:***

1. L’app visualizza la schermata del Regression Tree (*analoga a quella mostrata nello UC2*)

***Scenari alternativi:***

1. Se si prova ad eseguire UC3 senza aver prima mai eseguito UC2, viene visualizzata la seguente schermata di errore:



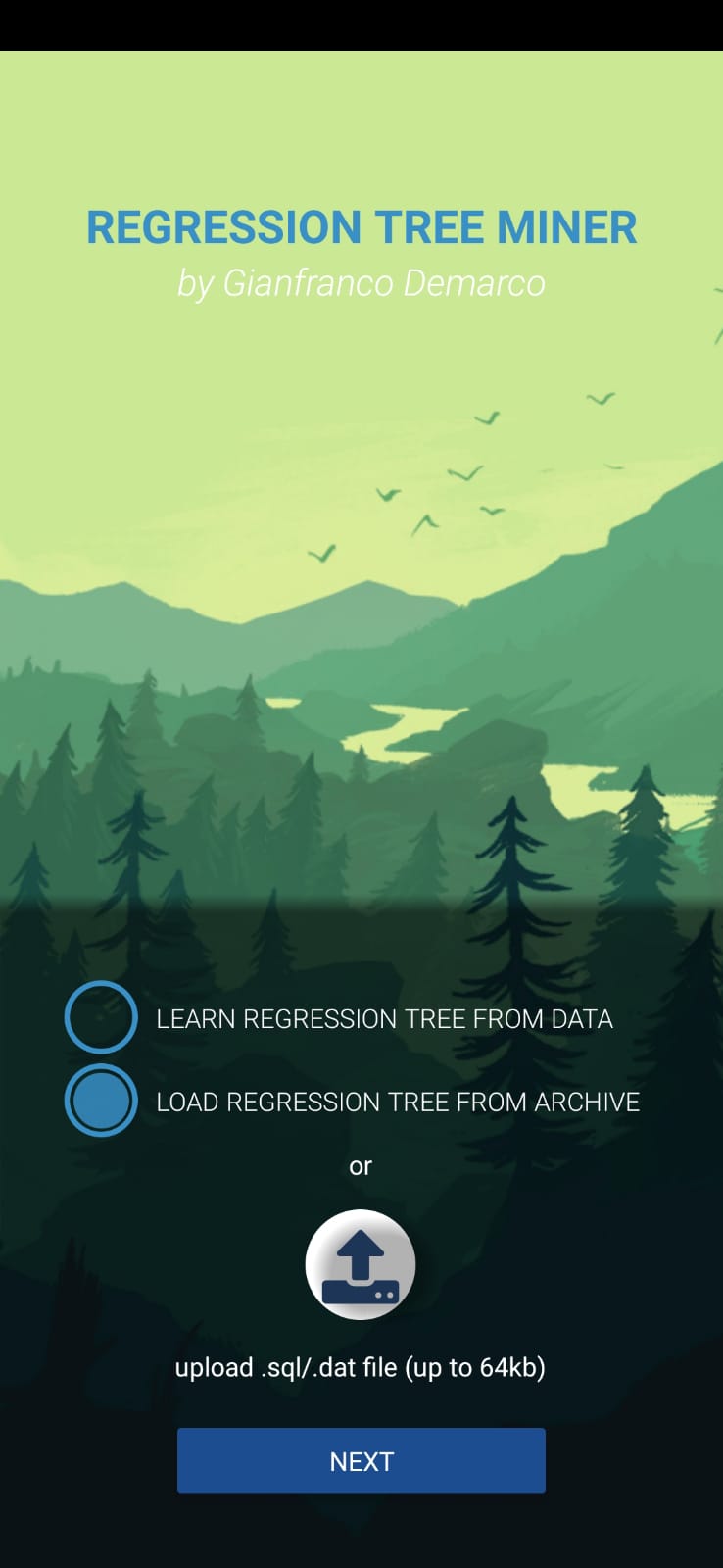
**UC4**: Caricamento di un nuovo dataset

***Pre-condizioni:***

*Analoghe a UC2*

***Flusso di esecuzione:***

* 1. Premere sull’icona di upload

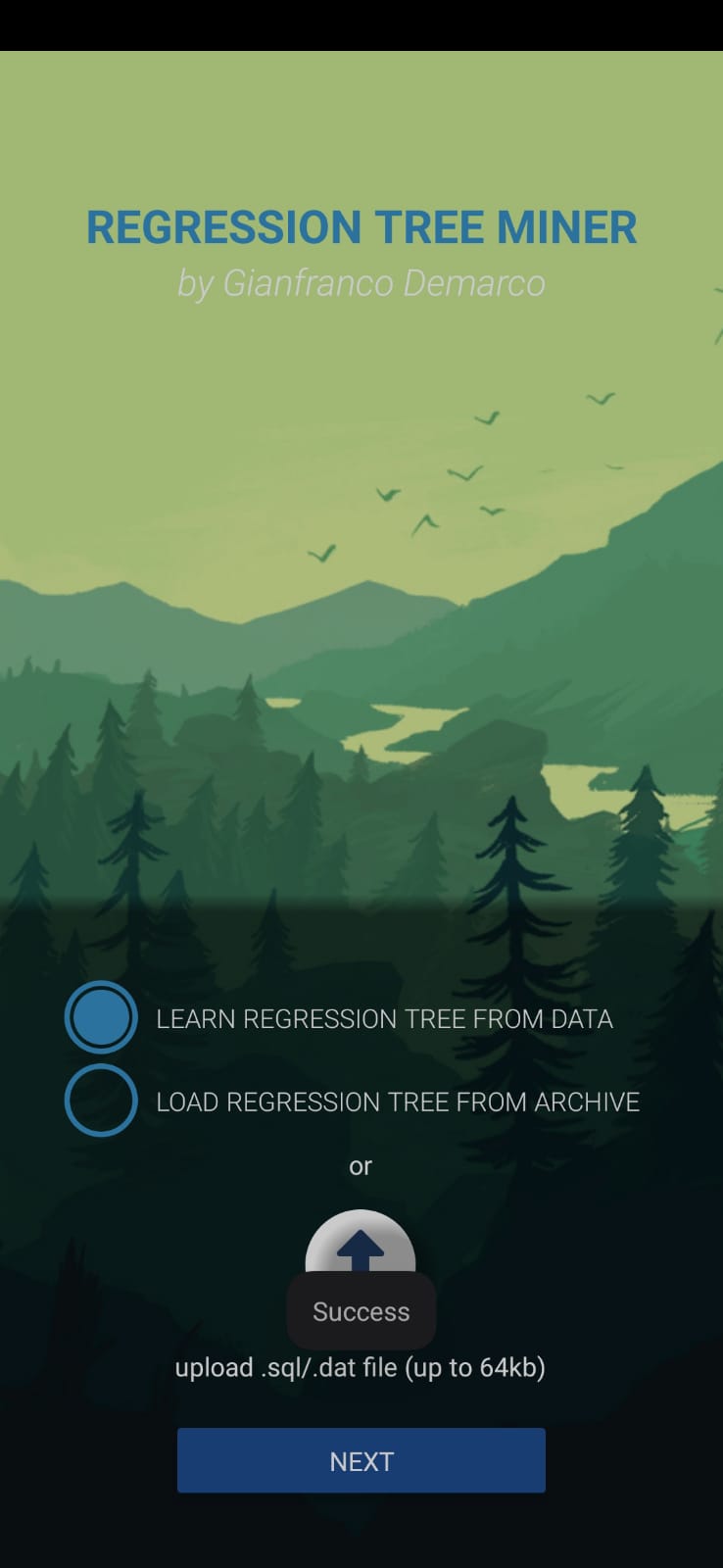


* 1. Viene aperto il gestore file del dispositivo. Selezionare un file con estensione .sql/.dat

ATTENZIONE! Non viene effettuata alcuna analisi semantica/sintattica dei file caricati.

***Post-condizioni:***

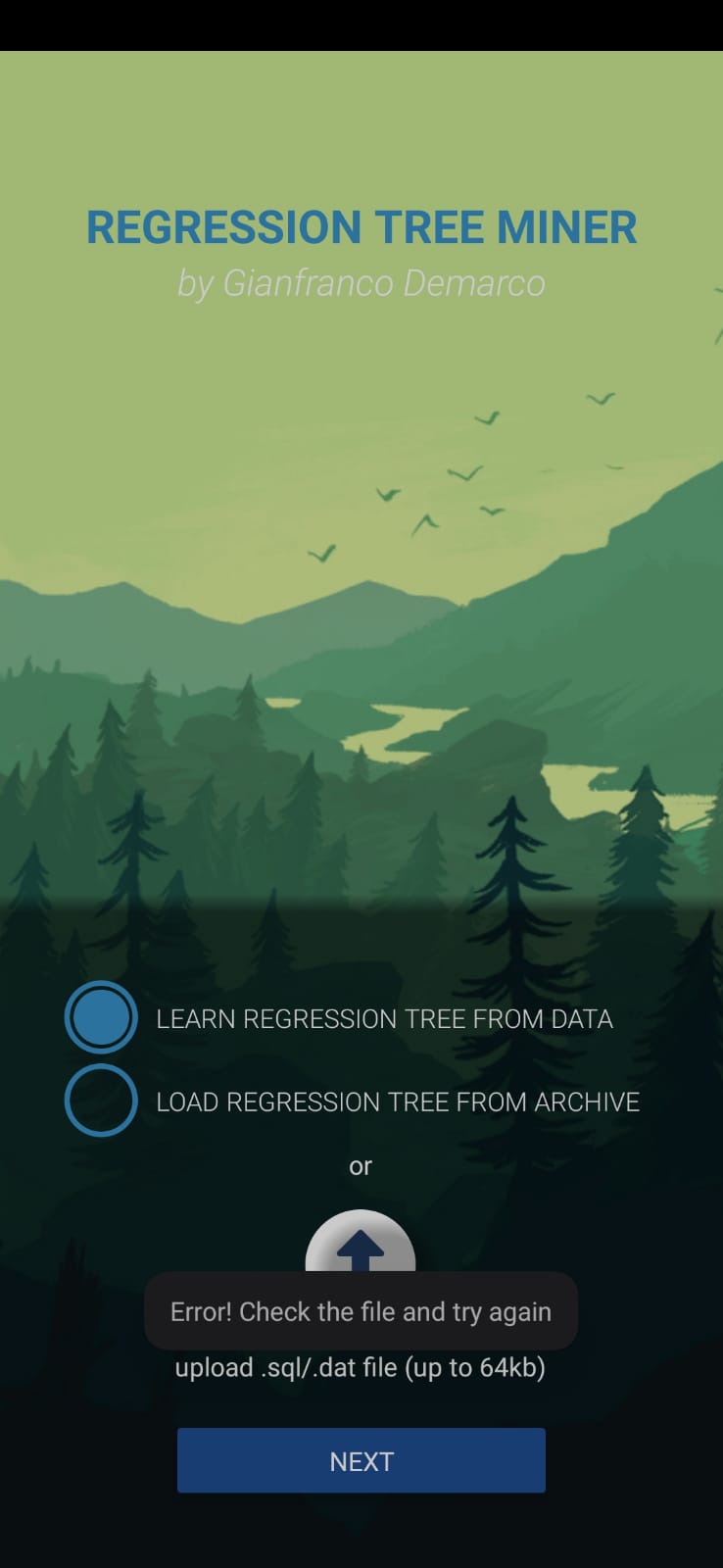
* 1. L’app ritorna sulla stessa schermata
  2. Viene visualizzato un messaggio di successo



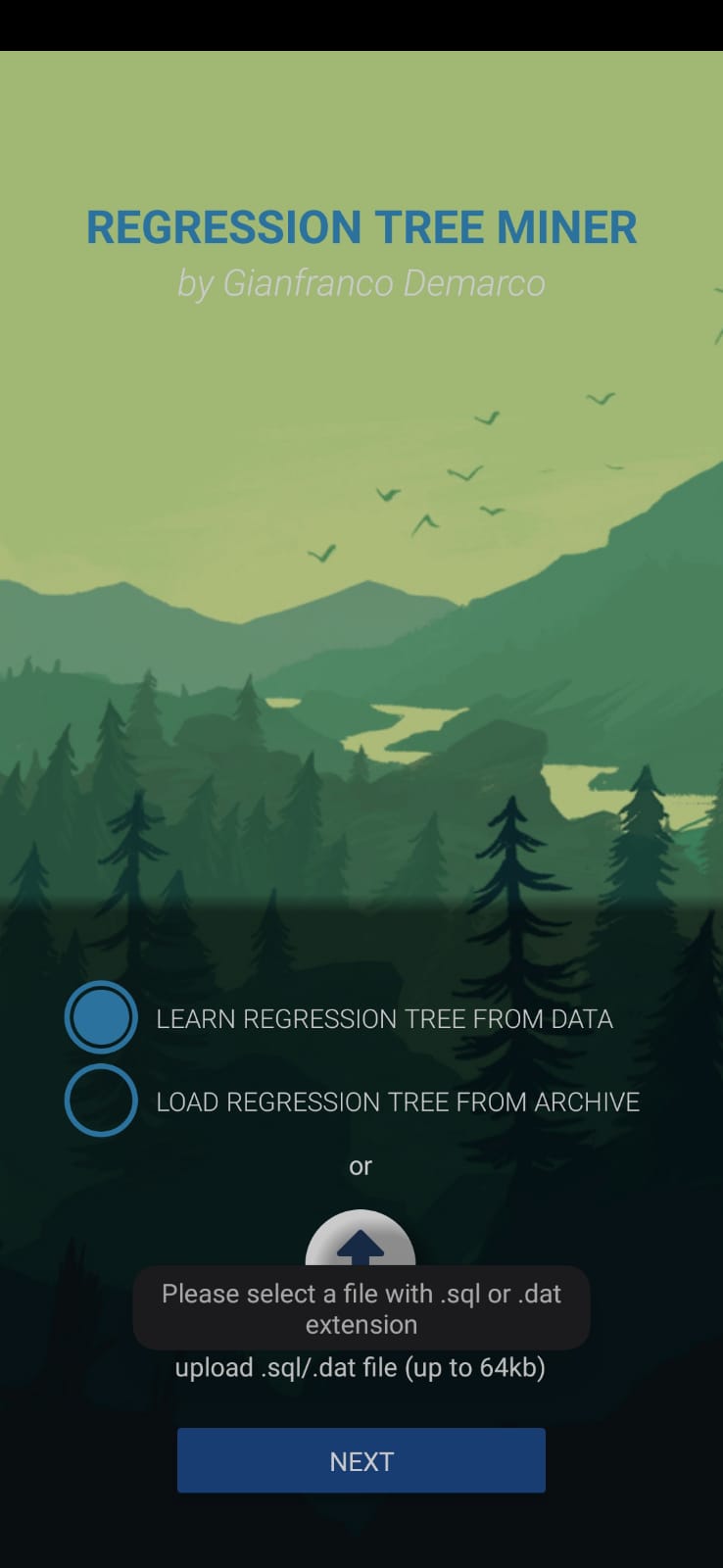
* 1. Il nuovo dataset sarà selezionabile durante le prossime esecuzioni dell’UC2

***Scenari alternativi:***

* 1. Al punto (2), a seconda del sistema operativo e delle impostazioni del dispositivo, potrebbe essere necessario fornire all’app l’autorizzazione di accedere al file system
  2. In caso di errore durante l’analisi del file, verrà notificato all’utente con un messaggio di errore



1. In caso il file selezionato dall’utente abbia un’estensione errata, verrà segnalato con un messaggio di errore



**UC5**: Visualizzazione regole e struttura albero

***Pre-condizioni:***

* 1. E’ stato eseguito uno tra UC2 e UC3
  2. E’ visualizzata la seguente schermata:

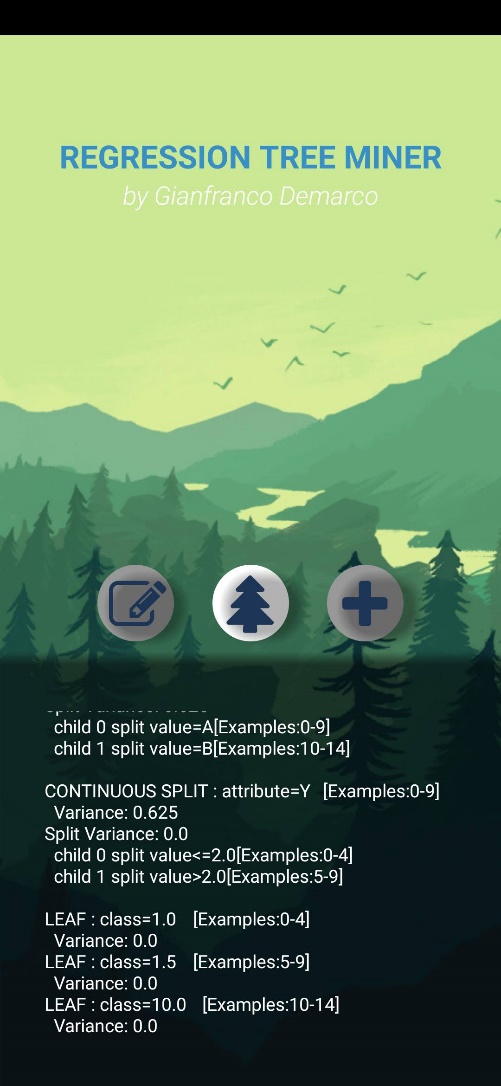


***Flusso di esecuzione:***

* 1. Premere l’icona centrale (albero)

***Post-condizioni:***

* 1. Quando viene visualizzata la schermata, nella sezione inferiore sono visualizzate le regole che rappresentano gli split dell’albero selezionato
  2. Dopo il punto (1), nella sezione inferiore è visualizzata la struttura dell’albero:



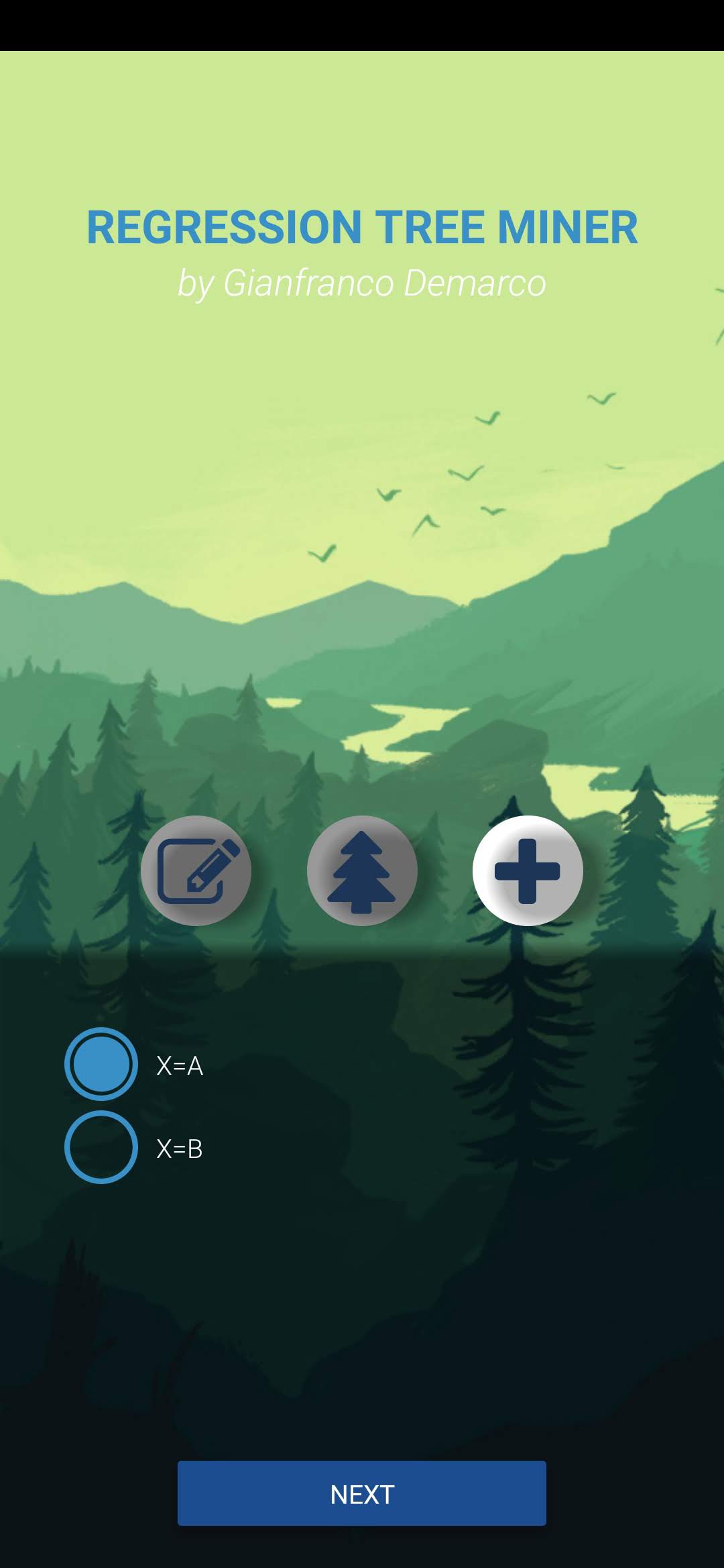
**UC6**: Predizione di un nuovo valore

***Pre-condizioni:*** *Analoghe a UC5*

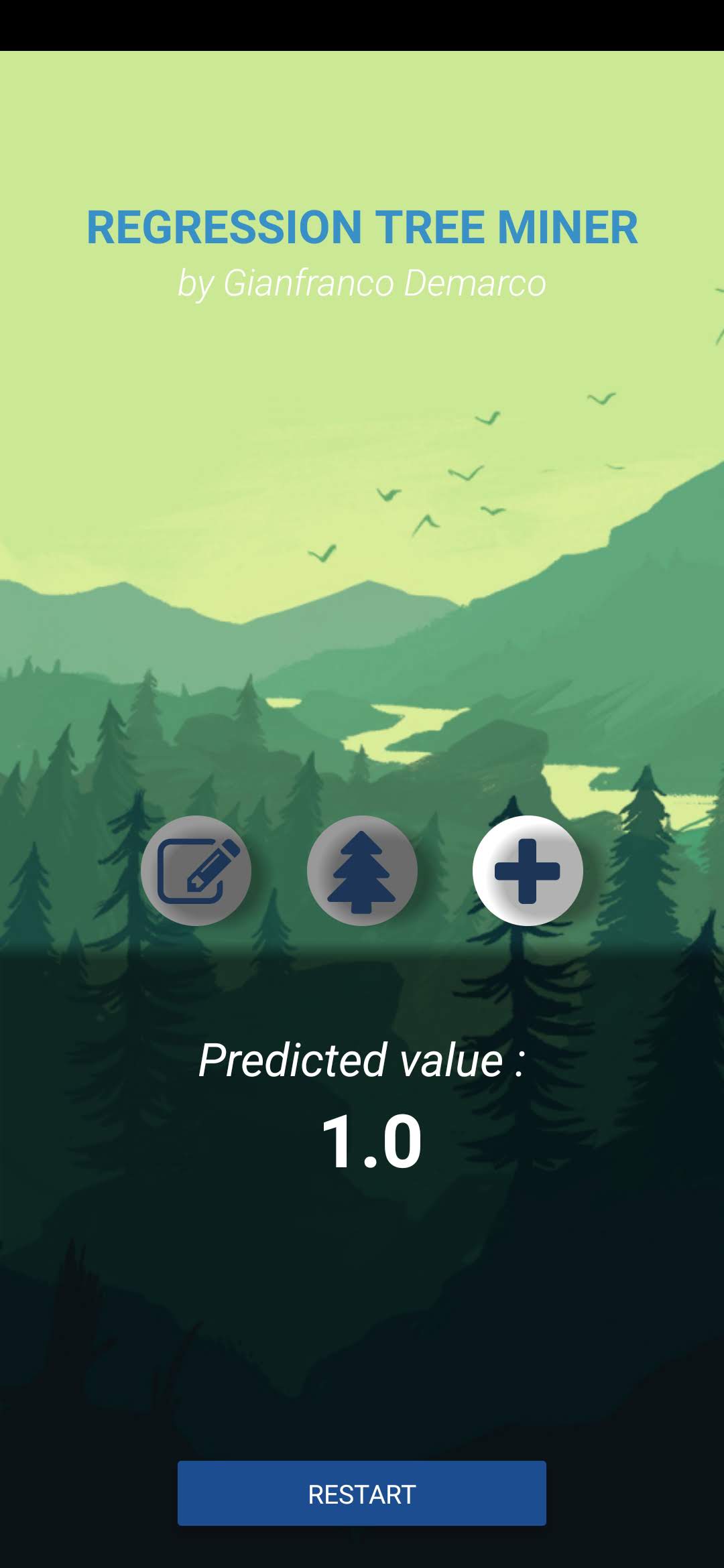
***Flusso di esecuzione:***

1. Premere l’icona a destra (+)

2. Vengono visualizzate le scelte corrispondenti agli split dell’albero scelto



* 1. Selezionare una delle opzioni e premere “NEXT”
  2. Ripetere 3 finché non si arriva ad un nodo foglia.
  3. Viene visualizzata la schermata con il valore predetto:



***Post-condizioni:***

* 1. Premendo “RESTART”, vengono visualizzate nuovamente le prime opzioni e si può procedere con una nuova predizione

**6. Javadoc**

Al codice sorgente dell’applicativo server è allegata la documentazione Javadoc.

Sono documentati i package Server, Data, Tree, Database.

**7. Link Utili**

Di seguito alcuni dei riferimenti utilizzati per lo sviluppo dell’app in ReactNative:

<https://it.wikipedia.org/wiki/React_(web_framework)>

<https://en.wikipedia.org/wiki/React_Native>

<https://www.npmjs.com/package/react-native-tcp>

<https://www.npmjs.com/package/react-native-simple-radio-button>

<https://material.io/resources/icons/?icon=expand_more&style=baseline>

<https://github.com/react-native-training/react-native-fonts>

<https://www.instamobile.io/android-development/generate-react-native-release-build-android/>

<https://medium.com/androiddevelopers/picking-your-compilesdkversion-minsdkversion-targetsdkversion-a098a0341ebd>

<https://medium.com/@appstud/add-a-splash-screen-to-a-react-native-app-810492e773f9>

[https://romannurik.github.io/AndroidAssetStudio/icons-launcher.html](https://romannurik.github.io/AndroidAssetStudio/icons-launcher.html7)

*Parte del codice è di proprietà della Prof.ssa Annalisa Appice, Università degli Studi di Bari -* [*http://www.di.uniba.it/~appice/*](http://www.di.uniba.it/~appice/)