

Modelli probabilistici per le decisioni:  
SmartHouse

Matteo Angelo Costantini, matricola: 795125  
Alessandro Longhi, matricola: 794235

# Contents

<b>1</b>	<b>Introduzione</b>	<b>3</b>
<b>2</b>	<b>Analisi del dataset</b>	<b>4</b>
2.1	Analisi del dataset A . . . . .	4
2.2	Analisi del dataset B . . . . .	5
2.3	Discretizzazione dei dati . . . . .	5
2.4	Preprocessing dei dati . . . . .	6
2.4.1	Conversione dei dataset iniziale . . . . .	6
2.4.2	Creazione dei dataset finali . . . . .	6
2.5	Dataset finali . . . . .	7
<b>3</b>	<b>Creazione del modello HMM</b>	<b>9</b>
3.1	Hidden Markov Models . . . . .	9
3.2	Creazione delle matrici . . . . .	9
3.2.1	Creazione della matrice delle probabilità iniziali . . . . .	9
3.2.2	Creazione della matrice delle transizioni . . . . .	9
3.2.3	Creazione della matrice delle emissioni . . . . .	10
<b>4</b>	<b>Sperimentazione</b>	<b>10</b>
4.1	Scelta del training . . . . .	10
4.2	Analisi dei risultati . . . . .	11
4.2.1	Misure di qualità utilizzate . . . . .	11
4.2.2	Capacità predittiva del modello . . . . .	12
<b>5</b>	<b>GUI</b>	<b>19</b>
<b>6</b>	<b>Conclusioni</b>	<b>21</b>

# 1 Introduzione

I miglioramenti ottenuti in ambito medico hanno portato ad un sensibile aumento dell'aspettativa di vita di ogni individuo, causando un generale invecchiamento della popolazione; il quale rappresenta un importante argomento di studi nell'ambito della sanità. Infatti, aumentando il numero di individui anziani, aumenta anche la necessità di studiare e trovare una soluzione ad alcuni problemi associati alla vecchiaia.

Focalizzarsi sui problemi dovuti all'anzianità significa, quindi, trovare dei modi efficienti per fornire supporto e cure agli individui più anziani direttamente a casa loro; tutto ciò è reso possibile grazie all'uso di sistemi di monitoraggio all'interno delle case, in quanto permettono sia l'aumento della sicurezza dell'abitante, sia il senso di protezione di quest'ultimo, favorendone l'autonomia.

Monitorare le azioni quotidiane degli individui è uno dei task più importanti all'interno di quest'ambito, in quanto permette di descrivere il benessere degli abitanti delle case, sia fisico che mentale. Il monitoraggio delle attività avviene grazie all'uso di alcuni sensori, i quali vengono posti in diversi luoghi all'interno di ogni abitazione.

Questo progetto si colloca nell'ambito appena descritto: avendo a disposizione dei dataset contenenti le rilevazioni delle attività e dei sensori in due abitazioni differenti, il lavoro svolto consiste nella definizione di un modello HMM in grado di inferire le attività data una sequenza di osservazioni derivanti dai sensori. Una volta creata la struttura HMM, è stata effettuata un'analisi riguardante la capacità predittiva del modello.

La relazione verrà presentata seguendo il seguente ordine:

- Verranno inizialmente presentati i dataset a disposizione, descrivendone anche i passaggi delle modifiche che sono state fatte al fine di semplificarne l'utilizzo
- In secondo luogo, verrà discussa la creazione del modello HMM, descrivendone la creazione delle matrici necessarie per la sua definizione
- Al termine di queste fasi, verrà descritta la sperimentazione svolta, alla quale seguiranno le discussioni dei risultati ottenuti
- Infine, verrà presentata l'interfaccia grafica creata per effettuare i test

## 2 Analisi del dataset

I dataset a nostra disposizione erano quattro, due per ogni abitazione, rappresentanti rispettivamente le attività ed i sensori rilevati all'interno dell'abitazione.

Le abitazioni considerate sono due, etichettate con "OrdonezA" ed "OrdonezB".

### 2.1 Analisi del dataset A

Per l'abitazione "OrdonezA" sono stati forniti due dataset: uno denominato "OrdonezA\_ADLs" contenente le attività rilevate, ed uno denominato "OrdonezA\_Sensors" contenente i sensori rilevati. Nel primo dataset, quello delle attività, sono descritti i seguenti campi:

- Start\_time: data ed ora di inizio della rilevazione di un'attività
- End\_time: data ed ora di fine della rilevazione di un'attività
- Activity: attività rilevata nel periodo considerato

in particolare, nella colonna "Activity" sono stati rilevati 10 valori diversi, cioè: Leaving, Toileting, Showering, Sleeping, Breakfast, Lunch, Dinner, Snack, Spare\_Time/TV e Grooming.

Il secondo dataset di "OrdonezA" è dedicato alla raccolta dei sensori rilevati. In quest'ultimo sono presenti i seguenti campi:

- Start\_time: data ed ora di inizio della rilevazione di un sensore
- End\_time: data ed ora di fine della rilevazione di un sensore
- Location: collocazione precisa in cui è posizionato il sensore rilevato
- Type: tipologia del sensore rilevato
- Place: camera dell'abitazione nella quale è posizionato il sensore rilevato

in particolare, nella colonna "location" sono stati rilevati 12 sensori in 4 camere diverse, che, divisi in base al tipo, sono:

- tipo PIR: Shower, Basin, Cooktop
- tipo Magnetic: Maindoor, Fridge, Cabinet, Cupboard
- tipo Flush: Toilet
- tipo Pressure: Seat, Bed
- tipo Electric: Microwave, Toaster

Per entrambi i dataset di A, sono state considerate le rilevazioni ottenute per un periodo di 14 giorni, dal 28-11-2011 al 11-12-2011.

## 2.2 Analisi del dataset B

Per l'abitazione "OrdonezB" sono stati forniti due dataset: uno denominato "OrdonezB\_ADLS" contenente le attività rilevate, ed uno denominato "OrdonezB\_Sensors" contenente i sensori rilevati.

Nel primo dataset, quello delle attività, sono descritti i seguenti campi:

- Start\_time: data ed ora di inizio della rilevazione di un'attività
- End\_time: data ed ora di fine della rilevazione di un'attività
- Activity: attività rilevata nel periodo considerato

in particolare, nella colonna "Activity" sono stati rilevati 10 valori diversi, cioè: Leaving, Toileting, Showering, Sleeping, Breakfast, Lunch, Dinner, Snack, Spare\_Time/TV e Grooming.

Il secondo dataset di "OrdonezB" è dedicato alla raccolta dei sensori rilevati. In quest'ultimo sono presenti i seguenti campi:

- Start\_time: data ed ora di inizio della rilevazione di un sensore
- End\_time: data ed ora di fine della rilevazione di un sensore
- Location: collocazione precisa in cui è posizionato il sensore rilevato
- Type: tipologia del sensore rilevato
- Place: camera dell'abitazione nella quale è posizionato il sensore rilevato

in particolare, nella colonna "location" sono stati rilevati 12 sensori in 5 camere diverse, che, divisi in base al tipo, sono:

- tipo PIR: Shower, Basin, Door Kitchen, Door Bathroom, Door Bedroom
- tipo Magnetic: Maindoor, Fridge, Cupboard
- tipo Flush: Toilet
- tipo Pressure: Seat, Bed
- tipo Electric: Microwave

Per entrambi i dataset di B, sono state considerate le rilevazioni ottenute per un periodo di 21 giorni, dal 11-11-2012 al 2-12-2012.

## 2.3 Discretizzazione dei dati

La lettura dei dati da parte dei sensori e la conseguente rilevazione delle attività vengono effettuate iterativamente, di conseguenza i dati descritti nei dataset iniziali presentano una distribuzione continua; per questo motivo, per ottenere una corretta formattazione temporale dei dati, è necessario effettuare un processo di discretizzazione del tempo.

Il tempo viene quindi diviso in timeslices, intervalli regolari spazati da una predeterminata granularità di tempo  $\Delta t$ .

Le rilevazioni di un sensore per ogni timeslice  $t$  sono denotati come  $x_t^i$  indicante se il sensore  $i$  si è attivato almeno una volta tra il tempo  $t$  ed il tempo  $t + \Delta t$ , con  $x_t^i \in \{0, 1\}$ .

In una casa con  $N$  sensori, verrà quindi definito un vettore di osservazioni  $\vec{x}_t = (x_t^1, x_t^2, \dots, x_t^{N-1}, x_t^N)^T$  per ogni timeslice.

Ogni istanza del dataset sarà rappresentato da un'etichetta che corrisponderà all'attività rilevata in quel segmento di tempo.

L'attività nel timeslice  $t$ , che rappresenta lo stato nel quale si trova il sistema in quel periodo di tempo, è denotato con  $y_t \in \{1, \dots, Q\}$  per  $Q$  possibili stati; di conseguenza ogni istanza del dataset finale sarà rappresentata da un'etichetta  $y_t$  indicante l'attività e da un vettore di osservazioni  $x_t$ , risultante da una mappatura basata sulla sovrapposizione dell'unità di tempo nella quale è stato rilevato un'attività e si sono attivati i sensori.

Sono stati infine introdotti l'attività "No activity" ed il vettore dei sensori interamente a 0, per rappresentare il fatto che in una timeslice non vengano rilevate delle attività o dei sensori.

## 2.4 Preprocessing dei dati

In seguito all'elaborazione dei dati sono stati ottenuti due dataset, uno per "OrdonezA" ed uno per "OrdonezB".

### 2.4.1 Conversione dei dataset iniziale

Prima di descrivere la struttura dei dataset ottenuti, è necessario precisare che prima è stata effettuata una conversione del formato dei dataset iniziali; infatti questi sono stati forniti in formato testuale, ma durante la fase di preprocessing sono stati convertiti in file .csv in modo da renderne più semplice la modifica. Bidogna sottolineare come siano state precedentemente fatte delle modifiche manuali ai file testuali, in modo da semplificarne la conversione in formato .csv.

Ai fini di rendere più semplice l'analisi nelle fasi successive, sono state apportate alcune modifiche ai valori contenuti nei dataset, in particolare:

- le date di inizio e di fine rilevazione sono state convertite in timestamps
- i valori contenuti nelle restanti colonne, presenti sottoforma di stringhe, sono stati mappati su valori interi (ad esempio, in "OrdonezA" nella colonna "Activity", l'attività "Breakfast" è rappresentata dal valore 0, e così via)

Durante queste modifiche, è stato effettuato anche un controllo sulla consistenza di ogni riga, in particolare, sono state eliminate le righe che presentavano una data di inizio rilevazione successiva alla data di fine rilevazione.

### 2.4.2 Creazione dei dataset finali

Una volta effettuate le modifiche descritte precedentemente, è stato svolto, per ogni abitazione, un lavoro di unione del dataset riguardante i sensori e quello riguardante le attività, in modo da ottenere i dataset finali.

Durante questa fase sono stati introdotte le timeslices, che, come sarà presentato in seguito, sono state poste a 60 secondi nella maggior parte della sperimentazione; tuttavia questo parametro è facilmente modificabile.

A causa delle timeslices, quindi, le righe nel dataset risultante sono aumentate, in quanto vengono considerati tutti gli intervalli di grandezza  $\Delta t$  partendo dalle 00:00 del primo giorno fino alle 23:59 dell'ultimo.

A questo punto ogni riga è stata completata con lo stato attivo in quella timeslice, che ne rappresenterà la label  $y_t$ , dal vettore dei sensori attivi in quella timeslice  $\vec{x}_t$ , identificati da "Location", "Type" e "Place" ed infine da un valore che ne indica il momento della giornata.

Si ricorda che è stata aggiunta un'attività denominata "No Activity" per riuscire ad etichettare tutte quelle timeslices nelle quali non veniva rilevata nessuna attività; inoltre è stato posto un vettore di soli 0 nel caso in cui in una timeslice non vengano trovati dei sensori accesi.

## 2.5 Dataset finali

Alla fine del preprocessing vengono generati due dataset finali: "OrdonezA" ed "OrdonezB".

Ognuno di questi dataset contiene i seguenti campi:

- Timestamp: data ed ora di inizio della timeslice
- Activity: valore intero che rappresenta una determinata attività rilevata nella timeslice
- Sensors: vettore di bit avente 1 nelle posizioni sulle quali sono mappati i sensori attivi in quella timeslice
- Period: valore intero rappresentante il momento della giornata nel quale è situata la timeslice

Avendo modellato le attività come interi di seguito viene riportata l'associazione tra ogni attività e il suo corrispettivo valore. Inoltre, nel dataset A non è presente l'attività 'Dinner', quindi i due dataset non hanno la stessa numerazione. Le attività per il dataset A sono:

0. Breakfast
1. Grooming
2. Leaving
3. Lunch
4. Showering
5. Sleeping
6. Snack
7. Spare Time - TV
8. Toileting
9. No activity

Per quanto riguarda le attività del dataset B invece, i valori sono quelli del dataset A incrementati di 1, ad eccezione di Breakfast che rimane invariato e di 'Dinner' che prima era assente:

0. Breakfast
1. Dinner
2. Grooming
3. Leaving
4. Lunch
5. Showering
6. Sleeping
7. Snack
8. Spare Time - TV
9. Toileting
10. No activity



## 3 Creazione del modello HMM

### 3.1 Hidden Markov Models

L' HMM è un modello probabilistico definito in termini di una variabile osservabile  $x_t$  e di una variabile nascosta  $y_t$  ad ogni istante discreto di tempo  $t$ .

Nel caso considerato in questo progetto, la variabile osservabile è formata dalle osservazioni dei sensori al tempo  $t$ , mentre la variabile nascosta è l'attività da riconoscere.

L'HMM è definito dalle due seguenti assunzioni:

- la variabile nascosta al tempo  $t$ , denominata  $y_t$ , dipende solamente dalla variabile nascosta precedente, ovvero  $y_{t-1}$
- la variabile osservabile al tempo  $t$ , denominata  $x_t$ , dipende solamente dalla variabile nascosta al tempo  $t$ , ovvero  $y_t$

Di seguito verranno descritti i fattori che specificano il funzionamento di un modello HMM.

La distribuzione di probabilità a priori  $P(y_t)$  esprime la probabilità che lo stato  $y_t$  sia lo stato iniziale della sequenza.

La distribuzione di osservazione  $P(x_t|y_t)$  è la probabilità che lo stato  $y_t$  generi l'osservazione  $x_t$ .

La distribuzione di probabilità di transizione  $P(y_t|y_{t-1})$  indica la probabilità di passare da uno stato all'altro. Questi valori sono ricavabili dalla *Conditional Probability Table*.

Queste tre distribuzioni descritte precedentemente sono sufficienti alla definizione di un modello.

### 3.2 Creazione delle matrici

Per creare le matrici è necessario avere un trainset che permetta di trovare le probabilità necessarie alla creazione della matrice. Di seguito verrà spiegato come sono state ottenute le varie matrici.

#### 3.2.1 Creazione della matrice delle probabilità iniziali

Di seguito verrà introdotto il metodo usato per calcolare la matrice delle probabilità iniziali, che sarà una matrice  $1 \times n$ , dove  $n$  è il numero degli stati nascosti(nel nostro caso, le attività).

Per trovare questa matrice, è stato effettuato il conteggio del numero di volte in cui uno stato nascosto si presentava all'interno del trainset; una volta ottenuti i valori, questi sono stati divisi per il numero di elementi totali del trainset, ottenendo così le probabilità a priori.

#### 3.2.2 Creazione della matrice delle transizioni

Di seguito verrà introdotto il metodo usato per calcolare la matrice delle transizioni, che sarà una matrice  $n \times n$ , dove  $n$  è il numero degli stati nascosti(nel nostro caso, le attività).

In questa matrice, il generico elemento  $M[i][j]$  indica la probabilità di essere giunti nello stato  $j$ , partendo dallo stato  $i$ . Per calcolare la matrice viene effettuato il conteggio del numero di volte che si passa dallo stato  $i$  allo stato  $j$  nel trainset; questo valore verrà poi diviso per il numero totale di volte in cui si è passato dallo stato  $i$ . Infine, qualora ci siano delle probabilità pari a zero per una transizione da uno stato  $i$  a uno stato  $j$ , questo valore è stato alterato ponendolo uguale a un valore infinitesimale non nullo.

### 3.2.3 Creazione della matrice delle emissioni

Di seguito verrà introdotto il metodo usato per calcolare la matrice delle emissioni, che sarà una matrice  $n \times m$ , dove  $n$  è il numero degli stati nascosti (nel nostro caso, le attività) ed  $m$  il numero delle osservazioni (nel nostro caso, i sensori).

In questa matrice, il generico elemento  $M[i][j]$  indica la probabilità di emettere l'osservazione  $j$ , partendo dallo stato  $i$ . Per calcolare la matrice, viene effettuato il conteggio del numero di volte, all'interno del trainset, che dallo stato  $i$  viene emessa l'osservazione  $j$ ; questo valore verrà poi diviso per il numero totale di volte in cui si è passati da  $i$ . Come per il caso precedente, se un elemento della matrice ha valore esattamente zero, è stato modificato in modo da avere probabilità estremamente basse, ma non comunque esattamente pari a zero.

## 4 Sperimentazione

### 4.1 Scelta del training

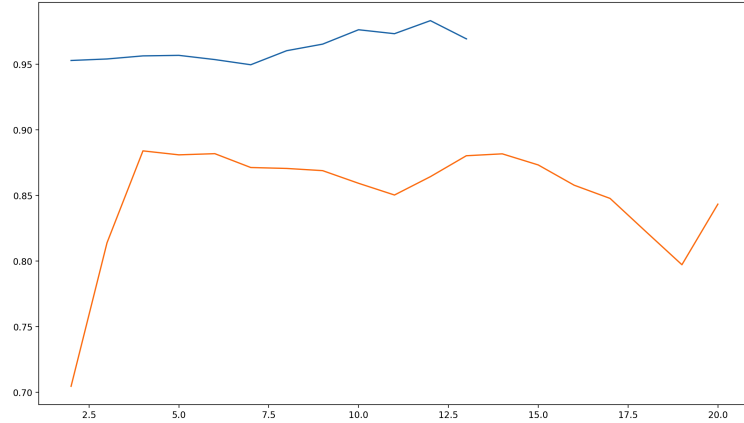
Per la suddivisione delle attività e dei sensori è stato scelto un timeslice di 60 secondi. Questa scelta è un compromesso che permette di rilevare sia eventi particolarmente lunghi che eventi di durata minore e permette di avere dei buoni risultati.

Una prima precisazione riguarda la scelta dei giorni da utilizzare come trainset e come testset: trattandosi di serie temporali non è possibile scegliere gli eventi in modo casuale in quanto spezzerebbero la dipendenza tra due attività consecutive. Per questo motivo è stato deciso di utilizzare i primi giorni del dataset come trainset e la parte finale come testset. Per quanto riguarda il momento esatto in cui è stato deciso di effettuare la suddivisione tra trainset e testset è stato scelto di "tagliare" la serie temporale a cavallo di due giorni e non secondo una percentuale fissata in modo da non interferire con la routine giornaliera e non interrompere attività in maniera irregolare.

Dopo questa scelta è stata effettuata una prima analisi per valutare le prestazioni dell'HMM al variare della dimensione del trainset. In figura 1 viene mostrata l'accuracy ottenuta sia per il dataset A che per il dataset B in funzione del numero di giorni utilizzati come trainset. Come si può vedere sono sufficienti pochi giorni per raggiungere una buona accuracy. Questo è dovuto al fatto che molto probabilmente le giornate si ripetono con una routine particolarmente definita e bastano quindi pochi giorni di osservazioni affinché il modello riesca a prevedere con buona precisione una sequenza di stati.

Fatta questa osservazione è stato deciso di utilizzare come trainset i primi cinque giorni in quanto, come si può vedere, l'andamento della curva si è sta-

bilizzato dopo il primo picco iniziale presente in corrispondenza del terzo e del quarto giorno.



**Figure 1.** Accuracy in funzione del numero di giorni utilizzati come trainset. In blu è riportata l'accuracy del dataset A, in arancione quella del dataset B

## 4.2 Analisi dei risultati

### 4.2.1 Misure di qualità utilizzate

L'esecuzione dell'algoritmo di Viterbi su una sequenza di osservazioni, ha permesso di generare una sequenza di stati nascosti che, con una maggiore probabilità, ha portato alla realizzazione delle osservazioni date come input all'algoritmo.

Una volta ottenuta la sequenza di stati frutto della previsione da parte dell'algoritmo di Viterbi, è stato possibile effettuare il confronto tra quest'ultima e la sequenza di stati reali, in modo da determinare la qualità dei risultati ottenuti.

Per svolgere il controllo della qualità dei risultati sono stati testati i seguenti parametri:

- Accuracy
- Precision
- Recall
- F-Measure

Il test sulla Accuracy permette di trovare la percentuale di correttezza della sequenza di stati nascosti prevista. Di conseguenza, esprime, data una generica posizione  $i$  all'interno della sequenza di stati, se l' $i$ -esimo stato nascosto previsto dall'algoritmo coincide con quello reale.

Prima di descrivere le misure seguenti, è utile introdurre i termini vero positivo, vero negativo, falso positivo e falso negativo, i quali sono usati per confrontare la previsione dell'i-esimo stato nascosto rispetto il corretto stato alla posizione desiderata. Ad esempio:

- Vero positivo: l'i-esimo elemento corretto è sleeping e l'algoritmo ha predetto sleeping
- Vero negativo: l'i-esimo elemento corretto non è sleeping e l'algoritmo ha predetto non sleeping
- Falso positivo: l'i-esimo elemento corretto non è sleeping e l'algoritmo ha predetto sleeping
- Falso negativo: l'i-esimo elemento corretto è sleeping e l'algoritmo ha predetto non sleeping

A questo punto è possibile introdurre le ultime misure di test; Precision e Recall verranno quindi definiti come segue:

$$Precision = \frac{Veropositivo}{VeroPositivo + Falsopositivo}$$

$$Recall = \frac{Veropositivo}{VeroPositivo + Falsonegativo}$$

Infine, il test F-Measure permette di misurare l'accuratezza di un test, attraverso l'uso di Precision e Recall, ed è descritto dalla formula:

$$F - Measure = 2 * \frac{Precision * Recall}{Precision + Recall}$$

#### 4.2.2 Capacità predittiva del modello

Per valutare la capacità predittiva del modello abbiamo svolto tre sperimentazioni diverse, tutte basandosi sul modello generato utilizzando come trainset i primi cinque giorni di osservazione di ognuno dei dataset (per i motivi citati precedentemente). Gli esperimenti svolti sono i seguenti:

1. Predizione su un sottoinsieme del testset comprendente solo i primi tre giorni successivi ai giorni contenuti nel trainset.
2. Predizione sul testset nella sua interezza, ovvero dal giorno successivo al termine del trainset all'ultimo giorno di rilevazioni.
3. Predizione su un campione generato casualmente basandosi sulle distribuzioni di probabilità osservate nel dataset.

**Predizione a breve termine** Come anticipato in precedenza nella prima coppia di test effettuati sono stati utilizzati, per allenare il modello, i primi cinque giorni contenuti nel dataset, sia per il test su A, sia per il test su B; inoltre, il modello è stato testato in entrambi i casi attraverso la sequenza di osservazioni rilevate nei tre giorni seguenti, ovvero dal sesto all'ottavo giorno. Le matrici di confusione ottenute sono riportate in figura 2 per il dataset A e

in figura 3 per il dataset b. Mentre in tabella 1 e 2 sono riportate le misure di Precision, Recall ed F-Measure rispettivamente per i dataset A e B.

Per quanto riguarda il dataset A, la sequenza di stati predetta presenta un valore di **accuracy di 0,949**; ciò indica che la sequenza di stati che l'algoritmo ha predetto è molto simile a quella reale.

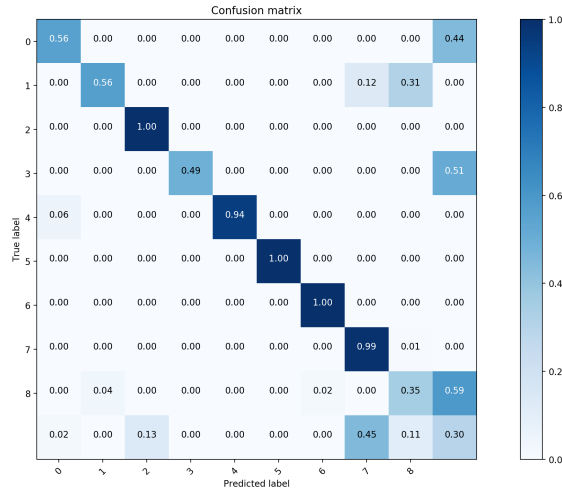
Dalla tabella 1 e dalla figura 2, si possono notare sia gli stati in cui l'algoritmo di Viterbi ha fornito una previsione migliore, sia quelli in cui l'algoritmo è stato meno preciso. Come si può dedurre, nella sequenza prevista dall'algoritmo, gli stati 2, 4, 5, 6 e 7 presentano un livello di predizione pressochè perfetto; mentre gli stati 0, 1, 3, 8 e 9 presentano una minore accuratezza; infatti nella sequenza prevista, questi vengono spesso scambiati, rispettivamente, con i valori 9, 8, 9, 9 e 7.

Per quanto riguarda il dataset B, la sequenza di stati predetta presenta un valore di **accuracy di 0,927**; ciò indica che la sequenza di stati che l'algoritmo ha predetto è molto simile a quella reale, anche se ad un livello minore di A.

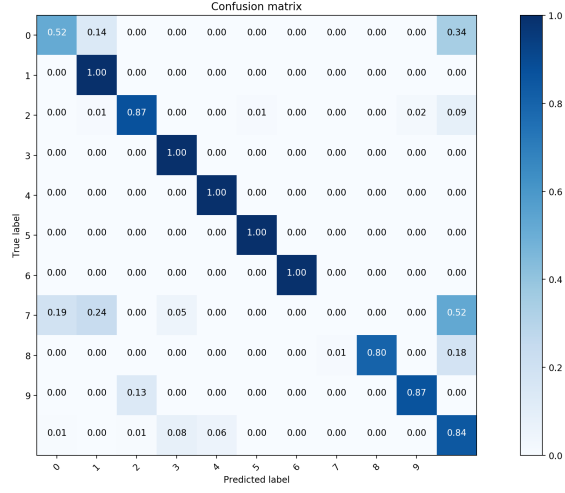
Anche in questo caso, dalla tabella 2 e dalla figura 3, si possono notare sia gli stati in cui l'algoritmo di Viterbi ha fornito una previsione migliore, sia quelli in cui l'algoritmo è stato meno preciso.

In questo dataset, a differenza del caso precedente, la maggior parte degli stati viene prevista con un buon livello di accuratezza, infatti la maggior parte di essi viene predetto correttamente con una probabilità maggiore o uguale di 0,8. Ciò non vale per gli stati 0, che viene spesso scambiato con lo stato 10, e soprattutto con lo stato 7, il quale non viene mai previsto correttamente.

Da questo esperimento è possibile affermare che il modello creato è in grado di effettuare una previsione dettagliata di una sequenza di stati nascosti, utilizzando una sequenza di osservazioni. Ciò è presumibilmente dovuto al fatto che la combinazione di osservazioni, rappresentate dai bit di sensori attivi in una timeslice, è in grado di identificare in modo preciso una determinata attività, rendendone possibile la previsione.



**Figure 2.** Matrice di confusione per il dataset A, testset di 3 giorni



**Figure 3.** Matrice di confusione per il dataset B, testset di 3 giorni

	precision	recall	f1-score	support
0	0.79	0.56	0.65	27
1	0.75	0.56	0.64	16
2	0.95	1.00	0.98	471
3	1.00	0.49	0.66	45
4	1.00	0.94	0.97	18
5	1.00	1.00	1.00	1783
6	0.25	1.00	0.40	1
7	0.95	0.99	0.97	1726
8	0.34	0.35	0.35	54
9	0.44	0.30	0.35	179
avg/total	0.94	0.95	0.94	4320

**Table 1.** Misure di Precision, Recall ed F-Measure per il dataset A, trainset di 3 giorni

	precision	recall	f1-score	support
0	0.68	0.52	0.59	29
1	0.18	1.00	0.30	3
2	0.89	0.87	0.88	94
3	0.97	1.00	0.98	1219
4	0.60	1.00	0.75	45
5	0.94	1.00	0.97	15
6	1.00	1.00	1.00	1452
7	0.00	0.00	0.00	21
8	1.00	0.80	0.89	899
9	0.81	0.87	0.84	15
10	0.69	0.84	0.76	528
avg/total	0.94	0.93	0.93	4320

**Table 2.** Misure di Precision, Recall ed F-Measure per il dataset B, trainset di 3 giorni

**Predizione a lungo termine** Nella seconda coppia di test effettuati sono stati utilizzati, per allenare il modello, i primi cinque giorni contenuti nel dataset, sia per il test su A, sia per il test su B; inoltre, il modello è stato testato in entrambi i casi attraverso la sequenza di osservazioni rilevate seguenti, ovvero dal sesto al quattordicesimo giorno nel caso di A, e dal sesto al ventunesimo giorno nel caso di B. Le matrici di confusione ottenute sono riportate in figura 4 per il dataset A e in figura 5 per il dataset b. Mentre in tabella 3 e 4 sono riportate le misure di Precision, Recall ed F-Measure rispettivamente per i dataset A e B.

Per quanto riguarda il dataset A, la sequenza di stati predetta presenta un valore di **accuracy di 0,957**; ciò indica che la sequenza di stati che l'algoritmo ha predetto è più simile a quella reale rispetto a quella del test precedente.

Dalla tabella 3 e dalla figura 4, si possono notare sia gli stati in cui l'algoritmo di Viterbi ha fornito una previsione migliore, sia quelli in cui l'algoritmo è stato meno preciso. In questo test, cinque stati sono stati previsti correttamente con un livello di accuratezza pari o superiore a 0,8. Bisogna sottolineare come, tra gli stati che hanno ottenuto un punteggio minore, non vi è più lo stato 1; tuttavia a questo gruppo si è aggiunto lo stato 6, che presenta una precisione di previsione molto più bassa rispetto al precedente test.

Per quanto riguarda il dataset B, la sequenza di stati predetta presenta un valore di **accuracy di 0,891**; ciò indica che la sequenza di stati che l'algoritmo ha predetto è molto simile a quella reale, tuttavia in minore quantità rispetto a quella ottenuta nel test precedente.

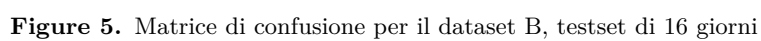
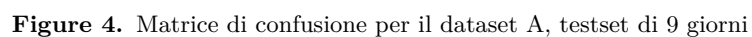
Anche in questo caso, dalla tabella 4 e dalla figura 5, si possono notare sia gli stati in cui l'algoritmo di Viterbi ha fornito una previsione migliore, sia quelli in cui l'algoritmo è stato meno preciso.

In questo dataset, a differenza del caso di A, tutti gli stati vengono previsti con un livello di accuratezza simile a quello ottenuto nel primo esperimento; tuttavia è presente una generale diminuzione della precisione con la quale è avvenuta la previsione, che permette di spiegare il minor valore di accuracy ottenuto in questo caso.

Da questo esperimento è possibile dedurre che il modello è in grado di effettuare la previsione di una sequenza di stati, data una sequenza di osservazioni. Più precisamente, è stato ottenuto un risultato migliore sul dataset riguardante l'abitazione A, mentre il dataset dell'abitazione B ha portato a risultati leggermente peggiori.

	precision	recall	f1-score	support
0	0.80	0.71	0.75	85
1	0.94	0.84	0.88	87
2	0.87	1.00	0.93	1474
3	0.92	0.51	0.65	174
4	0.90	0.99	0.94	72
5	1.00	1.00	1.00	5170
6	0.14	0.11	0.12	9
7	0.98	1.00	0.99	5263
8	0.50	0.56	0.53	94
9	0.53	0.32	0.40	532
avg/total	0.95	0.96	0.95	12960

**Table 3.** Misure di Precision, Recall ed F-Measure per il dataset A, trainset di 9 giorni



**Table 4.** Misure di Precision, Recall ed F-Measure per il dataset B, trainset di 16 giorni

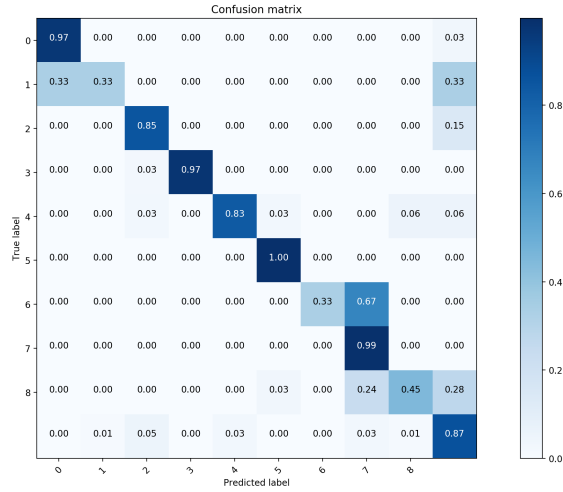


**Predizione con campione casuale** Nell'ultima coppia di test effettuati sono stati utilizzati, per allenare il modello, i primi cinque giorni contenuti nel dataset, sia per il test su A, sia per il test su B; inoltre, il modello è stato testato in entrambi i casi attraverso la sequenza di osservazioni generate a caso, contenenti 3000 minuti. Le matrici di confusione ottenute sono riportate in figura 6 per il dataset A e in figura 8 per il dataset b. Mentre in tabella 5 e 6 sono riportate le misure di Precision, Recall ed F-Measure rispettivamente per i dataset A e B.

In quest'ultimo esperimento sono stati ottenuti i migliori valori di accuracy, ovvero **accuracy di 0,968** per il dataset A e **accuracy di 0,967** per il dataset B.

Dalle tabelle 5 e 6 e dalle figure 6 e 8, si possono notare gli stati che sono stati predetti con una maggiore accuratezza.

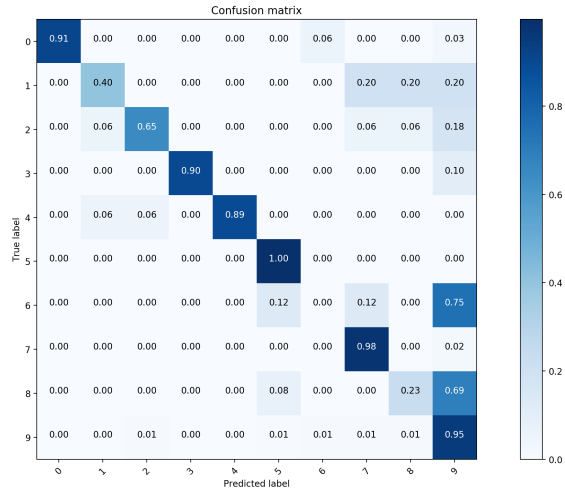
I valori migliori di accuratezza ottenuti in questo esperimento erano prevedibili, in quanto il campione generato si basa sulle matrici ricavate dal trainset utilizzato per allenare il modello, di conseguenza vengono emesse, anche se in modo randomico, delle sequenze facilmente prevedibili dal modello.



**Figure 6.** Matrice di confusione per il dataset A, testset casuale di 3000 campioni

	precision	recall	f1-score	support
0	0.97	0.97	0.97	30
1	0.20	0.33	0.25	3
2	0.89	0.85	0.87	137
3	0.97	0.97	0.97	116
4	0.83	0.83	0.83	36
5	1.00	1.00	1.00	640
6	0.33	0.33	0.33	3
7	0.99	0.99	0.99	1790
8	0.62	0.45	0.52	29
9	0.83	0.87	0.85	216
avg/total	0.97	0.97	0.97	3000

**Table 5.** Misure di Precision, Recall ed F-Measure per il dataset A, testset casuale di 3000 campioni

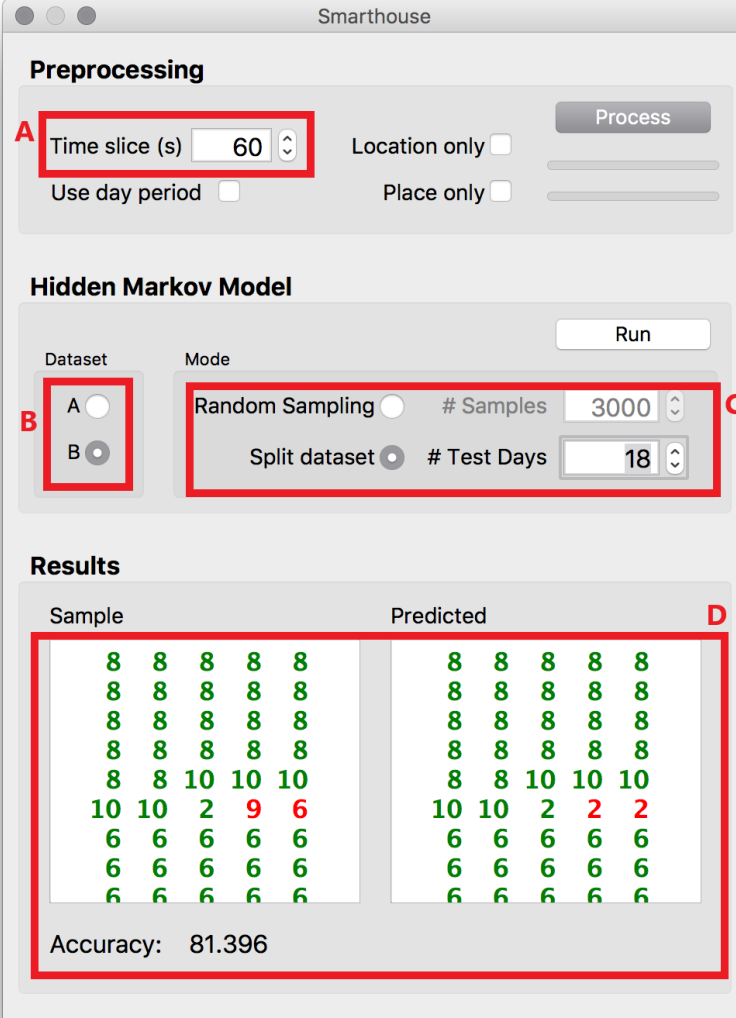


**Figure 7.** Matrice di confusione per il dataset B, testset di 3000 campioni

	precision	recall	f1-score	support
0	0.98	0.91	0.95	68
1	0.33	0.40	0.36	5
2	0.61	0.65	0.63	17
4	1.00	0.90	0.95	108
5	0.89	0.89	0.89	18
6	1.00	1.00	1.00	1370
7	0.00	0.00	0.00	8
8	0.99	0.98	0.98	751
9	0.23	0.23	0.23	13
10	0.93	0.95	0.94	642
avg/total	0.97	0.97	0.97	3000

**Table 6.** Misure di Precision, Recall ed F-Measure per il dataset B, testset casuale di 3000 campioni

## 5 GUI



**Preprocessing**

**A** Time slice (s)  Location only ☐ Process  
Use day period ☐ Place only ☐

**Hidden Markov Model**

**B** Dataset: A ☐ B ☒ Mode: Random Sampling ☐ # Samples  **C**  
Split dataset ☒ # Test Days

**Results**

Sample	Predicted	<b>D</b>
8 8 8 8 8	8 8 8 8 8	
8 8 8 8 8	8 8 8 8 8	
8 8 8 8 8	8 8 8 8 8	
8 8 8 8 8	8 8 8 8 8	
8 8 10 10 10	8 8 10 10 10	
10 10 2 9 6	10 10 2 2 2	
6 6 6 6 6	6 6 6 6 6	
6 6 6 6 6	6 6 6 6 6	
6 6 6 6 6	6 6 6 6 6	

Accuracy: 81.396

**Figure 8.** Interfaccia grafica

- Punto A: sezione per precisare l'ampiezza del timeslice. In questa sezione possono essere settati dei parametri riguardanti il preprocessing dei dati, dedicati alla scelta delle colonne del dataset da utilizzare per identificare ogni istanza
- Punto B: sezione dedicata alla scelta del dataset da utilizzare

- Punto C: sezione dedicata alle modalità con cui eseguire il test; è possibile scegliere se testare il modello sulla divisione in testset e trainset oppure se utilizzare una sequenza generata randomicamente. Inoltre è possibile settare anche la lunghezza, in giorni, del testset o il numero delle istanze da generare randomicamente.
- Punto D: sezione in cui vengono presentati i risultati, nel riquadro a sinistra è presente la sequenza di stati corretta, mentre sulla destra quella predetta. Ogni stato è colorato di verde, nel caso sia stato predetto correttamente, rosso se viceversa. In basso viene presentato il valore di accuracy riscontrato.

## 6 Conclusioni

In questa relazione è stato creato e discusso un modello HMM collocato nell'ambito del servizio sanitario; questo modello, infatti, è stato creato attraverso dei dati ottenuti grazie alle rilevazioni di attività e di sensori all'interno di alcune abitazioni di anziani. Quest'ambito di ricerca è molto importante, in quanto un netto miglioramento delle tecnologie in ambito medico ha portato ad un conseguente innalzamento dell'età media di un individuo. La presenza di tecnologie in grado di rilevare le attività all'interno di un'abitazione quindi, permette la sicurezza degli individui più anziani, promuovendone l'autonomia.

Nella relazione svolta, è stata prima svolta la presentazione del dataset utilizzato per la creazione dell'Hidden Markov Model, a cui è seguita la creazione di quest'ultimo. Nell'ultima parte, è stata invece descritta la sperimentazione eseguita, della quale ne sono stati discussi i risultati.