Documentazione della Web App

Castelli Marco

Anno accademico 2021-2022

Contents

1 Introduzione

La web application sviluppata permette di generare, tramite il «LookBack Apriori Algorithm», un insieme di regole a partire da un'insieme di dati e dei setting definiti dall'utente. Dopodichè è possibile inserire una query per effettuare il match delle regole precedentemente generate e la query inserita dall'utente.

Questa web application che rappresenta il server servirà poi per la creazione di un software di più grandi dimensioni. Il software sarà formato da una mobile app che manderà delle richieste API al server, esso genererà le regole prendendo i dati dati un database e ritornerà un risultato all'utente.

1.1 LookBack Apriori Algorithm

L'algoritmo è un estensione del classico Algoritmo Apriori, infatti effettua un processo di data mining dei dati utilizzando anche i valori temporali dei dati generati.

I dati processati hanno i valori delle attività fisica e della qualità del riposo dell'utente negli ultimi giorni, in particolare viene riportato per quanto tempo un attività è stata fatta. L'algoritmo quindi analizza i dati dell'utente e restituisce come risultato una lista di regole.

Queste regole sono personalizzate per l'utente stesse e rappresentano che attività dovrebbero essere svolte per dormire bene.

2 Struttura delle cartelle

Il software è formato da un file «app.py» che rappresenta le routes delle varie pagine html e in cui vengono richiamate le varie funzioni per gestire i dati ottenuti dal form.

Tutte le funzioni utilizzate sono state organizzate in un file «utilities.py» mentre l'algoritmo di data mining, ossia «LookBack Apriori Algorithm», è definito in un file a parte «LookBack Apriori Algorithm.py»

Le pagine html si trovano all'interno della cartella «templates» mentre nella cartella «static» è presente il codice css.

All'interno della cartella «setting» sono presenti tutte le regole che sono state generate durante l'utilizzo del software, il nome di ogni file specifica quali sono stati i setting per generare le regole, in particolare si specifca quale dataset è stato utilizzato e i parametri del setting: sleep value, temporal window, min support, min confidence.

Nella cartella «Data» sono presenti delle sottocartelle che contengono i dati di alcuni utenti.

3 Descrizione file principali

3.1 «app.py»

Prima delle «routes» vengono dichiarate tutte le librerie che verrano utilizzate e dopodichè vengono definite delle variabili globali in modo da essere disponibili all'interno delle routes.

La prima route è «settingRules» in cui viene semplicemente richiamata la pagina «settingRules.html» in questa pagina html si compila il form con le informazioni per il setting: dataset , sleep value, temporal window, min support, min confidence.

Dopo aver compilato il form i dati vengono passatti alla route «rulesGeneration», qui vengono effettuati vari controlli sulla correttezza dei dati, qualora non lo fossero l'utente viene reindirizzatto alla pagina «settingRules.html». Se i dati sono corretti si verifica se è già presente un file con quei setting in modo da non dover rigenerare le regole, se non lo fosse si controlla anche se è possibile generare le regole, a partire da un file già presente ma con diversi setting.

Se non è possibile nessuno dei due casi allora bisogna necessariamente richiamare la funzione di generazione delle regole «LookBack Apriori Algorithm.py» che restituisce una lista contenti tutte le regole generate dopodiche esse vengono salvate sul file.

Alla fine si viene reindirizzati alla pagina «rulesGeneration.html» che mostra le prime venti regole generate, ordinate per support, completezza e dimensione, vengono inoltre mostrati i setting utilizzati e il numero totale delle regole generate.

Nella navbar è presente la voce «Match Query» una volta premuta si viene reindirizzati alla pagina «matchQueryForm.html» in cui l'utente ha la possibilià di inserire una query secondo un determinato formato.

Una volta inserita la query e premuto l'apposito bottone si viene reindirizzatti alla route «matchingQuery» in cui viene controllata la correttezza della query inserita e successivamente viene richiamata la funzione di matching che mostra il risultato all'interno della pagina «matchingQuery.html».

L'utente ha sempre a disposizione la navbar in cui può tornare a visualizzare le regole generate oppure a creare nuove regole con nuovi setting.

3.2 «LookBack Apriori Algorithm.py»

Il codice dell'algoritmo è stato messo sotto forma di funzione in modo tale che i parametri del setting gli vengano passati per generare le regole secondo onerosi calcoli. Viene richiamata la funzione di ordinamento sulle regole ed infine vengono restituite tutte le regole generate.

4 Descrizione funzioni presenti nel file utilities

4.1 dataRange

Richiamata nella route «settingRules» verifica che i dati inseriti nel form appartengano ad un determinato range di valori.

Restituisce un messaggio di errore e un se i setting sono errati o meno.

4.2 isQueryValid

Richiamata nella route «matchingQuery» verifica che la query inserita sia ben formata ossia contenga come attività HA,MA,LA,ZL,R con annesso un valore temporale (es:

t2) e che quel valore temporale non superi la temporal window che l'utente ha inserito nei setting iniziali.

Restituisce un booleano e un messaggio.

4.3 addCriteria

Richiamata all'interno della funzione rulesSorting, altera la struttura di ogni elemento della lista di regole aggiungendo un campo completeness e size. Utilità: facilita la successiva operazione di ordinamento. Restituisce la lista di regole modificata.

4.4 rulesSorting

Richiamata all'interno del file «LookBack Apriori Algorithm.py» restituisce una nuova lista in cui le regole sono ordinate per support, completeness, size.

4.5 splitFile

Richiamata in due diverse route quando abbiamo necessità di leggere le regole dal file, ad esempio inseriamo un setting che è già presente nel file oppure un setting implicito in un altro file.

Scorre il file e memorizza tutte le informazioni in esso presenti, in particolare i valori dei setting, la sola lista di regole (camp 'rule' della struttura) e l'intera struttura delle regole.

Restituisce le informazioni precedenti

4.6 alreadySetting

Richiamata nella route «rulesGeneration».

Verifica se nella cartella «Setting» è già presente un file con il nome passato come parametro.

4.7 rulesImplicitGeneration

Richiamata nella route «rulesGeneration».

Verifica se esiste un file in setting dal quale è possibile prendere un sottoinsieme.

Utilità: inseriamo un setting che può essere stato generato all'interno di un'altro setting.

4.8 saveSetting

Richiamata nella route «rulesGeneration».

Consente di creare un file con il nome del setting in cui sono contenute tutte le regole generate e il setting.

4.9 takeRuleImplicit

Richiamata nella route «rulesGeneration» e «matchingQuery». Usata quando troviamo un file che contiene le regole specificate in un nuovo setting, per ottenere un sottoinsieme delle regole.

Consente di ottenere solamente le regole con support e confidence specificate.

Resituisce le sole regole e l'intera struttura delle regole.

4.10 findMatching

Funzione principale per il matching delle regole, in essa vengono fatte delle operazioni preliminari sulla query e dopodichè vengono chiamate le funzioni per il matching una alla volta.

Restituisce se trova la regola altrimenti null.

4.11 ExactMatch

Primo criterio per il matching: scorre tutta la lista di regole e se trova una regole che è identica alla query dell'utente la restituisce.

4.12 Match

Secondo criterio per il matching: scorre tutta la lista di regole e se trova una regole che contenuta nella query dell'utente la restituisce.

4.13 PartialMatch

Terzo criterio per il matching: scorre tutta la lista di regole e se trova una regole che ha un match parziale con la query dell'utente la restituisce.

4.14 SimilarMatch

Quarto criterio per il matching: scorre tutta la lista di regole e se trova una regole che ha un match simile con la query dell'utente la restituisce.

4.15 isSimilar

Funzione utilizzata all'interno delle funzioni PartialMatch e SimilarMatch, consente di verificare se un insieme di attività in una dato tempo (t2) sono simili rispetto ad un altro insieme nello stesso tempo.

Resituisce una lista con i valori 'Similar', 'Partial' e 'False', che serviranno per controllare il tipo di match.

4.16 splitActivity

Richiamata in tutte le funzioni di match, consente a partire da una stringa (la regola) di ottenere una lista formata dalle attività in t1,t2,t3.

4.17 ruleAntecedent

Consente di ottenere l'antecedente della regola cioè tutto ciò prima di '->'. Viene utilizzata per il matching delle regole.