



UNIVERSITÀ
DI TRENTO

Dipartimento di Ingegneria Industriale
Laboratorio didattico di misure meccaniche e
termiche

Anno Accademico 2021/2022

3 luglio 2022

Luca Rigoni

matricola 209523

luca.rigoni-1@studenti.unitn.it

1. Introduzione

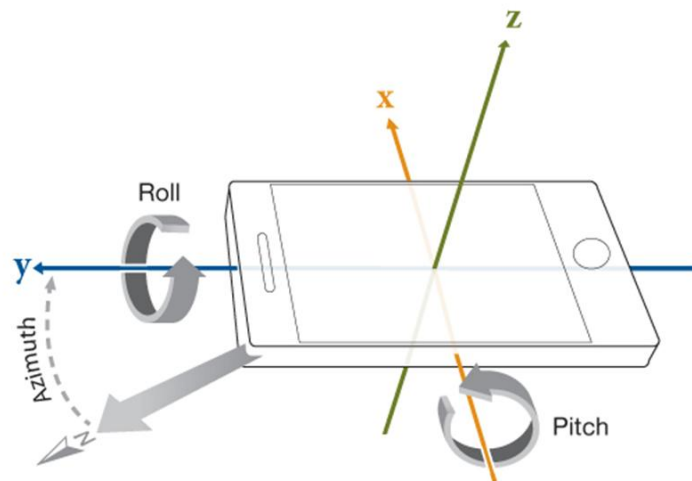
Il machine learning è un ramo dell'intelligenza artificiale che utilizza metodi statistici per migliorare le performance di un algoritmo nell'identificare pattern nei dati.

Con questa tipologia di AI è possibile allenare il calcolatore con molti dati (le quali attività sono note), in modo che - in un secondo momento - la macchina si predispona l'abilità di apprendere informazioni da nuovi dati in maniera autonoma, senza istruzioni esplicite.

Questo è possibile perché l'apprendimento automatico è strettamente legato al riconoscimento di pattern, sulla quale l'algoritmo sarà poi in grado di sviluppare delle predizioni in maniera autonoma.

2. Finalità dello studio

L'attività didattica ha lo scopo di utilizzare il comune smartphone per svolgere delle acquisizioni relative al segnale di accelerazione durante lo svolgimento di specifiche azioni.



Il database fornito contiene dati di accelerazione lungo le tre componenti x, y e z relativi a 30 soggetti mentre svolgono sei attività:

- 1 = Cammino (walk)
- 2 = Salita scale (walking upstairs)
- 3 = Discesa scale (walking downstairs)
- 4 = Sedersi (sitting)
- 5 = Fermo in piedi (standing)
- 6 = Fermo sdraiato (laying)

Sarà necessario analizzare i dati nel dominio del tempo e della frequenza per evidenziare le caratteristiche del segnale. Tali features dovranno successivamente essere utilizzate per addestrare il classificatore in grado di distinguere le diverse azioni.

2.1 Caricamento e visualizzazione dei dati

Il primo passaggio consiste nel caricare i dati all'interno della zona di lavoro su Matlab. Se si osservano i dati è possibile evincere che più precisamente il database è una struttura (struct) composta da 30 righe (una per soggetto) e 2 colonne, nelle quali si trovano l'indice che identifica il numero di attività svolta (da 1 a 6) e una matrice di tre colonne, i cui valori corrispondono alla misura dell'accelerazione nelle tre componenti.

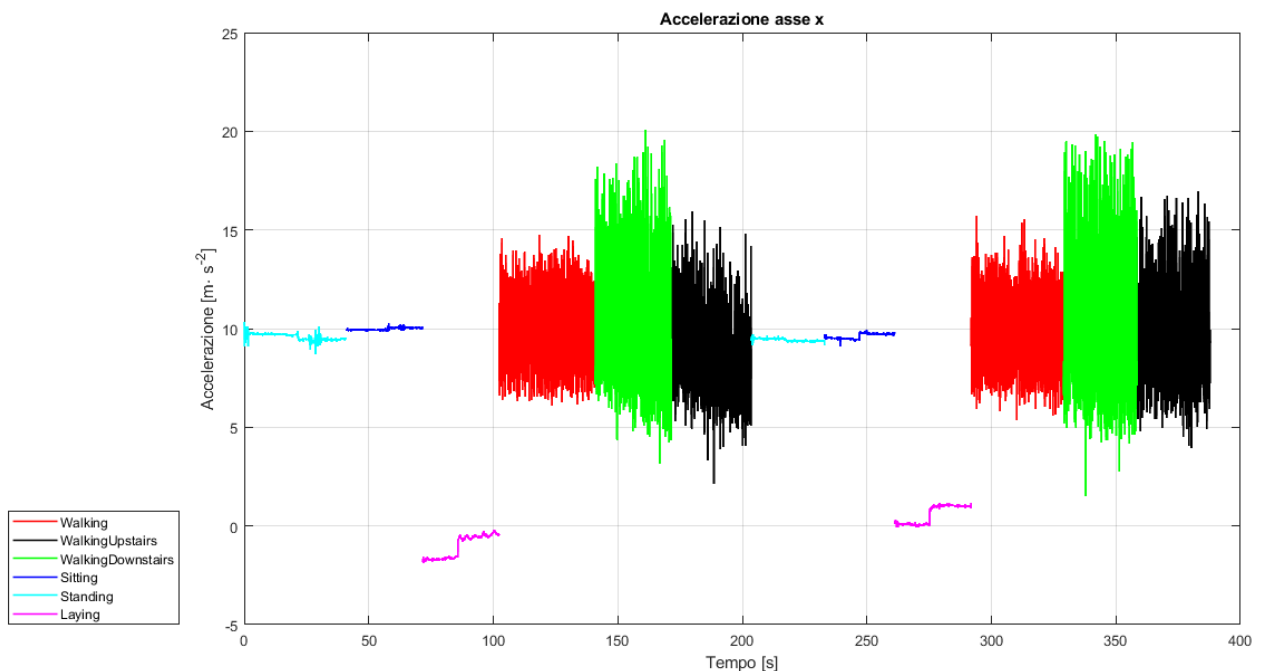
A questo punto è possibile realizzare una coppia di specifiche funzioni: "Get_acceleration()" e "Visualizzazione_accelerazioni()".

La prima prende in input la struttura dei dati iniziali, la componente di accelerazione che si vuole visualizzare, il singolo soggetto preso in esame e la frequenza di campionamento ed ha come output i vettori accelerazione e quello che identifica l'attività (caratterizzati dalla stessa dimensione) e l'asse dei tempi. Di seguito viene riportato il codice Matlab:

```
function [acc,act_label,time_axis] = Get_acceleration(subject, direction, subjects, fs)

    if direction == 'x'
        dir = 1;
    elseif direction == 'y'
        dir = 2;
    elseif direction == 'z'
        dir = 3;
    end
    acc = subjects(subject).totalacc(:,dir);
    act_label = subjects(subject).actid;
    time_axis = (1:length(subjects(subject).totalacc(:,dir)))/fs;
end
```

La seconda funzione prende come ingresso gli output della prima in modo da visualizzare i dati estratti. Si riposta di seguito un esempio della visualizzazione dei dati grezzi:



È evidente che i soggetti hanno svolto diverse attività nell'arco di tempo registrato.

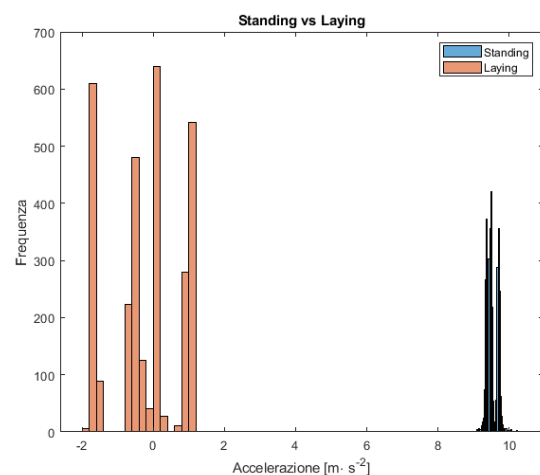
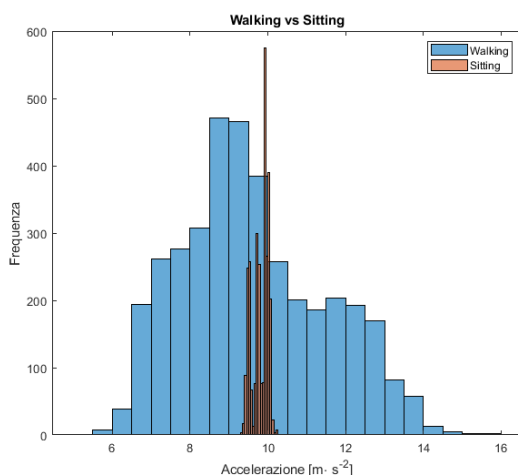
Questa prima osservazione è buona per avere un quadro generale. È però un'analisi limitativa, infatti non permette all'utente di osservare le diverse accelerazioni (i dati sono troppo condensati) e non permette al calcolatore di riconoscere immediatamente il tipo di attività svolta.

2.2 Istogrammi

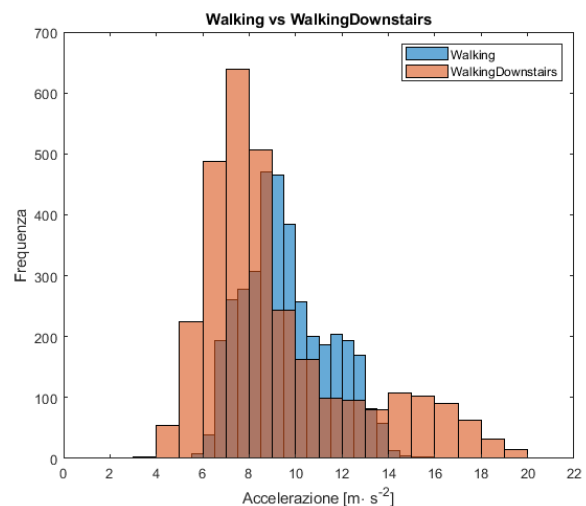
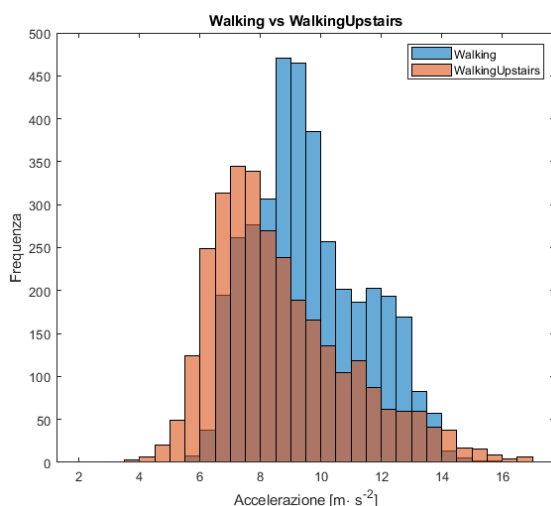
È per questo motivo che si decide di analizzare i dati su una tipologia diversa di grafico. Gli istogrammi permettono di identificare visivamente per ogni attività svolta di ciascun soggetto la frequenza con la quale una determinata accelerazione si presenta. Questa analisi permette di fare un importante passo in avanti, perché si può identificare l'accelerazione che caratterizza una determinata attività.

Si noti come confrontando a due a due gli istogrammi delle diverse attività, in alcuni casi i grafici sono ben distinti. Lo si può osservare, ad esempio, confrontando l'attività della camminata con quella dello stare seduti.

Oppure un caso più interessante è quello del confronto della attività di stare in piedi e stare distesi, perché il sensore di accelerazione viene ruotato, quindi la accelerazione di gravità viene rappresentata in un'altra componente rispetto a quella scelta per la visualizzazione di esempio.



Si noti però che questa analisi non permette di determinare con esattezza il tipo di attività svolta, è presente ancora una buona probabilità di errore. Di seguito vengono riportati alcuni esempi:



È evidente che nei casi rappresentati per le attività comparate il sensore di accelerazione ha registrato dati e frequenze non distanti tra loro, infatti i corrispondenti istogrammi sono largamente sovrapposti.

In questo caso si parla di compatibilità delle variabili, risulta quindi necessaria un'analisi più approfondita.

2.3 Analisi nel dominio della frequenza

Per un'analisi più approfondita si vuole osservare lo spettro delle accelerazioni corrispondenti alle diverse azioni. Si utilizza il potente strumento della trasformata di Fourier.

La trasformata di Fourier è un particolare caso della trasformata di Laplace, la quale ha un significato fisico ben definito ed ha come ipotesi iniziale che il segnale abbia transitori finiti. Questo tipo di trasformata viene applicato per le misure che variano nel tempo, poiché la dinamica è un problema semplificabile in termini stazionari tempo varianti.

La trasformata di Fourier è una formula matematica che mette in relazione il segnale campionato nel tempo con lo stesso segnale campionato in frequenza. Nell'elaborazione del segnale, la trasformata di Fourier è in grado di rivelare importanti caratteristiche di un segnale, ossia le sue componenti in frequenza.

In Matlab si utilizza la funzione "fft()", la quale computa la trasformata di Fourier discreta usando "l'algoritmo veloce della trasformata" (fast Fourier transform). Prende in ingresso un vettore (il segnale prodotto dall'insieme campionato di accelerazioni) e ne restituisce la sua trasformata.

Si presta particolare attenzione all'asse delle frequenze, il quale si normalizza alla frequenza di Nyquist, perché ci si aspetta una componente continua della trasformata, la quale verrà poi esclusa con un filtro.

Si riporta la funzione Matlab che calcola la trasformata del segnale:

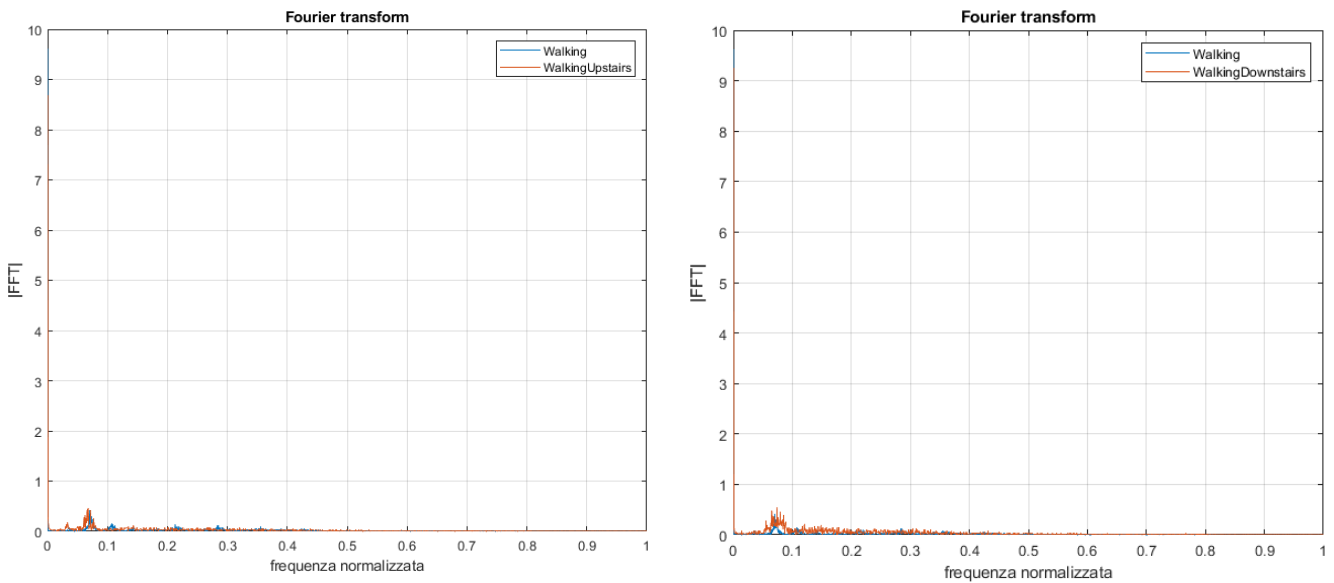
```
function [ff,s_FFT] = Plot_FFT(acc, fs)

    % numero di campioni
    N = length(acc);

    s_FFT = fft(acc)/length(acc);
    %frequenza di Nyquist = sampling frequency / 2
    fN = fs/2;
    Tc = 1/fs;
    ff = 0:1/(Tc*N*fN):1;    %asse delle frequenze
    %ff = (0:N-1)*fN/N;

    plot(ff,abs(s_FFT(1:floor(N/2)+1)));
    grid on
    title('Fourier transform');
    ylabel('|FFT|');    xlabel('frequenza normalizzata');
end
```

Si confrontano quindi i grafici delle trasformate:



Come ci si aspettava è presente un picco visibile a frequenza nulla, il quale non permette l'osservazione dei grafici per frequenze più alte dove sono presenti i picchi di interesse. Come previsto, si vuole applicare un filtro.

2.4 Analisi nel dominio del tempo

Il passaggio successivo consiste nella realizzazione di un filtro passa alto, in modo da eliminare la componente continua del segnale (a frequenze basse) e visualizzare solamente la componente dinamica (a frequenze più alte).

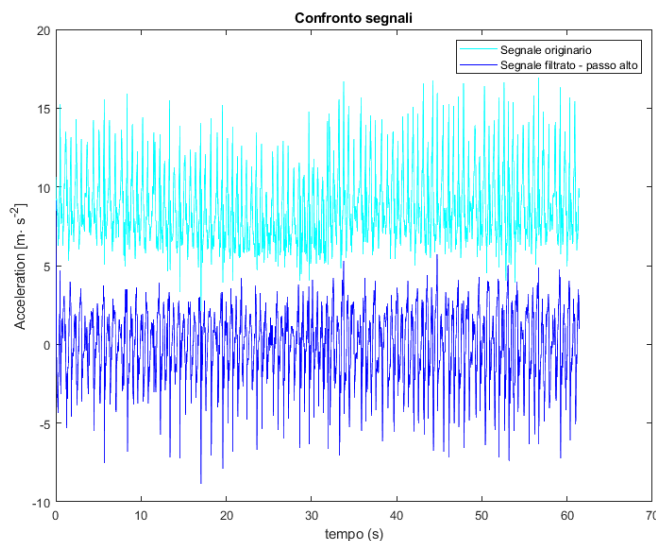
Il filtro si applica quindi ai segnali di accelerazione divisi per attività. In Matlab si genera la funzione "Filtro":

```
function [B, A] = Filtro(acceleration_w, fs, order, tau)
%FILTRO --> Filtro Passa Alto
% per eliminare componente continua

[B, A] = butter(order, tau, 'high');
%asse dei tempi
t = (1/fs)*(0:length(acceleration_w)-1);
%filtro il segnale secondo i coefficienti trovati e lo plotto
acc_filt = filter(B,A,acceleration_w);
figure;
plot(t, acceleration_w, 'c')
hold on
plot(t, acc_filt, 'b')
title('Confronto segnali')
xlabel('tempo (s)');
ylabel('Acceleration [m\cdot s^{-2}]')
legend('Segnale originario', 'Segnale filtrato - passo alto')
hold off
end
```

La funzione ha come input la accelerazione per una sola attività scelta, la frequenza di campionamento, e due costanti. Questi ultimi due parametri sono l'ordine del filtro e la costante di taglio, e sono stati scelti in modo che andassero bene per tutti i segnali di accelerazione, definiscono rispettivamente la selettività (pendenza) e la frequenza di taglio. Dopo alcuni tentativi si stabilisce il filtro sia del quinto ordine e la costante di taglio pari a 0,03.

Si possono comparare quindi i segnali prima e dopo il filtraggio:



È rappresentato in figura il caso della Salita scale nella componente x.

Si può notare che è la direzione della forza di gravità, perché il segnale è caratterizzato da accelerazioni che oscillano attorno a valori di 9.81 m/s².

Siccome la gravità rappresenta la componente continua del segnale, viene esclusa dal filtro passa alto.

2.5 Funzione per ricavare le features

L'obiettivo di questo passaggio consiste nell'estrarre da ogni segnale le sue caratteristiche fondamentali, che verranno successivamente usate per la classificazione.

Risulta necessario effettuare una riorganizzazione delle informazioni. I dati sono stati riorganizzati in modo da avere ciascuna azione suddivisa in 'n' set (campioni statistici) in riga, ciascuna da 500 dati campionati. Ciascun set è un campione statistico. I set serviranno poi per allenare il classificatore. I dati sono organizzati in una cell array di sei elementi. Ogni posizione corrisponde ad una specifica attività (elencata nell'introduzione). Ognuna delle tre componenti di accelerazione realizza una matrice 'n x 500', quindi ogni elemento del cell array è organizzato in forma di matrice tridimensionale, con dimensioni 'n x 3 x 500'.

A questo punto i dati relativi ai diversi soggetti vengono messi tutti assieme. Le registrazioni di ciascuno non sono più distinguibili. Il calcolatore dovrà lavorare su tutti i soggetti indistintamente.

Le caratteristiche da estrarre dal segnale per ogni direzione sono:

- Media
- Deviazione standard
- Valore del picco principale
- Posizione del picco principale

Ora è possibile procedere a scrivere la funzione, la quale prende in ingresso i dati del cell array, la frequenza di campionamento, i parametri A e B del filtro passa alto. Questi due ultimi parametri hanno lo scopo di calcolare la deviazione standard in maniera più accurata. Come uscita la funzione ritornerà la matrice “m x 13”, nella quale le prime 12 colonne sono caratterizzate dalle 4 features per ognuna delle 3 direzioni, mentre l’ultima colonna memorizza l’indice della specifica attività svolta (da 1 a 6). Si riporta uno spezzone di codice Matlab, che rappresenta la parte fondamentale dell’estrazione dei dati:

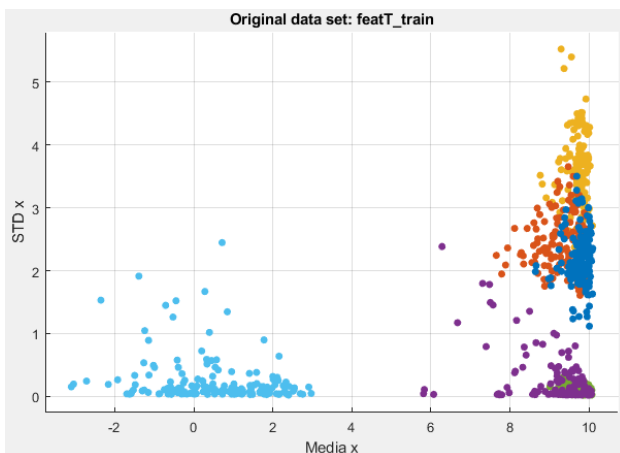
```
features = zeros(1t,13);
g = 1;

for i = 1 : 6 % indicaotre attività
    for j = 1 : length(stat_c{i}(:,1,1)) % indicatore sample number
        for k = 1 : 3 % indicatore componente
            sampleStat = stat_c{i}(j,:,k); % individuo il campione statistico
            Af = mean(sampleStat); % --> media
            Bf = std(sampleStat); % --> dev. standard
            sampleStat_fil = filter(b,a,sampleStat);
            sampleStat_FFT = fft(sampleStat_fil); % trasformata del segnale campionato
            L = length(sampleStat_fil);
            fn = fs/2; % freq di Nyquist
            P2 = abs(sampleStat_FFT/L);
            P1 = P2(1:L/2+1);
            P1(2:end-1) = 2*P1(2:end-1);
            f = (fs*(0:(L/2))/L)/fn;
            [Mf,Indice] = max(P1); % --> valore del max, posizione del max
            Indice_max = f(Indice);
```

2.6 Algoritmo di classificazione

In questa fase viene realizzato il classificatore, partendo dalle features precedentemente estratte.

Siccome lo scopo è quello di testare il classificatore, è possibile suddividere i dati che si hanno a disposizione in due campioni: il primo campione è composto dall’ 80% dei dati, presi in maniera casuale, mentre il secondo campione è composta dal restante 20% dei dati, in modo che non ci siano ripetizioni. La funzione “prevision” ha proprio questo scopo. La funzione prende in ingresso il set totale dei dati e ne restituisce i due campioni divisi. Si noti che il primo campione è composto da 13 colonne, mentre del



secondo se ne hanno solamente 12. La differenza si trova nel fatto che il primo deve essere usato per allenare il classificatore, mentre il secondo verrà usato per testarlo. Sarà quindi il classificatore a riempire l’ultima colonna, cercando di indovinare la tipologia di attività alla quale corrisponde il segnale analizzato.

Per far capire la non banalità del problema, a fianco viene riportato un esempio (sulla coordinata ‘x’) di come alcuni dati sono differenziabili da altri che invece si sovrappongono e sono più probabilmente soggetti ad errori.

2.7 Conclusioni

In ultima analisi si riporta la matrice di confusione:

1	37	3	1			
2	1	24				
3		6	34			
4				28	5	
5				8	55	
6						45
	1	2	3	4	5	6

Predicted Class

Questa matrice permette di fare un confronto tra la predizione dell'attività fatta dal classificatore e l'attività che era stata veramente svolta. Si può notare che Matlab indica di avere un'accuratezza del 85.8%.

Quello che si può evincere dal grafico quindi è che la gran parte delle attività sono state identificate correttamente (indicate in blu sulla diagonale della matrice), mentre le attività che non sono state riconosciute accuratamente si trovano all'esterno della diagonale.