



UNIVERSITÀ
DEGLI STUDI
FIRENZE

**Scuola di
Ingegneria**

Studente: Federico Magini

Data: 08/04/2021

Corso in Intelligenza Artificiale

Professore: Paolo Frasconi

RICONOSCIMENTO DI ATTIVITA' UMANE

1. INTRODUZIONE

Il riconoscimento delle attività umane consiste nell'individuare lo stato di un utente in un determinato momento: all'utente vengono attaccati al corpo alcuni sensori che producono dei segnali di monitoraggio continui. Questi segnali vengono poi elaborati e vengono identificate delle caratteristiche. In base a queste caratteristiche viene effettuata la classificazione dell'attività umana.

Nel seguente esercizio i segnali vengono classificati in 6 differenti attività:

Standing, Laying, Sitting, Walking, Walking Upstairs, Walking Downstairs, in accordo con il lavoro svolto da [Anguita et al. 2012](#).

2. FORMALIZZAZIONE DEL PROBLEMA

Viene fornito un [Dataset](#), proveniente dal campionamento dei segnali, i quali provengono, a loro volta, dai sensori.

L'obiettivo è utilizzare l'apprendimento supervisionato per classificare le attività umane, tramite l'algoritmo di Random Forest e costruire la matrice di confusione.

3. REALIZZAZIONE ELABORATO

La prima cosa da fare è importare le librerie Pandas e Sklearn, come descritto nel file README.md.

Dopo aver importato le librerie è necessario caricare il Dataset all'interno del progetto. Il Dataset deve trovarsi nella stessa cartella dove è contenuto il codice Python.

Il Dataset è partizionato randomicamente in due sottoinsiemi:

- Training Data
- Test Data.

La parte del Training Data costituisce circa il 70% del Dataset e consiste in:

1. X_{train} = Dati provenienti dal campionamento dei segnali provenienti dai sensori. Questa parte serve per addestrare il classificatore.
2. Y_{train} = Classificazioni corrispondenti ai dati in X_{train} .

La parte del Test Data costituisce circa il 30% del Dataset e consiste in:

3. X_{test} = Dati provenienti dal campionamento dei segnali provenienti dai sensori. Questa parte serve per effettuare i test al fine di valutare la capacità del classificatore di riconoscere l'attività corretta.
4. Y_{test} = Classificazioni corrispondenti ai dati in X_{test} .

I dati all'interno del Dataset (in X) sono separati mediante un numero variabile di spazi bianchi. Questo problema è stato risolto mediante l'utilizzo del separatore `sep='s+'` il quale distingue due valori consecutivi a prescindere dal numero di spazi bianchi che li separano.

Per caricare i dati dal file di testo, inoltre, è stata usata la funzione `read_csv()` contenuta all'interno della libreria Pandas.

A questo punto si dichiara il Classificatore Random Forest. L'implementazione di Random Forest è quella contenuta all'interno della libreria Sklearn.

Si addestra il classificatore tramite la funzione `fit` sui dati X_{train} e Y_{train} e poi si fa predire al classificatore i restanti dati contenuti in X_{test} .

Infine si effettua l'analisi dei risultati tramite la funzione `report()`, la quale costruisce la matrice di confusione e stampa alcune informazioni sull'accuratezza della predizione del classificatore.

4.CONCLUSIONI

Si riporta la stampa che risulta dall'esecuzione del codice:

```

Contingency matrix:
[[483  5  8  0  0  0]
 [ 33 431  7  0  0  0]
 [ 23  47 350  0  0  0]
 [  0  0  0 446 45  0]
 [  0  0  0  44 488  0]
 [  0  0  0  0  0 537]]

Classification report:
              precision    recall  f1-score   support

activity_labels      0.90      0.97      0.93       496

      accuracy              0.93       2947
    macro avg       0.93      0.92      0.93       2947
   weighted avg       0.93      0.93      0.93       2947

```

Figura 1: Stampa della Matrice di Confusione e informazioni sulla precisione della previsione

La Matrice di Confusione ha come righe e colonne le 6 attività nel seguente ordine: Walking, Walking Upstairs, Walking Downstairs, Standing, Sitting, Laying.

Si può notare che l'attività *Laying* viene sempre indovinata dal classificatore, 537 volte su 537.

In generale, si può notare che sia il valore di Precision che di Recall sono molto alti, e producono infatti un f1-score molto alto, dove f1-score è la media armonica tra Precision e Recall. Perché sia alta, bisogna che siano alti sia Precision che Recall.

I risultati ottenuti sono quindi compatibili con quelli ottenuti da [Anguita et al. 2012](#)

Random Forest è quindi un buon classificatore.