Well being and Machine Learning

Simone Naldini

mathema

29-05-2019

Well being

"*Health is a state of complete positive physical, mental, and social well-being and not merely the absence of disease or infirmity*".

(Preamble to the Constitution of the World Health Organization (WHO), 1946)

Introduzione

Campo di applicazione**: benessere personale**

Scopo: utilizzo della tecnologia come elemento aggregante dal punto di vista sociale, fornendo alle persone che vivono sole un supporto al miglioramento dello stato di salute, partendo dalla sfera emotiva (solitudine e problematiche conseguenti)

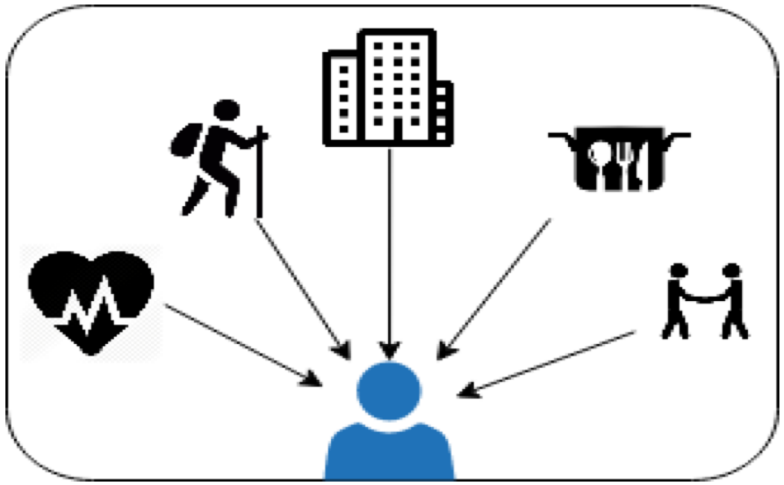
Utenti: tutte le persone che vivono sole, con particolare riguardo alle persone anziane, ancora autosufficienti

Approccio: elaborazione di suggerimenti personalizzati, per incentivare le persone a impegnarsi in attività sociali e migliorare le relazioni interpersonali

Introduzione

Campo di applicazione: **benessere personale**

Ogni utente è associato ad un **profilo** che lo identifica in base alle sue necessità e caratteristiche personali
La *profilazione* degli utenti fa riferimento a cinque principali aree di interesse:



- Salute
- Mobilità
- Vita domestica
- Relazioni interpersonali
- Relazioni di gruppo

Contesto di utilizzo

Una possibile soluzione

La soluzione ipotizzata è quella di **proporre** all'utente consigli ottenuti **ragionando** (anche con tecniche di AI), sulla base del suo profilo, su quanto si è **imparato** sul suo comportamento nel tempo e **osservando** quello che sta facendo in quell'istante.

> *Fornire suggerimenti solo sulla base di quella che 'dovrebbe' essere la best-practice, senza contare la condizione specifica dell'utente, potrebbe non portare benefici, o in alcuni casi, avere effetti deleteri*

Burzagli, Laura & Naldini, Simone. (2018). Elderly People and Loneliness: An Innovative Approach. DOI: 10.1007/978-3-319-94274-2_55

Condizioni iniziali

- il **profilo** dell'utente e tutte le caratteristiche che lo compongono,

- lo **stato** in cui si trova l'utente

Sulla base di queste rilevazioni, è possibile effettuare una prima **selezione e proposta** dei suggerimenti, non sensibile ai cambiamenti (analisi **statica)**, in quanto non considera le **decisioni** (preferenze) da parte dell'utente.

Da qui la necessità di inserire un processo ricorsivo.

Soluzioni applicative

Contesto di sviluppo: Framework **NodeJS**

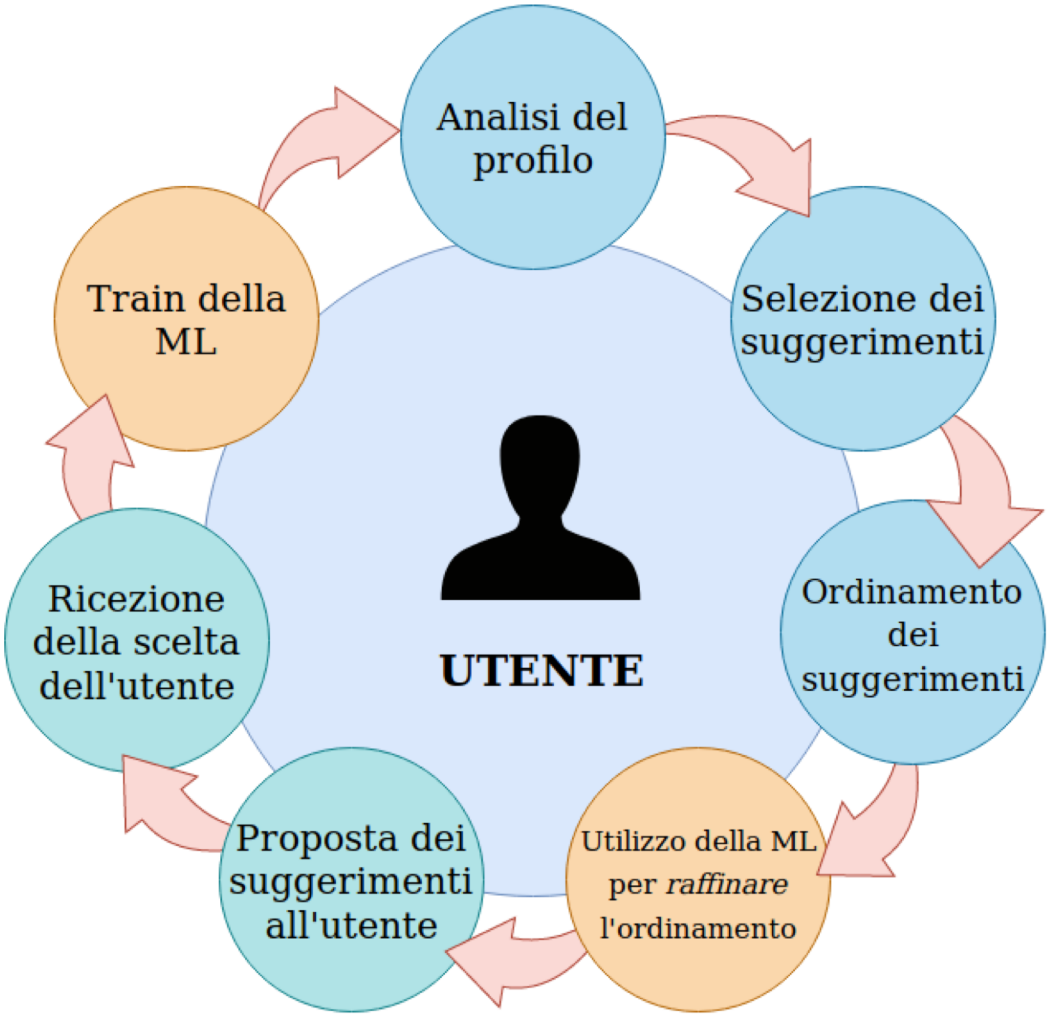
interfaccia grafica con out HTML,
compatibilità con altre piattaforme di sviluppo (R),
.....

Valutazioni delle soluzioni di ML già disponibili sui repository **npm**

- **limdu** (https://github.com/erelsgl/limdu)
  - algoritmi disponibili: Reti neurali, Winnow, MultiLabelClassification, Bayes, SVM, Threshold

- **machine_learning** (https://github.com/junku901/machine_learning)
  - algoritmi disponibili: regressione logistica, SVN, KNN, DecisionTree

Modello


Analisi del profilo
Selezione dei suggerimenti
Ordinamento dei suggerimenti
Utilizzo della ML per *raffinare* l'ordinamento
Proposta dei suggerimenti all'utente
Ricezione della scelta dell'utente
Train della ML
UTENTE

Realizzazione

Installazione del modulo

Essendo moduli presenti sul repository **npm** per installarli è sufficiente il comando *npn install*.
Nel caso di **limdu**

```
> npm install limdu
```

E per utilizzarlo basta eseguire un *include* nel codice Node

```
var limdu = require('limdu');
```

Realizzazione

Inizializzazione della ML

Il setup della ML consiste nella realizzazione di una nuova istanza dell'algoritmo, che diviene immediatamente richiamabile all'interno del codice.

Nel nostro caso, è sufficiente:

- decidere il tipo di ML (*bayes*, *winnow*, *treeshold, neural network),*
- l'algoritmo che vogliamo utilizzare

```
var nome_ml = new limdu.classifiers.NeuralNetwork();
```

Realizzazione

Training della ML

La fase di addestramento, in questo caso, consiste nel passare i dati in formato JSON alla ML:

```
nome_ml.trainBatch([
    {input:
        {
            'label_1' : valore,
            'label_2' : valore,
            'label_3' : valore,
        },
        output: output_rilevato
    }
]);
```

Realizzazione

Test (classificazione)

Con il comando *classify* è possibile sottoporre a test la ML, *chiedendole* di classificare un nuovo **sample**, sulla base dei dati che ha già acquisito.

```
nome_ml.classify({
    'label_1':valore,
    'label_2':valore,
    'label_3':valore
})
```

Esempio pratico 1

Contesto

Si vuole **valutare** il grado di accettazione da parte di un utente, **di** uno specifico suggerimento, come ad esempio "*fare una passeggiata nel parco*"

Si progetta una ML capace di indicare se e quanto un suggerimento potrebbe essere adatto ad un determinato profilo d'utente, in un determinato momento, sulla base delle esperienze pregresse.

Esempio pratico 1

Contesto

I fattori presi in considerazione sono:

- Il livello di **mobilità** del soggetto
- la **preferenza** del soggetto rispetto alle attività all'aria aperta
- la condizione operativa dell'utente (quanto il soggetto è **impegnato** al momento)

Come risposta ci aspettiamo una **probabilità**

Esempio pratico 2

Contesto

Si vuole cercare di sapere se sottoporre un suggerimento ad un utente sia o meno un'opzione valida, basandosi sulle scelte precedentemente effettuate.
Anche in questo caso il suggerimento sarà quello della "passeggiata nel parco".

Si vuole quindi avere uno strumento, che dato i parametri dello specifico caso, sia in gradi di classificare il suggerimento come *adatto* o *non adatto* a quel determinato utente.

Esempio pratico 2

Contesto

I fattori presi in considerazione sono:

- Il livello di **mobilità** del soggetto
- lo stato di **solitudine** del soggetto
- le condizioni **meteo** al momento di sottoporre il suggerimento
- la **condizione** operativa dell'utente (quanto il soggetto è **impegnato** al momento)

In output ci aspettiamo una risposta 'booleana': **vero o falso**

Esempio pratico 3

Contesto

In input si ha solamente un testo in *linguaggio naturale*, senza ulteriori dati a corollario.

In output ci aspettiamo una classificazione rispetto ad uno dei suggerimenti che già fanno parte del sistema

Well being and Machine Learning

GRAZIE PER L'ATTENZIONE!

simone.naldini@mathema.com